

**TUGAS JURNAL
MODUL 14**



Disusun Oleh :

Pradana Argo Pangestu – 2311104079

Kelas : SE-07-02

Dosen :

Yudha Islami Sulistya

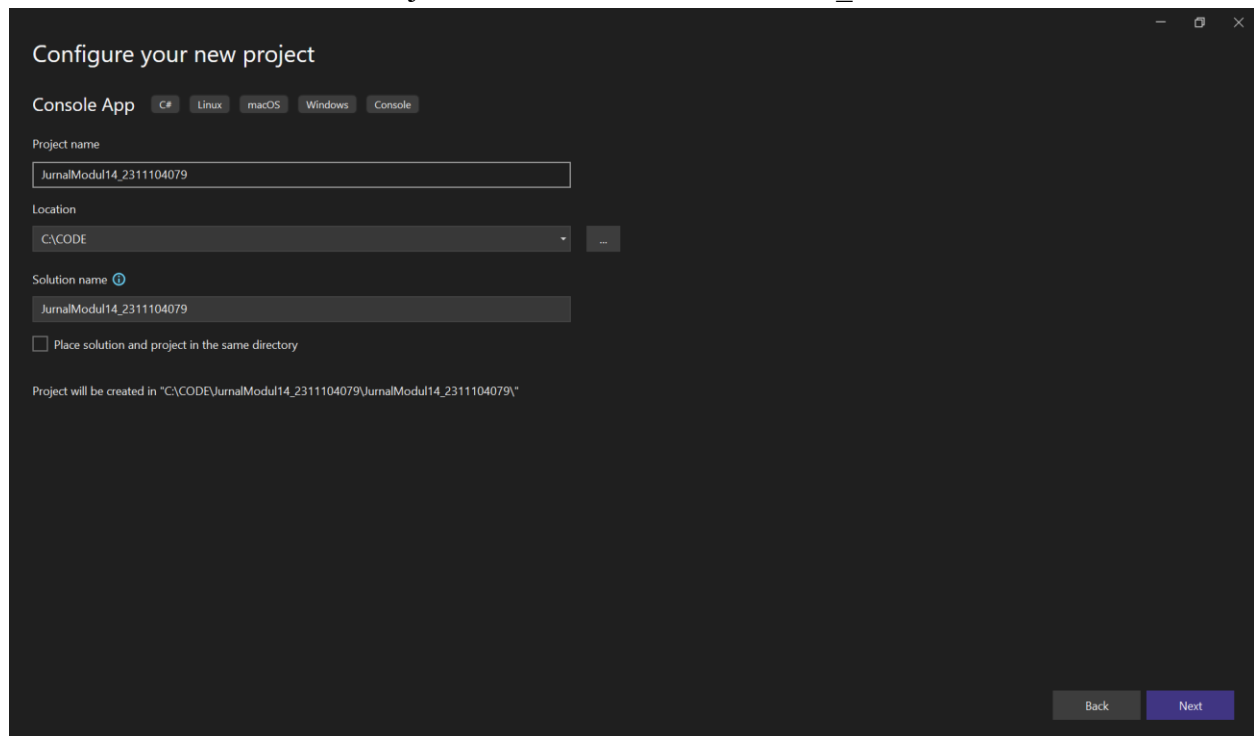
**PROGRAM STUDI SOFTWARE ENGINEERING
DIREKTORAT KAMPUS PURWOKERTO
TELKOM UNIVERSITY PURWOKERTO
2025**

I. Link Github

- https://github.com/Pradana123/KPL_PradanaAP_2311104079/tree/main/14_Clean_Code

II. Alur Pengerjaan

1. Membuat Project Console APP JurnalModul14_2311104079



The screenshot shows the 'Configure your new project' dialog in Visual Studio. The 'Console App' template is selected. The project name is 'JurnalModul14_2311104079', the location is 'C:\CODE', and the solution name is 'JurnalModul14_2311104079'. The checkbox 'Place solution and project in the same directory' is unchecked. A message at the bottom states: 'Project will be created in "C:\CODE\JurnalModul14_2311104079\JurnalModul14_2311104079"'. Navigation buttons 'Back' and 'Next' are at the bottom right.

Configure your new project

Console App C# Linux macOS Windows Console

Project name
JurnalModul14_2311104079

Location
C:\CODE

Solution name ⓘ
JurnalModul14_2311104079

☐ Place solution and project in the same directory

Project will be created in "C:\CODE\JurnalModul14_2311104079\JurnalModul14_2311104079"

Back Next

2. Code pada Class “PusatDataSingleton .cs”

```
using System;
using System.Collections.Generic;

namespace PusatDataSingleton
{
    public class PusatData
    {
        private static PusatData? _instance;
        private List<string> data = new();

        // Constructor private agar tidak bisa diinstansiasi dari luar
        private PusatData() { }

        // Method untuk mendapatkan satu-satunya instance (Singleton)
        public static PusatData GetInstance()
        {
            if (_instance == null)
                _instance = new PusatData();
            return _instance;
        }

        public void TambahkanData(string item)
        {
            data.Add(item);
        }
    }
}
```

```
public void HapusSebuahData(int index)
{
    if (index >= 0 && index < data.Count)
        data.RemoveAt(index);
}

public void PrintSemuaData()
{
    Console.WriteLine("Isi Data:");
    foreach (var item in data)
        Console.WriteLine("- " + item);
}
}
```

Penjelasan Singkat:

Class `PusatData` adalah implementasi dari pola desain Singleton yang bertugas sebagai pusat penyimpanan data berupa list string. Konstruktor class ini dibuat `private` agar objek `PusatData` tidak dapat dibuat secara langsung dari luar class, sehingga hanya ada satu instance yang dapat diakses melalui method `GetInstance()`. Method `GetInstance()` akan membuat instance baru jika belum ada, atau mengembalikan instance yang sudah ada sebelumnya. Di dalam class ini terdapat sebuah list data untuk menyimpan elemen string. Method `TambahkanData(string item)` berfungsi untuk menambahkan data ke dalam list, sedangkan `HapusSebuahData(int index)`

menghapus data berdasarkan indeks yang diberikan jika indeks tersebut valid. Terakhir, method `PrintSemuaData()` digunakan untuk mencetak seluruh isi data ke konsol dengan format yang rapi.

3. Memasukkan Code pada Class “Program .cs”

```
using System;
using PusatDataSingleton;

class Program
{
    static void Main()
    {
        var pusat = PusatData.GetInstance();

        pusat.TambahkanData("Joko");
        pusat.TambahkanData("Bambang");
        pusat.TambahkanData("Rian");

        Console.WriteLine("Data awal:");
        pusat.PrintSemuaData();

        pusat.HapusSebuahData(2); // hapus "Rian"

        Console.WriteLine("\nData setelah penghapusan:");
        pusat.PrintSemuaData();

        Console.ReadKey();
    }
}
```

Penjelasan singkat:

Class Program merupakan titik masuk (entry point) dari aplikasi konsol ini. Di dalam method `Main()`, program mengambil instance tunggal dari class `PusatData` melalui method `GetInstance()`—ini memastikan bahwa hanya ada satu objek `PusatData` yang digunakan sepanjang aplikasi (konsep

Singleton). Selanjutnya, data berupa string seperti "Joko", "Bambang", dan "Rian" ditambahkan ke dalam pusat data tersebut menggunakan method `TambahkanData()`. Program kemudian menampilkan semua data yang ada dengan memanggil `PrintSemuaData()`. Setelah itu, program menghapus data pada indeks ke-2 (data "Rian") dengan method `HapusSebuahData()`. Terakhir, program kembali menampilkan data yang tersisa, dan menunggu input dari pengguna agar console tidak langsung tertutup.

III. Hasil Output

```
Data awal:  
Isi Data:  
- Joko  
- Bambang  
- Rian  
  
Data setelah penghapusan:  
Isi Data:  
- Joko  
- Bambang
```

IV. Kesimpulan

Kode program yang kamu buat terdiri dari dua bagian utama, yaitu class Program yang berfungsi sebagai titik awal eksekusi aplikasi dan class PusatData yang mengimplementasikan pola desain Singleton untuk pengelolaan data secara tunggal. Class Program menjalankan aplikasi dengan mengambil instance tunggal dari PusatData, menambahkan data, menampilkan data, menghapus data berdasarkan indeks, dan menampilkan kembali data yang tersisa. Sementara itu, class PusatData mengatur penyimpanan data dalam sebuah list dan memastikan hanya ada satu instance yang aktif melalui konstruktor private dan method `GetInstance()`. Dari sisi clean code, kode ini sudah cukup baik karena menggunakan penamaan yang jelas dan deskriptif, memisahkan tanggung jawab dengan baik, serta struktur kode yang mudah dibaca dan dipahami. Namun, terdapat beberapa aspek yang masih bisa diperbaiki untuk meningkatkan kualitasnya, seperti menambahkan validasi dan penanganan kesalahan saat menghapus data untuk menghindari kesalahan tidak terdeteksi, memperbaiki nama method agar lebih singkat dan konsisten, menambahkan komentar yang informatif, dan mempertimbangkan aspek thread safety jika aplikasi dikembangkan ke kondisi multi-threading. Dengan melakukan perbaikan tersebut, kode akan menjadi lebih robust, mudah dipelihara, dan siap untuk digunakan dalam aplikasi yang lebih kompleks. Jika kamu ingin, aku dapat membantu membuat versi kode yang lebih bersih dan kuat sesuai prinsip clean code.