

TP Modul 13

Nama : Muhammad Shafiq Rasuna

Nim : 2311104043

Link Repo : https://github.com/rasunaaa/KPL_MUHAMMAD-SHAFIQ-RASUNA_2311104043_SE0702/tree/main/13_Design_Pattern_Implementation%2014

Class SayaTubeVideo

Class program

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

// Interface Observer
public interface IObservable
{
    void Update(string message);
}

// Interface Subject
public interface ISubject
{
    void Attach(IObservable observer);
    void Detach(IObservable observer);
    void Notify(string message);
}

// Concrete Subject
public class NewsAgency : ISubject
{
    private List<IObservable> observers = new List<IObservable>();

    public void Attach(IObservable observer) => observers.Add(observer);
    public void Detach(IObservable observer) => observers.Remove(observer);

    public void Notify(string message)
    {
        foreach (var observer in observers)
            observer.Update(message);
    }

    // Trigger perubahan
    public void AddNews(string news)
    {
        Console.WriteLine("News Added: " + news);
        Notify(news);
    }
}
```

```

}

// Concrete Observer
public class NewsReader : IObservable
{
    private string _name;

    public NewsReader(string name)
    {
        _name = name;
    }

    public void Update(string message)
    {
        Console.WriteLine($"{_name} received news: {message}");
    }
}

// Main Program
class Program
{
    static void Main()
    {
        // Membuat subject
        var agency = new NewsAgency();

        // Membuat observers
        var reader1 = new NewsReader("Shafiq");
        var reader2 = new NewsReader("Alya");

        // Attach observers
        agency.Attach(reader1);
        agency.Attach(reader2);

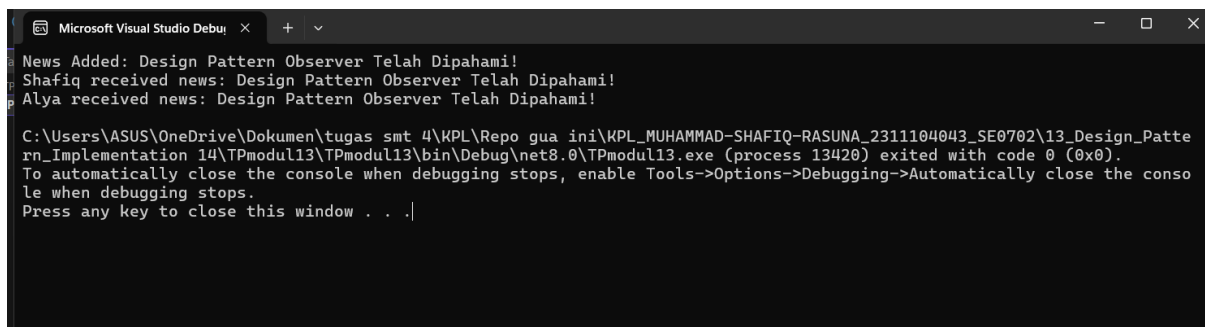
        // Tambahkan berita
        agency.AddNews("Design Pattern Observer Telah Dipahami!");
    }
}

```

Observer Pattern adalah salah satu design pattern perilaku (behavioral) yang digunakan untuk menciptakan hubungan satu-ke-banyak antar objek, sehingga ketika satu objek (disebut *subject*) mengalami perubahan status, maka seluruh objek yang bergantung padanya (disebut *observers*) akan diberi notifikasi secara otomatis. Contoh nyata dari penerapan pattern ini adalah pada sistem notifikasi aplikasi cuaca, di mana ketika data cuaca berubah di server pusat (subject), maka tampilan di perangkat pengguna (observers) akan langsung diperbarui. Implementasi dari Observer Pattern biasanya dimulai dengan membuat antarmuka *ISubject* yang berisi metode untuk menambah, menghapus, dan memberi notifikasi ke observer (*Attach()*, *Detach()*, dan *Notify()*). Selanjutnya, dibuat antarmuka *IObservable* yang berisi metode *Update()* untuk menerima data dari subject. Kemudian, kelas konkret seperti

NewsAgency mengimplementasikan ISubject, menyimpan daftar observer, dan memanggil Update() saat data berubah. Kelas seperti NewsReader berperan sebagai observer yang menanggapi notifikasi dengan menampilkan pesan atau melakukan tindakan tertentu. Dalam fungsi Main(), kita membuat objek subject dan beberapa observer, lalu menghubungkannya dengan Attach(), dan memicu perubahan melalui AddNews(). Kelebihan dari pola ini adalah kemampuannya menciptakan struktur program yang longgar (loose coupling), fleksibel, dan mudah diperluas. Namun, kelemahannya adalah potensi kompleksitas ketika jumlah observer sangat banyak dan kesulitan dalam melacak urutan notifikasi. Dengan memahami dan menerapkan Observer Pattern, pengembang dapat membangun sistem yang reaktif dan dinamis dalam merespons perubahan data secara otomatis.

Hasil output nya :



```
Microsoft Visual Studio Debug Console
News Added: Design Pattern Observer Telah Dipahami!
Shafiq received news: Design Pattern Observer Telah Dipahami!
Alya received news: Design Pattern Observer Telah Dipahami!

C:\Users\ASUS\OneDrive\Dokumen\tugas smt 4\KPL\Repo gua ini\KPL_MUHAMMAD-SHAFIQ-RASUNA_2311104043_SE0702\13_Design_Pattern_Implementation 14\TPmodul13\TPmodul13\bin\Debug\net8.0\TPmodul13.exe (process 13420) 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 . . .|
```