

**TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK**

MODUL 13

Design_Pattern_Implementation



Disusun Oleh :

Ahmad Junaidi / 2211104002

SE-06-01

Asisten Praktikum:

Naufal El Kamil Aditya Pratama Rahman

Imelda

Dosen Pengampu :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

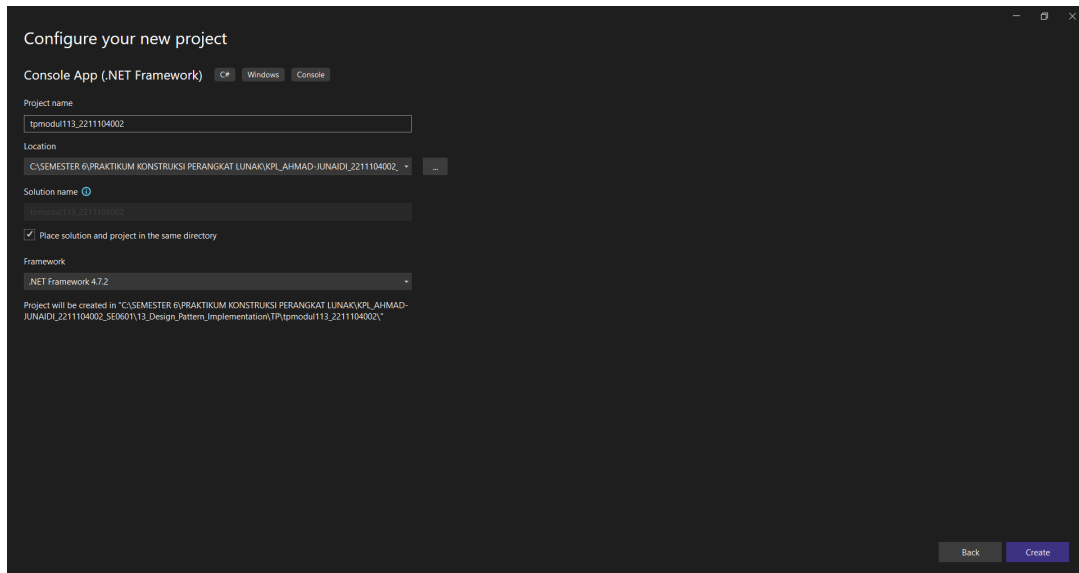
2025

TUGAS PENDAHULUAN

1. Link Github Repository

https://github.com/Ahmadjunaidi101105/KPL_AHMAD-JUNAIDI_2211104002_SE0601/tree/main/10_Library_Construction

2. MEMBUAT PROJECT GUI BARU



3. 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):

- Berikan salah satu contoh kondisi dimana design pattern “Observer” dapat digunakan

Jawab: Design pattern *Observer* cocok digunakan ketika ada satu objek (subject) yang statusnya diamati oleh banyak objek lainnya (observer). Misalnya, aplikasi berita: saat ada berita baru, semua subscriber (observer) akan menerima notifikasi.

- Berikan penjelasan singkat mengenai langkah-langkah dalam mengimplementasikan design pattern Observer”

jawab:

- Buat interface *IObserver* (observer) dan *ISubject* (publisher).
- Class yang menjadi observer harus mengimplementasikan *IObserver*.
- Class yang diamati mengimplementasikan *ISubject* dan menyimpan daftar observer.

- Saat ada perubahan data di subject, ia akan *notify* semua observer untuk memperbarui diri.
- c. Berikan kelebihan dan kekurangan dari design pattern “Observer”

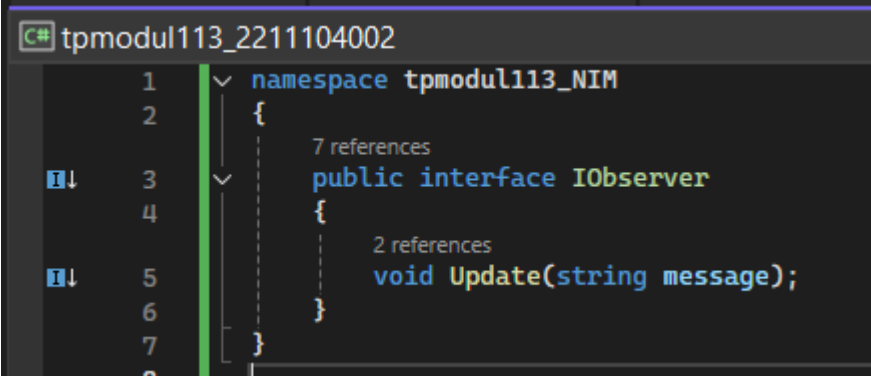
jawab:

- Kelebihan:
 - Memisahkan antara data dan tampilan/penerima.
 - Memudahkan sinkronisasi data ke banyak bagian.
- Kekurangan:
 - Bisa menjadi rumit jika terlalu banyak observer.
 - Sulit dilacak saat ada bug karena notifikasi terjadi secara tidak langsung.

4. IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN OBSERVER

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. Pada project yang telah dibuat sebelumnya, tambahkan kode yang mirip atau sama dengan contoh kode yang diberikan di halaman web tersebut
 - IObserver.cs



```
1 namespace tpmodul113_NIM
2 {
3     7 references
4     public interface IObserver
5     {
6         2 references
7         void Update(string message);
8     }
9 }
```

- ISubject.cs

```

1  using System.Collections.Generic;
2
3  namespace tpmodul113_NIM
4  {
5      1 reference
6      public interface ISubject
7      {
8          3 references
9          void Attach(IObserver observer);
10         1 reference
11         void Detach(IObserver observer);
12         2 references
13         void Notify();
14     }
15 }

```

- NewsAgency.cs (Concrete Subject)

```

tpmodul113_2211104002
1  using System.Collections.Generic;
2
3  namespace tpmodul113_NIM
4  {
5      1 reference
6      public class NewsAgency : ISubject
7      {
8          private List<IObserver> observers = new List<IObserver>();
9          private string news;
10
11         2 references
12         public void SetNews(string value)
13         {
14             news = value;
15             Notify();
16         }
17
18         3 references
19         public void Attach(IObserver observer)
20         {
21             observers.Add(observer);
22         }
23
24         1 reference
25         public void Detach(IObserver observer)
26         {
27             observers.Remove(observer);
28         }
29
30         2 references
31         public void Notify()
32         {
33             foreach (var observer in observers)
34             {
35                 observer.Update(news);
36             }
37         }
38     }
39 }

```

- Subscriber.cs (Concrete Observer)

```

1 using System;
2
3 namespace tpmodul113_NIM
4 {
5     3 references
6     public class Subscriber : IObservable
7     {
8         private string name;
9
10        2 references
11        public Subscriber(string name)
12        {
13            this.name = name;
14        }
15
16        2 references
17        public void Update(string message)
18        {
19            Console.WriteLine($"{name} menerima berita: {message}");
20        }
21    }
22 }

```

- Program.cs

```

1 using System;
2
3 namespace tpmodul113_NIM
4 {
5     0 references
6     class Program
7     {
8         0 references
9         static void Main(string[] args)
10        {
11            var agency = new NewsAgency();
12
13            var sub1 = new Subscriber("Ahmad");
14            var sub2 = new Subscriber("Junaidi");
15
16            agency.Attach(sub1);
17            agency.Attach(sub2);
18
19            agency.SetNews("Cuaca hari ini cerah.");
20            agency.SetNews("Harga BBM naik.");
21        }
22    }
23 }

```

- b. Jalankan program tersebut dan pastikan tidak ada error pada saat project dijalankan

```

Microsoft Visual Studio Debug
Ahmad menerima berita: Cuaca hari ini cerah.
Junaidi menerima berita: Cuaca hari ini cerah.
Ahmad menerima berita: Harga BBM naik.
Junaidi menerima berita: Harga BBM naik.

C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\WPL_AHMAD-JUNAIDI_2211104002_SE0601\13_Design_Pattern_Implementation\
TP\tpmodul113_2211104002\bin\Debug\tpmodul113_2211104002.exe (process 18500) 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 . . .

```

```
1 using System;
2
3 namespace tpmodul113_NIM
4 {
5     class Program
6     {
7         static void Main(string[] args)
8         {
9             var agency = new NewsAgency();
10
11             var sub1 = new Subscriber("Ahmad");
12             var sub2 = new Subscriber("Junaidi");
13
14             agency.Attach(sub1);
15             agency.Attach(sub2);
16
17             agency.SetNews("Cuaca hari ini cerah.");
18             agency.SetNews("Harga BBM naik.");
19         }
20     }
21 }
22
```

Output

Show output from: Debug

'tpmodul113_2211184002.exe' (CLR v4.0.30319: DefaultDomain): Loaded 'C:\WINDOWS\Microsoft.Net\assembly\GAC_32\mscorlib\v4.0.0.0_b77a5c561934e889\mscorlib.dll'. Skipped loa

'tpmodul113_2211184002.exe' (CLR v4.0.30319: DefaultDomain): Loaded 'C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT LUNAK\KPL_AHMAD-JUNAIDI_2211184002_SE06081\13_Design_Pattern_Im

The program '[18500] tpmodul113_2211184002.exe' has exited with code 0 (0x0).

c. Jelaskan tiap baris kode yang terdapat di bagian method utama atau “main”

Berikut adalah penjelasan tiap baris kode dalam method Main pada file Program.cs untuk implementasi design pattern Observer:

- `var agency = new NewsAgency();` ➡ Membuat objek agency dari kelas NewsAgency, yang berperan sebagai subject (publisher) dalam pattern Observer. Objek ini akan menyimpan data berita dan memberi tahu observer saat berita berubah.
- `var sub1 = new Subscriber("Ahmad");` ➡ Membuat observer pertama (sub1) dengan nama "Ahmad". Objek ini akan menerima notifikasi dari agency ketika berita berubah.
- `var sub2 = new Subscriber("Junaidi");` ➡ Membuat observer kedua (sub2) dengan nama "Junaidi".
- `agency.Attach(sub1); agency.Attach(sub2);` ➡ Menambahkan sub1 dan sub2 ke daftar observer milik agency. Artinya, kedua subscriber ini mendaftar sebagai penerima notifikasi.
- `agency.SetNews("Cuaca hari ini cerah.");` ➡ Mengubah isi berita di agency menjadi "Cuaca hari ini cerah." Saat method SetNews() dipanggil, method Notify() juga dipanggil secara otomatis, sehingga kedua observer (Ahmad dan Junaidi) menerima berita tersebut.
- `agency.SetNews("Harga BBM naik.");` ➡ Mengubah berita lagi menjadi "Harga BBM naik.", dan men-trigger notifikasi ulang ke semua observer yang masih aktif/terdaftar.