

JURNAL MODUL 13

Muhammad Daniel Anugrah Pratama
2311104063
SE07-02

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 JM13
8  {
9      6 references
10     class PusatDataSingleton
11     {
12         private static PusatDataSingleton _instance;
13         6 references
14         public List<string> DataTersimpan { get; private set; }
15
16         1 reference
17         private PusatDataSingleton()
18         {
19             DataTersimpan = new List<string>();
20         }
21
22         2 references
23         public static PusatDataSingleton GetDataSingleton()
24         {
25             if (_instance == null)
26             {
27                 _instance = new PusatDataSingleton();
28             }
29             return _instance;
30         }
31
32         2 references
33         public List<string> GetSemuaData()
34         {
35             return DataTersimpan;
36         }
37
38         2 references
39         public void PrintSemuaData()
40         {
41             foreach (string data in DataTersimpan)
42             {
43                 Console.WriteLine(data);
44             }
45         }
46
47         3 references
48         public void AddSebuahData(string input)
49         {
50             DataTersimpan.Add(input);
51         }
52
53         1 reference
54         public void HapusSebuahData(int index)
55         {
56             if (index >= 0 && index < DataTersimpan.Count)
57             {
58                 DataTersimpan.RemoveAt(index);
59             }
60         }
61     }
62 }
```

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 JM13;
2
3  0 references
4  public class Program
5  {
6      0 references
7      public static void Main(string[] args)
8      {
9          var data1 = PusatDataSingleton.GetDataSingleton();
10         var data2 = PusatDataSingleton.GetDataSingleton();
11
12         data1.AddSebuahData("Daniel");
13         data1.AddSebuahData("Desi");
14         data1.AddSebuahData("Reza");
15
16         Console.WriteLine("Data di data2:");
17         data2.PrintSemuaData();
18
19         data2.HapusSebuahData(2);
20
21         Console.WriteLine("\nSetelah penghapusan, data di data1:");
22         data1.PrintSemuaData();
23
24         Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
25         Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
26     }
27 }
```

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 "Reza") 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:
```

```
Daniel
```

```
Desi
```

```
Reza
```

```
Setelah penghapusan, data di data1:
```

```
Daniel
```

```
Desi
```

```
Jumlah data di data1: 2
```

```
Jumlah data di data2: 2
```