

## TP Modul 14

Nama : Kafka Putra Riyadi

Kelas : SE 07-02

NIM : 2311104041

### 1. Source code class program.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Threading;
4
5 namespace RefactoringGuru.DesignPatterns.Observer.Refactored
6 {
7     public interface IObservable
8     {
9         void Update(IObservable observable);
10     }
11
12     public interface ISubject
13     {
14         void Attach(IObservable observable);
15         void Detach(IObservable observable);
16         void Notify();
17     }
18
19     public class Subject : ISubject
20     {
21         public int State { get; set; } = 0;
22
23         private readonly List<IObservable> _observers = new();
24
25         public void Attach(IObservable observable)
26         {
27             Console.WriteLine("Subject: Attached an observer.");
28             _observers.Add(observable);
29         }
30
31         public void Detach(IObservable observable)
32         {
33             _observers.Remove(observable);
34             Console.WriteLine("Subject: Detached an observer.");
35         }
36
37         public void Notify()
38         {
39             Console.WriteLine("Subject: Notifying observers...");
40             foreach (var observer in _observers)
41             {
42                 observer.Update(this);
43             }
44         }
45
46         public void SomeBusinessLogic()
47         {
48             Console.WriteLine($"Subject: I'm doing something important.");
49             State = new Random().Next(0, 10);
50
51             Thread.Sleep(15);
52
53             Console.WriteLine($"Subject: My state has just changed to: {State}");
54             Notify();
55         }
56     }
57
58     public class ConcreteObserverA : IObservable
59     {
60         public void Update(IObservable observable)
61         {
62             if ((observable as Subject)?.State < 5)
63             {
64                 Console.WriteLine("ConcreteObserverA: Reacted to the event.");
65             }
66         }
67     }
68
69     public class ConcreteObserverB : IObservable
70     {
71         public void Update(IObservable observable)
72         {
73             var state = (observable as Subject)?.State;
74             if (state == 0 || state >= 2)
75             {
76                 Console.WriteLine("ConcreteObserverB: Reacted to the event.");
77             }
78         }
79     }
80
81     public class Program
82     {
83         public static void Main(string[] args)
84         {
85             var subject = new Subject();
86
87             var observerA = new ConcreteObserverA();
88             subject.Attach(observerA);
89
90             var observerB = new ConcreteObserverB();
91             subject.Attach(observerB);
92
93             subject.SomeBusinessLogic();
94             subject.SomeBusinessLogic();
95
96             subject.Detach(observerB);
97
98             subject.SomeBusinessLogic();
99         }
100     }
101
102 }
```

Penjelasan :

Code dari class program.cs merupakan implementasi dari Design Pattern Observer dalam bahasa C#, yang digunakan untuk memungkinkan suatu objek (Subject) memberi tahu sejumlah objek lain (Observer) ketika terjadi perubahan pada dirinya. ISubject adalah antarmuka yang mendefinisikan kontrak untuk objek yang dapat diawasi, dengan metode Attach(), Detach(), dan Notify(). Subject mengimplementasikan antarmuka ini, menyimpan state (State) dan daftar observer yang terdaftar. Setiap kali method SomeBusinessLogic() dijalankan, State akan berubah secara acak dan memicu pemanggilan Notify() untuk memberitahu semua observer. Kelas ConcreteObserverA dan ConcreteObserverB masing-masing mengimplementasikan antarmuka IObserver dan bereaksi terhadap perubahan State berdasarkan kondisi tertentu. Dalam Main(), dua observer didaftarkan ke Subject, lalu Subject menjalankan logikanya beberapa kali sehingga observer dapat memberikan respons secara otomatis saat perubahan terjadi. Pola ini mencerminkan prinsip pemisahan kekhawatiran (separation of concerns) dan mendukung arsitektur yang fleksibel dan mudah diperluas.

Outputannya :

```
Subject: Attached an observer.  
Subject: Attached an observer.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 8  
Subject: Notifying observers...  
ConcreteObserverB: Reacted to the event.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 8  
Subject: Notifying observers...  
ConcreteObserverB: Reacted to the event.  
Subject: Detached an observer.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 7  
Subject: Notifying observers...
```