TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK

MODUL XIII

DESIGN PATTERN IMPLEMENTATION

Disusun Oleh :

Rizky Hanifa Afania

2211104017

SE-06-01

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

TELKOM UNIVERSITY PURWOKERTO

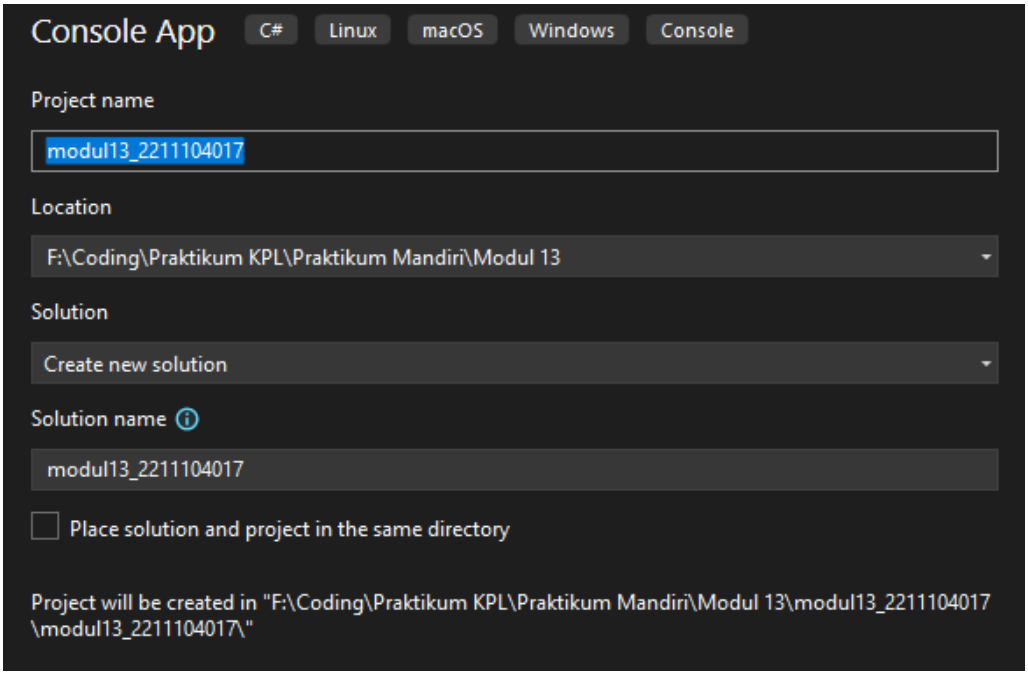
TUGAS JURNAL 13

1. MEMBUAT PROJECT GUI BARU

Buka IDE misalnya dengan Visual Studio

A. Misalnya menggunakan Visual Studio, buatlah project baru dengan nama modul113_NIM

B. Project yang dibuat bisa berupa console atau sejenisnya



2. MENJELASKAN SALAH SATU DESIGN PATTERN

Buka halaman web https://refactoring.guru/design-patterns/catalog kemudian baca design pattern dengan nama "Observer", dan jawab pertanyaan berikut ini (dalam Bahasa Indonesia):

A. Berikan salah DUA contoh kondisi dimana design pattern "Singleton" dapat digunakan?

Jawab:

- Pengaturan Konfigurasi Aplikasi

  Singleton dapat digunakan untuk menyimpan pengaturan konfigurasi aplikasi yang hanya perlu dimuat satu kali dan digunakan di seluruh aplikasi. Contoh: setting database connection atau pengaturan API.

- Pencatatan Log Aplikasi (Logging)

  Singleton digunakan untuk memastikan hanya ada satu instance logger yang bertanggung jawab mencatat log dari seluruh aplikasi, sehingga memudahkan dalam pemantauan dan debugging.

B. Berikan penjelasan singkat mengenai langkah-langkah dalam mengimplementasikan design pattern “Singleton” .

   Jawab:

   - Membuat class dengan konstruktor privat: konstruktor privat mencegah pembuatan instance dari luar class.
   - Membuat atribut statis untuk menyimpan instance: atribut ini menyimpan satu-satunya instance dari class.
   - Membuat method statis untuk mendapatkan instance: method ini akan memeriksa apakah instance sudah ada. Jika belum, instance baru akan dibuat dan disimpan di atribut statis tersebut.
   - Memastikan class tidak dapat diwarisi (optional): pada beberapa bahasa, class Singleton dapat dibuat sebagai sealed (final) untuk mencegah pewarisan.

C. Berikan kelebihan dan kekurangan dari design pattern “Singleton”

   Jawab:

   Kelebihan

   - Efisiensi memori: karena hanya ada satu instance yang digunakan, sehingga penggunaan memori lebih hemat.
   - Mudah diakses di seluruh aplikasi: instance dapat diakses dari mana saja tanpa perlu membuat objek baru.
   - Menghindari konflik data: karena hanya ada satu instance, data yang disimpan konsisten di seluruh aplikasi.

Kekurangan:

- Kesulitan dalam pengujian (testing): singleton sulit untuk diuji karena instance bersifat global dan tidak mudah direset.
- Dapat menjadi bottleneck: jika terlalu banyak operasi dijalankan pada Singleton, ini dapat menyebabkan kemacetan (bottleneck).
- Kurang fleksibel: tidak cocok jika aplikasi memerlukan beberapa instance dari class yang sama.

3. IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN SINGLETON

Buka halaman web berikut https://refactoring.guru/design-patterns/observer dan scroll ke bagian "Code Examples", pilih kode yang akan dilihat misalnya C# dan ikuti langkah-langkah berikut:

A. Dengan contoh yang sudah diberikan, buatlah sebuah class dengan design pattern singleton dengan nama "PusatDataSingleton".

B. Class "PusatDataSingleton" mempunyai dua atribut yaitu "DataTersimpan" yang mempunyai tipe berupa List<string> dan property singleton dengan nama "_instance" dengan tipe data "PusatDataSingleton" itu sendiri.

C. Class tersebut juga memiliki beberapa method yaitu:

- Konstruktor dari kelas tersebut yang mengisi atribut "DataTersimpan" dengan list kosong.
- GetDataSingleton() yang mengembalikan "_instance" jika tidak null dan memanggil konstruktor terlebih dahulu apabila nilainya masih null.
- GetSemuaData() yang mengembalikan list dari property "DataTersimpan".
- PrintSemuaData() yang melakukan print satu per satu dari string yang ada di list "DataTersimpan".
- AddSebuahData(string input) yang menambahkan satu data baru "input" ke dalam list "DataTersimpan".

- HapusSebuahData(int index) yang menghapus sebuah data berdasarkan index tertentu.

4. IMPLEMENTASI PROGRAM UTAMA

Tambahkan beberapa implementasi di program/method utama atau “main”:

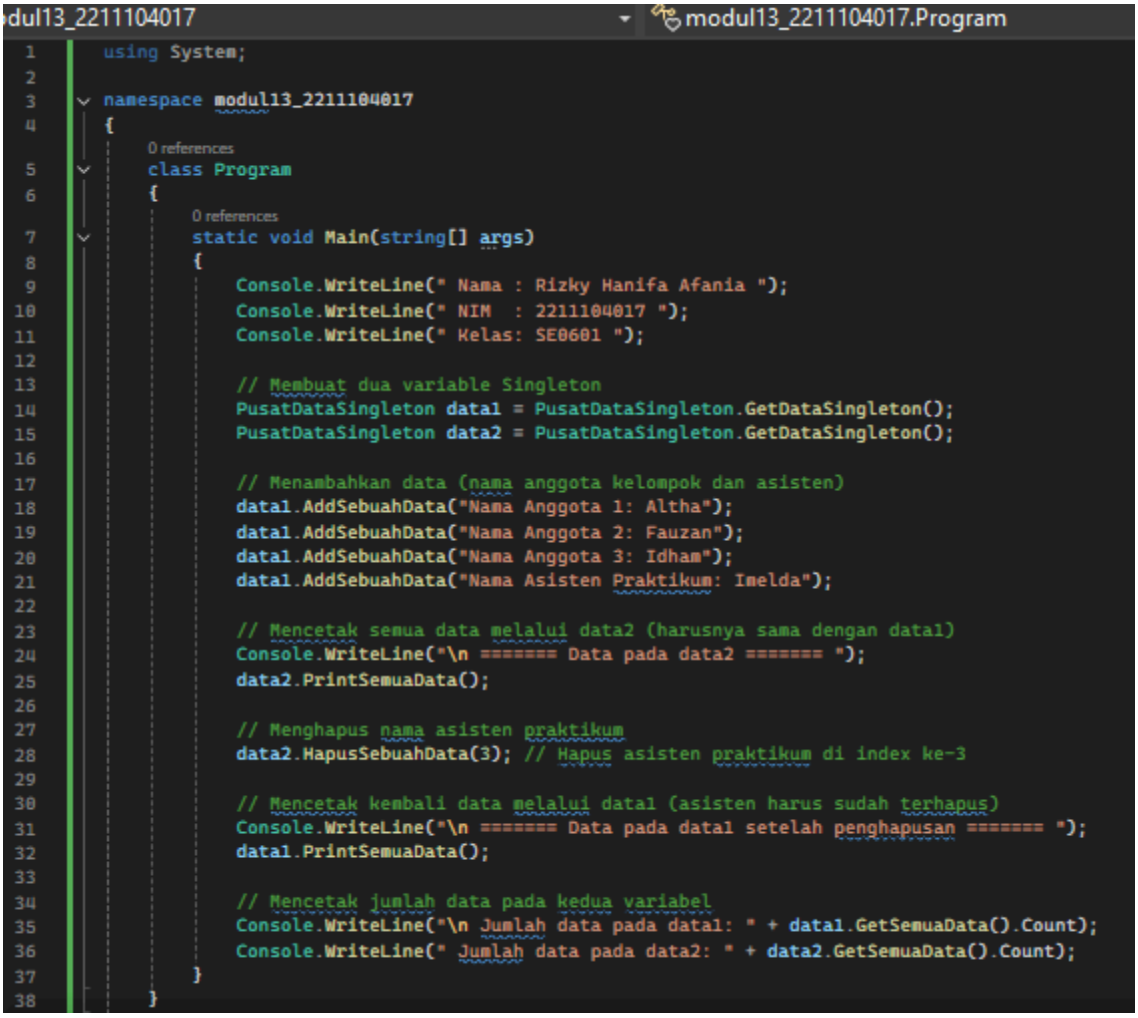
A. Buatlah dua variable dengan tipe “PusatDataSingleton” bernama data1 dan data2.
B. Isi kedua variable tersebut dengan hasil keluaran dari GetDataSingleton().
C. Pada data1 lakukan pemanggilan method AddSebuahData() beberapa kali dengan input nama anggota kelompok dan asisten praktikum.
D. Pada data2 panggil method PrintSemuaData(), pastikan keluaran dari hasil print data2 menampilkan nama-nama anggota kelompok dan asisten praktikum.
E. Pada data2 panggil HapusSebuahData() untuk menghapus nama asisten praktikum anda sekarang.
F. Pada data1 panggil PrintSemuaData(), dan seharusnya nama asisten praktikum anda tidak muncul di hasil print tersebut.
G. Langkah terakhir, pada data1 dan data2 panggil GetSemuaData() dan lakukan print dari jumlah “Count” atau elemen yang ada di list pada data1 dan data2.

Source Code:

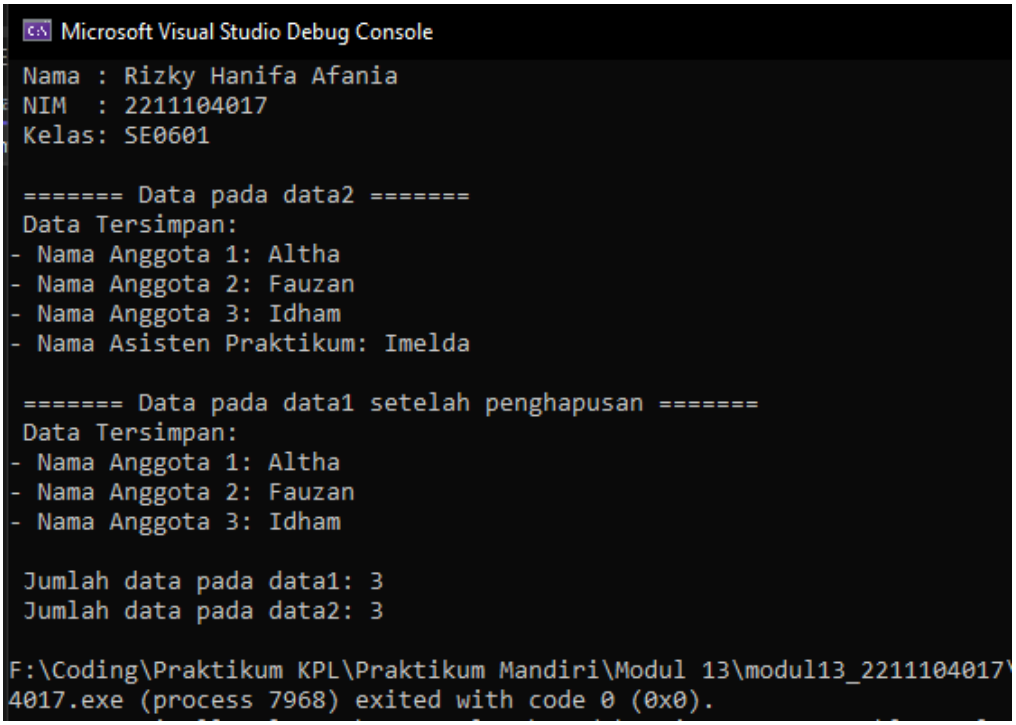
- File PusatDataSingleton.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace modul13_2211104017
{
    8 references
    public class PusatDataSingleton
    {
        // Atribut Singleton dan List
        private static PusatDataSingleton _instance;
        private List<string> DataTersimpan;

        // Konstruktor Private
        1 reference
        private PusatDataSingleton()
        {
            DataTersimpan = new List<string>();
        }

        // Method Singleton
        2 references
        public static PusatDataSingleton GetDataSingleton()
        {
            if (_instance == null)
            {
                _instance = new PusatDataSingleton();
            }
            return _instance;
        }

        // Method untuk mendapatkan semua data
        2 references
        public List<string> GetSemuaData()
        {
            return DataTersimpan;
        }

        // Method untuk mencetak semua data
        2 references
        public void PrintSemuaData()
        {
            Console.WriteLine(" Data Tersimpan:");
            foreach (string data in DataTersimpan)
            {
                Console.WriteLine("- " + data);
            }
        }

        // Method untuk menambahkan data
        4 references
        public void AddSebuahData(string input)
        {
            DataTersimpan.Add(input);
        }

        // Method untuk menghapus data berdasarkan index
        1 reference
        public void HapusSebuahData(int index)
        {
            if (index >= 0 && index < DataTersimpan.Count)
            {
                DataTersimpan.RemoveAt(index);
            }
            else
            {
                Console.WriteLine("Index tidak valid.");
            }
        }
    }
}
```

- File Program.cs



Hasil:



Penjelasan

Pada file PusatDataSingleton.cs, terdapat class bernama PusatDataSingleton yang menggunakan pola desain Singleton. Pola Singleton berguna untuk memastikan bahwa hanya ada satu instance dari class ini yang akan digunakan di seluruh program. Class ini memiliki atribut DataTersimpan berupa list string untuk menyimpan data, serta _instance yang menyimpan instance tunggal dari class. Metode GetDataSingleton() memastikan bahwa instance hanya akan dibuat satu kali, dan metode lainnya seperti AddSebuahData, HapusSebuahData, dan PrintSemuaData digunakan untuk menambahkan, menghapus, dan mencetak data dari list tersebut.

Pada file Program.cs, digunakan untuk menguji cara kerja Singleton dengan membuat dua variabel (data1 dan data2) yang sebenarnya adalah instance yang sama dari class PusatDataSingleton. Data ditambahkan melalui data1, tetapi karena kedua variabel mengacu pada instance yang sama, data tersebut juga terlihat di data2. Ketika data dihapus dari data2, perubahan juga terlihat pada data1, membuktikan bahwa kedua variabel tersebut adalah satu instance yang sama. Program ini kemudian mencetak jumlah data dari kedua variabel yang hasilnya selalu sama.