

**TUGAS PENDAHULUAN 14
KONSTRUKSI PERANGKAT LUNAK**

**MODUL XIV
CLEAN CODE**



Disusun Oleh :

Nadia Putri Rahmaniar / 2211104012

S1 SE-06-01

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
TELKOM UNIVERSITY PURWOKERTO**

TUGAS JURNAL XIV

1. MEMBUAT PROJECT MODUL

Buka IDE misalnya dengan Visual Studio

- A. Copy salah satu folder tugas pendahuluan yang dimiliki sebelumnya (dari modul sampai modul 10), kemudian rename folder hasil copy-paste tersebut dengan tpmodul14_NIM (coba pilih tugas pendahuluan yang paling sederhana)
- B. 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:

- A. Naming convention
 - Variable / Property / Attribute
 - Method / Function / Procedure
- B. White space dan indentation
- C. Variable / attribute declarations
- D. Comments

Jawab:

Saya menyalin tugas pendahuluan modul lima tentang “Generics” dan rename dengan nama folder tpmodul14_2211104012.

a. 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>("2211104002");
        data.PrintData();
    }
}
```

b. 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>("2211104002");
        studentData.DisplayData();
    }
}
```

Data yang tersimpan adalah: 2211104012

Penjelasan

Implementasi kode ini telah ditingkatkan secara signifikan melalui praktik Clean Code dan pemanfaatan Generic Class, yang secara kolektif meningkatkan keterbacaan serta kemampuan adaptasi kode.

Fokus utamanya ada pada kelas `GenericData<T>`. Kelas ini dirancang untuk dapat menyimpan satu data dengan tipe apa pun (diwakili oleh `T`, yang disebut generik). Di dalamnya, terdapat field privat bernama `_data` yang bertugas menyimpan nilai tersebut. Kelas ini juga dilengkapi dengan konstruktor untuk inisialisasi data dan metode `DisplayData()` yang bertanggung jawab menampilkan data yang tersimpan.

Dalam metode `Main()` (yang biasanya merupakan titik awal eksekusi program), sebuah objek dari kelas `GenericData<T>` ini dibuat khusus untuk menyimpan Nomor Induk Mahasiswa (NIM). Data NIM tersebut kemudian ditampilkan menggunakan metode `DisplayData()` yang telah disediakan.

Secara keseluruhan, penggunaan nama kelas, variabel, dan metode yang lebih deskriptif, ditambah dengan komentar kode yang relevan, telah membuat kode ini jauh lebih mudah dipahami oleh siapa pun yang membacanya. Selain itu, sifat generik dari `GenericData<T>` memungkinkan kode ini untuk digunakan kembali dengan berbagai tipe data lainnya di masa mendatang tanpa perlu menulis ulang seluruh logika yang sama, sehingga sangat efisien dan fleksibel.