

## TP MODUL 14

Nama : Farhan Kurniawan

NIM : 2311104073

Kelas : SE-07-02

Link Github : [https://github.com/farkurr/KPL\\_FARHANKURNIAWAN\\_2311104073\\_SE-07-02.git](https://github.com/farkurr/KPL_FARHANKURNIAWAN_2311104073_SE-07-02.git)

Dari TP Modul: 8

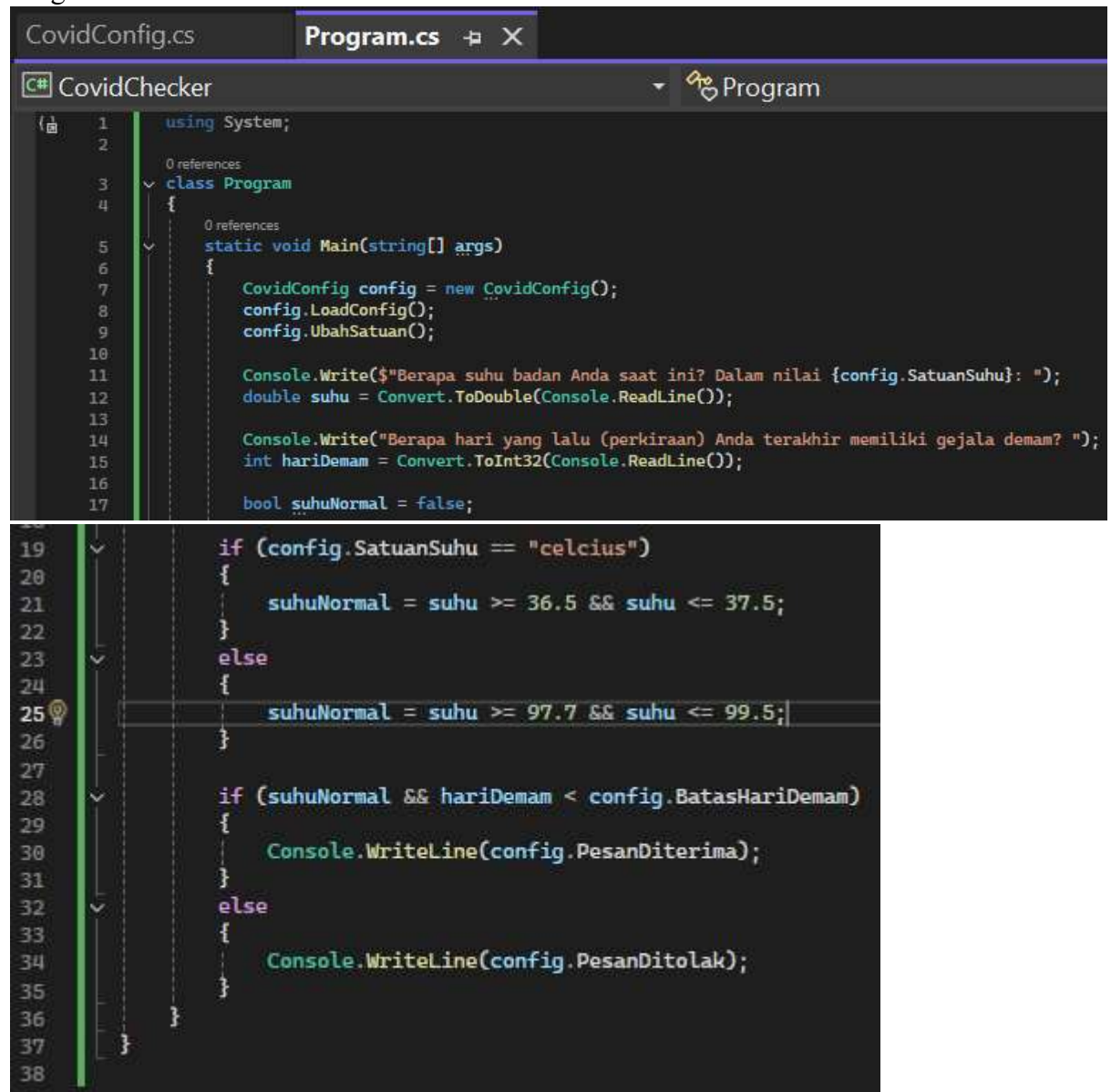
### 1. CovidConfig.cs

```
CovidConfig.cs Program.cs
C# CovidChecker CovidConfig
1 using System;
2 using System.IO;
3 using System.Text.Json;
4
5 4 references
6 public class CovidConfig
7 {
8     6 references
9     public string SatuanSuhu { get; set; } = "celcius";
10    3 references
11    public int BatasHariDemam { get; set; } = 14;
12    3 references
13    public string PesanDitolak { get; set; } = "Anda tidak diperbolehkan masuk ke dalam gedung ini";
14    3 references
15    public string PesanDiterima { get; set; } = "Anda dipersilahkan untuk masuk ke dalam gedung ini";
16    private const string ConfigPath = "covid_config.json";
17
18
19
20
21 1 reference
22 public void LoadConfig()
23 {
24     if (File.Exists(ConfigPath))
25     {
26         string jsonString = File.ReadAllText(ConfigPath);
27         CovidConfig? config = JsonSerializer.Deserialize<CovidConfig>(jsonString);
28
29         if (config != null)
30         {
31             SatuanSuhu = config.SatuanSuhu;
32             BatasHariDemam = config.BatasHariDemam;
33             PesanDitolak = config.PesanDitolak;
34             PesanDiterima = config.PesanDiterima;
35         }
36     }
37 }
38
39 1 reference
40 public void UbahSatuan()
41 {
42     SatuanSuhu = SatuanSuhu == "celcius" ? "fahrenheit" : "celcius";
43 }
44 }
```

File CovidConfig.cs berisi kelas CovidConfig yang berfungsi sebagai tempat penyimpanan dan pengelolaan konfigurasi sistem pemeriksaan suhu tubuh. Kelas ini menyimpan data berupa satuan suhu (celcius atau fahrenheit), batas maksimum hari sejak terakhir demam, serta pesan yang akan ditampilkan kepada pengguna apakah

mereka diperbolehkan masuk atau tidak. File ini juga menyediakan dua metode utama: LoadConfig() untuk membaca data konfigurasi dari file covid\_config.json menggunakan deserialisasi JSON, serta UbahSatuan() untuk mengubah satuan suhu secara otomatis dari celcius ke fahrenheit atau sebaliknya. Kelas ini berperan penting dalam menjaga agar konfigurasi aplikasi tetap fleksibel dan mudah diubah tanpa mengedit kode program utama.

## 2. Program.cs



```
1  using System;
2
3  0 references
4  class Program
5  {
6      0 references
7      static void Main(string[] args)
8      {
9          CovidConfig config = new CovidConfig();
10         config.LoadConfig();
11         config.UbahSatuan();
12
13         Console.WriteLine($"Berapa suhu badan Anda saat ini? Dalam nilai {config.SatuanSuhu}: ");
14         double suhu = Convert.ToDouble(Console.ReadLine());
15
16         Console.WriteLine("Berapa hari yang lalu (perkiraan) Anda terakhir memiliki gejala demam? ");
17         int hariDemam = Convert.ToInt32(Console.ReadLine());
18
19         bool suhuNormal = false;
20
21         if (config.SatuanSuhu == "celcius")
22         {
23             suhuNormal = suhu >= 36.5 && suhu <= 37.5;
24         }
25         else
26         {
27             suhuNormal = suhu >= 97.7 && suhu <= 99.5;
28         }
29
30         if (suhuNormal && hariDemam < config.BatasHariDemam)
31         {
32             Console.WriteLine(config.PesanDiterima);
33         }
34         else
35         {
36             Console.WriteLine(config.PesanDitolak);
37         }
38     }
39 }
```

File Program.cs merupakan titik masuk (entry point) dari aplikasi console ini, yang bertanggung jawab menjalankan logika utama pemeriksaan suhu tubuh dan gejala demam berdasarkan konfigurasi dari CovidConfig. Program akan memuat konfigurasi awal, mengubah satuan suhu, meminta input dari pengguna mengenai suhu tubuh dan waktu terakhir mengalami demam, lalu mengevaluasi apakah kondisi tersebut memenuhi syarat untuk diizinkan masuk. Logika pemeriksaan menyesuaikan nilai suhu normal berdasarkan satuan suhu yang digunakan, dan hasil evaluasi akan menentukan

apakah pengguna menerima pesan diterima atau ditolak. File ini mengatur alur interaksi pengguna dan merupakan jantung dari proses pengambilan keputusan dalam aplikasi.

3. Perubahan sebelum dan setelah di clean code:

1. File: CovidConfig.cs

- Sebelum Clean Code:

File ini berisi kelas CovidConfig yang menyimpan konfigurasi COVID seperti satuan suhu, batas hari demam, serta pesan diterima/ditolak. Atribut-atributnya menggunakan penamaan gaya snake\_case, seperti satuan\_suhu dan batas\_hari\_demam, yang tidak sesuai dengan konvensi PascalCase di C#. Selain itu, whitespace dan indentation masih benar, namun kurang komentar yang menjelaskan fungsi dari masing-masing metode. Penggunaan var di Deserialize juga tidak aman tanpa pengecekan null, yang berpotensi menyebabkan error saat runtime.

- Setelah Clean Code:

File ini telah diperbaiki mengikuti C# naming convention (PascalCase), misalnya satuan\_suhu menjadi SatuanSuhu. Komentar telah ditambahkan untuk menjelaskan fungsi metode LoadConfig dan UbahSatuan. Penggunaan JsonSerializer.Deserialize kini menggunakan pengecekan null (dengan CovidConfig? config), sehingga lebih aman. Secara keseluruhan, file ini sekarang lebih rapi, terbaca, dan sesuai dengan standar .NET yang direkomendasikan.

2. File: Program.cs

- Sebelum Clean Code:

File ini berisi method Main() sebagai titik masuk aplikasi, yang berfungsi membaca konfigurasi dari CovidConfig, menerima input pengguna (suhu dan hari demam), lalu mengevaluasi apakah pengguna diizinkan masuk. Namun, nama variabel config, suhu, dan hariDemam sudah cukup baik, tetapi penggunaan conditional seperti if (config.satuan\_suhu == "celcius") menunjukkan bahwa variabel satuan\_suhu masih belum menggunakan gaya PascalCase. Selain itu, tidak ada komentar yang menjelaskan logika pemeriksaan suhu dan gejala.

- Setelah Clean Code:

File ini telah di-refactor agar lebih mudah dibaca dan sesuai konvensi. Semua pemanggilan properti dari CovidConfig telah menggunakan gaya PascalCase seperti config.SatuanSuhu. Komentar tambahan diberikan untuk menjelaskan setiap langkah logika program, dari mulai input hingga validasi kondisi. Struktur kode juga ditata dengan rapi dengan indentasi konsisten, memisahkan blok logika menggunakan whitespace agar lebih jelas.