

## JURNAL MODUL 13

Nama : Kafka Putra Riyadi

Kelas : SE 07-02

NIM : 231110401

### 1. Source code class PusatDataSingleton.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace JurnalModul13_231110401;
8
9  public class PusatDataSingleton
10 {
11     private static PusatDataSingleton _instance;
12     public List<string> DataTersimpan { get; private set; }
13
14     private PusatDataSingleton()
15     {
16         DataTersimpan = new List<string>();
17     }
18
19     public static PusatDataSingleton GetDataSingleton()
20     {
21         if (_instance == null)
22         {
23             _instance = new PusatDataSingleton();
24         }
25         return _instance;
26     }
27
28     public List<string> GetSemuaData()
29     {
30         return DataTersimpan;
31     }
32
33     public void PrintSemuaData()
34     {
35         foreach (string data in DataTersimpan)
36         {
37             Console.WriteLine(data);
38         }
39     }
40
41     public void AddSebuahData(string input)
42     {
43         DataTersimpan.Add(input);
44     }
45
46     public void HapusSebuahData(int index)
47     {
48         if (index >= 0 && index < DataTersimpan.Count)
49         {
50             DataTersimpan.RemoveAt(index);
51         }
52     }
53 }
54
```

Penjelasan :

Class *PusatDataSingleton* merupakan implementasi dari design pattern Singleton yang memastikan hanya ada satu instance dari class ini selama program berjalan. Class ini memiliki properti `DataTersimpan` berupa list string yang digunakan untuk menyimpan data secara global. Constructor dibuat `private` agar tidak bisa diakses dari luar class, dan instance hanya dibuat satu kali melalui method static `GetDataSingleton()`. Selain itu, class ini menyediakan method untuk menambah data (`AddSebuahData`), menghapus data berdasarkan index (`HapusSebuahData`), menampilkan semua data ke konsol (`PrintSemuaData`), dan mengambil seluruh data dalam bentuk list (`GetSemuaData`). Implementasi ini cocok digunakan ketika aplikasi membutuhkan pusat data tunggal yang dapat diakses oleh banyak bagian program secara konsisten.

## 2. Source code class program.cs

```
1 using JurnalModul13_2311104041;
2
3 public class Program
4 {
5     public static void Main(string[] args)
6     {
7         var data1 = PusatDataSingleton.GetDataSingleton();
8         var data2 = PusatDataSingleton.GetDataSingleton();
9
10        data1.AddSebuahData("Kafka");
11        data1.AddSebuahData("Adit");
12        data1.AddSebuahData("Bu Nia");
13
14        Console.WriteLine("Data di data2:");
15        data2.PrintSemuaData();
16
17        data2.HapusSebuahData(2);
18
19        Console.WriteLine("\nSetelah penghapusan, data di data1:");
20        data1.PrintSemuaData();
21
22        Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
23        Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
24    }
25 }
26
```

Penjelasan :

Program ini menguji implementasi pola Singleton dengan menggunakan class *PusatDataSingleton* yang menyimpan data dalam satu instance. Dua variabel (`data1` dan `data2`) dibuat menggunakan method `GetDataSingleton()`, yang keduanya akan menunjuk pada instance yang sama. Kemudian, program menambahkan tiga data ke `data1` dan mencetaknya melalui `data2`, membuktikan bahwa data disimpan bersama. Setelah itu, data pada index ke-2 (nama "Bu Nia") dihapus melalui `data2`, lalu hasilnya ditampilkan kembali melalui `data1` untuk menunjukkan bahwa perubahan pada satu instance memengaruhi yang lain. Terakhir, jumlah elemen dalam list ditampilkan untuk memastikan bahwa instance benar-benar tunggal dan datanya konsisten.

Outputannya :

```
Data di data2:  
Kafka  
Adit  
Bu Nia  
  
Setelah penghapusan, data di data1:  
Kafka  
Adit  
  
Jumlah data di data1: 2  
Jumlah data di data2: 2
```