

**TUGAS JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**MODUL XIV  
CLEAN CODE**



**Disusun Oleh :**  
**Atika Aji Hadiyani**  
**2211104003**  
**SE06-01**

**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\_2211104003.

a. Source Code sebelum di refactor

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

namespace tp_modul9.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 = "Atika Aji Hadiyani", Nim = "2211104003" },
            new Mahasiswa { Nama = "Kholifahdina", Nim = "2211104004" },
            new Mahasiswa { Nama = "Afad Fath Musyarof Halim", Nim = "2211104030" },
            new Mahasiswa { Nama = "Rachmat Taufik", Nim = "2211104028" },
            new Mahasiswa { Nama = "Noor Alam Fauzan", Nim = "2211104031" },
        };

        // 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_2211104003.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    0 references
    public class MahasiswaController : ControllerBase
    {
        9 references
        public class Mahasiswa
        {
            5 references
            public string Nama { get; set; }
            5 references
            public string Nim { get; set; }
        }

        private static readonly List<Mahasiswa> _mahasiswaList = new()
        {
            new Mahasiswa { Nama = "Atika Aji Hadiyani", Nim = "2211104003" },
            new Mahasiswa { Nama = "Kholifahdina", Nim = "2211104004" },
            new Mahasiswa { Nama = "Afad Fath Musyarof Halim", Nim = "2211104030" },
            new Mahasiswa { Nama = "Rachmat Taufik", Nim = "2211104028" },
            new Mahasiswa { Nama = "Noor Alam Fauzan", Nim = "2211104031" }
        };

        // GET: api/mahasiswa
        [HttpGet]
        0 references
        public ActionResult<IEnumerable<Mahasiswa>> GetAllMahasiswa()
        {
            return Ok(_mahasiswaList);
        }

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

            return Ok(_mahasiswaList[id]);
        }

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

            _mahasiswaList.RemoveAt(id);
            return NoContent();
        }
    }
}
```

### c. Penjelasan

Berikut adalah penjelasan mengenai clean code yang sudah di refactor

#### 1. Konvensi Penamaan

##### a. Variabel / Properti / Atribut

- Variabel list yang sebelumnya bernama listMahasiswa kini diubah menjadi `_mahasiswaList`. Penambahan garis bawah (`_`) di awal menunjukkan bahwa variabel ini bersifat private dan statis.
- Nama properti seperti Nama dan Nim tetap menggunakan PascalCase sesuai standar penulisan properti di C#.

##### b. Method / Fungsi / Prosedur

- Nama method dibuat lebih deskriptif dan memakai PascalCase, contohnya:
  - Method `Get()` menjadi `GetAllMahasiswa()` yang lebih jelas menggambarkan fungsi untuk mengambil seluruh data mahasiswa.
  - Method `Post()` diubah menjadi `AddMahasiswa()` agar lebih spesifik menandakan penambahan data mahasiswa.
  - Method `Delete()` menjadi `DeleteMahasiswa()`, sehingga jelas bahwa fungsi ini bertugas menghapus data mahasiswa.

#### 2. Penggunaan White Space dan Indentasi

- Indentasi pada kode sudah diperbaiki menjadi lebih rapi dan konsisten, dengan menggunakan 4 spasi per level indentasi pada blok kode seperti kelas, method, dan kondisi if-else.
- Ditambahkan spasi kosong antar method agar kode tidak terlihat padat dan lebih mudah dibaca oleh developer lain.

#### 3. Deklarasi Variabel dan Atribut

- Variabel list mahasiswa sekarang dideklarasikan sebagai private static readonly `List<Mahasiswa> _mahasiswaList = new()`.

- private berarti variabel ini hanya dapat diakses di dalam kelas tersebut.
- static menunjukkan bahwa data ini bersifat sama di semua instance controller dan tidak tergantung pada objek tertentu.
- readonly memastikan bahwa referensi list tidak bisa diganti dengan objek lain, walaupun isi list masih dapat dimodifikasi (tambah/hapus elemen).

#### **4. Penambahan Komentar**

- Komentar singkat ditambahkan di atas setiap endpoint untuk menjelaskan tujuan dan fungsi masing-masing method, misalnya:

*csharp*

*SalinEdit*

*// GET: api/mahasiswa*

*// Mengambil semua data mahasiswa*

- Komentar ini sangat membantu pembaca kode agar cepat memahami fungsi setiap bagian tanpa harus membaca detail implementasi method.