TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK

MODUL IX
API DESIGN & CONSTRUCTION USING SWAGGER

Disusun Oleh:

Aorinka Anendya Chazanah / 2211104013

S1 SE-06-01

Dosen Pengampu:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

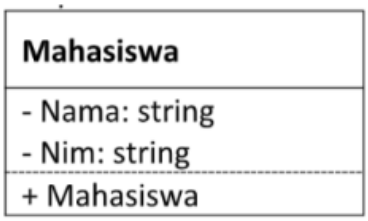
2025

TUGAS PENDAHULUAN

1. IMPLEMENTASI WEB API

Dari master/main branch dan class utama, buatlah program/aplikasi web API dari spesifikasi sebagai berikut ini:

A. API yang dibuat menggunakan data dari kelas Mahasiswa.



B. API yang dibuat mempunyai lokasi sebagai berikut '/api/mahasiswa', URL domain boleh dari port mana saja, misalnya https://localhost:5001/api/mahasiswa (port bebas)

C. Secara default, program yang dibuat memiliki array/list mahasiswa dari anggota kelompok anda (tuliskan nama anda di urutan pertama/paling atas), contohnya:

i. Nama: "LeBron James", Nim: "1302000001"

ii. Nama: "Stephen Curry", Nim: "1302000002"

iii. dst.

D. Gunakan teknologi API sehingga program tersebut dapat menerima HTTP request sebagai berikut:

- GET /api/mahasiswa: mengembalikan output berupa list/array dari semua objek mahasiswa yang tersimpan
- GET /api/mahasiswa/{index}: mengembalikan output berupa objek mahasiswa untuk index ke-'index'
- POST /api/mahasiswa: menambahkan objek mahasiswa baru dengan menyertakan nama dan nim
- DELETE /api/mahasiswa/{index}: menghapus objek mahasiswa dengan index ke 'index'

E. Impementasi yang dibuat tidak menggunakan database, cukup disimpan sebagai suatu variable, dan gunakan “static” di variable tersebut yang menyimpan list/array dari objek-objek mahasiswa

F. Dalam pembuatan program/aplikasi ini, anda dapat mengasumsikan bahwa input dari user selalu benar dan sesuai dengan tipe data yang diharapkan

JAWABAN

- **Source code**

a. File MahasiswaController.cs

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

namespace tpmodul9_2211104013.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    0 references
    public class MahasiswaController : ControllerBase
    {
        9 references
        public class Mahasiswa
        {
            4 references
            public string Nama { get; set; }
            4 references
            public string Nim { get; set; }
        }

        private static List<Mahasiswa> listMahasiswa = new List<Mahasiswa>
        {
            new Mahasiswa { Nama = "Aorinka Anendya Chazanah", Nim = "2211104013" },
            new Mahasiswa { Nama = "Muhammad Abdul Aziz", Nim = "2211104026" },
            new Mahasiswa { Nama = "Fadhila Agil Permana", Nim = "2211104006" },
            new Mahasiswa { Nama = "Muhammad Luthfi Hamdani", Nim = "2211104020" },
        };

        [HttpGet]
        0 references
        public IEnumerable<Mahasiswa> Get()
        {
            return listMahasiswa;
        }

        [HttpGet("{id}")]
        0 references
        public ActionResult<Mahasiswa> Get(int id)
```

```
        public ActionResult<Mahasiswa> Get(int id)
        {
            if (id < 0 || id >= listMahasiswa.Count)
                return NotFound();
            return listMahasiswa[id];
        }

        [HttpPost]
        0 references
        public void Post([FromBody] Mahasiswa mhs)
        {
            listMahasiswa.Add(mhs);
        }

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

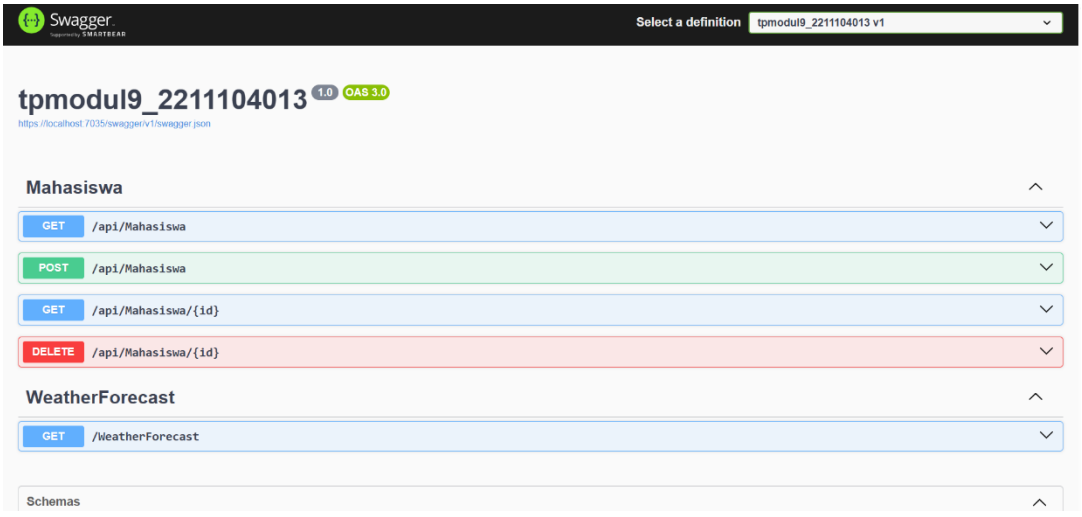
b. File Mahasiswa.cs dalam folder Models

```
namespace tpmodul9_2211104013.Models
{
    0 references
    public class Mahasiswa
    {
        0 references
        public string Nama { get; set; }
        0 references
        public string Nim { get; set; }
    }
}
```

c. File program.cs

```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

➢ Hasil running program



- **Penjelasan**

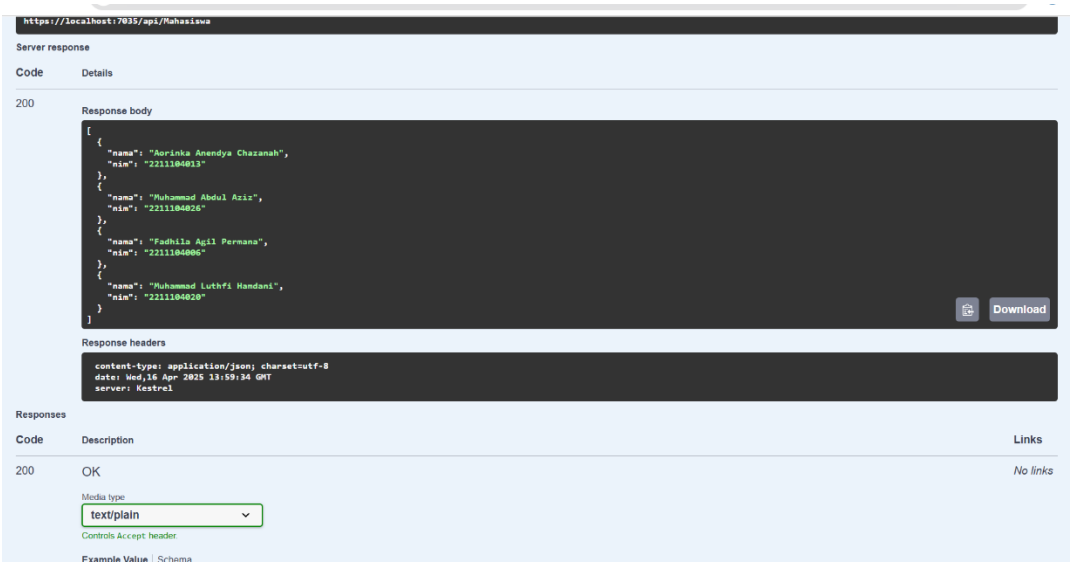
  Program ini merupakan implementasi Web API sederhana dengan ASP.NET Core yang berfungsi untuk mengelola data mahasiswa. Data disimpan dalam sebuah list statis bernama listMahasiswa, yang berisi informasi nama dan NIM beberapa mahasiswa. Pengelolaan permintaan HTTP dilakukan melalui sebuah controller bernama MahasiswaController.

  Controller ini menyediakan beberapa endpoint untuk melayani permintaan pengguna, seperti GET untuk menampilkan seluruh data mahasiswa atau data mahasiswa tertentu berdasarkan indeks, POST untuk menambahkan data baru, dan DELETE untuk menghapus data berdasarkan indeks. Setiap endpoint diberi atribut seperti [HttpGet], [HttpPost], dan [HttpDelete] untuk menyesuaikan dengan jenis permintaan yang diterima. Seluruh data disimpan secara sementara dalam memori aplikasi, sehingga data akan hilang ketika aplikasi dihentikan atau dijalankan ulang karena tidak menggunakan database permanen.

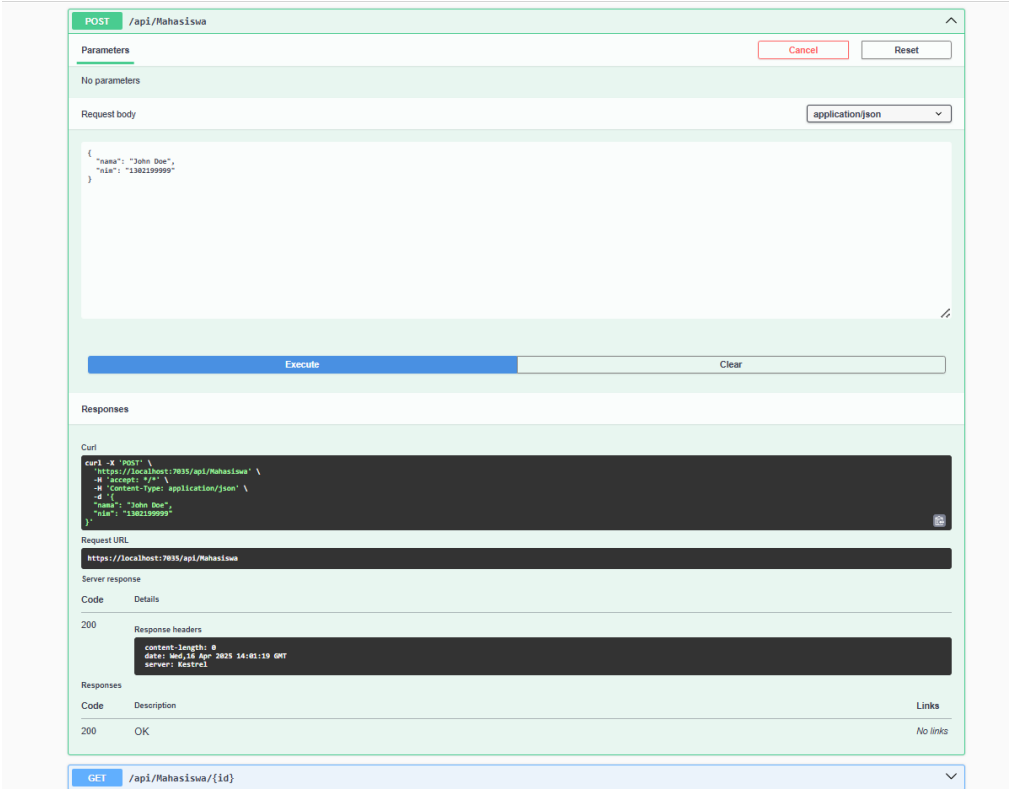
2. MENDEMONSTRASI WEB API

Beberapa skenario yang harus dicoba untuk memastikan jika program telah berjalan dengan baik. Buatlah dokumen yang berisi semua screenshot dari hasil uji coba scenario yang disebutkan pada list berikut ini:

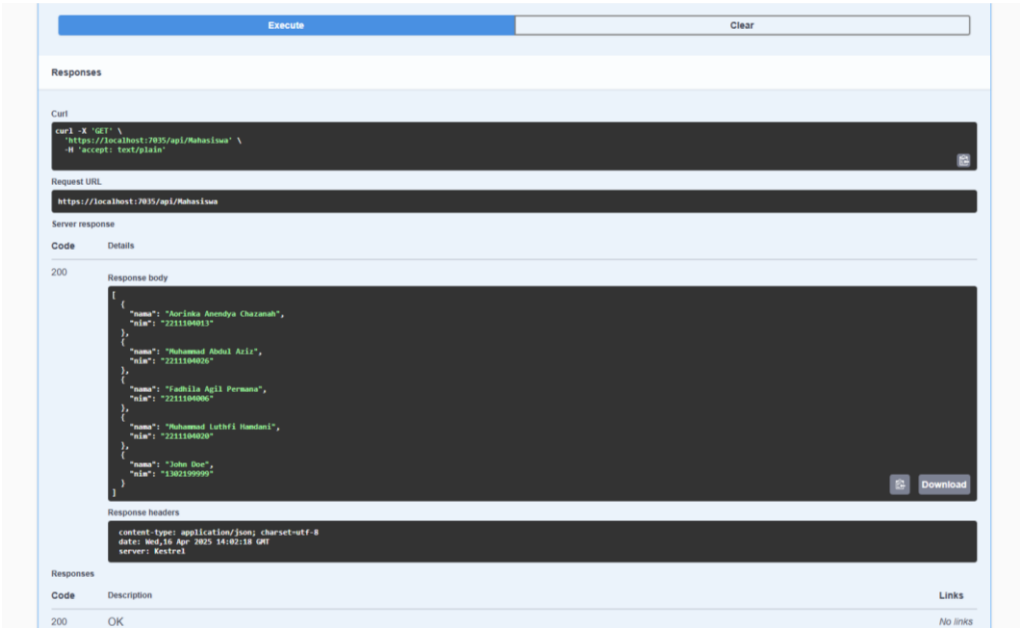
A. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok)



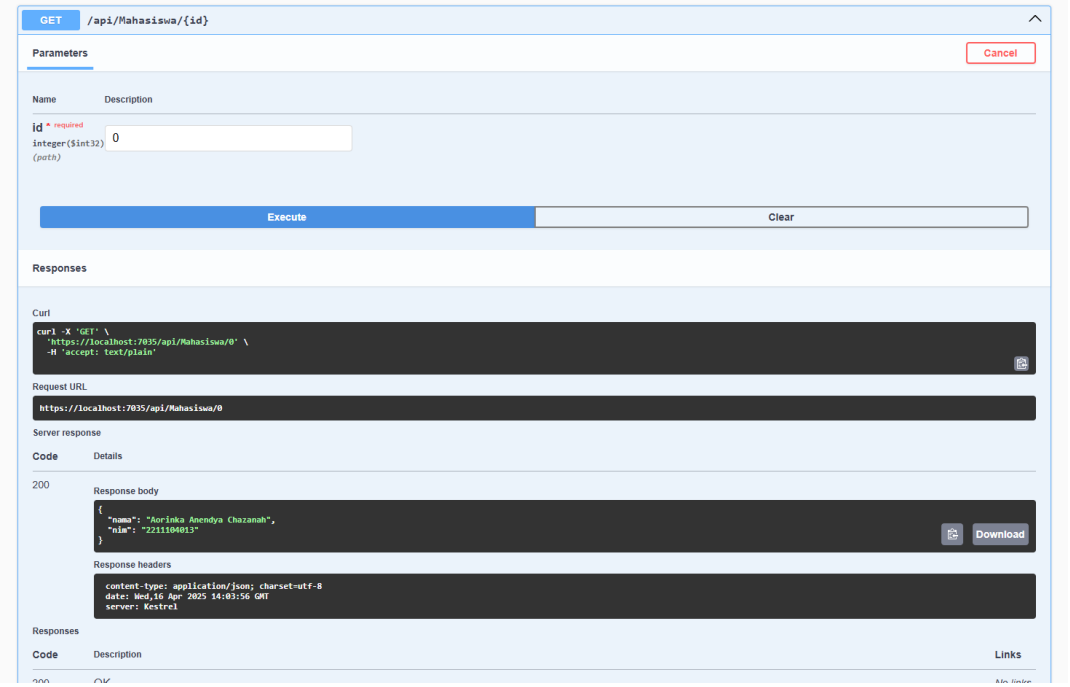
B. Menambahkan mahasiswa => Nama: “John Doe” dan NIM: “1302199999” dengan “POST /api/mahasiswa”



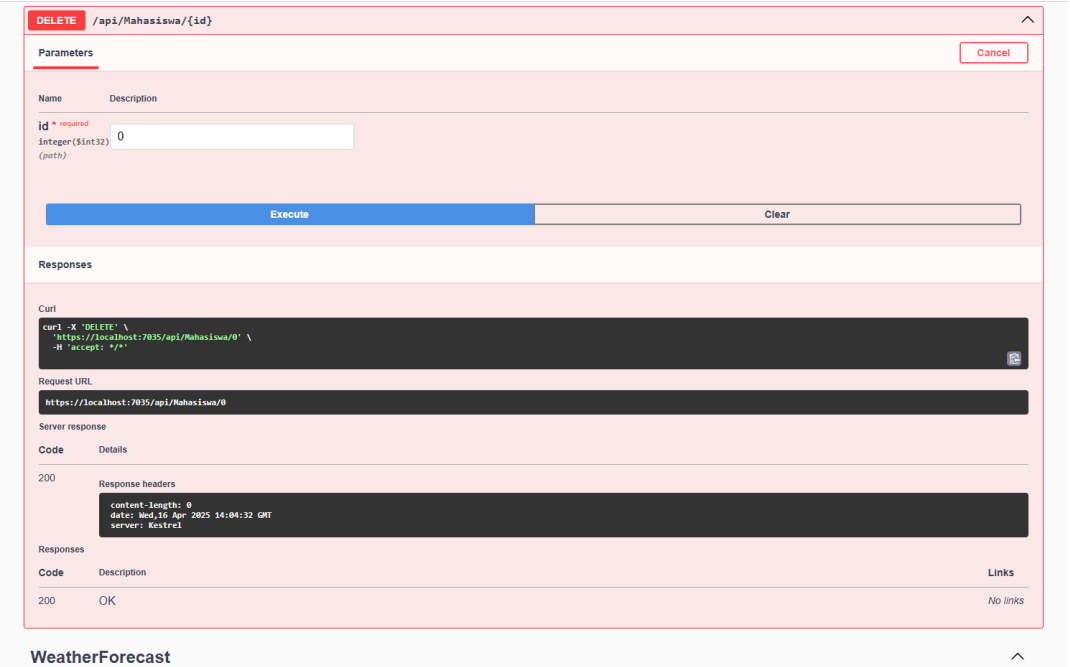
C. Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:



D. Mencoba meminta mahasiswa dengan index 0, “GET /api/mahasiswa/0” yang seharusnya mengeluarkan nama dan nim anda:



E. Menghapus objek mahasiswa dengan index ke-0 dengan “DELETE /api/mahasiswa/0”



F. Cek list/array dari semua mahasiswa sekali lagi dengan “GET /api/mahasiswa”, pastikan nama anda sudah tidak muncul di list tersebut: