

**JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 12  
Design Pattern Implementation**



**Disusun Oleh :  
Andera Singgih Pratama  
2211104007  
SE0601**

**Asisten Praktikum :  
Naufal El Kamil Aditya Pratama Rahman  
Imelda**

**Dosen Pengampu :  
Yudha Islami Sulistya, S.Kom., M.Cs.**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## Output Program

```
Terminal - modul13_2211104007

=== Data dari data2 ===
Andera Singgih pratama
Rizky Hanifa
Idham Cholid
Fauzan Mubarak
Imelda
Naufal
=== Data dari data1 setelah penghapusan Imelda & Naufal ===
Andera Singgih pratama
Rizky Hanifa
Idham Cholid
Fauzan Mubarak
Jumlah data di data1: 4
Jumlah data di data2: 4
```

## Code Program

```
1  using System;
2  using System.Collections.Generic;
3
4  public class PusatDataSingleton
5  {
6      private static PusatDataSingleton _instance;
7      public List<string> DataTersimpan { get; private set; }
8
9      // Konstruktor private
10     private PusatDataSingleton()
11     {
12         DataTersimpan = new List<string>();
13     }
14
15     // Method Singleton
16     public static PusatDataSingleton GetDataSingleton()
17     {
18         if (_instance == null)
19         {
20             _instance = new PusatDataSingleton();
21         }
22         return _instance;
23     }
24
25     // Mengembalikan semua data
26     public List<string> GetSemuaData()
27     {
28         return DataTersimpan;
29     }
30
31     // Menampilkan semua data
32     public void PrintSemuaData()
33     {
34         if (DataTersimpan.Count == 0)
35         {
36             Console.WriteLine("Data kosong.");
37         }
38         else
39         {
40             foreach (var data in DataTersimpan)
41             {
42                 Console.WriteLine(data);
43             }
44         }
45     }
46 }
```

```

        if (index >= 0 && index < DataTersimpan.Count)
        {
            DataTersimpan.RemoveAt(index);
        }
        else
        {
            Console.WriteLine("Index tidak valid!");
        }
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        // A & B: Buat dua variabel singleton dan isi dari GetDataSingleton()
        PusatDataSingleton data1 = PusatDataSingleton.GetDataSingleton();
        PusatDataSingleton data2 = PusatDataSingleton.GetDataSingleton();

        // C: Tambah data anggota kelompok dan asisten praktikum ke data1
        data1.AddSebuahData("Andera Singgih pratama");
        data1.AddSebuahData("Rizky Hanifa");
        data1.AddSebuahData("Idham Cholid");
        data1.AddSebuahData("Fauzan Mubarak");
        data1.AddSebuahData("Imelda"); // Asisten praktikum
        data1.AddSebuahData("Naufal"); // Asisten praktikum

        // D: Print semua data dari data2
        Console.WriteLine("=== Data dari data2 ===");
        data2.PrintSemuaData();

        // E: Hapus nama asisten praktikum ("Imelda" dan "Naufal") dari data2
        int indexImelda = data2.GetSemuaData().IndexOf("Imelda");
        if (indexImelda != -1)
            data2.HapusSebuahData(indexImelda);

        int indexNaufal = data2.GetSemuaData().IndexOf("Naufal");
        if (indexNaufal != -1)
            data2.HapusSebuahData(indexNaufal);

        // F: Print data kembali dari data1 (harusnya Imelda & Naufal sudah tidak ada)
        Console.WriteLine("=== Data dari data1 setelah penghapusan Imelda & Naufal ===");
        data1.PrintSemuaData();

        // G: Print jumlah elemen di data1 dan data2
        Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
        Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
    }
}

```

## Penjelasan Singkat

Program ini merupakan implementasi dari design pattern Singleton dalam bahasa C#. Tujuan utamanya adalah memastikan hanya ada satu objek dari class `PusatDataSingleton` yang digunakan bersama di seluruh program. Kelas `PusatDataSingleton` memiliki atribut `DataTersimpan` berupa list string yang menyimpan data, seperti nama anggota kelompok dan asisten praktikum. Pada method `Main`, dibuat dua variabel `data1` dan `data2` yang keduanya mengambil instance dari method `GetDataSingleton()`. Karena menggunakan Singleton, kedua variabel tersebut merujuk pada objek yang sama. Data anggota kelompok dan asisten praktikum ditambahkan melalui `data1`, lalu ditampilkan lewat `data2`. Selanjutnya, data asisten praktikum ("Imelda" dan "Naufal") dihapus melalui `data2`. Ketika data ditampilkan kembali lewat `data1`, data yang terhapus tidak muncul lagi, membuktikan bahwa kedua variabel mengakses instance yang sama. Terakhir, program mencetak jumlah total data yang tersisa dari kedua variabel. Program ini menunjukkan penggunaan pola Singleton untuk berbagi data yang konsisten di seluruh bagian aplikasi.