

TP 13 KONTRUKSI PERANGKAT LUNAK

Nama : Rifqi M Ramdani

NIM : 2311104044

Kelas : SE-07-02

LINK GITHUB :

https://github.com/Ramdaniiii/KPL_RifqiMRamdani_2311104044_SE0702.git

MENJELASKAN SALAH SATU DESIGN PATTERN

Observer pattern cocok digunakan pada **aplikasi berita atau media sosial**, di mana pengguna akan menerima notifikasi saat ada update baru dari akun atau topik yang mereka ikuti. Misalnya, saat akun Instagram yang kita follow mengunggah foto baru, kita langsung mendapat notifikasi.

Jawba

1. Buat interface Observer dengan method Update().
2. Buat interface Subject dengan method Attach(), Detach(), dan Notify().
3. Implementasikan ConcreteSubject yang menyimpan data dan daftar observer.
4. Implementasikan ConcreteObserver yang menerima notifikasi dan bereaksi.
5. Di program utama, buat objek subject dan observer, lalu daftarkan observer ke subject.

Kelebihan

- Mengurangi keterikatan antar objek
- Mudah menambahkan observer baru
- Reaktif terhadap perubahan secara otomatis

Kekurangan

- Sulit melacak siapa yang menerima notifikasi
- Bisa berdampak pada performa jika terlalu banyak observer
- Debugging bisa lebih rumit

Implementasi kode

```
Program.cs
C# tpm modul113_2311104044 Subscriber

1 using System;
2 using System.Collections.Generic;
3
4 // Interface untuk observer yang menerima update berita
5 public interface IObserver
6 {
7     void Update(string message);
8 }
9
10 // Interface untuk subject yang mengelola observer
11 public interface ISubject
12 {
13     void Attach(IObserver observer);
14     void Detach(IObserver observer);
15     void Notify();
16 }
17
18 // Implementasi subject: NewsAgency
19 public class NewsAgency : ISubject
20 {
21     // Daftar observer yang terdaftar
22     private readonly List<IObserver> observers = new List<IObserver>();
23
24     // Berita terbaru yang akan dikirim ke observer
25     private string latestNews;
26
27     public void Attach(IObserver observer)
28     {
29         if (observer == null) throw new ArgumentNullException(nameof(observer));
30         if (!observers.Contains(observer))
31             observers.Add(observer);
32     }
33
34     public void Detach(IObserver observer)
35     {
36         if (observer == null) throw new ArgumentNullException(nameof(observer));
37         observers.Remove(observer);
38     }
39
40     public void Notify()
41     {
42         // Jika tidak ada berita, tidak perlu notify
43         if (string.IsNullOrEmpty(latestNews)) return;
44
45         foreach (var observer in observers)
46         {
47             observer.Update(latestNews);
48         }
49     }
50
51     // Menambahkan berita baru dan notify observer
```

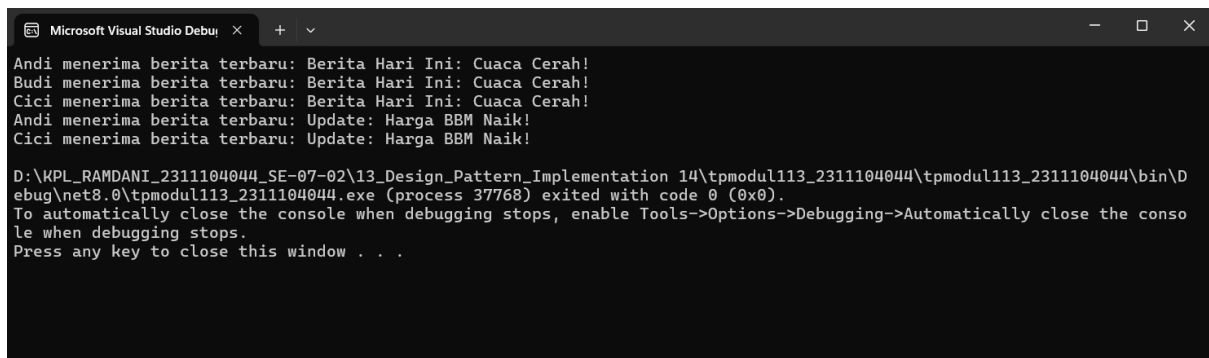
```

51 // Menambahkan berita baru dan notify observer
52 public void AddNews(string news)
53 {
54     if (string.IsNullOrEmpty(news)) return;
55
56     latestNews = news;
57     Notify();
58 }
59
60
61 // Implementasi observer: Subscriber
62 public class Subscriber : IObservable
63 {
64     private readonly string name;
65
66     public Subscriber(string name)
67     {
68         if (string.IsNullOrEmpty(name)) throw new ArgumentException("Name cannot be empty", nameof(name));
69         this.name = name;
70     }
71
72     public void Update(string message)
73     {
74         Console.WriteLine($"{name} menerima berita terbaru: {message}");
75     }
76 }
77
78 // Program utama untuk menjalankan demo
79 class Program
80 {
81     static void Main()
82     {
83         var agency = new NewsAgency();
84
85         var sub1 = new Subscriber("Andi");
86         var sub2 = new Subscriber("Budi");
87         var sub3 = new Subscriber("Cici");
88
89         agency.Attach(sub1);
90         agency.Attach(sub2);
91         agency.Attach(sub3);
92
93         agency.AddNews("Berita Hari Ini: Cuaca Cerah!");
94
95         agency.Detach(sub2);
96
97         agency.AddNews("Update: Harga BBM Naik!");
98     }
99 }
100

```

mengimplementasikan pola desain Observer, di mana NewsAgency bertindak sebagai subject yang mengelola daftar observer (Subscriber). Setiap kali berita baru ditambahkan melalui metode AddNews, NewsAgency akan memberitahu semua subscriber yang terdaftar dengan memanggil metode Update mereka, sehingga para subscriber dapat menerima dan menampilkan berita terbaru. Attach dan Detach memungkinkan penambahan dan penghapusan subscriber dari daftar pengamat secara dinamis. Pada program utama, tiga subscriber dibuat dan didaftarkan ke NewsAgency, kemudian mereka menerima update berita pertama, setelah itu satu subscriber dilepas dan update kedua hanya diterima oleh dua subscriber yang tersisa.

Maka Outputnya

A screenshot of the Microsoft Visual Studio Debug Console window. The window has a dark background and a title bar that reads "Microsoft Visual Studio Debug Console". The output text is as follows:

```
Andi menerima berita terbaru: Berita Hari Ini: Cuaca Cerah!  
Budi menerima berita terbaru: Berita Hari Ini: Cuaca Cerah!  
Cici menerima berita terbaru: Berita Hari Ini: Cuaca Cerah!  
Andi menerima berita terbaru: Update: Harga BBM Naik!  
Cici menerima berita terbaru: Update: Harga BBM Naik!  
  
D:\KPL_RAMDANI_2311104044_SE-07-02\13_Design_Pattern_Implementation 14\tpmodul113_2311104044\tpmodul113_2311104044\bin\Debug\net8.0\tpmodul113_2311104044.exe (process 37768) 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 . . .
```