

**RAKTIKUM KONSTRUKSI PERANGKAT LUNAK  
TUGAS PENDAHULUAN 09**

**Api Design & Construction Using Swagger**



**Telkom  
University**

disusun Oleh:  
Alfian Mutakim  
2211104024

SE0601

**S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY  
2025**

## 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.

Mahasiswa
- Nama: string
- Nim: string
+ 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:
- Nama: “LeBron James”, Nim: “1302000001”
  - Nama: “Stephen Curry”, Nim: “1302000002”
  - dst
- d. Gunakan teknologi API sehingga program tersebut dapat menerima HTTP request sebagai berikut:

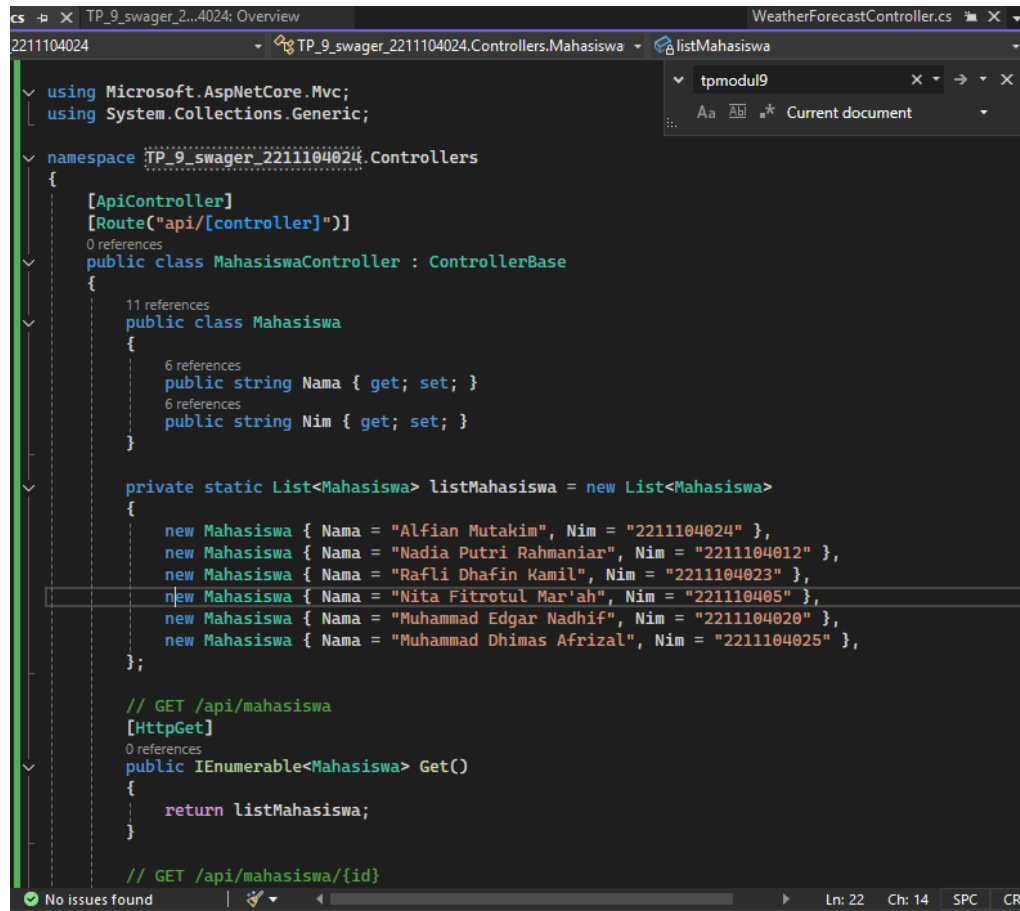
Mahasiswa	
GET	/api/Mahasiswa
POST	/api/Mahasiswa
GET	/api/Mahasiswa/{id}
DELETE	/api/Mahasiswa/{id}

- 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. Implementasi 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.

Jawab:

- Source Code

DataMahasiswa.cs



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

namespace TP_9_swager_2211104024.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 = "Alfian Mutakim", Nim = "2211104024" },
            new Mahasiswa { Nama = "Nadia Putri Rahmانيar", Nim = "2211104012" },
            new Mahasiswa { Nama = "Rafli Dhafin Kamil", Nim = "2211104023" },
            new Mahasiswa { Nama = "Nita Fitrotul Mar'ah", Nim = "221110405" },
            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}
    }
}
```

## 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();
}

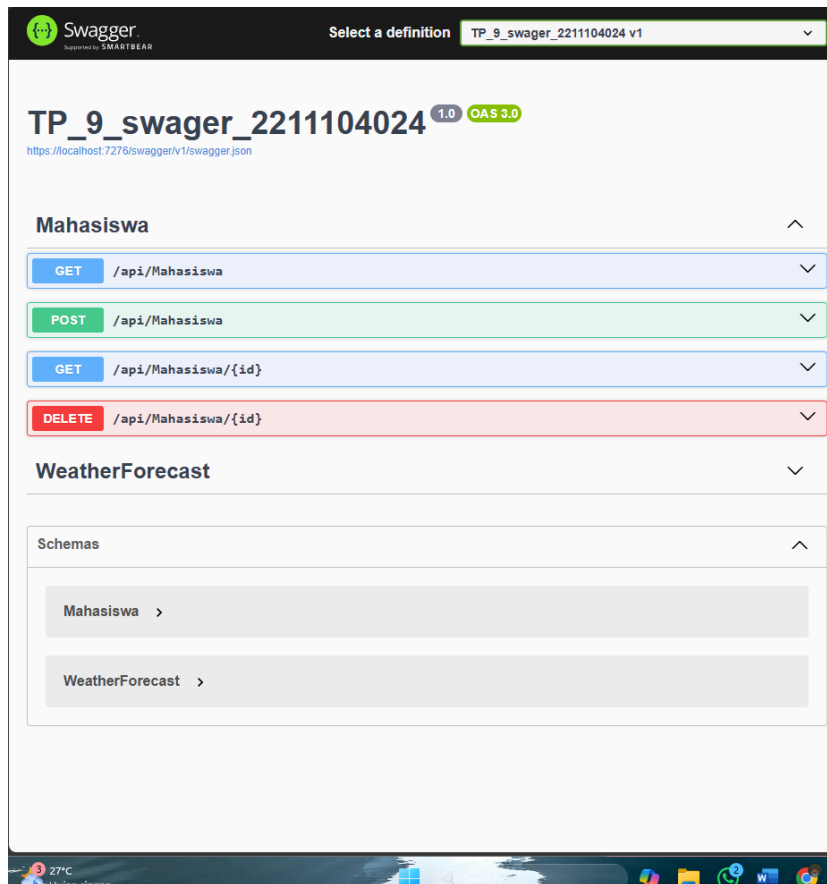
app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

### - Screenshot hasil run



### - Penjelasan

Aplikasi ini merupakan contoh sederhana dari implementasi Web API menggunakan ASP.NET Core, yang dirancang untuk mengelola informasi mahasiswa. Data mahasiswa disimpan sementara dalam sebuah list statis bernama listMahasiswa, yang berisi pasangan nama dan NIM.

Pengolahan permintaan HTTP dilakukan melalui sebuah controller bernama MahasiswaController. Controller ini menangani berbagai rute, seperti GET untuk mengambil data mahasiswa (baik seluruh data maupun berdasarkan indeks), POST untuk menambahkan data baru, dan DELETE untuk menghapus data mahasiswa berdasarkan indeks tertentu.

Setiap endpoint diatur menggunakan atribut seperti [HttpGet], [HttpPost], dan [HttpDelete], yang menunjukkan jenis permintaan HTTP yang didukung. Misalnya, permintaan ke GET /api/mahasiswa akan mengembalikan semua data dalam daftar, sementara POST /api/mahasiswa akan menambahkan data baru yang dikirim pengguna.

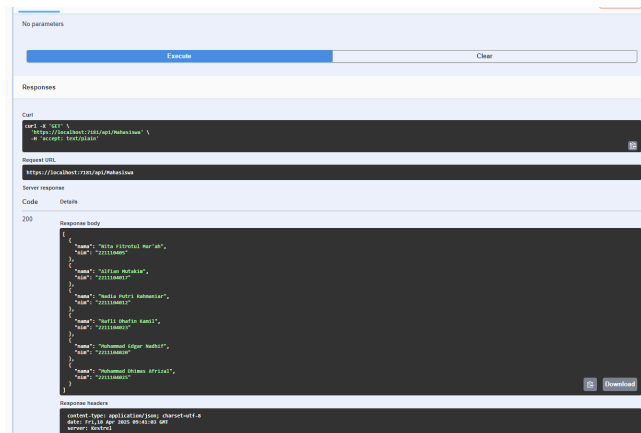
Perlu dicatat bahwa data mahasiswa hanya disimpan di memori dan tidak menggunakan basis data permanen. Artinya, data akan hilang saat aplikasi dihentikan atau dijalankan ulang, karena list bersifat statis dan tidak terhubung ke penyimpanan eksternal.

## 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 skenario yang disebutkan pada list berikut ini:

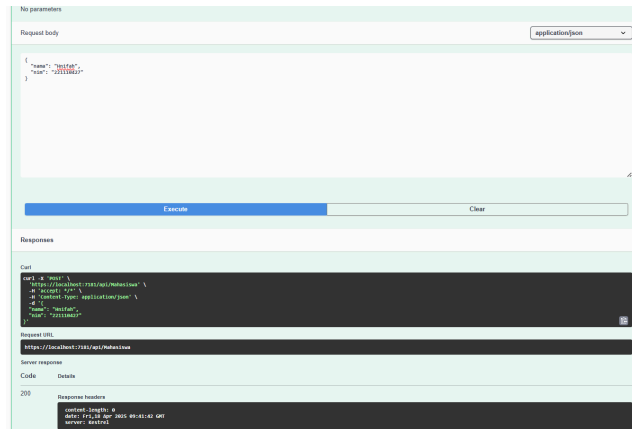
- Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok)

### Hasil



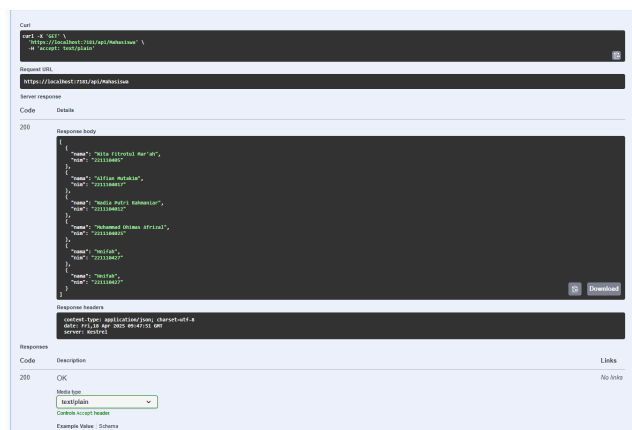
- Menambahkan mahasiswa => Nama: “John Doe” dan NIM: “1302199999” dengan “POST /api/mahasiswa”

### Hasil



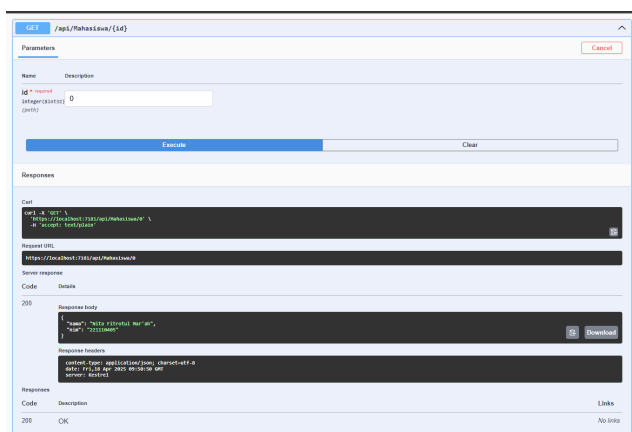
- c. Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:

## Hasil



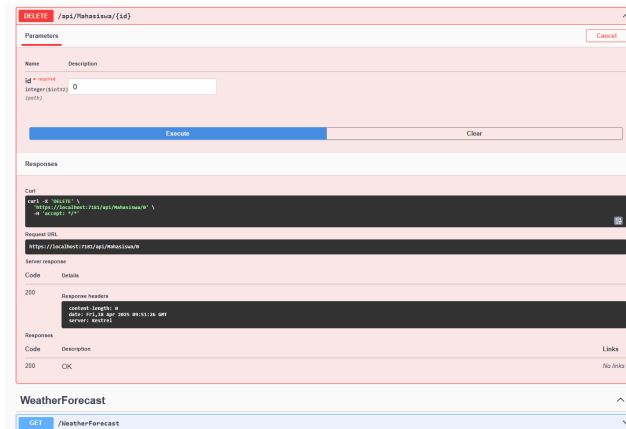
- d. Mencoba meminta mahasiswa dengan index 0, “GET /api/mahasiswa/0” yang seharusnya mengeluarkan nama dan nim anda:

## Hasil



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

## Hasil



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

## Hasil

