

TP Modul 13

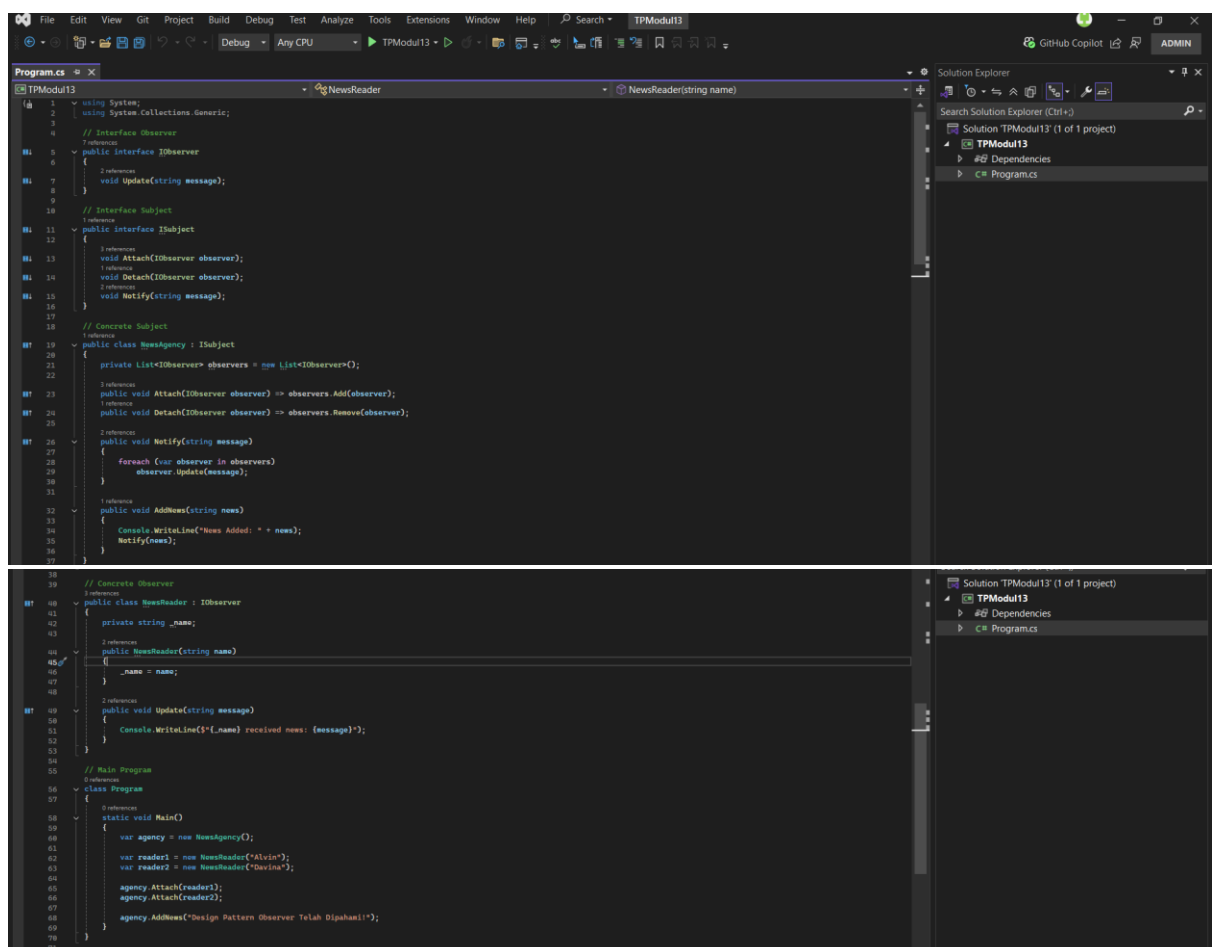
Nama : Alvin Bagus Firmansyah

Kelas : SE 07 02

NIM : 2311104070

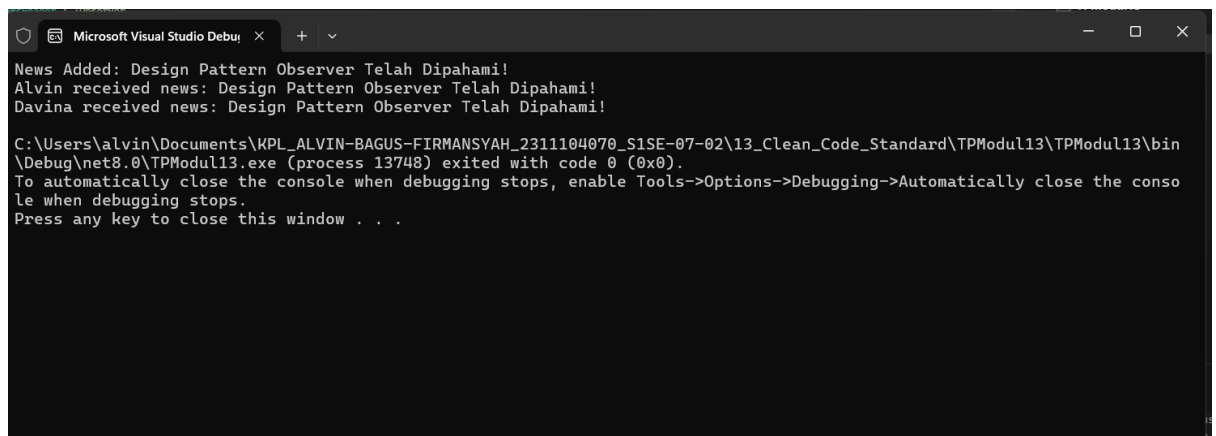
1. Class Program.cs

Program ini menerapkan **Observer Design Pattern** yang terdiri dari dua peran utama, yaitu *Subject* dan *Observer*. Pada program ini, NewsAgency berperan sebagai *Subject* yang bertanggung jawab menyimpan daftar observer dan memberitahu mereka jika ada informasi baru. Dua interface digunakan untuk mendefinisikan struktur komunikasi, yaitu *ISubject* untuk subject dan *IObserver* untuk observer. Kelas NewsReader merupakan implementasi dari observer yang menerima notifikasi berita terbaru. Dalam method Main, objek NewsAgency dibuat lalu dua observer yaitu "Alvin" dan "Davina" ditambahkan. Ketika berita baru ditambahkan menggunakan method AddNews, semua observer yang terdaftar akan otomatis menerima dan mencetak pesan berita tersebut. Pola ini sangat berguna untuk sistem yang membutuhkan sinkronisasi data secara real-time tanpa ketergantungan langsung antar objek.



```
1 using System;
2 using System.Collections.Generic;
3
4 // Interface Observer
5 public interface IObserver
6 {
7     void Update(string message);
8 }
9
10 // Interface Subject
11 public interface ISubject
12 {
13     void Attach(IObserver observer);
14     void Detach(IObserver observer);
15     void Notify(string message);
16 }
17
18 // Concrete Subject
19 public class NewsAgency : ISubject
20 {
21     private List<IObserver> observers = new List<IObserver>();
22
23     public void Attach(IObserver observer) => observers.Add(observer);
24     public void Detach(IObserver observer) => observers.Remove(observer);
25
26     public void Notify(string message)
27     {
28         foreach (var observer in observers)
29             observer.Update(message);
30     }
31
32     public void AddNews(string news)
33     {
34         Console.WriteLine("News Added: " + news);
35         Notify(news);
36     }
37 }
38
39 // Concrete Observer
40 public class NewsReader : IObserver
41 {
42     private string _name;
43
44     public NewsReader(string name)
45     {
46         _name = name;
47     }
48
49     public void Update(string message)
50     {
51         Console.WriteLine($"{_name} received news: {message}");
52     }
53 }
54
55 // Main Program
56 class Program
57 {
58     static void Main()
59     {
60         var agency = new NewsAgency();
61
62         var reader1 = new NewsReader("Alvin");
63         var reader2 = new NewsReader("Davina");
64
65         agency.Attach(reader1);
66         agency.Attach(reader2);
67
68         agency.AddNews("Design Pattern Observer Telah Dipahami!");
69     }
70 }
71
```

Hasil Outputnya :



The image shows a screenshot of the 'Microsoft Visual Studio Debug Console' window. The window has a title bar with the text 'Microsoft Visual Studio Debug Console' and standard window controls (minimize, maximize, close). The console output is as follows:

```
News Added: Design Pattern Observer Telah Dipahami!  
Alvin received news: Design Pattern Observer Telah Dipahami!  
Davina received news: Design Pattern Observer Telah Dipahami!  
  
C:\Users\alvin\Documents\KPL_ALVIN-BAGUS-FIRMANSYAH_2311104070_SISE-07-02\13_Clean_Code_Standard\TPModul13\TPModul13\bin  
\Debug\net8.0\TPModul13.exe (process 13748) 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 . . .
```