

MODUL 14 KONTRUKSI PERANGKAT LUNAK

Nama: Rifqi M Ramdani

NIM : 2311104044

Kelas : SE-07-02

Link Github:

Sebelum direfactoring

```
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 }
```

```

49     }
50
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 : IObserver
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

```

Sesudah direfactoring

```
1  using System;
2  using System.Collections.Generic;
3
4  // Interface untuk observer yang menerima update berita
5
6  public interface IObserver
7  {
8      void Update(string message);
9  }
10 // Interface untuk subject yang mengelola observer
11
12 public interface ISubject
13 {
14     void Attach(IObserver observer);
15     void Detach(IObserver observer);
16     void Notify();
17 }
18 // Implementasi subject: NewsAgency
19
20 public class NewsAgency : ISubject
21 {
22     // Menyimpan daftar observer yang terdaftar
23     private readonly List<IObserver> _observers = new();
24     // Menyimpan berita terbaru yang akan dikirimkan ke observer
25     private string _latestNews;
26
27     public void Attach(IObserver observer)
28     {
29         if (observer == null)
30             throw new ArgumentNullException(nameof(observer));
31
32         if (!_observers.Contains(observer))
33             _observers.Add(observer);
34     }
35
36     public void Detach(IObserver observer)
37     {
38         if (observer == null)
39             throw new ArgumentNullException(nameof(observer));
40
41         _observers.Remove(observer);
42     }
43
44     public void Notify()
45     {
46         // Jika tidak ada berita, tidak perlu memberi notifikasi
47         if (string.IsNullOrEmpty(_latestNews)) return;
48
49         foreach (var observer in _observers)
50         {
51             observer.Update(_latestNews);
52         }
53     }
54 }
```

```
52     }
53 }
54
55 // Menambahkan berita baru dan memberi notifikasi ke semua observer
56 public void AddNews(string news)
57 {
58     if (string.IsNullOrEmpty(news)) return;
59     _latestNews = news;
60     Notify();
61 }
62
63 // Implementasi observer: Subscriber
64
65 public class Subscriber : IObserver
66 {
67     private readonly string _name;
68
69     public Subscriber(string name)
70     {
71         if (string.IsNullOrEmpty(name))
72             throw new ArgumentException("Name cannot be empty", nameof(name));
73         _name = name;
74     }
75
76     public void Update(string message)
77     {
78         Console.WriteLine($"{_name} menerima berita terbaru: {message}");
79     }
80 }
81
82 // Program utama untuk menjalankan demo
83
84 public class Program
85 {
86     public static void Main()
87     {
88         var agency = new NewsAgency();
89
90         var andi = new Subscriber("Andi");
91         var budi = new Subscriber("Budi");
92         var cici = new Subscriber("Cici");
93
94         agency.Attach(andi);
95         agency.Attach(budi);
96         agency.Attach(cici);
97
98         agency.AddNews("Berita Hari Ini: Cuaca Cerah!");
99
100        agency.Detach(budi);
101
102        agency.AddNews("Update: Harga BBM Naik!");
103    }
104 }
105
106 }
```

1. Naming Convention

Pastikan semua nama menggunakan PascalCase untuk class, method, dan properti, serta camelCase untuk variabel lokal dan parameter.

Sudah benar:

```
NewsAgency, AddNews, latestNews
```

```
Attach(IObserver observer)
```

```
Update(string message)
```

Tidak perlu ubah apapun, naming convention sudah baik.

2. Whitespace dan Indentation

Pastikan:

Tidak ada spasi berlebih.

Setiap blok kode {} ter-indentasi dengan baik.

Contoh sudah baik:

```
csharp
```

```
CopyEdit
```

```
foreach (var observer in observers)
```

```
{
```

```
    observer.Update(latestNews);
```

```
}
```

3. Variable/Attribute Declarations

Sudah menggunakan pendekatan deklarasi yang jelas:

```
csharp
```

```
CopyEdit
```

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

```
private string latestNews;
```

Jika ingin lebih ekspresif:

```
csharp
```

```
CopyEdit
```

```
private readonly List<IObserver> _observers = new();
```

```
private string _latestNews;
```

(menggunakan prefix _ adalah praktik umum di banyak tim .NET modern)

4. Comments

Tambahkan komentar untuk menjelaskan fungsi dari class, method, atau blok penting.

Contoh yang bisa ditambahkan:

```
csharp
```

```
CopyEdit
```

```
// Menyimpan daftar subscriber (observer) yang akan menerima notifikasi berita
```

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

```
// Method untuk menambahkan berita baru dan memberitahu semua subscriber
```

```
public void AddNews(string news)
```

```
{
```

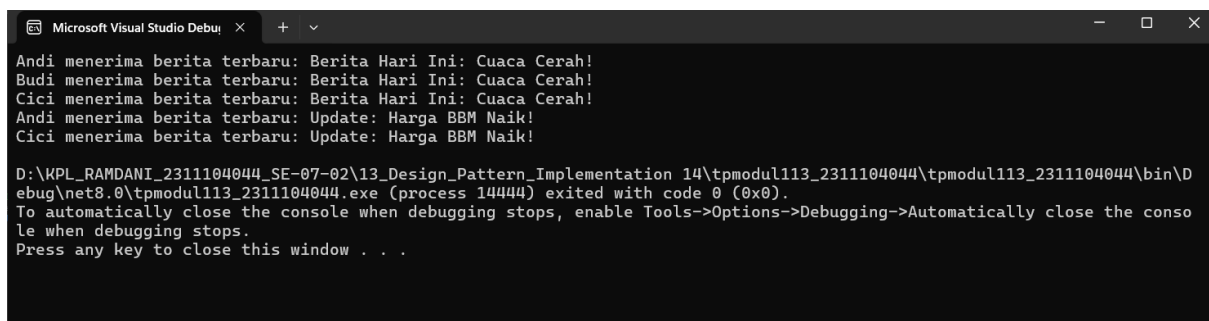
```
    if (string.IsNullOrEmpty(news)) return;
```

```
    latestNews = news;
```

```
    Notify();
```

```
}
```

Output

A screenshot of the Microsoft Visual Studio Debug Console window. The window title is "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 14444) 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 . . .
```