

Laporan Tugas Jurnal Modul 14.

PusatDataSingleton

```
using System;
using
System.Collections.Generic;
using System.Linq; using
System.Text;
using System.Threading.Tasks;

namespace modul13;
public class PusatDataSingleton
{
    private static PusatDataSingleton _instance;
    public List<string> DataTersimpan { get; private set; }

    private PusatDataSingleton()
    {
        DataTersimpan = new List<string>();
    }
    public static PusatDataSingleton GetDataSingleton()
    {
        if (_instance == null)
        {
            _instance = new PusatDataSingleton();
        }
        return _instance;
    }
    public List<string> GetSemuaData()
    {
        return DataTersimpan;
    }
    public void PrintSemuaData()
    {
        foreach (string data in DataTersimpan)
        {
            Console.WriteLine(data);
        }
    }
    public void AddSebuahData(string input)
    {
        DataTersimpan.Add(input);
    }
    public void HapusSebuahData(int index)
    {
        if (index >= 0 && index < DataTersimpan.Count)
        {
            DataTersimpan.RemoveAt(index);
        }
    }
}
```

Penjelasan:

Kelas PusatDataSingleton adalah:

1. **Sebuah Singleton:** Ia memastikan hanya ada satu objek PusatDataSingleton yang ada di seluruh aplikasi, dan menyediakan metode GetDataSingleton() untuk mengakses objek tunggal tersebut.
2. **Sebuah Pengelola Data:** Objek tunggal ini bertanggung jawab untuk menyimpan dan mengelola sebuah daftar string (DataTersimpan).
3. **Titik Akses Global:** Karena merupakan Singleton, daftar data ini dapat diakses dan dimodifikasi dari berbagai bagian aplikasi melalui instance tunggal PusatDataSingleton, menyediakan semacam "pusat data" bersama.

2. Program.cs

```
using modul13;

public class Program
{
    public static void Main(string[] args)
    {
        var data1 = PusatDataSingleton.GetDataSingleton();
        var data2 = PusatDataSingleton.GetDataSingleton();

        data1.AddSebuahData("Ikhsan");
        data1.AddSebuahData("Denis");          data1.AddSebuahData("Joni");

        Console.WriteLine("Data di data2:");
        data2.PrintSemuaData();

        data2.HapusSebuahData(2);

        Console.WriteLine("\nSetelah penghapusan, data di data1:");
        data1.PrintSemuaData();

        Console.WriteLine($"Jumlah data di data1:
{data1.GetSemuaData().Count}");
        Console.WriteLine($"Jumlah data di data2:
{data2.GetSemuaData().Count}");
    }
}
```

Penjelasan:

Kode program ini secara efektif mendemonstrasikan sifat utama dari Singleton pattern:

- Hanya ada satu instance dari kelas PusatDataSingleton di seluruh aplikasi.

- Semua bagian dari program yang meminta instance PusatDataSingleton akan **menerima referensi ke instance tunggal yang sama tersebut.**
- Akibatnya, **data yang disimpan dan dimodifikasi melalui satu referensi ke Singleton akan terlihat dan konsisten ketika diakses melalui referensi lain ke Singleton yang sama, karena pada dasarnya mereka semua bekerja pada satu objek dan satu set data yang sama (DataTersimpan).**

Hasil run

```
Data di data2:
Ikhsan
Denis
Joni

Setelah penghapusan, data di data1:
Ikhsan
Denis

Jumlah data di data1: 2
Jumlah data di data2: 2

D:\KPL_Muhammad-Ikhsan-Al-Hakim_2311104064_SE-07-02\13_Design_Pattern_Implementation\modul13\modul13\bin\Debug\net8.0\modul13.exe (process 141428) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```