

**JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 12  
Design Pattern Implementation**



**Disusun Oleh :  
Mohammad Dhimas Afrizal  
2211104023  
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_2211104023

=== Data dari data2 ===
Mohammad Dhimas Afrizal
Juna
Luthfi
Aziz
Imelda
Naufal
=== Data dari data1 setelah penghapusan Imelda & Naufal ===
Mohammad Dhimas Afrizal
Juna
Luthfi
Aziz
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 }
```

```

46
47 // Menambahkan data baru
48 public void AddSebuahData(string input)
49 {
50     DataTersimpan.Add(input);
51 }
52
53 // Menghapus data berdasarkan index
54 public void HapusSebuahData(int index)
55 {
56     if (index >= 0 && index < DataTersimpan.Count)
57     {
58         DataTersimpan.RemoveAt(index);
59     }
60     else
61     {
62         Console.WriteLine("Index tidak valid!");
63     }
64 }
65
66 public class Program
67 {
68     public static void Main(string[] args)
69     {
70         // A & B: Buat dua variabel singleton dan isi dari GetDataSingleton()
71         PusatDataSingleton data1 = PusatDataSingleton.GetDataSingleton();
72         PusatDataSingleton data2 = PusatDataSingleton.GetDataSingleton();
73
74         // C: Tambah data anggota kelompok dan asisten praktikum ke data1
75         data1.AddSebuahData("Mohammad Dhimas Afrizal");
76         data1.AddSebuahData("Juna");
77         data1.AddSebuahData("Luthfi");
78         data1.AddSebuahData("Aziz");
79         data1.AddSebuahData("Imelda"); // Asisten praktikum
80         data1.AddSebuahData("Naufal"); // Asisten praktikum
81
82         // D: Print semua data dari data2
83         Console.WriteLine("=== Data dari data2 ===");
84         data2.PrintSemuaData();
85
86         // E: Hapus nama asisten praktikum ("Imelda" dan "Naufal") dari data2
87         int indexImelda = data2.GetSemuaData().IndexOf("Imelda");
88         if (indexImelda != -1)
89             data2.HapusSebuahData(indexImelda);
90
91         int indexNaufal = data2.GetSemuaData().IndexOf("Naufal");
92         if (indexNaufal != -1)
93             data2.HapusSebuahData(indexNaufal);
94
95         // F: Print data kembali dari data1 (harusnya Imelda & Naufal sudah tidak ada)
96         Console.WriteLine("=== Data dari data1 setelah penghapusan Imelda & Naufal ===");
97         data1.PrintSemuaData();
98
99         // G: Print jumlah elemen di data1 dan data2
100        Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
101        Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
102    }
103 }

```

## 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.