

**RAKTIKUM KONSTRUKSI PERANGKAT LUNAK  
TUGAS PENDAHULUAN 14**

**Clean Code**



**Telkom  
University**

disusun Oleh:  
Nita Fitrotul Mar'ah  
2211104005  
SE0601

**Dosen Pengampu :**  
Yudha Islami Sulistya, S.Kom., M.Cs.

**S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## 1. MEMBUAT PROJECT MODUL

Buka IDE misalnya dengan Visual Studio

- Copy salah satu folder tugas pendahuluan yang dimiliki sebelumnya (dari modul 2 sampai modul 10), kemudian rename folder hasil copy-paste tersebut dengan tpmodul14\_NIM (coba pilih tugas pendahuluan yang paling sederhana)
- Misalnya menggunakan Visual Studio, bukalah project/folder yang di-copy sebelumnya.

## 2. REFACTORING DENGAN STANDAR CODE

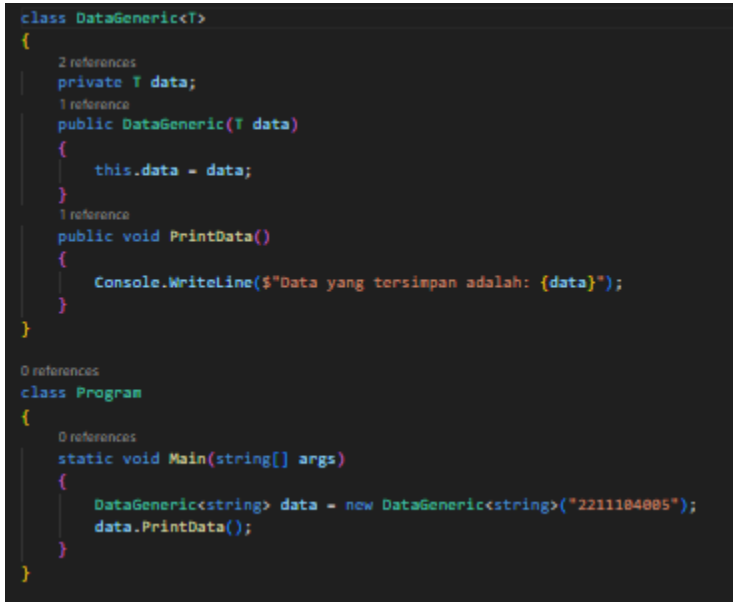
Dengan mengikuti standard code yang digunakan (misal C# dengan standar dari .NET), pastikan kode yang dikumpulkan memenuhi faktor-faktor berikut:

- Naming convention
  - Variable / Property / Attribute
  - Method / Function / Procedure
- White space dan indentation
- Variable / attribute declarations
- Comments

### Jawab:

Saya menyalin tugas pendahuluan modul lima tentang “Generics” dan rename dengan nama folder tpmodul14\_2211104005.

- Source Code sebelum di refactor



```
class DataGeneric<T>
{
    2 references
    private T data;
    1 reference
    public DataGeneric(T data)
    {
        this.data = data;
    }
    1 reference
    public void PrintData()
    {
        Console.WriteLine($"Data yang tersimpan adalah: {data}");
    }
}

0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        DataGeneric<string> data = new DataGeneric<string>("2211104005");
        data.PrintData();
    }
}
```

- Source Code sesudah di refactor

```

using System;

// Kelas generik untuk menyimpan dan menampilkan data
3 references
public class GenericData<T>
{
    2 references
    private T _data;

    // Konstruktor untuk menyimpan data
    1 reference
    public GenericData(T data)
    {
        _data = data;
    }

    // Method untuk mencetak data yang tersimpan
    1 reference
    public void DisplayData()
    {
        Console.WriteLine($"Data yang tersimpan adalah: {_data}");
    }
}

// Kelas utama Program
0 references
class Program
{
    0 references
    static void Main()
    {
        // Membuat objek GenericData dengan NIM dan menampilkan data
        GenericData<string> studentData = new GenericData<string>("2211104005");
        studentData.DisplayData();
    }
}

```

c. Hasil

```

Data yang tersimpan adalah: 2211104005

D:\KPL\14_Clean_Code\TP\tpmodul14_2211104

```

d. Penjelasan

Kode hasil refaktor menggunakan prinsip clean code dan generic class untuk meningkatkan keterbacaan dan fleksibilitas. Class `GenericData<T>` menyimpan satu data bertipe generik `T` dengan field privat `_data`, lalu menyediakan konstruktor untuk inisialisasi dan method `DisplayData()` untuk menampilkannya. Pada method `Main()`, objek bertipe `GenericData<string>` dibuat untuk menyimpan NIM, dan data tersebut ditampilkan menggunakan method yang tersedia. Penggunaan nama class, variabel, dan method yang lebih deskriptif serta penambahan komentar menjadikan kode lebih mudah dipahami dan dapat digunakan kembali untuk tipe data lain tanpa harus menulis ulang logika.