

**LAPORAN JURNAL
MODUL 13**



Disusun Oleh :

Zaenarif Putra 'Ainurdin – 2311104049

Kelas :

SE-07-02

Dosen :

Yudha Islami Sulistya

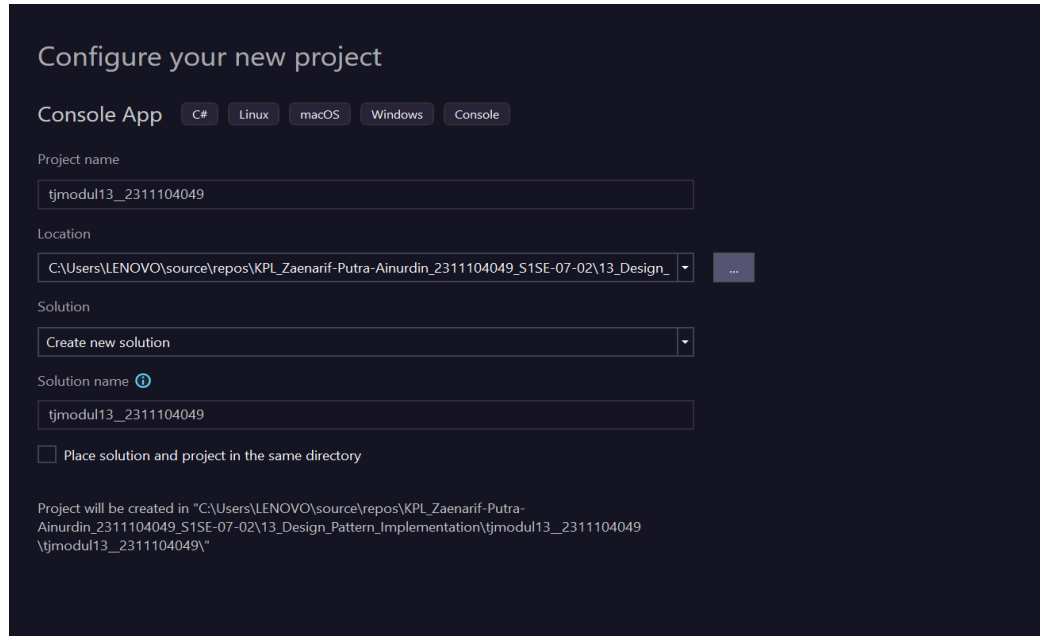
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY
PURWOKERTO
2025**

I. Link Github

https://github.com/zaenarifputra/KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02/tree/5baf947c85be8c62d087ec1ff418f582719264c8/10_Library_Construction/TJ/TjModul10_2311104049

II. Alur Pengerjaan

1. Membuat project baru yang dimana menggunakan Console App yang diberi nama "Tjmodul13_2311104049"



The screenshot shows the 'Configure your new project' dialog in Visual Studio. The 'Console App' template is selected. The project name is 'tjmodul13_2311104049'. The location is 'C:\Users\LENOVO\source\repos\KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02\13_Design_'. The solution is 'Create new solution'. The solution name is 'tjmodul13_2311104049'. The checkbox 'Place solution and project in the same directory' is unchecked. The final path shown is 'C:\Users\LENOVO\source\repos\KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02\13_Design_Pattern_Implementation\tjmodul13_2311104049\tjmodul13_2311104049'.

2. Kemudian Membuat sebuah class yang diberi nama *PusatDataSingleton.cs* yang dimana digunakan untuk mengimplementasikannya berikut penjelasan dari syntax dan juga apa saja yang digunakan pada class *PusatDataSingleton.cs*



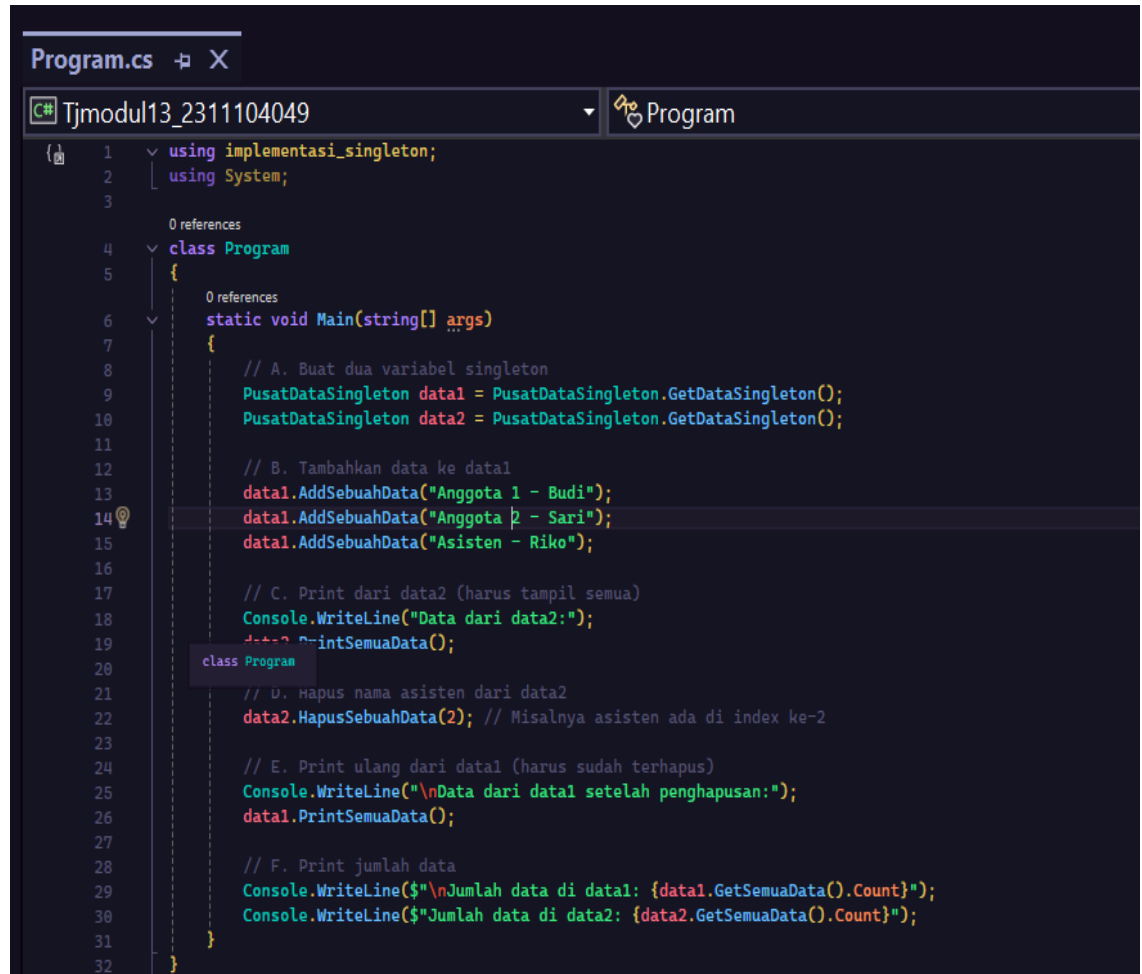
```

PusatDataSingleton.cs  Program.cs
[C#] Tjmodul13_2311104049  implement

4  namespace implementasi_singleton
5  {
6      8 references
7      public class PusatDataSingleton
8      {
9          private static PusatDataSingleton _instance = null;
10         public List<string> DataTersimpan;
11
12         1 reference
13         private PusatDataSingleton()
14         {
15             DataTersimpan = new List<string>();
16         }
17
18         2 references
19         public static PusatDataSingleton GetDataSingleton()
20         {
21             if (_instance == null)
22             {
23                 _instance = new PusatDataSingleton();
24             }
25             return _instance;
26         }
27
28         2 references
29         public List<string> GetSemuaData()
30         {
31             return DataTersimpan;
32         }
33
34         2 references
35         public void PrintSemuaData()
36         {
37             if (DataTersimpan.Count == 0)
38             {
39                 Console.WriteLine("(Tidak Terdapat Data Yang Disimpan)");
40             }
41             else
42             {
43                 for (int i = 0; i < DataTersimpan.Count; i++)
44                 {
45                     Console.WriteLine($"{i + 1}. {DataTersimpan[i]}");
46                 }
47             }
48         }
49
50         3 references
51         public void AddSebuahData(string input)
52         {
53             DataTersimpan.Add(input);
54         }
55
56         1 reference
57         public void HapusSebuahData(int index)
58         {
59             if (index >= 0 && index < DataTersimpan.Count)
60             {
61                 DataTersimpan.RemoveAt(index);
62             }
63             else
64             {
65                 Console.WriteLine("Index invalid. Tidak ada data yang dihapus.");
66             }
67         }
68
69         0 references
70         public override string ToString()
71         {
72             return $"PusatDataSingleton dengan {DataTersimpan.Count} data tersimpan.";
73         }
74     }
  
```

Syntax berikut merupakan implementasi pada penerapan Singleton, yang memastikan hanya ada satu objek dari kelas PusatDataSingleton yang digunakan di seluruh program. Kelas ini berfungsi sebagai pusat penyimpanan data berupa list string, dengan atribut DataTersimpan untuk menyimpan data dan _instance sebagai penampung satu-satunya instans Singleton. Di dalamnya terdapat beberapa method, yaitu GetDataSingleton() untuk mendapatkan instans Singleton, GetSemuaData() untuk mengambil seluruh data yang tersimpan, PrintSemuaData() untuk menampilkan semua data ke konsol dengan penanganan jika list kosong, AddSebuahData(string input) untuk menambahkan data baru ke dalam list, HapusSebuahData(int index) untuk menghapus data berdasarkan indeks yang diberikan, dan ToString() untuk menampilkan informasi jumlah data yang tersimpan dalam bentuk string. Kelas ini sangat bermanfaat saat dibutuhkan satu titik pusat penyimpanan data yang konsisten dan dapat diakses dari berbagai bagian dalam aplikasi.

3. Kemudian setelah menyelesaikan implementasi Singleton, kemudian selanjutnya pada main program pada *Program.cs* yang dimana digunakan untuk menggunakan yang tadi sudah di implementasikan pada *PusatDataSingleton.cs* berikut penjelasan syntax pada programnya :



```

1  using implementasi_singleton;
2  using System;
3
4  class Program
5  {
6      static void Main(string[] args)
7      {
8          // A. Buat dua variabel singleton
9          PusatDataSingleton data1 = PusatDataSingleton.GetDataSingleton();
10         PusatDataSingleton data2 = PusatDataSingleton.GetDataSingleton();
11
12         // B. Tambahkan data ke data1
13         data1.AddSebuahData("Anggota 1 - Budi");
14         data1.AddSebuahData("Anggota 2 - Sari");
15         data1.AddSebuahData("Asisten - Riko");
16
17         // C. Print dari data2 (harus tampil semua)
18         Console.WriteLine("Data dari data2:");
19         data2.PrintSemuaData();
20
21         // D. Hapus nama asisten dari data2
22         data2.HapusSebuahData(2); // Misalnya asisten ada di index ke-2
23
24         // E. Print ulang dari data1 (harus sudah terhapus)
25         Console.WriteLine("\nData dari data1 setelah penghapusan:");
26         data1.PrintSemuaData();
27
28         // F. Print jumlah data
29         Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
30         Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
31     }
32 }
  
```

Syntax berikut dimulai dengan membuat dua variabel, data1 dan data2, yang keduanya mendapatkan objek yang sama melalui method `GetDataSingleton()`, karena Singleton memastikan hanya ada satu instans yang digunakan bersama. Kemudian, tiga data ditambahkan melalui data1, dan data yang sama dapat langsung diakses serta ditampilkan melalui data2, membuktikan bahwa kedua variabel tersebut merujuk ke objek yang sama. Selanjutnya, data ketiga dihapus menggunakan data2, dan ketika isi data ditampilkan kembali melalui data1, data tersebut sudah hilang, yang semakin menegaskan bahwa perubahan pada satu variabel juga terlihat pada variabel lainnya karena keduanya menggunakan satu sumber data yang sama. Di akhir program, jumlah data pada data1 dan data2 juga ditampilkan dan hasilnya sama, karena keduanya mengakses list data yang identik dalam satu instans Singleton. Program ini menggambarkan dengan sederhana dan jelas bagaimana pola Singleton bekerja sebagai pusat penyimpanan data yang seragam dan konsisten di seluruh aplikasi.

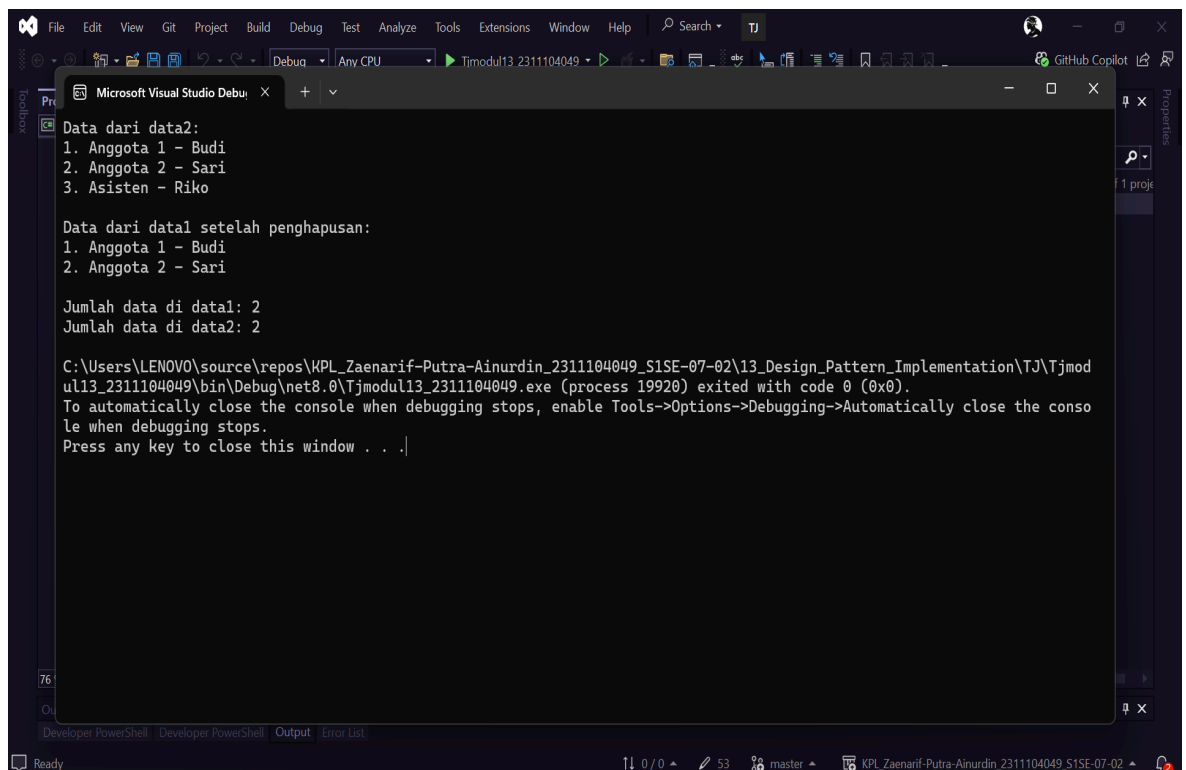
4. setelah semua implementasi yang dibutuhkan sudah berhasil di implementasikan semua selanjutnya melakukan push menuju cloud github yang dimana untuk melakukan push bisa melakukannya dengan perintah `git add .`, `git commit -m "mengimplementasikan GUI dan juga Unit Testing Pada TJ 12"` dan kemudian terakhir melakukan `git push -u origin master`. jika berhasil maka akan di tampilkan seperti gambar berikut :

KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02 / 13_Design_Pattern_Implementation / TJ / Tjmodul13_2311104049 /

 **zaenarifputra** mengimplementasikan singleton modul 13

| Name | Last commit message |
|-----------------------------|--|
| .. | |
| Program.cs | mengimplementasikan singleton modul 13 |
| PusatDataSingleton.cs | mengimplementasikan singleton modul 13 |
| Tjmodul13_2311104049.csproj | mengimplementasikan singleton modul 13 |

III. Hasil Running



```

Data dari data2:
1. Anggota 1 - Budi
2. Anggota 2 - Sari
3. Asisten - Riko

Data dari data1 setelah penghapusan:
1. Anggota 1 - Budi
2. Anggota 2 - Sari

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

C:\Users\LENOVO\source\repos\KPL_Zaenarif-Putra-Ainurdin_2311104049_S1SE-07-02\13_Design_Pattern_Implementation\TJ\Tjmodul13_2311104049\bin\Debug\net8.0\Tjmodul13_2311104049.exe (process 19920) 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 . . .
  
```

Kesimpulannya bisa memahami cara penggunaan dari singleton yang dimana juga method dan juga atribut yang digunakan serta output yang bisa dimanfaatkan dengan baik dan juga import menggunakan namespace agar bisa memahaminya lebih baik