

**TUGAS PENDAHULUAN  
KONSTRUKSI PERANGKAT LUNAK**

**MODUL XIV  
CLEAN CODE**



**Disusun Oleh :**

Rizky Hanifa Afania

2211104017

SE-06-01

**Dosen Pengampu :**

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

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

## **TUGAS PENDAHULUAN XIV**

### **1. MEMBUAT PROJECT MODUL**

Buka IDE misalnya dengan Visual Studio

- A. 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)
- 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\_2211104017.

a. Source Code sebelum di refactor

```
1  using System;
2
3  public class HaloGeneric
4  {
5      public static void SapaUser<T>(T user)
6      {
7          Console.WriteLine($"Halo user {user}");
8      }
9  }
10
11 public class DataGeneric<T>
12 {
13     private T data;
14
15     public DataGeneric(T data)
16     {
17         this.data = data;
18     }
19
20     public void PrintData()
21     {
22         Console.WriteLine($"Data yang tersimpan adalah: {data}");
23     }
24 }
25
26 class Program
27 {
28     static void Main()
29     {
30         HaloGeneric.SapaUser("Rizky Manifa Afania");
31
32         // Membuat objek DataGeneric dengan NIM
33         DataGeneric<string> data = new DataGeneric<string>("2211104017");
34         data.PrintData();
35     }
36 }
```

b. Source Code sesudah di refactor

```
1  using System;
2
3  // Kelas generik untuk menampilkan sapaan kepada user
4  public class GreetingUtility
5  {
6      public static void DisplayGreeting<T>(T userName)
7      {
8          Console.WriteLine($"Halo user {userName}");
9      }
10 }
11
12 // Kelas generik untuk menyimpan dan menampilkan data
13 public class GenericData<T>
14 {
15     private T _data;
16
17     // Konstruktor untuk menyimpan data
18     public GenericData(T data)
19     {
20         _data = data;
21     }
22
23     // Method untuk mencetak data yang tersimpan
24     public void DisplayData()
25     {
26         Console.WriteLine($"Data yang tersimpan adalah: {_data}");
27     }
28 }
29
30 // Kelas utama Program
31 class Program
32 {
33     static void Main()
34     {
35         // Menampilkan sapaan untuk user
36         GreetingUtility.DisplayGreeting("Rizky Hanifa Afania");
37
38         // Membuat objek GenericData dengan NIM dan menampilkan data
39         GenericData<string> studentData = new GenericData<string>("2211104017");
40         studentData.DisplayData();
41     }
42 }
```

c. Hasil

```
C:\> Microsoft Visual Studio Debug Console
Halo user Rizky Hanifa Afania
Data yang tersimpan adalah: 2211104017
F:\Coding\Praktikum KPL\Praktikum Mandiri\Modul 14_Clean Code\
```

#### **d. Penjelasan**

Berikut adalah penjelasan mengenai clean code yang sudah di refactor

##### **1. Naming Convention**

###### **a. Class Names:**

- HaloGeneric diubah menjadi GreetingUtility agar lebih deskriptif dan jelas.
- DataGeneric diubah menjadi GenericData agar lebih deskriptif

###### **b. Method Names:**

- SapaUser diubah menjadi DisplayGreeting agar sesuai standar PascalCase dan lebih jelas).
- PrintData diubah menjadi DisplayData agar lebih deskriptif

###### **c. Variable Names:**

- Variabel `_data` diberi prefiks `_` untuk menunjukkan bahwa itu adalah variabel private sesuai standar .NET.

##### **2. Variable / Property / Attribute**

- Atribut `_data` sudah menggunakan konvensi penamaan dengan prefiks `_` untuk variabel private.
- Variabel lokal `studentData` sudah jelas dan menggunakan camelCase agar sesuai standar untuk variabel lokal

##### **3. Method / Function / Procedure**

- Semua method menggunakan PascalCase, sesuai standar .NET.
- Metode memiliki nama yang deskriptif: `DisplayGreeting` dan `DisplayData` yang langsung menunjukkan fungsinya.

##### **4. White Space dan Indentation**

- Indentasi konsisten menggunakan 4 spasi agar sesuai standar Visual Studio
- Pemisahan antar class memiliki whitespace yang jelas.
- Setiap method memiliki spacing yang cukup untuk keterbacaan.

##### **5. Variable / Attribute Declarations**

- Atribut `_data` dideklarasikan secara private dengan tipe generik.

- b. Variable `studentData` diinisialisasi dengan tipe generik `GenericData<string>`.

## **6. Comments**

Comment ditambahkan di bagian:

- a. Setiap class untuk menjelaskan fungsinya.
- b. Method untuk menjelaskan fungsi metode tersebut.
- c. Bagian kode utama (Main) untuk menjelaskan proses yang dilakukan.