

## **TUGAS JURNAL MODUL 13**



**Disusun Oleh :**

**Pradana Argo Pangestu**

**2311104079**

**Kelas :SE-07-02**

**Dosen :**

**Yudha Islami Sulistya**

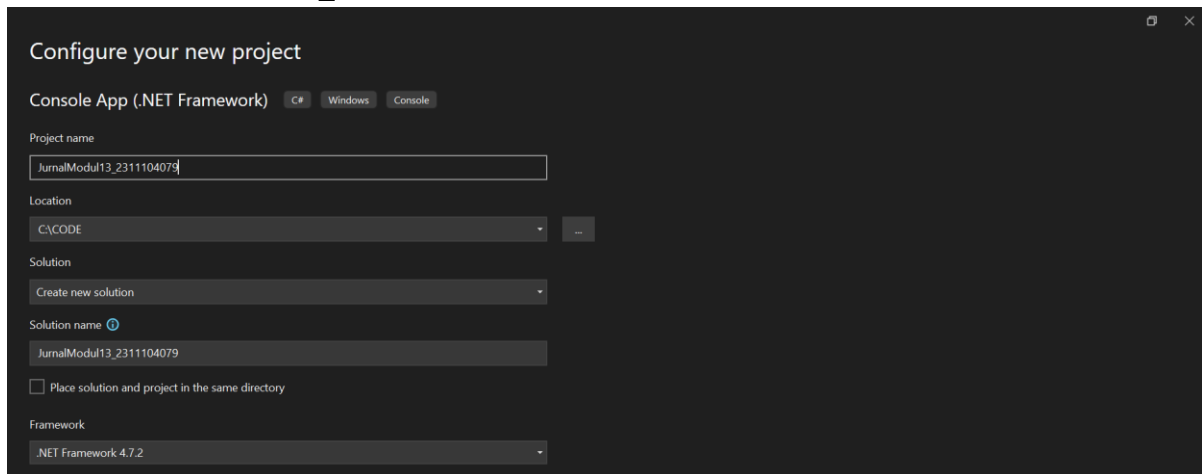
**PROGRAM STUDI SOFTWARE ENGINEERING  
DIREKTORAT KAMPUS PURWOKERTO  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. Link Github

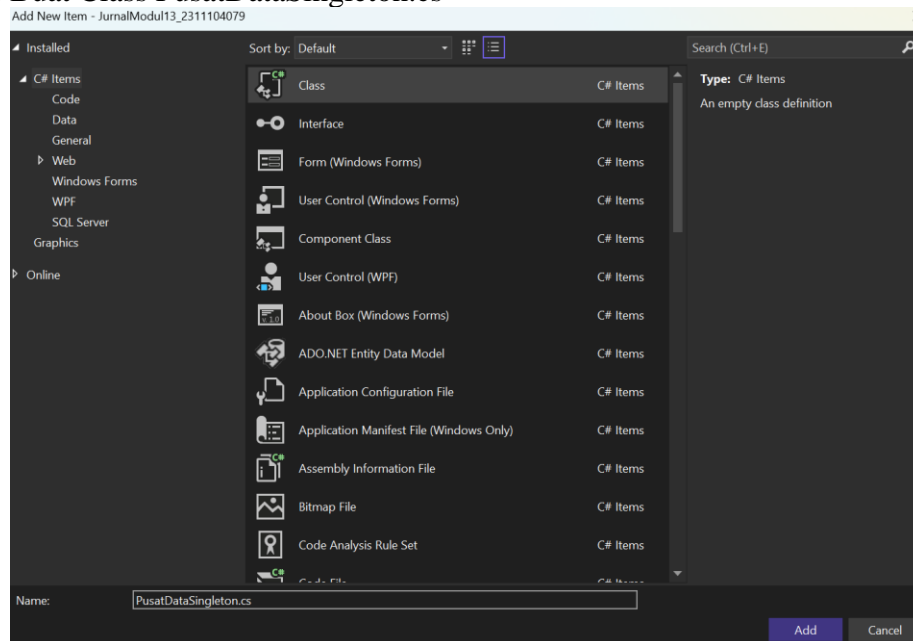
[https://github.com/Pradana123/KPL\\_PradanaAP\\_2311104079/tree/main/13\\_Design\\_Pattern\\_Implementation](https://github.com/Pradana123/KPL_PradanaAP_2311104079/tree/main/13_Design_Pattern_Implementation)

### I. Ss Output :

1. Membuat New Project Menggunakan Console App dengan project name JurnalModul13\_2311104079



2. Buat Class PusatDataSingleton.cs



3. Menambahkan syntax di class PusatDataSingleton.cs

```
1. using System;  
2. using System.Collections.Generic;  
3.  
4. public sealed class PusatDataSingleton
```

```
5. {
6.     private static PusatDataSingleton instance = null;
7.     private static readonly object padlock = new object();
8.
9.     public List<string> DataTersimpan { get; private set; }
10.
11.     private PusatDataSingleton()
12.     {
13.         DataTersimpan = new List<string>();
14.     }
15.
16.     public static PusatDataSingleton GetDataSingleton()
17.     {
18.         lock (padlock)
19.         {
20.             if (instance == null)
21.             {
22.                 instance = new PusatDataSingleton();
23.             }
24.             return instance;
25.         }
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.     public int GetSemuaData()
50.     {
51.         return DataTersimpan.Count;
52.     }
53. }
```

Penjelasan Singkat:

**Class PusatDataSingleton** merupakan penerapan **design pattern Singleton** dalam bahasa C#. Tujuan utamanya adalah untuk memastikan bahwa hanya ada **satu objek (instance)** dari class ini yang digunakan selama program berjalan. Objek tersebut dapat diakses secara **global** melalui method **GetDataSingleton()**.

Class ini memiliki satu properti utama bernama **DataTersimpan**, yaitu sebuah **List<string>** yang digunakan untuk menyimpan kumpulan data dalam bentuk teks.

Agar objek hanya bisa dibuat sekali, **konstruktornya dibuat private**, sehingga tidak bisa diakses dari luar class. Ini adalah ciri khas dari pola Singleton.

Untuk mendapatkan instance-nya, disediakan method **public static GetDataSingleton()**. Method ini akan **membuat objek baru hanya jika belum ada**, dan jika sudah ada, akan mengembalikan objek yang sama.

Class ini juga menyediakan beberapa method untuk mengelola data, yaitu:

- **AddSebuahData(string input)**: untuk menambahkan data baru,
- **HapusSebuahData(int index)**: untuk menghapus data berdasarkan indeks,
- **PrintSemuaData()**: untuk menampilkan semua data ke layar,
- **GetSemuaData()**: untuk mengambil semua data yang tersimpan.

Dengan pola ini, meskipun **GetDataSingleton()** dipanggil dari berbagai bagian program, kita tetap bekerja dengan **objek yang sama**. Hal ini membuat semua perubahan pada data bersifat **global dan konsisten** di seluruh program.

#### 4. Menambah syntax Class Program.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_2311104079
8. {
9.     class Program
10.    {
11.        static void Main(string[] args)
12.        {
13.            // A. Buat dua variable dengan tipe PusatDataSingleton
14.            PusatDataSingleton data1 =
                PusatDataSingleton.GetDataSingleton();
15.            PusatDataSingleton data2 =
                PusatDataSingleton.GetDataSingleton();
16.
17.            // C. Tambahkan data ke data1
18.            data1.AddSebuahData("Nama Anggota 1");
19.            data1.AddSebuahData("Nama Anggota 2");
20.            data1.AddSebuahData("Asisten Praktikum: Budi");
21.
22.            // D. Print semua data dari data2
23.            Console.WriteLine("Data sebelum penghapusan:");
24.            data2.PrintSemuaData();
25.
26.            // E. Hapus data asisten praktikum (index 2)
27.            data2.HapusSebuahData(2);
28.
29.            // F. Print semua data dari data1
30.            Console.WriteLine("\nData setelah penghapusan:");
31.            data1.PrintSemuaData();
32.
33.            // G. Print jumlah data
34.            Console.WriteLine($"Jumlah data di data1:
                {data1.GetSemuaData()}");
35.            Console.WriteLine($"Jumlah data di data2:
                {data2.GetSemuaData()}");
36.        }
37.    }
38. }
```

Penjelasan Singkat:

**Class Program** merupakan **titik awal eksekusi aplikasi**, yaitu melalui method **Main()**. Di dalam method ini, terdapat **simulasi penggunaan pola Singleton** untuk memastikan bahwa objek **PusatDataSingleton** benar-benar hanya ada satu dan digunakan bersama.

Pertama, dua variabel, yaitu **data1** dan **data2**, dibuat dan masing-masing diisi dengan hasil pemanggilan **GetDataSingleton()**. Karena menggunakan **pola Singleton**, kedua variabel ini akan **mengacu pada objek yang sama**.

Selanjutnya, beberapa data ditambahkan melalui **data1**, seperti **nama anggota kelompok** dan **nama asisten praktikum**. Kemudian, data yang sama ditampilkan menggunakan **data2**. Hasilnya tetap sama, karena **data2** menunjuk pada objek yang sama dengan **data1**.

Setelah itu, **data asisten** dihapus melalui **data2**, lalu data kembali ditampilkan menggunakan **data1**. Ini menunjukkan bahwa perubahan data berlaku secara **global**, karena kedua variabel mengakses **instance yang sama**.

Terakhir, program mencetak jumlah data dari **data1** dan **data2**. Hasilnya akan sama, yang kembali menegaskan bahwa **hanya ada satu instance** **PusatDataSingleton** yang digunakan dalam seluruh program.

## I. Hasil Running

### 1. Hasil Running



```
Microsoft Visual Studio Debu X + -
Data sebelum penghapusan:
Nama Anggota 1
Nama Anggota 2
Asisten Praktikum: Budi

Data setelah penghapusan:
Nama Anggota 1
Nama Anggota 2

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

C:\CODE\JurnalModul13_2311104079\JurnalModul13_2311104079\bin\Debug\JurnalModul13_2311104079.exe (process 6660) 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 . . .
```

## II. Kesimpulan

Penerapan **design pattern Singleton** pada class **PusatDataSingleton** berhasil memastikan bahwa hanya ada **satu objek yang digunakan selama program berjalan**. Hal ini terlihat dari penggunaan dua variabel,

data1 dan data2, yang ternyata **mengakses dan memanipulasi data dari instance yang sama.**

Dengan pendekatan ini, program dapat **menjaga konsistensi data secara global**, karena semua bagian program bekerja dengan objek yang sama.

Design pattern Singleton sangat bermanfaat dalam situasi di mana diperlukan **satu sumber data bersama**, seperti pada pengaturan aplikasi, koneksi ke database, atau sistem pencatatan log.

Namun, penting untuk menggunakan pola ini **dengan hati-hati**, karena bisa menimbulkan beberapa kelemahan, seperti:

- **Sulit untuk dilakukan unit testing**, karena ketergantungan pada satu instance.
- **Risiko tight coupling**, yaitu bagian program menjadi terlalu tergantung pada instance global.

Oleh karena itu, meskipun berguna, Singleton sebaiknya digunakan **secara bijak dan sesuai kebutuhan.**