Tugas Pendahuluan Modul 13
Konstruksi Perangkat Lunak
PRODI REKAYASA PERANGKAT LUNAK

Nama :
Reyner Atira Prasetyo
2311104057
S1SE-07-02

DIREKTORAT TELKOM UNIVERSITY
PURWOKERTO

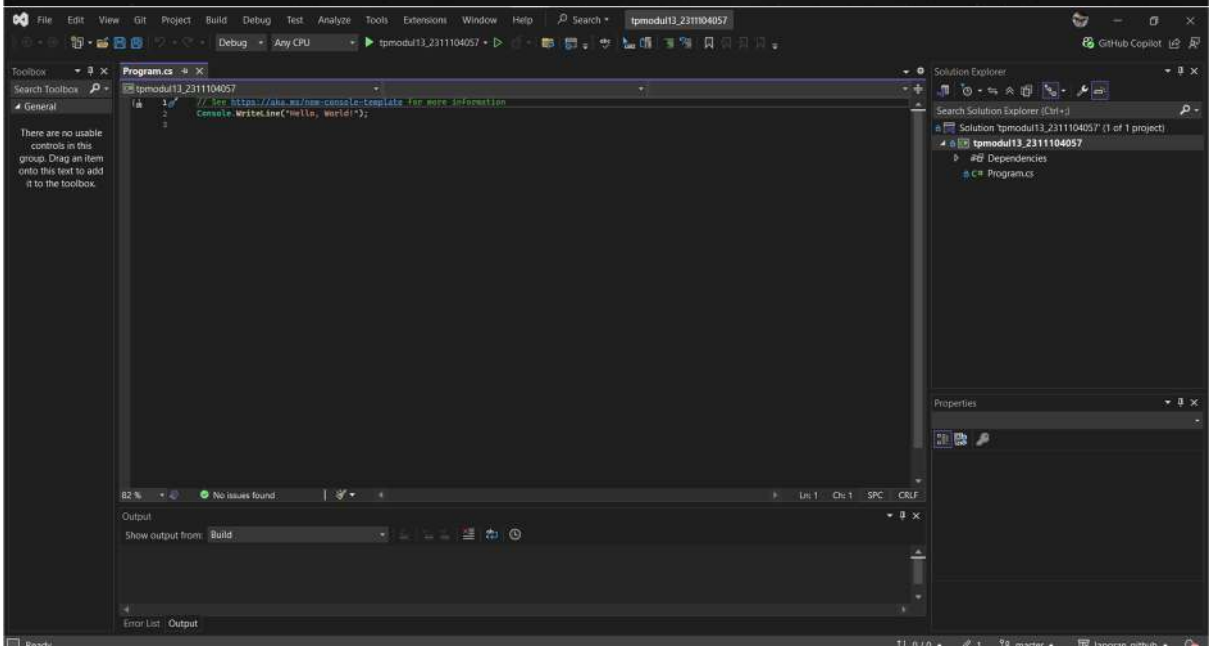
Tugas Pendahuluan Modul 13

1. Initialize project dan membuat GUI

```
D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±2 (0.103s)
git add 13_Design_Pattern_Implementation/tp\


D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±4 (0.15s)
git commit -m "tp 13: initialize project"
[master 47a0312] tp 13: initialize project
 3 files changed, 37 insertions(+)
 create mode 100644 13_Design_Pattern_Implementation/tp/tpmodul13_2311104057/tpmodul13_2311104057.sln
 create mode 100644 13_Design_Pattern_Implementation/tp/tpmodul13_2311104057/tpmodul13_2311104057/Program.cs
 create mode 100644 13_Design_Pattern_Implementation/tp/tpmodul13_2311104057/tpmodul13_2311104057/tpmodul13_2311104057.csproj


D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±1 (2.369s)
git push
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (20/20), 3.42 KiB | 1.14 MiB/s, done.
Total 20 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/reyneratira/KPL_REYNER-ATIRA-PRASETYO_2311104057_S1SE-07-02.git
   35c050f..47a0312  master -> master
```



2. Penjelasan Observer

A. Contoh penggunaan Observer Pattern

Observer pattern dapat digunakan dalam aplikasi cuaca, di mana objek WeatherStation (Subject) mengirimkan pembaruan data suhu/kelembaban ke Display (Observer) setiap kali ada perubahan.

B. Langkah implementasi Observer berdasarkan kasus di atas

1. Buat Interface Subject (ISubject): Memiliki method Attach, Detach, dan Notify
2. Buat Interface Observer (IObserver): Memiliki method Update()
3. Implementasi ConcreteSubject: Menyimpan state penting dan list observer. Bila state berubah, panggil Notify()
4. Implementasi ConcreteObserver: Menerapkan Update() untuk merespon perubahan dari Subject
5. Main Program: Hubungkan observer ke subject dan ubah state subject

C. Kelebihan dan Kekurangan

Kelebihan

1. Loose coupling (pengamat dan subject tidak saling tergantung)
2. Dapat menambahkan observer tanpa mengubah subject
3. Mendukung event-driven system

Kekurangan

1. Sulit men-debug saat banyak observer
2. Risiko notifikasi berulang jika tidak hati-hati
3. Urutan update observer tidak dijamin

3. Implementasi Design Pattern Observer

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

namespace ObserverPattern_WeatherStation
{
    // Observer interface
    public interface IObserver
    {
        void Update(float temperature);
    }

    // Subject interface
    public interface ISubject
    {
        void Attach(IObserver observer);
        void Detach(IObserver observer);
        void Notify();
    }

    // Concrete Subject
    public class WeatherStation : ISubject
    {
```

```
    private List<IObserver> observers = new List<IObserver>();
    private float temperature;

    public void Attach(IObserver observer)
    {
        observers.Add(observer);
    }

    public void Detach(IObserver observer)
    {
        observers.Remove(observer);
    }

    public void SetTemperature(float temp)
    {
        Console.WriteLine($"\n[WeatherStation] Suhu berubah menjadi {temp}°C");
        temperature = temp;
        Notify();
    }

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

// Concrete Observer 1
```

```
public class PhoneDisplay : IObserver
{
    public void Update(float temperature)
    {
        Console.WriteLine($"[PhoneDisplay] Menampilkan suhu: {temperature}°C");
    }
}

// Concrete Observer 2
public class WindowDisplay : IObserver
{
    public void Update(float temperature)
    {
        Console.WriteLine($"[WindowDisplay] Update suhu jendela: {temperature}°C");
    }
}

class Program
{
    static void Main(string[] args)
    {
        // Membuat subject
        WeatherStation station = new WeatherStation();

        // Membuat observer
        PhoneDisplay phoneDisplay = new PhoneDisplay();
        WindowDisplay windowDisplay = new WindowDisplay();

        // Menambahkan observer ke subject
        station.Attach(phoneDisplay);
```

```
            station.Attach(windowDisplay);

            // Simulasi perubahan suhu
            station.SetTemperature(25.5f);
            station.SetTemperature(30.0f);

            // Lepaskan salah satu observer
            station.Detach(windowDisplay);
            Console.WriteLine("\n[INFO] WindowDisplay dilepas dari observer list.\n");

            // Simulasi perubahan suhu lagi
            station.SetTemperature(28.0f);

            Console.WriteLine("\nTekan sembarang tombol untuk keluar...");
            Console.ReadKey();
        }
    }
}
```

Penjelasan Singkat:

1. WeatherStation = Subject yang menyimpan suhu dan memberitahu semua observer jika suhu berubah
2. PhoneDisplay dan WindowDisplay = Observer yang merespon ketika suhu berubah
3. Observer dapat dilepas/dipasang kapan pun dengan Attach / Detach

4. Implementasi di main Program

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Membuat subject
        WeatherStation station = new WeatherStation();

        // Membuat observer
        PhoneDisplay phoneDisplay = new PhoneDisplay();
        WindowDisplay windowDisplay = new WindowDisplay();

        // Menambahkan observer ke subject
        station.Attach(phoneDisplay);
        station.Attach(windowDisplay);

        // Simulasi perubahan suhu
        station.SetTemperature(25.5f);
        station.SetTemperature(30.0f);

        // Lepaskan salah satu observer
        station.Detach(windowDisplay);
        Console.WriteLine("\n[INFO] WindowDisplay dilepas dari observer list.\n");

        // Simulasi perubahan suhu lagi
        station.SetTemperature(28.0f);
        station.SetTemperature(22.5f);
        Console.ReadKey();
    }
}
```

```
Microsoft Visual Studio Debug Console

[WeatherStation] Suhu berubah menjadi 25,5°C
[PhoneDisplay] Menampilkan suhu: 25,5°C
[WindowDisplay] Update suhu jendela: 25,5°C

[WeatherStation] Suhu berubah menjadi 30°C
[PhoneDisplay] Menampilkan suhu: 30°C
[WindowDisplay] Update suhu jendela: 30°C

[INFO] WindowDisplay dilepas dari observer list.


[WeatherStation] Suhu berubah menjadi 28°C
[PhoneDisplay] Menampilkan suhu: 28°C

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github\13_Design_Pattern_Implementation\tp\tpmodul13_2311104057\tpmodul13_231
1104057\bin\Debug\net8.0\tpmodul13_2311104057.exe (process 248) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

PUSH

```
D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±2 (0.105s)
git add 13_Design_Pattern_Implementation/tp\

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±2 (0.164s)
git commit -m "tp 13: implementasi design pattern observer"
[master 82907fe] tp 13: implementasi design pattern observer
 1 file changed, 99 insertions(+), 2 deletions(-)

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±1 (2.509s)
git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.23 KiB | 631.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/reyneratira/KPL_REYNER-ATIRA-PRASETYO_2311104057_S1SE-07-02.git
   47a0312..82907fe  master -> master
```