

JURNAL
KONSTRUKSI PERANGKAT LUNAK
DESIGN PATTERN IMPLEMENTATION



Oleh :

Muhammad Idham Cholid

2211104016

SE0601

PROGRAM STUDI S1 SOFTWARE ENGINEERING

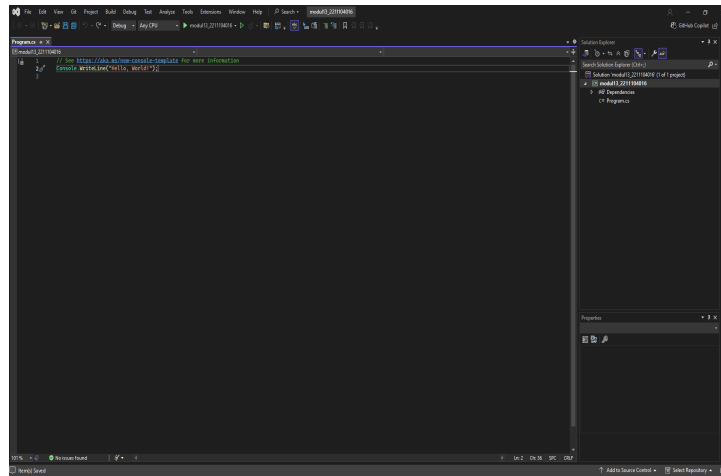
FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

1. MEMBUAT PROJECT GUI BARU

- Membuat Project baru dengan nama Modul13_2211104016



2. MENJELASKAN DESIGN PATTERN

- Logger system
digunakan untuk mencatat log sistem yang berjalan. Jika ada banyak instance logger, log bisa berantakan. Singleton memastikan hanya ada satu logger yang aktif.
- Database Connection Pool
Dalam aplikasi besar, koneksi ke database sebaiknya dikelola terpusat agar tidak membuka koneksi baru terus-menerus.

Langkah Implementasi Singleton:

- Buat konstruktor private.
- Tambahkan field static untuk menyimpan instance.
- Tambahkan method static untuk mengakses atau membuat instance.

Kelebihan Singleton

- Hemat resource, hanya satu instance.
- Akses global yang mudah.
- Cocok untuk shared data seperti konfigurasi.

Kekurangan Singleton

- Susah diuji karena bergantung pada state global.
- Langgar prinsip SRP, mengatur instance dan logika sendiri.
- Tidak thread-safe secara default, butuh logika tambahan.

3. IMPLEMENTASI DAN PEMAHAMAN PROGRAM

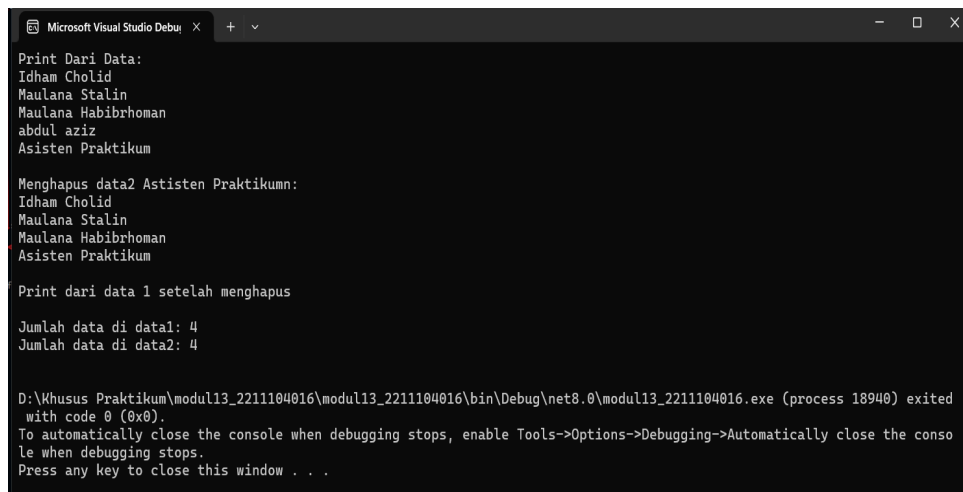
- PusatDataSingleton.cs

```
PusatDataSingleton.cs  Program.cs
modul13_2211104016
1  using System;
2  using System.Collections.Generic;
3
4  namespace modul13_2211104016
5  {
6      public class PusatDataSingleton
7      {
8          private static readonly Lazy<PusatDataSingleton> _instance =
9              new Lazy<PusatDataSingleton>(() => new PusatDataSingleton());
10
11          public static PusatDataSingleton Instance => _instance.Value;
12
13          private List<string> DataTersimpan;
14
15          private PusatDataSingleton()
16          {
17              DataTersimpan = new List<string>();
18          }
19
20          public List<string> GetSemuaData()
21          {
22              return DataTersimpan;
23          }
24
25          public void PrintSemuaData()
26          {
27              foreach (string data in DataTersimpan)
28              {
29                  Console.WriteLine(data);
30              }
31          }
32
33          public void AddSebuahData(string input)
34          {
35              DataTersimpan.Add(input);
36          }
37
38          public void HapusSebuahData(int index)
39          {
40              if (index >= 0 && index < DataTersimpan.Count)
41              {
42                  DataTersimpan.RemoveAt(index);
43              }
44          }
45      }
46  }
```

- Program.cs

```
modul13_2211104016  Program
1  using modul13_2211104016;
2  using System;
3
4  class Program
5  {
6      static void Main(string[] args)
7      {
8          var data1 = PusatDataSingleton.Instance;
9          var data2 = PusatDataSingleton.Instance;
10
11          data1.AddSebuahData("Idham Cholid");
12          data1.AddSebuahData("Maulana Stalin");
13          data1.AddSebuahData("Maulana Habibrhoman");
14          data1.AddSebuahData("abdul aziz");
15          data1.AddSebuahData("Asisten Praktikum");
16
17          Console.WriteLine("Print Dari Data:");
18          data2.PrintSemuaData();
19
20          data2.HapusSebuahData(3);
21
22          Console.WriteLine("\nMenghapus data2 Asisten Praktikum.");
23          data1.PrintSemuaData();
24
25          Console.WriteLine("\nPrint dari data 1 setelah menghapus");
26
27          Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
28          Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
29
30          Console.ReadLine();
31      }
32  }
```

- Output



```
Microsoft Visual Studio Debu  X + -
Print Dari Data:
Idham Cholid
Maulana Stalin
Maulana Habibrhoman
abdul aziz
Asisten Praktikum

Menghapus data2 Astisten Praktikum:
Idham Cholid
Maulana Stalin
Maulana Habibrhoman
Asisten Praktikum

Print dari data 1 setelah menghapus

Jumlah data di data1: 4
Jumlah data di data2: 4

D:\Khusus Praktikum\modul13_2211104016\modul13_2211104016\bin\Debug\net8.0\modul13_2211104016.exe (process 18940) exited
with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

- Penjelasan

Kelas PusatDataSingleton menerapkan pola Singleton, yaitu memastikan hanya ada satu instance yang digunakan selama program berjalan. Kelas ini menyimpan data string dalam list DataTersimpan dan hanya bisa diakses melalui method statis GetDataSingleton(). Beberapa method disediakan untuk menambah, menghapus, mencetak, dan mengambil semua data. Dalam program, dua variabel (data1 dan data2) dibuat dari instance Singleton yang sama. Saat data ditambah lewat data1 lalu dicek dan dihapus lewat data2, hasilnya tetap sinkron karena keduanya mengakses objek yang sama. Ini membuktikan pola Singleton berhasil menjaga konsistensi data secara global dalam aplikasi.