

Nama : Naufal Aflakh Wijayanto

Kelas : SE063

NIM : 2211104073

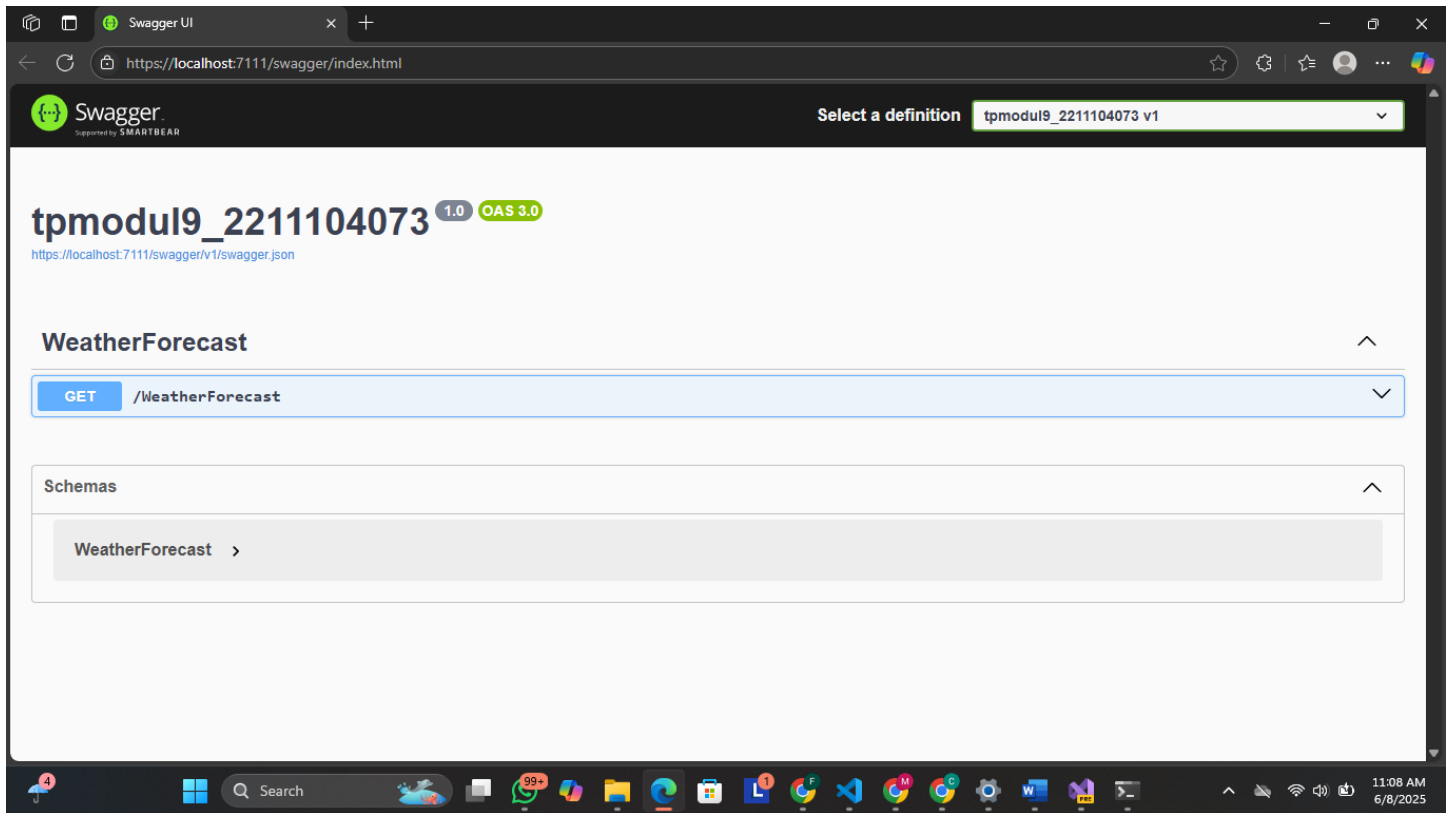
## TP MODUL 9

### 1. MEMBUAT PROJECT WEB API

Berhubung cara membuat project web api berbeda-beda untuk setiap bahasa pemrograman, langkah-langkah berikut hanya berlaku apabila dilakukan dengan menggunakan .NET dan Visual Studio. Untuk IDE dan bahasa pemrograman lain, yang terpenting adalah nama project yang dibuat yaitu "tpmodul8\_NIM".

- A. Buka visual studio yang sudah terinstall dengan ASP.NET dan .NET 5.0 SDK atau setelahnya
- B. Pilih New Project dan kemudian pilih ASP.NET Core Web API atau API (pastikan opsi 'Enable OpenAPI support' tercentang).
- C. Pastikan untuk memilih .NET versi 5.0 atau yang lebih baru.
- D. Masukkan nama projek "tpmodul9\_NIM".
- E. Langkah-langkah yang disertai gambar dapat dilihat pada link berikut ini (cukup dilihat pada bagian "Create a Web API project"):  
<https://docs.microsoft.com/en-us/aspnet/core/tutorials/min-web-api?view=aspnetcore-6.0&tabs=visual-studio>
- F. Setelah project tersebut selesai dibuat, coba run programnya, dan tunggu sampai program selesai di-compile.

## BUKTI Pengerjaan



## 2. 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.

### **MahasiswaController.cs**

```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using tpmodul9_2211104073.Models;

namespace tpmodul9_2211104073.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class MahasiswaController : ControllerBase
    {
        // Daftar mahasiswa statis
        private static List<Mahasiswa> mahasiswaList = new List<Mahasiswa>
        {
            new Mahasiswa("Naufal Aflakh Wijayanto", "2211104073"),
            new Mahasiswa("Muhammad Fariz Nur Hidayat", "2211104069"),
            new Mahasiswa("Putra Pratama Okta Riano", "2211104068"),
            new Mahasiswa("Allaya Daffa Zhilal", "2211104090")
        };

        // Endpoint GET untuk mengambil daftar mahasiswa
        [HttpGet]
        public ActionResult<IEnumerable<Mahasiswa>> GetMahasiswa()
        {
            return Ok(mahasiswaList);
        }

        // Endpoint GET untuk mengambil mahasiswa berdasarkan index
        [HttpGet("{index}")]
        public ActionResult<Mahasiswa> GetMahasiswaByIndex(int index)
        {
            if (index < 0 || index >= mahasiswaList.Count)
            {
                return NotFound(); // Mengembalikan 404 jika index tidak ditemukan
            }

            return Ok(mahasiswaList[index]);
        }

        // Endpoint POST untuk menambahkan mahasiswa baru
        [HttpPost]
        public ActionResult AddMahasiswa([FromBody] Mahasiswa mahasiswa)
        {
            mahasiswaList.Add(mahasiswa);
            return CreatedAtAction(nameof(GetMahasiswa), new { index = mahasiswaList.Count - 1 }, mahasiswa);
        }
    }
}
```

```
// Endpoint DELETE untuk menghapus mahasiswa berdasarkan index
[HttpDelete("{index}")]
public ActionResult DeleteMahasiswa(int index)
{
    if (index < 0 || index >= mahasiswaList.Count)
    {
        return NotFound(); // Mengembalikan 404 jika index tidak ditemukan
    }

    mahasiswaList.RemoveAt(index);
    return NoContent(); // Mengembalikan status 204 jika penghapusan berhasil
}
}
```

### PENJELASAN KODE:

Kode di atas mendefinisikan sebuah **API controller** bernama MahasiswaController dalam aplikasi ASP.NET Core yang berfungsi untuk mengelola data mahasiswa. Controller ini menggunakan daftar statis mahasiswaList yang berisi objek Mahasiswa, yang masing-masing memiliki atribut Nama dan Nim.

Terdapat beberapa endpoint API:

- **GET /api/mahasiswa**: Mengembalikan seluruh daftar mahasiswa dalam bentuk JSON.
- **GET /api/mahasiswa/{index}**: Mengembalikan data mahasiswa berdasarkan index yang diberikan, dengan pengecekan validitas index untuk menghindari error (mengembalikan 404 jika index tidak valid).
- **POST /api/mahasiswa**: Menambahkan mahasiswa baru ke dalam daftar berdasarkan data yang dikirimkan dalam body request, dan mengembalikan status 201 (Created) setelah penambahan.
- **DELETE /api/mahasiswa/{index}**: Menghapus mahasiswa berdasarkan index yang diberikan dan mengembalikan status 204 (No Content) setelah penghapusan, dengan pengecekan validitas index.

Controller ini menggunakan ActionResult untuk menangani respons HTTP yang dikirimkan kepada pengguna, baik itu data mahasiswa maupun status error seperti **404 Not Found** atau **204 No Content**.

### Mahasiswa.cs

```
namespace tpmodul9_2211104073.Models
{
    public class Mahasiswa
    {
        public string Nama { get; set; }
        public string Nim { get; set; }

        // Constructor untuk Mahasiswa
        public Mahasiswa(string nama, string nim)
        {
            Nama = nama;
            Nim = nim;
        }
    }
}
```

## PENJELASAN KODE

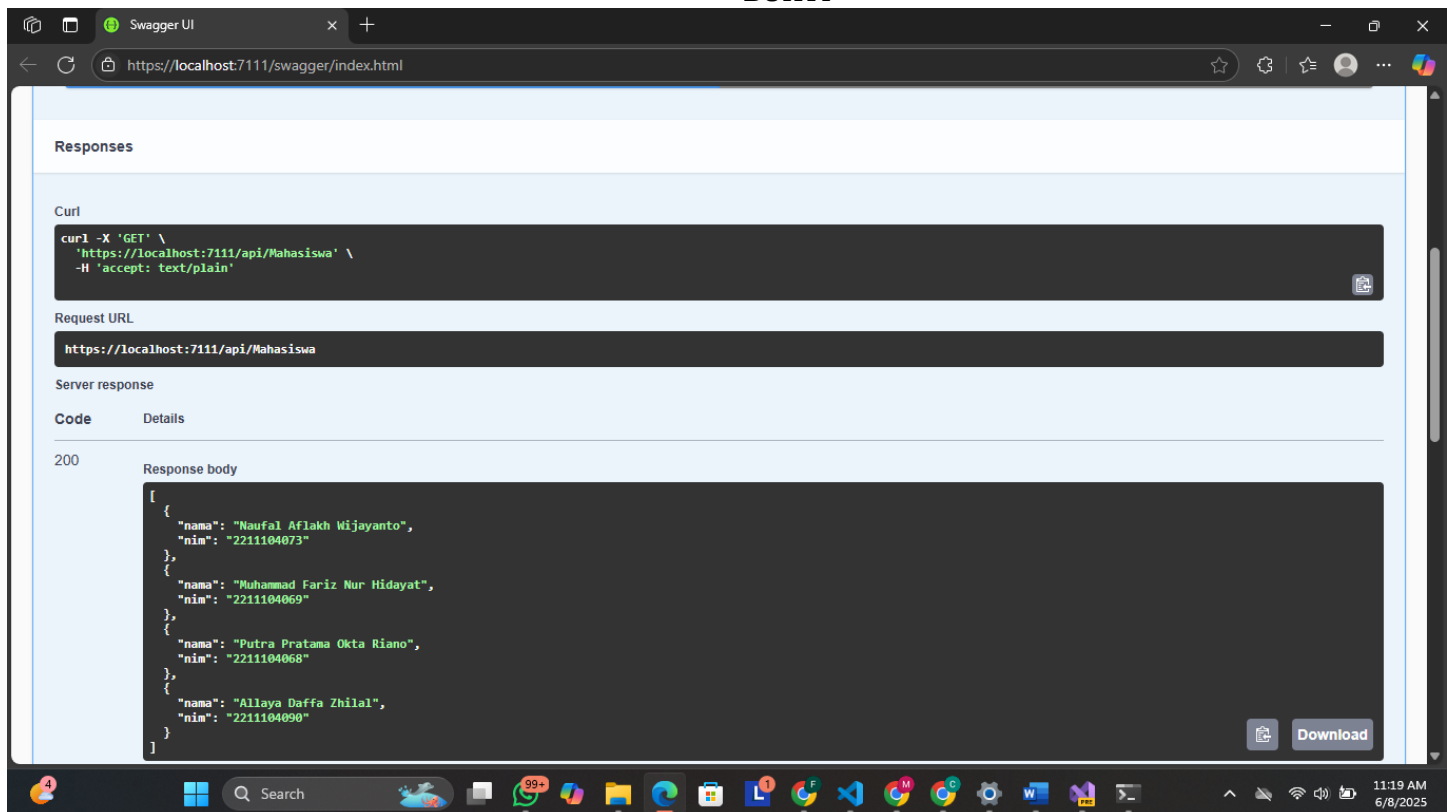
Kode di atas mendefinisikan sebuah kelas Mahasiswa dalam namespace `tpmodul9_2211104073.Models` yang merepresentasikan data dasar dari seorang mahasiswa. Kelas ini memiliki dua properti, yaitu Nama dan Nim (Nomor Induk Mahasiswa), yang masing-masing bertipe string. Untuk menginisialisasi nilai properti tersebut, digunakan sebuah *constructor* yang menerima dua parameter (nama dan nim). Kelas ini berguna dalam konteks aplikasi yang membutuhkan pengelolaan data mahasiswa secara terstruktur, seperti dalam sistem akademik atau API pendidikan.

### 3. 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:

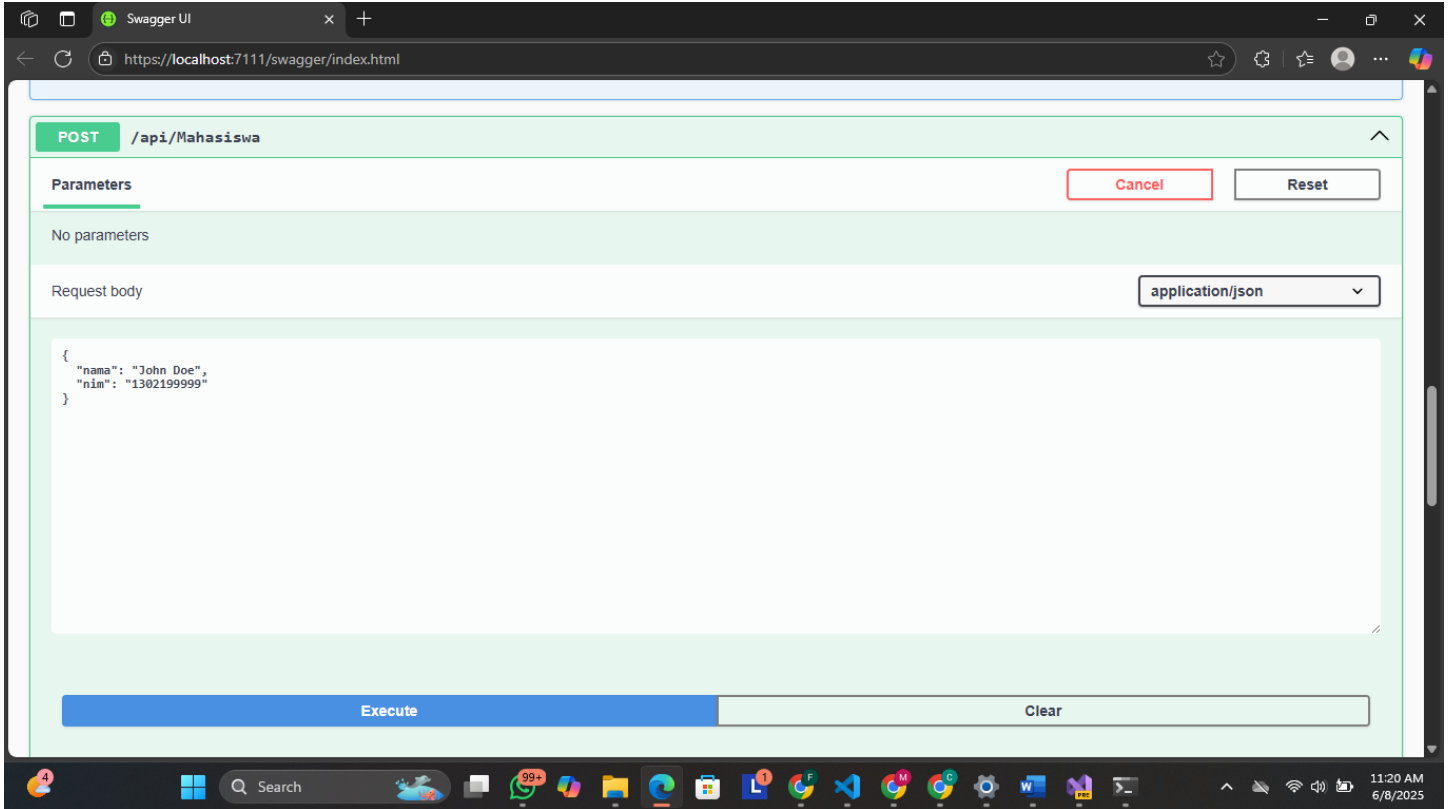
- A. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok):

#### BUKTI



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

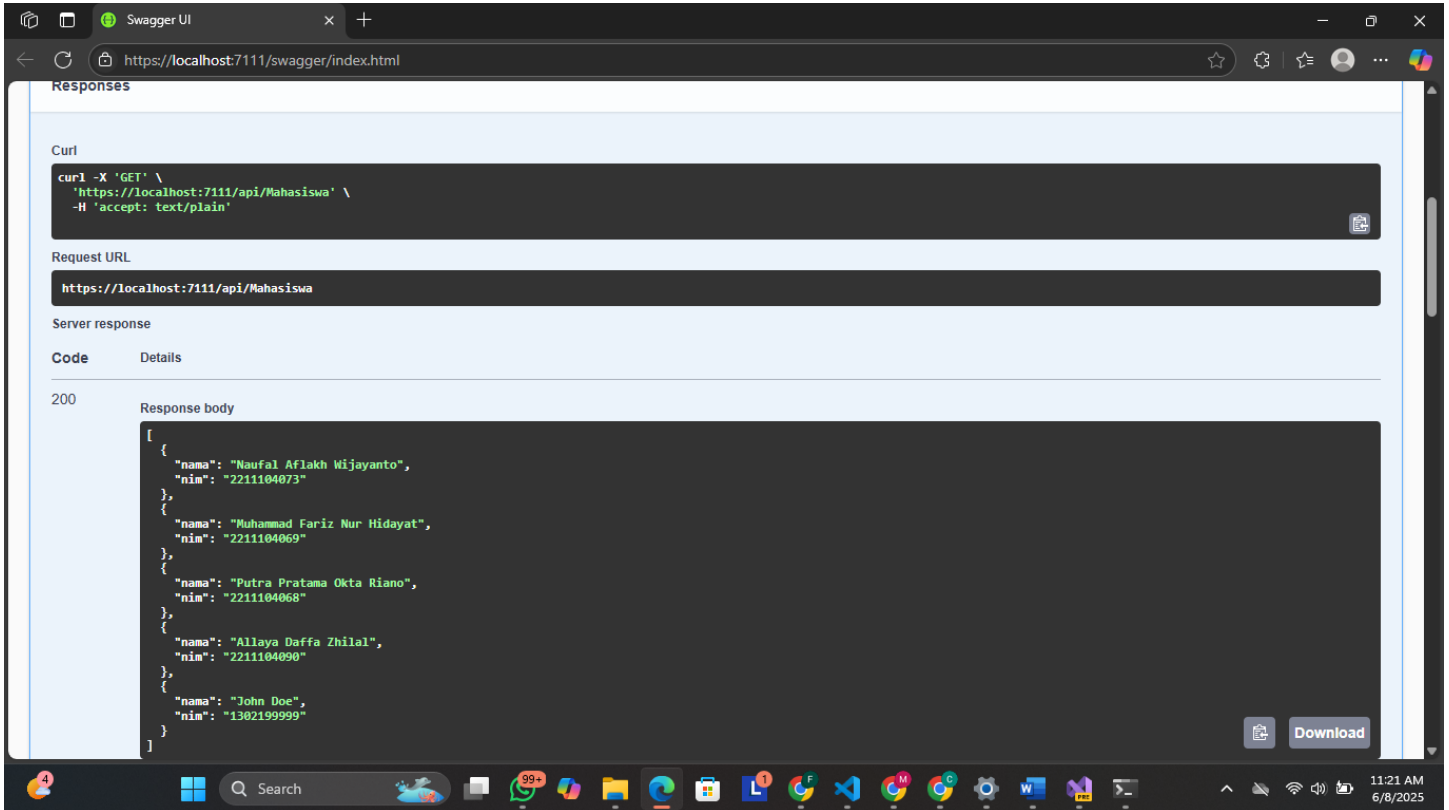
**BUKTI**





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

### BUKTI



The screenshot shows the Swagger UI interface in a web browser. The URL bar indicates the endpoint is `https://localhost:7111/swagger/index.html`. The 'Responses' section is expanded, showing the details for a 200 status code. The 'Request URL' is `https://localhost:7111/api/Mahasiswa`. The 'Server response' section shows a 200 status code and a JSON array of student data in the 'Response body'.

**Curl**

```
curl -X 'GET' \
  'https://localhost:7111/api/Mahasiswa' \
  -H 'accept: text/plain'
```

**Request URL**

```
https://localhost:7111/api/Mahasiswa
```

**Server response**

**Code** 200

**Details**

**Response body**

```
[
  {
    "nama": "Naufal Aflakh Wijayanto",
    "nim": "2211104073"
  },
  {
    "nama": "Muhammad Fariz Nur Hidayat",
    "nim": "2211104069"
  },
  {
    "nama": "Putra Pratama Okta Riano",
    "nim": "2211104068"
  },
  {
    "nama": "Alleya Daffa Zhilal",
    "nim": "2211104090"
  },
  {
    "nama": "John Doe",
    "nim": "1302199999"
  }
]
```

The Windows taskbar at the bottom shows the system clock as 11:21 AM on 6/8/2025.

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

### BUKTI

The screenshot shows the Swagger UI interface in a web browser. The URL bar indicates the page is at `https://localhost:7111/swagger/index.html`. The main content area displays the API endpoint `index` with a required `integer ($int32)` parameter in the path, which has been set to `0`. Below the parameter field are `Execute` and `Clear` buttons. The `Responses` section shows a `200` status code with a response body containing the student's name and NIM: `{ "nama": "Naufal Aflakh Wijayanto", "nim": "2211104073" }`. A `Download` button is visible next to the response body. The bottom of the image shows the Windows taskbar with various application icons and the system clock indicating 11:21 AM on 6/8/2025.

Name	Description
<code>index</code> * required <code>integer (\$int32)</code> (path)	<input type="text" value="0"/>

`Execute` `Clear`

**Responses**

Curl

```
curl -X 'GET' \
  'https://localhost:7111/api/Mahasiswa/0' \
  -H 'accept: text/plain'
```

Request URL

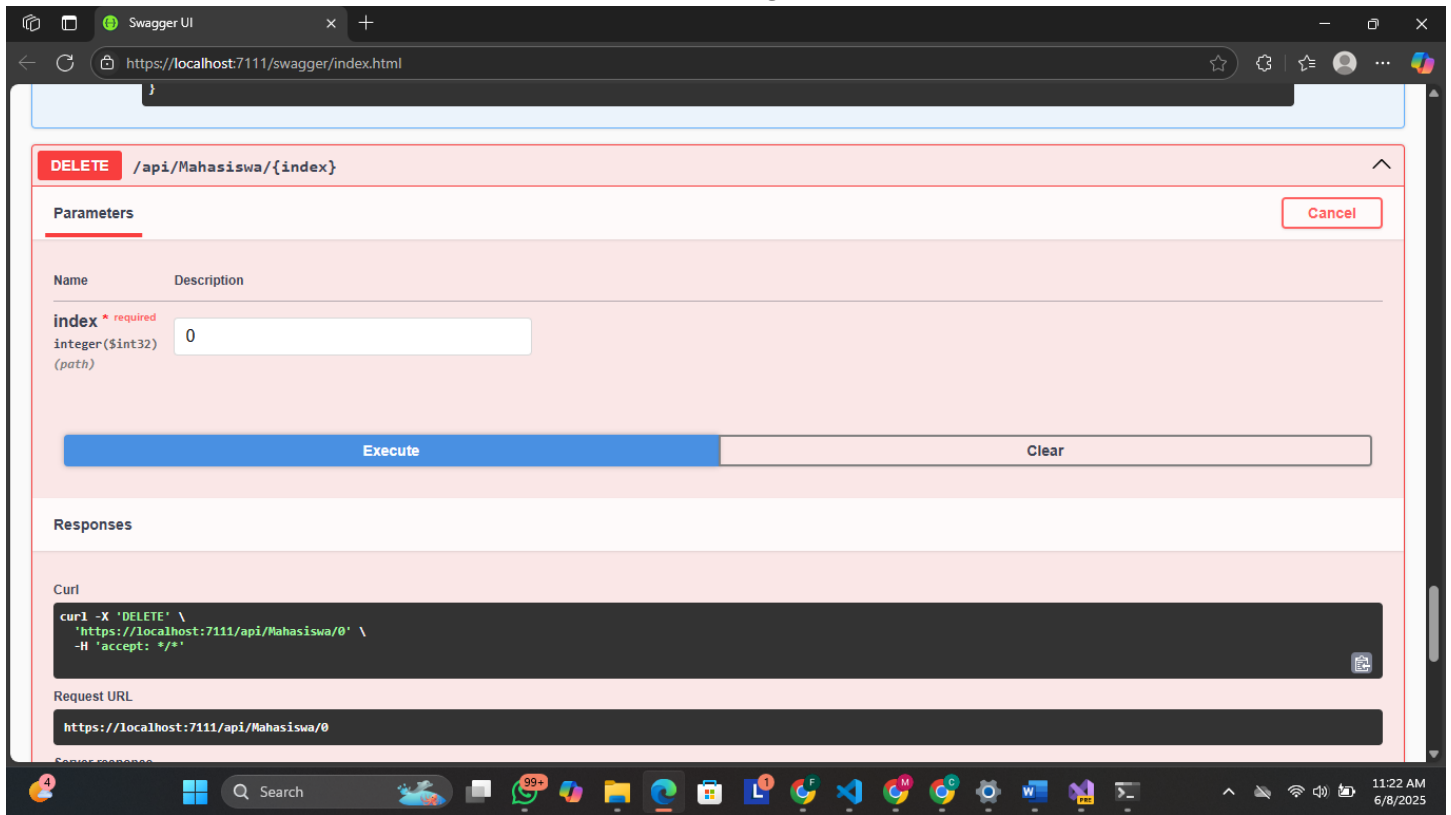
```
https://localhost:7111/api/Mahasiswa/0
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "nama": "Naufal Aflakh Wijayanto",   "nim": "2211104073" }</pre> <code>Download</code>

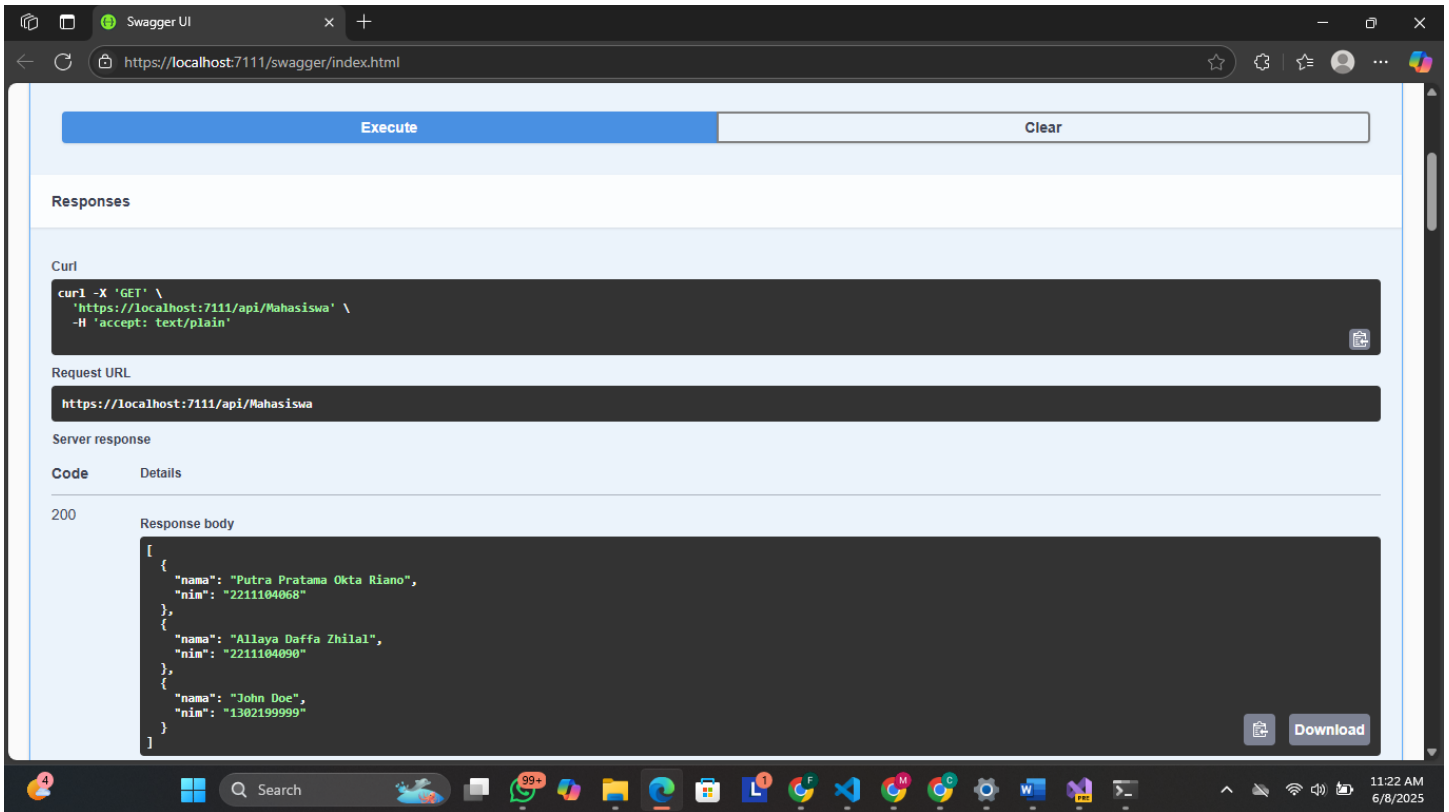
E. Menghapus objek mahasiswa dengan index ke-0 dengan “DELETE /api/mahasiswa/0”

### BUKTI



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

### BUKTI



The screenshot shows the Swagger UI interface in a web browser. The URL bar indicates the endpoint is `https://localhost:7111/swagger/index.html`. The interface includes an "Execute" button and a "Clear" button. Below these, the "Responses" section is visible. The "Curl" section shows the command used to execute the request: `curl -X 'GET' \ 'https://localhost:7111/api/Mahasiswa' \ -H 'accept: text/plain'`. The "Request URL" section shows the endpoint: `https://localhost:7111/api/Mahasiswa`. The "Server response" section shows a status code of 200. The "Response body" section displays a JSON array of three student objects. The first two objects have names and NIMs that match the user's information, while the third object is a placeholder "John Doe".

```
{
  {
    "nama": "Putra Pratama Okta Riano",
    "nim": "2211104068"
  },
  {
    "nama": "Allaya Daffa Zhilal",
    "nim": "2211104090"
  },
  {
    "nama": "John Doe",
    "nim": "1302199999"
  }
}
```

The Windows taskbar at the bottom shows the system time as 11:22 AM on 6/8/2025.