

**TUGAS JURNAL
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 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_2211104017.

a. Source Code sebelum di refactor

```
1  using Microsoft.AspNetCore.Mvc;
2  using System.Collections.Generic;
3
4  namespace modul14_2211104017.Controllers
5  {
6      [ApiController]
7      [Route("api/[controller]")]
8      public class MahasiswaController : ControllerBase
9      {
10         public class Mahasiswa
11         {
12             public string Nama { get; set; }
13             public string Nim { get; set; }
14         }
15
16         private static List<Mahasiswa> listMahasiswa = new List<Mahasiswa>
17         {
18             new Mahasiswa { Nama = "Rizky Hanifa Afania", Nim = "2211104017" },
19             new Mahasiswa { Nama = "Althafia Defiyandrea L.W", Nim = "2211104011" },
20             new Mahasiswa { Nama = "Muhammad Idham C", Nim = "2211104016" },
21             new Mahasiswa { Nama = "Fauzan Wahyu M", Nim = "2211104027" },
22             new Mahasiswa { Nama = "Andera Singgih P", Nim = "2211104007" },
23         };
24
25         // GET /api/mahasiswa
26         [HttpGet]
27         public IEnumerable<Mahasiswa> Get()
28         {
29             return listMahasiswa;
30         }
31
32         // GET /api/mahasiswa/{id}
33         [HttpGet("{id}")]
34         public ActionResult<Mahasiswa> Get(int id)
35         {
36             if (id < 0 || id >= listMahasiswa.Count)
37                 return NotFound();
38             return listMahasiswa[id];
39         }
40
41         // POST /api/mahasiswa
42         [HttpPost]
43         public void Post([FromBody] Mahasiswa mhs)
44         {
45             listMahasiswa.Add(mhs);
46         }
47
48         // DELETE /api/mahasiswa/{id}
49         [HttpDelete("{id}")]
50         public void Delete(int id)
51         {
52             if (id >= 0 && id < listMahasiswa.Count)
53             {
54                 listMahasiswa.RemoveAt(id);
55             }
56         }
57     }
58 }
```

b. Source Code sesudah di refactor

```
1  using Microsoft.AspNetCore.Mvc;
2  using System.Collections.Generic;
3
4  namespace modul14_2211104017.Controllers
5  {
6      [ApiController]
7      [Route("api/[controller]")]
8      public class MahasiswaController : ControllerBase
9      {
10         // Model Mahasiswa
11         public class Mahasiswa
12         {
13             public string Nama { get; set; }
14             public string Nim { get; set; }
15         }
16
17         // List statis data mahasiswa
18         private static readonly List<Mahasiswa> _mahasiswaList = new()
19         {
20             new Mahasiswa { Nama = "Rizky Hanifa Afania", Nim = "2211104017" },
21             new Mahasiswa { Nama = "Althafia Defiyandrea L.W", Nim = "2211104011" },
22             new Mahasiswa { Nama = "Muhammad Idham C", Nim = "2211104016" },
23             new Mahasiswa { Nama = "Fauzan Wahyu M", Nim = "2211104027" },
24             new Mahasiswa { Nama = "Andera Singgih P", Nim = "2211104007" }
25         };
26
27         // GET: api/mahasiswa
28         [HttpGet]
29         public ActionResult<IEnumerable<Mahasiswa>> GetAllMahasiswa()
30         {
31             return Ok(_mahasiswaList);
32         }
33     }
```

```
33
34         // GET: api/mahasiswa/{id}
35         [HttpGet("{id}")]
36         public ActionResult<Mahasiswa> GetMahasiswaById(int id)
37         {
38             if (id < 0 || id >= _mahasiswaList.Count)
39             {
40                 return NotFound();
41             }
42             return Ok(_mahasiswaList[id]);
43         }
44
45         // POST: api/mahasiswa
46         [HttpPost]
47         public ActionResult AddMahasiswa([FromBody] Mahasiswa mahasiswa)
48         {
49             _mahasiswaList.Add(mahasiswa);
50             return CreatedAtAction(nameof(GetMahasiswaById), new { id = _mahasiswaList.Count - 1 }, mahasiswa);
51         }
52
53         // DELETE: api/mahasiswa/{id}
54         [HttpDelete("{id}")]
55         public ActionResult DeleteMahasiswa(int id)
56         {
57             if (id < 0 || id >= _mahasiswaList.Count)
58             {
59                 return NotFound();
60             }
61             _mahasiswaList.RemoveAt(id);
62             return NoContent();
63         }
64     }
65 }
66
67 }
```

c. Penjelasan

Berikut adalah penjelasan mengenai clean code yang sudah di refactor

1. Naming Convention

a. Variable / Property / Attribute

- Sebelumnya variabel list bernama listMahasiswa → sekarang menjadi _mahasiswaList. Penambahan _ di awal menandakan bahwa variabel ini bersifat private dan statis.
- Nama-nama property seperti Nama dan Nim tetap menggunakan PascalCase, karena ini adalah standar penulisan property di C#.

b. Method / Function / Procedure

Nama method sekarang lebih deskriptif dan menggunakan PascalCase, misalnya:

- Get() → jadi GetAllMahasiswa() → lebih jelas bahwa ini mengambil semua data.
- Post() → jadi AddMahasiswa() → lebih jelas bahwa ini menambahkan mahasiswa.
- Delete() → jadi DeleteMahasiswa() → jelas bahwa ini untuk menghapus mahasiswa.

2. White Space dan Indentation

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

3. Variable / Attribute Declarations

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

- private: artinya hanya bisa diakses dalam class itu saja.
- static: karena datanya tidak berubah antar instance controller.

- readonly: supaya daftar ini tidak bisa diganti dengan list baru, hanya bisa ditambah/dihapus elemennya.

4. Comments

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 cepat memahami fungsi dari setiap bagian, tanpa harus membaca isi method satu per satu.