

**RAKTIKUM KONSTRUKSI PERANGKAT LUNAK
TUGAS PENDAHULUAN 09**

Api Design & Construction Using Swagger



**Telkom
University**

disusun Oleh:
Nita Fitrotul Mar'ah
2211104005

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 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);
            }
        }
    }
}
```

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();
}

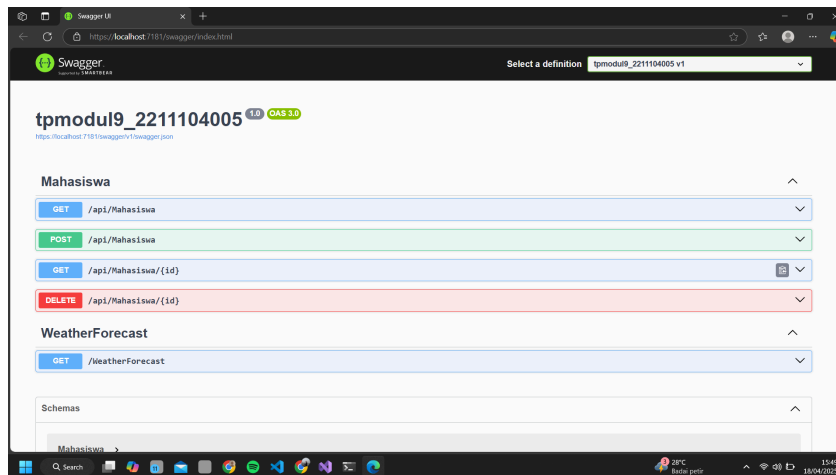
app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

- Screenshot hasil run



- Penjelasan

Program ini merupakan implementasi Web API sederhana menggunakan ASP.NET Core yang berfungsi untuk mengelola data mahasiswa. Data mahasiswa disimpan dalam sebuah list statis bernama listMahasiswa, yang berisi beberapa entri nama dan NIM.

Pengelolaan permintaan (request) HTTP dilakukan melalui sebuah controller bernama MahasiswaController. Controller ini menangani berbagai jenis endpoint: metode GET digunakan untuk menampilkan seluruh data mahasiswa atau mahasiswa berdasarkan indeks, POST untuk menambahkan data mahasiswa baru, serta DELETE untuk menghapus data berdasarkan indeks tertentu.

Setiap endpoint dikonfigurasi menggunakan atribut seperti [HttpGet], [HttpPost], dan [HttpDelete], yang menunjukkan tipe permintaan HTTP yang diterima. Contohnya,

ketika pengguna mengakses endpoint GET /api/mahasiswa, maka server akan mengembalikan seluruh isi daftar mahasiswa. Jika pengguna mengirimkan data melalui POST /api/mahasiswa, data tersebut akan ditambahkan ke dalam list.

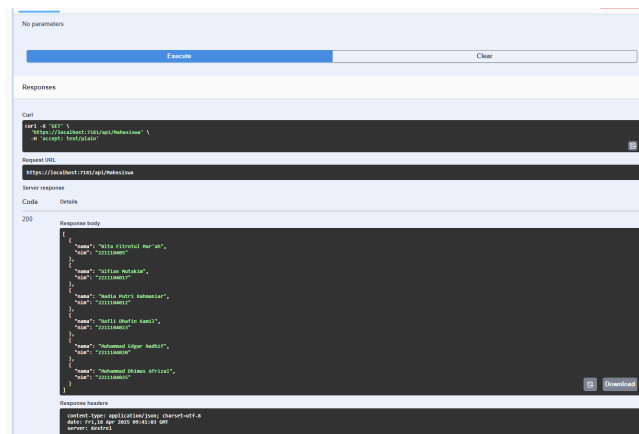
Perlu diketahui bahwa semua data disimpan secara sementara di dalam memori (tidak menggunakan basis data), sehingga data hanya tersedia selama aplikasi berjalan. Ketika server dimulai ulang, data akan kembali ke kondisi awal karena list bersifat statis.

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:

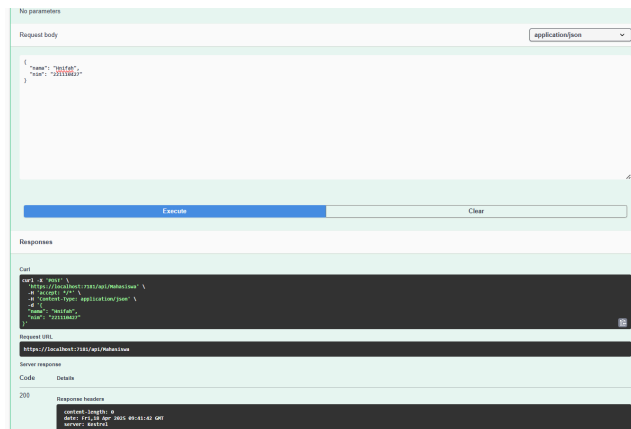
- Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok)

Hasil

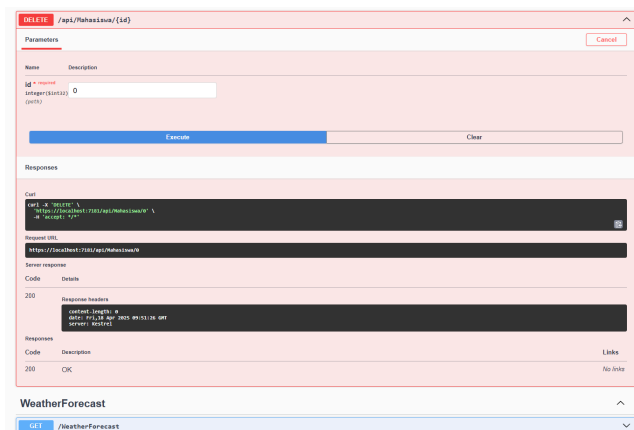


- Menambahkan mahasiswa => Nama: “John Doe” dan NIM: “1302199999” dengan “POST /api/mahasiswa”

Hasil



- ## Hasil



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

Hasil

