

**TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK**

**MODUL IX
API DESIGN & CONSTRUCTION USING SWAGGER**



**Disusun Oleh :
Atika Aji Hadiyani**

2211104003

SE06-01

**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.

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:

- i. Nama: “LeBron James”, Nim: “1302000001”
- i. Nama: “Stephen Curry”, Nim: “1302000002”
- iii. 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**

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

namespace tp_modul9.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 = "Atika Aji Hadiyani", Nim = "2211104003" },
            new Mahasiswa { Nama = "Kholifahdina", Nim = "2211104004" },
            new Mahasiswa { Nama = "Afad Fath Musyarof Halim", Nim = "2211104030" },
            new Mahasiswa { Nama = "Rachmat Taufik", Nim = "2211104028" },
            new Mahasiswa { Nama = "Noor Alam Fauzan", Nim = "2211104031" },
        };

        // GET /api/mahasiswa
        [HttpGet]
        public IEnumerable<Mahasiswa> Get()
        {
            return listMahasiswa;
        }
    }
}
```

```

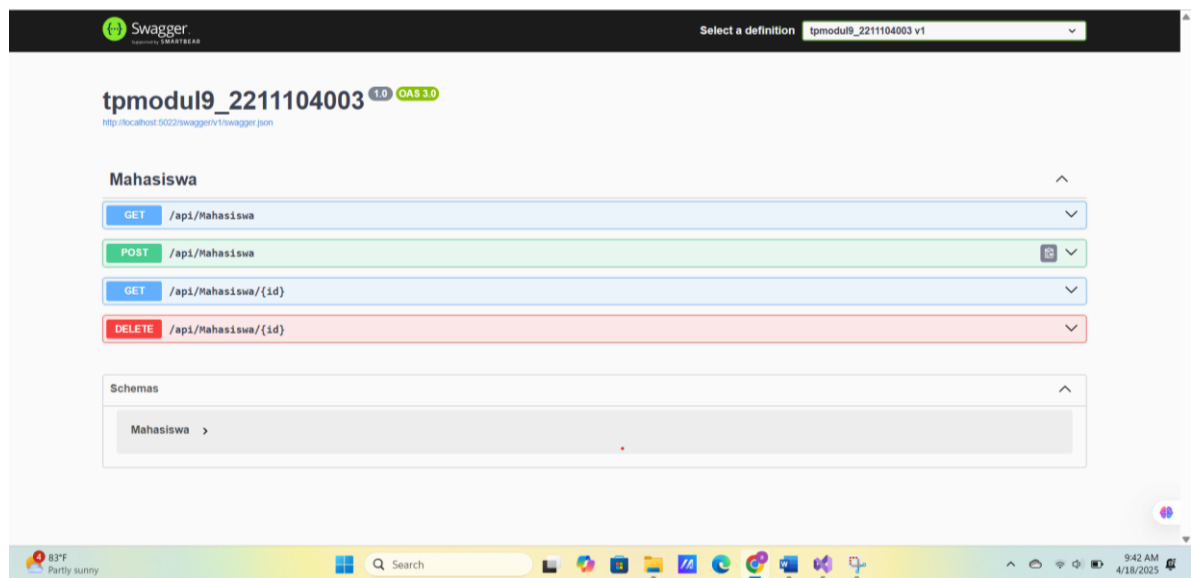
// GET /api/mahasiswa/{id}
[HttpGet("{id}")]
0 references
public ActionResult<Mahasiswa> Get(int id)
{
    if (id < 0 || id >= listMahasiswa.Count)
        return NotFound();
    return listMahasiswa[id];
}

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

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

```

- Screenshot hasil run



- Penjelasan

Program di atas merupakan implementasi Web API sederhana menggunakan ASP.NET Core yang difokuskan untuk mengelola informasi mahasiswa. Data mahasiswa disimpan dalam sebuah list statis bernama `listMahasiswa`, yang memuat elemen-elemen berupa nama dan NIM. Pengelolaan permintaan HTTP dilakukan oleh sebuah controller bernama `MahasiswaController`. Controller ini menyediakan beberapa endpoint: metode GET digunakan untuk menampilkan seluruh data mahasiswa atau data mahasiswa tertentu berdasarkan indeks, POST untuk menambahkan data baru, dan DELETE untuk menghapus data berdasarkan indeks yang diberikan.

Setiap endpoint dikaitkan dengan atribut penanda jenis permintaan HTTP seperti `[HttpGet]`, `[HttpPost]`, dan `[HttpDelete]`. Contohnya, permintaan ke endpoint GET `/api/mahasiswa` akan menghasilkan respon berupa seluruh isi list mahasiswa. Sementara itu, permintaan POST `/api/mahasiswa` dengan data mahasiswa baru akan menambahkan data tersebut ke dalam list. Perlu dicatat bahwa penyimpanan data dilakukan di dalam memori menggunakan struktur data list statis, sehingga data tidak bersifat permanen dan akan hilang apabila aplikasi dimatikan atau server mengalami restart.

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:

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

Hasil:



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

Hasil:

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** /api/mahasiswa
- Parameters:** No parameters
- Request body:** application/json
- Request body content:**

```
{  "nama": "John Doe",  "nim": "1302199999"}
```
- Execute button:** A blue button labeled "Execute".
- Responses:**
 - Request URL:** http://localhost:5022/api/mahasiswa
 - Server response:**
 - Code:** 200
 - Details:** Response headers: content-length: 0, date: Fri, 18 Apr 2025 02:46:42 GMT, server: Kestrel
 - Responses table:**

Code	Description	Links
200	OK	No links

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

Hasil:

The screenshot shows a REST client interface with the following details:

- Request URL:** http://localhost:5022/api/mahasiswa
- Server response:**
 - Code:** 200
 - Details:** Response body
- Response body content:**

```
{  {    "nama": "Atika Aji Hadiyahani",    "nim": "2211104003"  },  {    "nama": "Kholifahdina",    "nim": "2211104004"  },  {    "nama": "Afad Fath Musyarof Halim",    "nim": "2211104030"  },  {    "nama": "Rachmat Taufik",    "nim": "2211104028"  },  {    "nama": "Moer Alim Fauzan",    "nim": "2211104031"  },  {    "nama": "John Doe",    "nim": "1302199999"  }}
```
- Download button:** A button labeled "Download".

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

Hasil:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /api/mahasiswa/{id}
- Parameters:**
 - Name:** id (required)
 - Description:** Integer(\$int32) (path)
 - Value:** 0
- Buttons:** Execute, Clear
- Responses:**
 - Code:** 200
 - Response body:**

```
{  "nama": "Atika Aji Hadiyah",  "nim": "2211104003"}
```
 - Response headers:**

```
content-type: application/json; charset=utf-8date: Fri, 18 Apr 2025 03:06:09 GMTserver: Kestreltransfer-encoding: chunked
```

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

Hasil:

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** /api/mahasiswa/{id}
- Parameters:**
 - Name:** id (required)
 - Description:** Integer(\$int32) (path)
 - Value:** 0
- Buttons:** Execute, Clear
- Responses:**
 - Code:** 200
 - Response headers:**

```
content-length: 0date: Fri, 18 Apr 2025 03:07:17 GMTserver: Kestrel
```

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

Hasil:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /api/Mahasiswa
- Parameters:** No parameters
- Request URL:** http://localhost:5022/api/Mahasiswa
- Response Code:** 200
- Response Body:** A JSON array of student objects.

```
{
  "nama": "Kholifahdina",
  "nia": "2211104004"
},
{
  "nama": "Afad Fath Musyarof Halin",
  "nia": "2211104030"
},
{
  "nama": "Rachmat Taufik",
  "nia": "2211104028"
},
{
  "nama": "Noor Alam Fauzan",
  "nia": "2211104031"
},
{
  "nama": "John Doe",
  "nia": "1502199999"
}
]
```
- Response Headers:**

```
content-type: application/json; charset=utf-8
date: Fri, 18 Apr 2025 03:12:19 GMT
server: Kestrel
transfer-encoding: chunked
```