

**TUGAS JURNAL**  
**KONSTRUKSI PERANGKAT LUNAK**  
**MODUL XIV**  
**CLEAN CODE**



**Disusun Oleh:**  
**Maria Nathasya Desfera Pangestu**  
**2211104008**  
**SE0601**

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

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

## Tugas Jurnal

### 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 mencopy tugas pendahuluan modul sembilan API Design dan merename nama folder jadi modul14\_2211104008.

Source code sebelum refactoring

```

1  using Microsoft.AspNetCore.Mvc;
2  using System.Collections.Generic;
3  using tp_modul9.Models;
4
5  [ApiController]
6  [Route("api/[controller]")]
7  public class MahasiswaController : ControllerBase
8  {
9      private static List<Mahasiswa> listMahasiswa = new List<Mahasiswa>
10     {
11         new Mahasiswa { Nama = "Maria Nathasya", Nim = "2211104008" },
12         new Mahasiswa { Nama = "Alfian Mutakim", Nim = "2211104017" },
13         new Mahasiswa { Nama = "Nadia Putri Rahmانيar", Nim = "2211104012" },
14         new Mahasiswa { Nama = "Muhammad Edgar Nadhif", Nim = "2211104028" },
15         new Mahasiswa { Nama = "Muhammad Dhimas Afrizal", Nim = "2211104025" },
16     };
17
18     [HttpGet]
19     public IEnumerable<Mahasiswa> Get()
20     {
21         return listMahasiswa;
22     }
23
24     [HttpGet("{id}")]
25     public ActionResult<Mahasiswa> Get(int id)
26     {
27         if (id < 0 || id >= listMahasiswa.Count)
28             return NotFound();
29         return listMahasiswa[id];
30     }
31
32     [HttpPost]
33     public void Post([FromBody] Mahasiswa mhs)
34     {
35         listMahasiswa.Add(mhs);
36     }
37
38     [HttpDelete("{id}")]
39     public void Delete(int id)
40     {
41         if (id >= 0 && id < listMahasiswa.Count)
42             listMahasiswa.RemoveAt(id);
43     }
44 }

```

Source code setelah refactoring

```

1  using Microsoft.AspNetCore.Mvc;
2  using System.Collections.Generic;
3  using tp_modul9.Models;
4
5  [ApiController]
6  [Route("api/[controller]")]
7  0 references
8  public class MahasiswaController : ControllerBase
9  {
10     6 references
11     private static readonly List<Mahasiswa> MahasiswaList = new List<Mahasiswa>
12     {
13         new Mahasiswa { Nama = "Maria Nathasya", Nim = "2211104008" },
14         new Mahasiswa { Nama = "Alfian Mutakim", Nim = "2211104017" },
15         new Mahasiswa { Nama = "Nadia Putri Rahmaniar", Nim = "2211104012" },
16         new Mahasiswa { Nama = "Muhammad Edgar Nadhif", Nim = "2211104028" },
17         new Mahasiswa { Nama = "Muhammad Dhimas Afrizal", Nim = "2211104025" },
18     };
19
20     [HttpGet]
21     0 references
22     public ActionResult<IEnumerable<Mahasiswa>> GetAllMahasiswa()
23     {
24         return Ok(MahasiswaList);
25     }
26
27     [HttpGet("{id}")]
28     1 reference
29     public ActionResult<Mahasiswa> GetMahasiswaById(int id)
30     {
31         if (!IsValidId(id))
32             return NotFound();
33
34         return Ok(MahasiswaList[id]);
35     }
36
37     [HttpPost]
38     0 references
39     public ActionResult AddMahasiswa([FromBody] Mahasiswa mahasiswa)
40     {
41         if (mahasiswa == null)
42             return BadRequest("Mahasiswa cannot be null.");
43
44         MahasiswaList.Add(mahasiswa);
45         return CreatedAtAction(nameof(GetMahasiswaById), new { id = MahasiswaList.Count - 1 }, mahasiswa);
46     }
47
48     [HttpDelete("{id}")]
49     0 references

```

```

50     0 references
51     public ActionResult DeleteMahasiswa(int id)
52     {
53         if (!IsValidId(id))
54             return NotFound();
55
56         MahasiswaList.RemoveAt(id);
57         return NoContent();
58     }
59
60     2 references
61     private bool IsValidId(int id)
62     {
63         return id >= 0 && id < MahasiswaList.Count;
64     }
65 }
66
67
68

```

## Penjelasan

### a. Konvensi Penamaan

- Variabel / Properti / Atribut

Sebelumnya, variabel daftar bernama `listMahasiswa`, kini diubah menjadi `_mahasiswaList`. Penambahan tanda underscore (`_`) di awal menunjukkan bahwa variabel ini bersifat privat dan statis. Nama-nama properti seperti Nama dan Nim tetap menggunakan format PascalCase, sesuai dengan standar penulisan properti di C#.

### b. Metode / Fungsi / Prosedur

- `Get()` diubah menjadi `GetAllMahasiswa()`, yang lebih jelas menunjukkan bahwa ini mengambil semua data.
- `Post()` diubah menjadi `AddMahasiswa()`, yang lebih jelas bahwa ini berfungsi untuk menambahkan mahasiswa.
- `Delete()` diubah menjadi `DeleteMahasiswa()`, yang jelas menunjukkan bahwa ini digunakan untuk menghapus mahasiswa.

### c. White Space dan Indentation

Indentasi diperbaiki agar lebih rapi dan konsisten: setiap blok kode (dalam class, metode, if-else) ditulis dengan 4 spasi untuk indentasi. Ruang kosong antar metode ditambahkan agar kode tidak terlihat menumpuk dan lebih mudah dibaca.

### d. Deklarasi Variabel / Atribut

`listMahasiswa` sekarang ditulis sebagai: `private static readonly List<Mahasiswa> _mahasiswaList = new()`

- `Private` menunjukkan bahwa variabel ini hanya dapat diakses dalam class tersebut.
- `Static` karena data ini tidak berubah antar instance controller.
- `Readonly` agar daftar ini tidak dapat diganti dengan list baru, hanya bisa menambah atau menghapus elemennya

### e. Komentar

Komentar ditambahkan untuk menjelaskan setiap endpoint dan fungsi kode, seperti:

```
// GET: api/mahasiswa
```

```
// Mengambil semua data mahasiswa
```

Komentar ini membantu siapa pun yang membaca kode untuk dengan cepat memahami fungsi dari setiap bagian, tanpa perlu membaca isi metode satu per satu.