RAKTIKUM KONSTRUKSI PERANGKAT LUNAK
TUGAS JURNAL 14

Clean Code

disusun Oleh:
Nita Fitrotul Mar’ah
2211104005
SE0601

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

S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

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 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_2211104005.

   a. Source Code sebelum di refactor

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

namespace tpmodul9_2211104005.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 Mutakin", Nim = "2211104017" },
            new Mahasiswa { Nama = "Nadia Putri Rahmaniar", Nim = "2211104012" },
            new Mahasiswa { Nama = "Rafli Dhafin Kamil", Nim = "2211104023" },
            new Mahasiswa { Nama = "Muhammad Edgar Nadhif", Nim = "2211104020" },
            new Mahasiswa { Nama = "Muhammad Dhimas Afrizal", Nim = "2211104025" },
        };

        // 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_2211104005.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();
        }
    }
}
```

c. Penjelasan

Berikut ini merupakan penjelasan mengenai clean code yang telah dilakukan refactor:

1. Naming Convention (Konvensi Penamaan)
    a. Variable / Property / Attribute
        - Variabel yang sebelumnya dinamai listMahasiswa kini diubah menjadi _mahasiswaList. Penambahan awalan underscore (_) menunjukkan bahwa variabel ini bersifat private dan static.
        - Penamaan properti seperti Nama dan Nim tetap menggunakan format PascalCase, sesuai standar properti pada bahasa C#.
    b. Method / Function / Procedure
        - Nama metode dibuat lebih deskriptif dan tetap menggunakan PascalCase.

- Misalnya: Get() diubah menjadi GetAllMahasiswa() agar lebih jelas fungsinya untuk mengambil semua data mahasiswa, Post() diubah menjadi AddMahasiswa() untuk menandakan penambahan data mahasiswa, dan Delete() menjadi DeleteMahasiswa() untuk menunjukkan fungsinya menghapus data mahasiswa.

2. White Space dan Indentation (Spasi dan Indentasi)
   - Indentasi kini konsisten menggunakan 4 spasi untuk setiap blok kode, baik di dalam kelas, metode, maupun blok if-else, sehingga kode tampak lebih rapi dan terstruktur.
   - Ditambahkan juga spasi kosong antar metode agar kode tidak terlihat menumpuk sehingga lebih mudah dibaca.
3. Deklarasi Variabel / Atribut

   Deklarasi variabel listMahasiswa diubah menjadi:

   private static readonly List<Mahasiswa> _mahasiswaList = new();
   - private: Variabel hanya dapat diakses di dalam kelas tersebut.
   - static: Data bersifat tetap dan berbagi antar instance controller.
   - readonly: Daftar ini tidak bisa diganti dengan objek daftar baru, namun isi elemennya masih dapat dimodifikasi (misalnya ditambah atau dihapus).
4. Komentar
   - Komentar ditambahkan untuk memberikan penjelasan pada setiap endpoint dan bagian kode, contohnya: // GET: api/mahasiswa atau // Mengambil semua data mahasiswa.
   - Komentar ini membantu programmer lain untuk memahami tujuan setiap bagian kode tanpa harus membaca seluruh isi metode secara rinci.