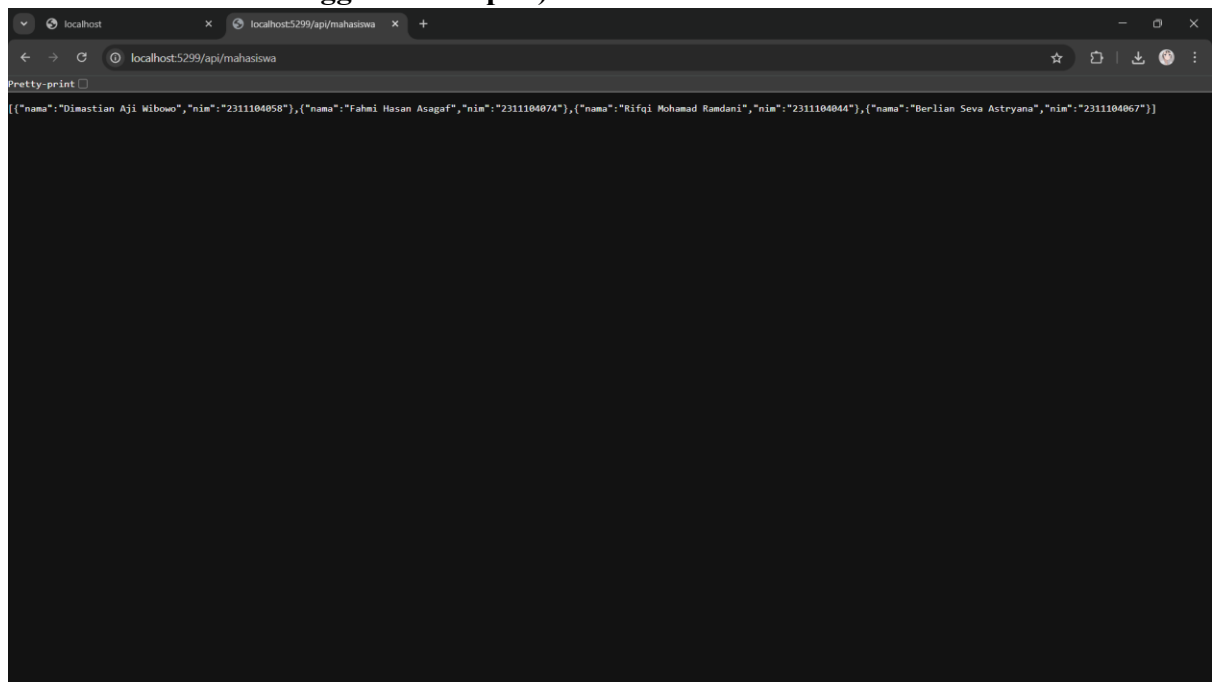


Nama : Dimastian Aji Wibowo

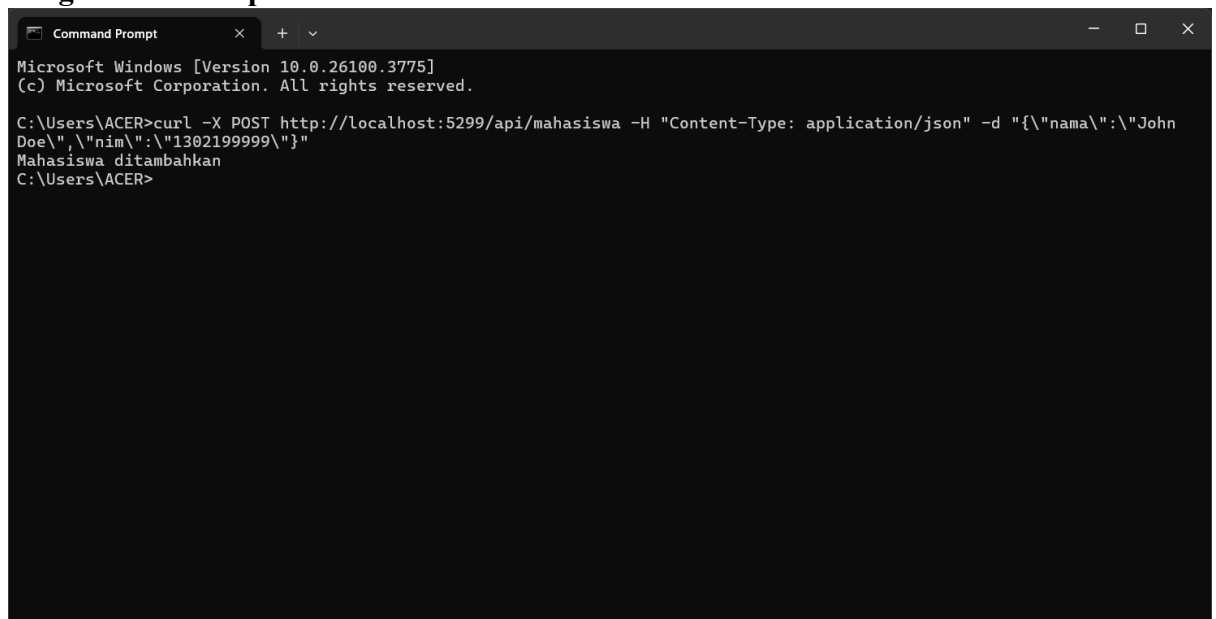
NIM : 2311104058

TP MODUL 9

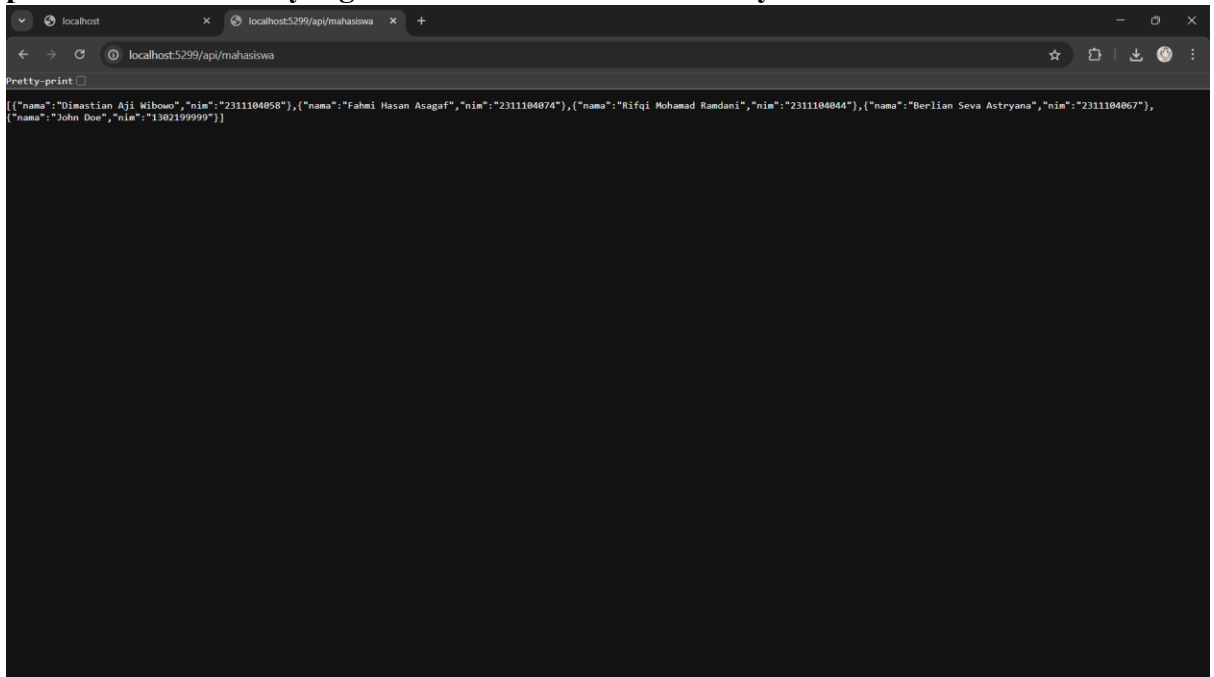
- A. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok):**



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

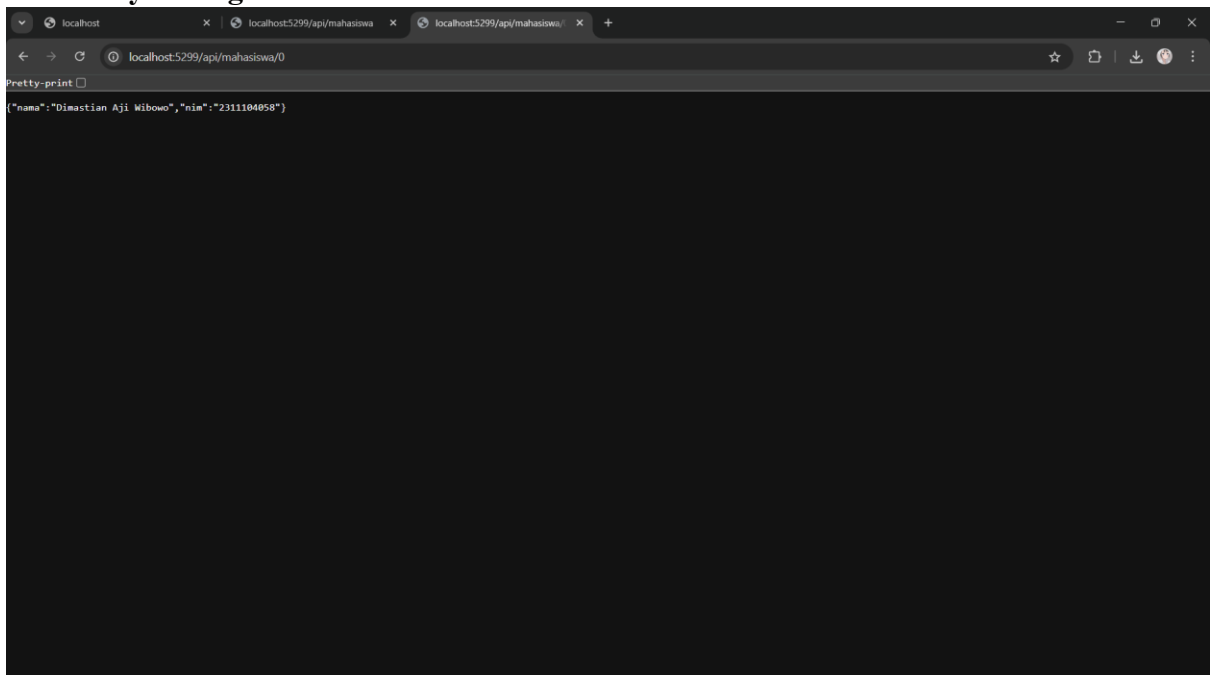


- C. Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:**



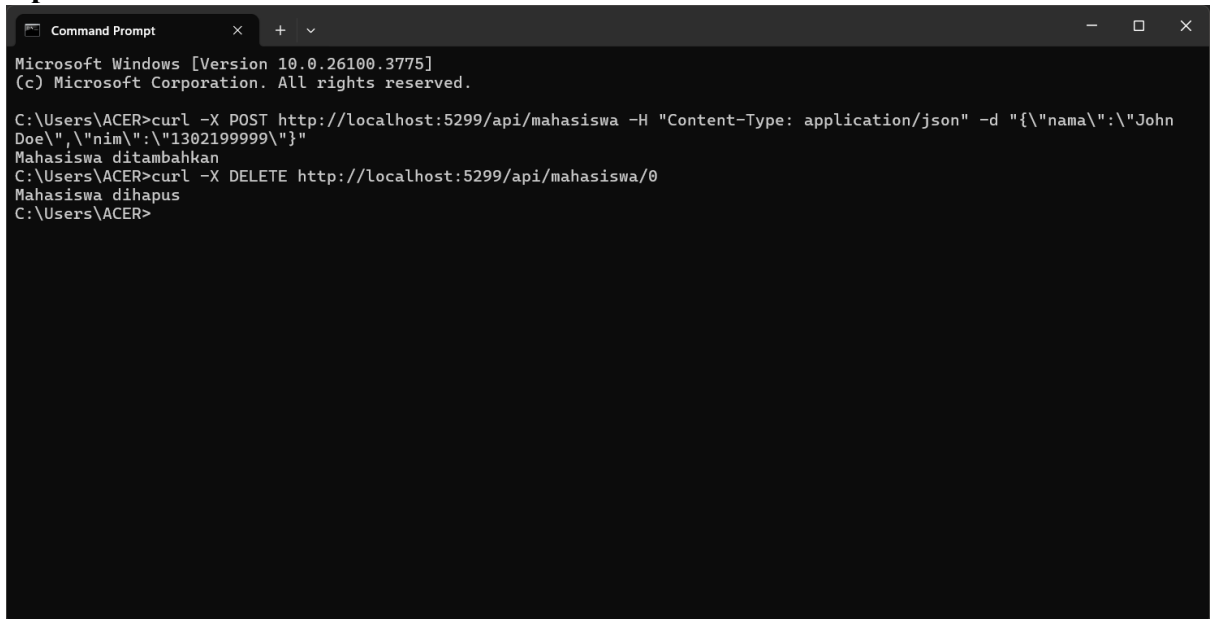
```
[[{"nama": "Dimastian Aji Wibowo", "nim": "2311104058"}, {"nama": "Fahmi Hasan Asagaf", "nim": "2311104074"}, {"nama": "Rifqi Mohamad Randani", "nim": "2311104044"}, {"nama": "Berlian Seva Astryana", "nim": "2311104067"}, {"nama": "John Doe", "nim": "1302199999"}]]
```

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



```
{"nama": "Dimastian Aji Wibowo", "nim": "2311104058"}
```

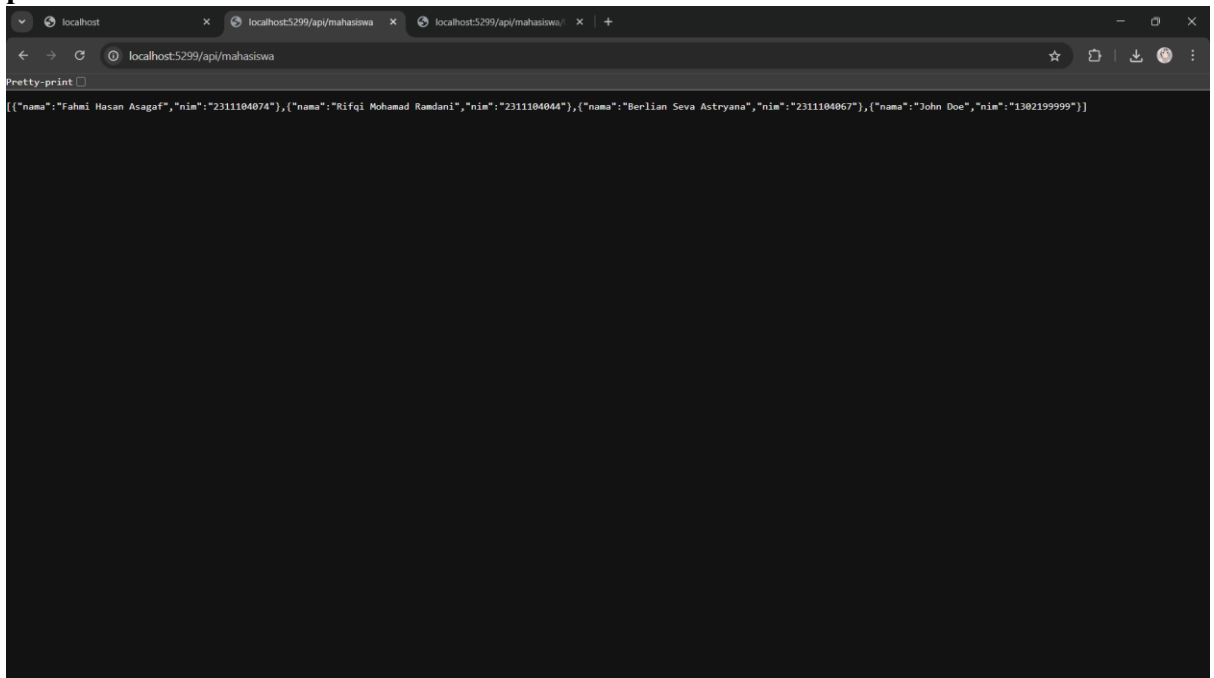
E. Menghapus objek mahasiswa dengan index ke-0 dengan “DELETE /api/mahasiswa/0”



```
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ACER>curl -X POST http://localhost:5299/api/mahasiswa -H "Content-Type: application/json" -d '{"nama":"John Doe","nim":"1302199999"}'
Mahasiswa ditambahkan
C:\Users\ACER>curl -X DELETE http://localhost:5299/api/mahasiswa/0
Mahasiswa dihapus
C:\Users\ACER>
```

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



```
localhost:5299/api/mahasiswa
[{"nama":"Fahmi Hasan Asagaf","nim":"2311104074"}, {"nama":"Rifqi Mohamad Randani","nim":"2311104044"}, {"nama":"Berlian Seva Astryana","nim":"2311104067"}, {"nama":"John Doe","nim":"1302199999"}]
```

Source Code

MahasiswaController.cs

```
using Microsoft.AspNetCore.Mvc;
using System.Collections.Generic;
using tpmodu19_2311104058;

namespace MahasiswaApi.Controllers
{
    [ApiController]
```

```

[Route("api/[controller]")]
public class MahasiswaController : ControllerBase
{
    private static List<Mahasiswa> mahasiswaList = new List<Mahasiswa>
    {
        new Mahasiswa("Dimastian Aji Wibowo", "2311104058"),
        new Mahasiswa("Fahmi Hasan Asagaf", "2311104074"),
        new Mahasiswa("Rifqi Mohamad Ramdani", "2311104044"),
        new Mahasiswa("Berlian Seva Astryana", "2311104067")
    };

    [HttpGet]
    public ActionResult<List<Mahasiswa>> GetAll()
    {
        return mahasiswaList;
    }

    [HttpGet("{index}")]
    public ActionResult<Mahasiswa> GetByIndex(int index)
    {
        if (index < 0 || index >= mahasiswaList.Count)
            return NotFound("Index tidak valid");

        return mahasiswaList[index];
    }

    [HttpPost]
    public ActionResult AddMahasiswa([FromBody] Mahasiswa mahasiswaBaru)
    {
        mahasiswaList.Add(mahasiswaBaru);
        return Ok("Mahasiswa ditambahkan");
    }

    [HttpDelete("{index}")]
    public ActionResult DeleteMahasiswa(int index)
    {
        if (index < 0 || index >= mahasiswaList.Count)
            return NotFound("Index tidak valid");

        mahasiswaList.RemoveAt(index);
        return Ok("Mahasiswa dihapus");
    }
}

```

Kode di atas adalah sebuah implementasi API menggunakan ASP.NET Core untuk mengelola data mahasiswa. API ini memiliki beberapa endpoint yang memungkinkan pengguna untuk melakukan operasi dasar CRUD (Create, Read, Update, Delete). Terdapat sebuah controller bernama MahasiswaController yang menangani permintaan HTTP terkait data mahasiswa.

1. GetAll() - Mengembalikan daftar semua mahasiswa dalam bentuk JSON.
2. GetByIndex(int index) - Mengembalikan data mahasiswa berdasarkan indeks yang diberikan. Jika indeks tidak valid, maka mengembalikan respons NotFound.
3. AddMahasiswa(Mahasiswa mahasiswaBaru) - Menambahkan mahasiswa baru ke dalam daftar. Data mahasiswa dikirimkan melalui body permintaan HTTP dalam format JSON.
4. DeleteMahasiswa(int index) - Menghapus mahasiswa berdasarkan indeks yang diberikan. Jika indeks tidak valid, respons NotFound akan dikembalikan.

Controller ini menggunakan atribut [ApiController] untuk memastikan bahwa pengaturan konvensi dan validasi otomatis berjalan dengan benar, serta [Route("api/[controller]")] untuk menentukan URL endpoint API.

Mahasiswa.cs

```
namespace tpmodul9_2311104058
{
    public class Mahasiswa
    {
        public string Nama { get; set; }
        public string Nim { get; set; }

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

Kode di atas mendefinisikan sebuah class bernama Mahasiswa di dalam namespace tpmodul9_2311104058. Class ini merepresentasikan data dasar dari seorang mahasiswa, yaitu nama dan NIM (Nomor Induk Mahasiswa).

Terdapat dua properti publik:

1. Nama (tipe string) untuk menyimpan nama mahasiswa, dan
2. Nim (tipe string) untuk menyimpan NIM mahasiswa.

Class ini juga memiliki sebuah konstruktor yang menerima dua parameter (nama dan nim) dan menginisialisasi properti Nama dan Nim sesuai dengan nilai yang diberikan. Class ini digunakan sebagai model data dalam API yang telah dijelaskan sebelumnya.

Program.cs

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
var app = builder.Build();
app.MapControllers();
app.Run();
```

Kode tersebut adalah bagian dari konfigurasi utama untuk menjalankan aplikasi web berbasis ASP.NET Core. Berikut penjelasan singkatnya:

1. WebApplication.CreateBuilder(args)
Membuat instance builder yang digunakan untuk mengatur dan mengkonfigurasi layanan serta middleware yang dibutuhkan oleh aplikasi.
2. builder.Services.AddControllers()
Menambahkan layanan controller ke dalam dependency injection container, yang memungkinkan aplikasi untuk menggunakan fitur-fitur API controller seperti routing, model binding, dan lainnya.
3. var app = builder.Build()
Membangun aplikasi dari konfigurasi yang telah disiapkan di builder.
4. app.MapControllers()
Mendaftarkan semua route controller yang ada di aplikasi agar dapat diakses melalui endpoint HTTP.

5. `app.Run()`

Menjalankan aplikasi web dan mulai menerima permintaan HTTP.

Secara keseluruhan, kode ini adalah setup minimal untuk menjalankan sebuah Web API di ASP.NET Core.