

NAMA: AMMAR DZAKI NANDANA
KELAS: SE 07 02
NIM: 2311104071

JURNAL MODUL 13

KONSTRUKSI PERANGKAT LUNAK

PusatDataSingleton.cs

```
1  using System;
2  using System.Collections.Generic;
3
4  namespace modul13_2311104071 // ← NIM disesuaikan
5  {
6      public sealed class PusatDataSingleton
7      {
8          private static PusatDataSingleton _instance;
9          public List<string> DataTersimpan;
10
11          private PusatDataSingleton()
12          {
13              DataTersimpan = new List<string>();
14          }
15
16          public static PusatDataSingleton GetDataSingleton()
17          {
18              if (_instance == null)
19              {
20                  _instance = new PusatDataSingleton();
21              }
22              return _instance;
23          }
24
25          public List<string> GetSemuaData()
26          {
27              return DataTersimpan;
28          }
29
30          public void PrintSemuaData()
31          {
32              Console.WriteLine("Isi data:");
33              foreach (var data in DataTersimpan)
34              {
35                  Console.WriteLine($"- {data}");
36              }
37          }
38
39          public void AddSebuahData(string input)
40          {
41              DataTersimpan.Add(input);
42          }
43
44          public void HapusSebuahData(int index)
45          {
46              if (index >= 0 && index < DataTersimpan.Count)
47              {
48                  DataTersimpan.RemoveAt(index);
49              }
50              else
51              {
52                  Console.WriteLine("Index tidak valid.");
53              }
54          }
55      }
56  }
57
```

Program.cs

```
1  using System;
2
3  namespace modul13_2311104071 // ← NIM disesuaikan
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              PusatDataSingleton data1 = PusatDataSingleton.GetDataSingleton();
10             PusatDataSingleton data2 = PusatDataSingleton.GetDataSingleton();
11
12             Console.Write("Masukkan jumlah anggota kelompok (tidak termasuk asisten): ");
13             int jumlahAnggota = int.Parse(Console.ReadLine());
14
15             for (int i = 0; i < jumlahAnggota; i++)
16             {
17                 Console.Write($"Masukkan nama anggota ke-{i + 1}: ");
18                 string anggota = Console.ReadLine();
19                 data1.AddSebuahData(anggota);
20             }
21
22             Console.Write("Masukkan nama asisten praktikum: ");
23             string asisten = Console.ReadLine();
24             data1.AddSebuahData(asisten);
25
26             Console.WriteLine("\nIsi data dari data2:");
27             data2.PrintSemuaData();
28
29             Console.WriteLine("\nMenghapus nama asisten praktikum...");
30             data2.HapusSebuahData(data2.GetSemuaData().Count - 1);
31
32             Console.WriteLine("\nIsi data dari data1 setelah penghapusan:");
33             data1.PrintSemuaData();
34
35             Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
36             Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
37         }
38     }
39 }
40
```

Bagian Teori & Tanya-Jawab (design-pattern *Singleton*)

A. Dua contoh kondisi penggunaan Singleton

- Mengelola koneksi database tunggal yang harus dibagikan ke seluruh aplikasi agar konsisten dan hemat sumber daya.
- Menyediakan configuration manager global: satu objek memuat pengaturan aplikasi (layaknya variabel global, tapi terkontrol).

B. Langkah implementasi singkat

- Tambah field statis privat untuk menampung instance.

- Sediakan metode statis publik (factory) yang mengembalikan instance.
- Terapkan lazy initialization di metode tersebut.
- Jadikan konstruktor privat agar instansiasi langsung dicegah.
- Ubah kode klien memakai metode statis, bukan new.

C. 3 Kelebihan & 3 Kekurangan

Kelebihan

- Menjamin hanya satu instance—mencegah inkonsistensi.
- Akses global yang sederhana.
- Inisialisasi satu kali—hemat memori.

Kekurangan

- Melanggar *Single Responsibility*—kelas menangani instansiasi & logika sendiri.
- Bisa menyembunyikan desain buruk (menggantikan dependency injection).
- Rentan masalah *multithreading* tanpa mekanisme *locking/double-checked locking*.

Penjelasan Kode

`PusatDataSingleton.cs`: implementasi pola Singleton—metode `GetDataSingleton()` memastikan satu-satunya objek, menyediakan operasi CRUD pada koleksi `List<string>`.

`Program.cs`: mendemonstrasikan bahwa variabel `data1` dan `data2` menunjuk instance sama; pembacaan input, penambahan, penghapusan, dan verifikasi daftar.