

**TUGAS JURNAL  
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 2 sampai modul 13), kemudian rename folder hasil copy-paste tersebut dengan modul14\_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 sembilan tentang “API Design” dan rename dengan nama folder modul14\_2211104012.

a. Source Code sebelum di refactor

```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;

namespace modul14_2211104012.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class MahasiswaController : ControllerBase
    {
        public class Mahasiswa
        {
            public string Nama { get; set; }
            public string Nim { get; set; }
        }

        private static List<Mahasiswa> listMahasiswa = new
List<Mahasiswa>
        {
            new Mahasiswa { Nama = "Nita Fitrotul Mar'ah", Nim =
"2211104005" },
            new Mahasiswa { Nama = "Alfian Mutakim", Nim =
"2211104017" },
            new Mahasiswa { Nama = "Nadia Putri Rahmaniar", Nim =
"2211104012" },
            new Mahasiswa { Nama = "Rafla Dafin Kamil", Nim =
"2211104023" },
            new Mahasiswa { Nama = "Muhammad Edgar Nadhif", Nim =
"22111040028" },
            new Mahasiswa { Nama = "Muhammad Dhimas Afrizal", Nim =
"22111040025" },
        };

        // GET /api/mahasiswa
        [HttpGet]
        public IEnumerable<Mahasiswa> Get()
        {
            return listMahasiswa;
        }
    }
}
```

```

    }

    // GET /api/mahasiswa/{id}
    [HttpGet("{id}")]
    public ActionResult<Mahasiswa> Get(int id)
    {
        if (id < 0 || id >= listMahasiswa.Count)
            return NotFound();
        return listMahasiswa[id];
    }

    // POST /api/mahasiswa
    [HttpPost]
    public void Post([FromBody] Mahasiswa mhs)
    {
        listMahasiswa.Add(mhs);
    }

    // DELETE /api/mahasiswa/{id}
    [HttpDelete("{id}")]
    public void Delete(int id)
    {
        if (id >= 0 && id < listMahasiswa.Count)
        {
            listMahasiswa.RemoveAt(id);
        }
    }
}
}

```

b. Source Code sesudah di refactor

```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;

namespace modul14_2211104012.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class MahasiswaController : ControllerBase
    {
        // Model Mahasiswa
        public class Mahasiswa
        {
            public string Nama { get; set; }
            public string Nim { get; set; }
        }

        // List statis data mahasiswa
        private static readonly List<Mahasiswa> _mahasiswaList =
new()
        {
            new Mahasiswa { Nama = "Nadia Putri Rahmaniar", Nim =
"2211104012" },
            new Mahasiswa { Nama = "Nita Fitrotul Mar'ah", Nim =
"2211104005" },
            new Mahasiswa { Nama = "Alfian Mutakim", Nim =
"2211104020" },
            new Mahasiswa { Nama = "Edgar Nadhif", Nim =
"2211104021" },
            new Mahasiswa { Nama = "Dhimas Afrizal", Nim =
"22111040019" }
        };

        // GET: api/mahasiswa
        [HttpGet]
        public ActionResult<IEnumerable<Mahasiswa>>
GetAllMahasiswa()
        {

```

```

        return Ok(_mahasiswaList);
    }

    // GET: api/mahasiswa/{id}
    [HttpGet("{id}")]
    public ActionResult<Mahasiswa> GetMahasiswaById(int id)
    {
        if (id < 0 || id >= _mahasiswaList.Count)
        {
            return NotFound();
        }

        return Ok(_mahasiswaList[id]);
    }

    // POST: api/mahasiswa
    [HttpPost]
    public ActionResult AddMahasiswa([FromBody] Mahasiswa
mahasiswa)
    {
        _mahasiswaList.Add(mahasiswa);

        return CreatedAtAction(nameof(GetMahasiswaById), new {
id = _mahasiswaList.Count - 1 }, mahasiswa);
    }

    // DELETE: api/mahasiswa/{id}
    [HttpDelete("{id}")]
    public ActionResult DeleteMahasiswa(int id)
    {
        if (id < 0 || id >= _mahasiswaList.Count)
        {
            return NotFound();
        }

        _mahasiswaList.RemoveAt(id);

        return NoContent();
    }

```

```
}  
}
```

## Penjelasan

Berikut adalah penjelasan mengenai clean code yang sudah di refactor

### 1. Naming Convention

#### a. Variable / Property / Attribute

- Variabel yang sebelumnya bernama listMahasiswa kini menjadi `_mahasiswaList`. Penambahan `_` (underscore) di awal menandakan bahwa variabel tersebut bersifat private dan statis.
- Nama properti seperti Nama dan Nim tetap menggunakan PascalCase, sesuai standar penulisan properti di C#.

#### b. Method / Function / Procedure

- Nama metode dibuat lebih deskriptif dan menggunakan PascalCase.
- Contohnya: `Get()` diubah menjadi `GetAllMahasiswa()` (lebih jelas untuk mengambil semua data), `Post()` menjadi `AddMahasiswa()` (lebih jelas untuk menambahkan mahasiswa), dan `Delete()` menjadi `DeleteMahasiswa()` (jelas untuk menghapus mahasiswa).

### 2. White Space dan Indentation

- Indentasi: Indentasi kini menggunakan 4 spasi untuk setiap blok kode (dalam kelas, metode, atau pernyataan if-else) agar kode terlihat rapi dan konsisten.
- Spasi Kosong: Penambahan spasi kosong antar metode membuat kode tidak terlihat menumpuk dan meningkatkan keterbacaan.

### 3. Variable / Attribute Declarations

Deklarasi listMahasiswa kini menjadi `private static readonly List<Mahasiswa> _mahasiswaList = new();`

- `private`: Menandakan bahwa variabel hanya bisa diakses dalam kelas itu saja.

- static: Menunjukkan bahwa data tidak berubah antar instansi kontroler.
- readonly: Memastikan daftar ini tidak dapat diganti dengan daftar baru, hanya elemennya yang bisa ditambah atau dihapus.

#### **4. Comments**

- Penjelasan Kode: Komentar ditambahkan untuk menjelaskan setiap endpoint dan fungsi kode, contohnya: // GET: api/mahasiswa atau // Mengambil semua data mahasiswa.
- Komentar ini membantu pembaca kode untuk cepat memahami fungsi setiap bagian tanpa harus membaca isi metode secara detail.