

Haza Zaidan Zidna Fann

2311104056

SISE 0702

https://github.com/hazazaidan/KPL_Haza_Zaidan_Zidna_Fann_2311104056_SISE0702

1. Contoh Kondisi Penggunaan Singleton

- **Contoh 1:** Logger — untuk mencatat log aplikasi agar semua komponen menulis log ke satu objek saja.
- **Contoh 2:** Koneksi ke database — agar aplikasi hanya memiliki satu koneksi aktif ke database yang dikelola terpusat.

2. Langkah Implementasi Singleton

1. Buat class dengan konstruktor **private** agar tidak bisa di-instantiate dari luar.
2. Tambahkan **static field** yang menyimpan instance Singleton.
3. Tambahkan **public static method** yang mengembalikan instance tersebut. Jika belum ada, inisialisasi terlebih dahulu.

3. Kelebihan & Kekurangan Singleton

Kelebihan:

- Menghemat memori karena hanya satu instance dibuat.
- Konsistensi data karena semua modul menggunakan instance yang sama.
- Cocok untuk resource yang mahal dibuat berulang (misal koneksi DB).

Kekurangan:

- Sulit untuk di-test karena instance global tidak mudah dikontrol.
- Bisa menyebabkan tight coupling antar class.
- Jika tidak thread-safe, bisa menyebabkan bug di aplikasi multi-thread.



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

Program.cs

```
1 class Program
2 {
3     static void Main(string[] args)
4     {
5         var data1 = PusatDataSingleton.GetDataSingleton();
6         var data2 = PusatDataSingleton.GetDataSingleton();
7
8         // Tambahkan data anggota kelompok dan asisten
9         data1.AddSebuahData("Jono");
10        data1.AddSebuahData("Jafar");
11        data1.AddSebuahData("Asisten: Budi");
12
13        Console.WriteLine("Data di data2 (setelah penambahan):");
14        data2.PrintSemuaData();
15
16        // Hapus asisten
17        data2.HapusSebuahData(2);
18
19        Console.WriteLine("\nData di data1 (setelah penghapusan asisten):");
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
```

