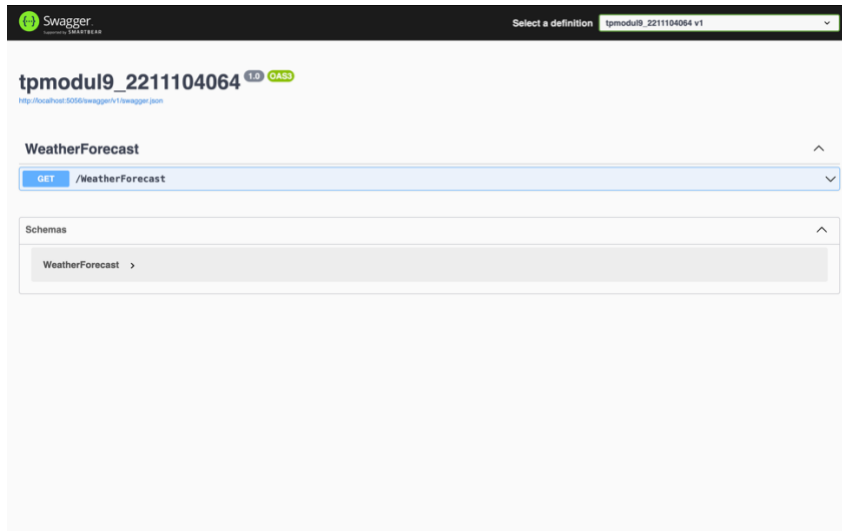
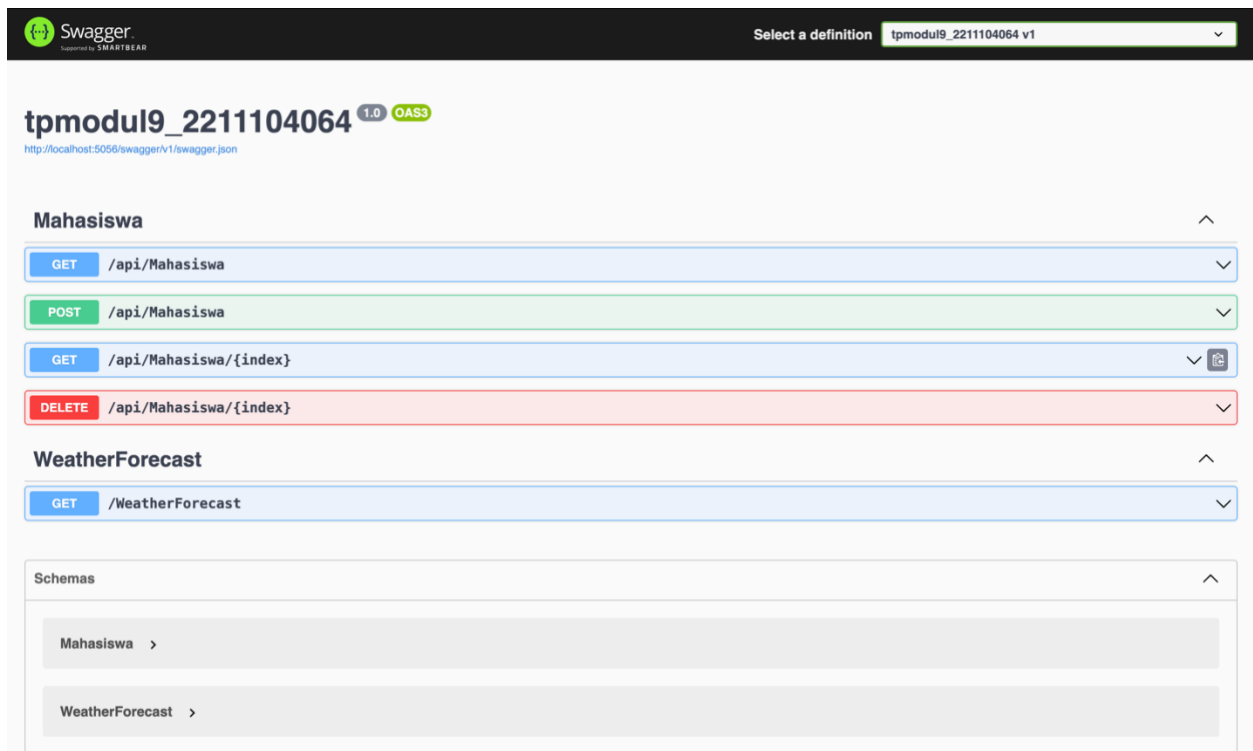


1. MEMBUAT PROJECT WEB API



2. IMPLEMENTASI WEB API



3. MENDEMONSTRASI WEB API

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

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

- Execute** button (highlighted in blue) and **Clear** button.
- Responses** section.
- Curl** section with the command:

```
curl -X 'GET' \
'http://localhost:5056/api/Mahasiswa' \
-H 'accept: text/plain'
```
- Request URL** section with the URL: `http://localhost:5056/api/Mahasiswa`
- Server response** section.
- Code** section showing status **200**.
- Details** section.
- Response body** section showing a JSON array of student data:

```
[
  {
    "nama": "Julian",
    "nim": "2211104064"
  },
  {
    "nama": "Fathur",
    "nim": "2211104070"
  },
  {
    "nama": "Samud",
    "nim": "2211104062"
  },
  {
    "nama": "Aditya sendi",
    "nim": "2211104067"
  },
  {
    "nama": "Dimas",
    "nim": "2211104086"
  }
]
```
- Response headers** section showing:

```
content-type: application/json; charset=utf-8
date: Tue, 10 Jun 2025 13:39:07 GMT
server: Kestrel
transfer-encoding: chunked
```
- Download** button.

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

tpmodul9_2211104064 1.0 OAS3

<http://localhost:5056/swagger/v1/swagger.json>

Mahasiswa

GET /api/Mahasiswa

POST /api/Mahasiswa

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{  
  "nama": "John Doe",  
  "nim": "1302199999"  
}
```



Execute

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

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5056/api/Mahasiswa' \
-H 'accept: text/plain'
```

Request URL

```
http://localhost:5056/api/Mahasiswa
```

Server response

Code

Details

200

Response body

```
[
  {
    "nama": "Julian",
    "nim": "2211184064"
  },
  {
    "nama": "Fathur",
    "nim": "2211184070"
  },
  {
    "nama": "Samud",
    "nim": "2211184062"
  },
  {
    "nama": "Aditya sendi",
    "nim": "2211184067"
  },
  {
    "nama": "Dimas",
    "nim": "2211184086"
  },
  {
    "nama": "John Doe",
    "nim": "1302199999"
  }
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 10 Jun 2025 13:43:07 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

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

GET /api/Mahasiswa/{index}

Parameters

Cancel

Name	Description
index * required	
integer(\$int32)	0
(path)	

ExecuteClear

Responses

Curl

curl -X 'GET' \n'http://localhost:5056/api/Mahasiswa/0' \n-H 'accept: text/plain'

Request URL

http://localhost:5056/api/Mahasiswa/0

Server response

Code	Details
200	<div><div>Response body</div><div>{\n "nama": "Julian",\n "nim": "2211104064"\n}</div><div>Response headers</div><div>content-type: application/json; charset=utf-8\ndate: Tue, 10 Jun 2025 13:44:24 GMT\nserver: Kestrel\ntransfer-encoding: chunked</div></div>

Download

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

DELETE

/api/Mahasiswa/{index}

Parameters

Cancel

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

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5056/api/Mahasiswa/0' \
-H 'accept: */*'
```

Request URL

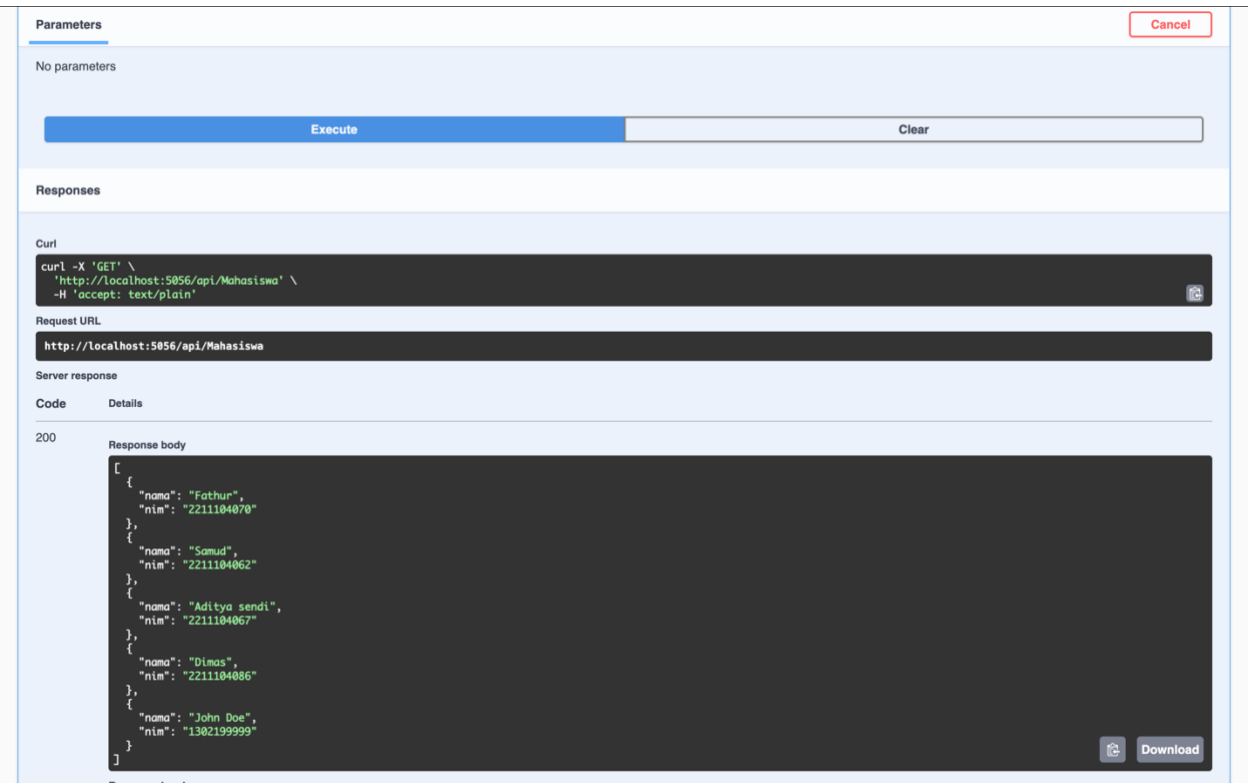
```
http://localhost:5056/api/Mahasiswa/0
```

Server response

Code	Details
200	<div>Response body<pre>{ "message": "Mahasiswa dihapus." }</pre><div>Download</div></div> <div>Response headers<pre>content-type: application/json; charset=utf-8 date: Tue, 10 Jun 2025 13:46:00 GMT server: Kestrel transfer-encoding: chunked</pre></div>

Responses

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



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

- Parameters:** No parameters.
- Execute:** A blue button to execute the request.
- Clear:** A button to clear the request.
- Responses:** A section showing the response details.
 - Curl:** `curl -X 'GET' \ 'http://localhost:5056/api/Mahasiswa' \ -H 'accept: text/plain'`
 - Request URL:** `http://localhost:5056/api/Mahasiswa`
 - Server response:**
 - Code:** 200
 - Details:** Response body

The response body is a JSON array of student objects:

```
[
  {
    "nama": "Fathur",
    "nim": "2211104070"
  },
  {
    "nama": "Samud",
    "nim": "2211104062"
  },
  {
    "nama": "Aditya sendi",
    "nim": "2211104067"
  },
  {
    "nama": "Dimas",
    "nim": "2211104086"
  },
  {
    "nama": "John Doe",
    "nim": "1302199999"
  }
]
```

Mahasiswa.cs

```
using System;
namespace tpmodul9_2211104064.Models
{
    public class Mahasiswa
    {
        public string Nama { get; set; }
        public string Nim { get; set; }
    }
}
```

Penjelasan kode:

Kode di atas mendefinisikan sebuah **class Mahasiswa** di dalam namespace `tpmodul9_2211104064.Models`. Class ini digunakan untuk merepresentasikan data mahasiswa, dan memiliki dua properti publik: `Nama` dan `Nim`, keduanya bertipe data string. Properti `Nama` menyimpan nama mahasiswa, sedangkan `Nim` menyimpan Nomor Induk Mahasiswa. Class ini nantinya akan digunakan dalam Web API untuk mengelola data mahasiswa, seperti menambah, menampilkan, dan

menghapus data. Model Mahasiswa ini penting karena menjadi struktur dasar dari objek yang dikirim atau diterima melalui endpoint API.

MahasiswaController.cs

```
using Microsoft.AspNetCore.Mvc;
using tpmodul9_2211104064.Models;
using System.Collections.Generic;

namespace tpmodul9_2211104064.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class MahasiswaController : ControllerBase
    {
        private static List<Mahasiswa> daftarMahasiswa = new
        List<Mahasiswa>
        {
            new Mahasiswa { Nama = "Julian", Nim = "2211104064" },
            new Mahasiswa { Nama = "Fathur", Nim = "2211104070" },
            new Mahasiswa { Nama = "Samud", Nim = "2211104062" },
            new Mahasiswa { Nama = "Aditya sendi", Nim = "2211104067" },
            new Mahasiswa { Nama = "Dimas", Nim = "2211104086" }
        };

        // GET: api/mahasiswa
        [HttpGet]
        public ActionResult<IEnumerable<Mahasiswa>> GetAll()
        {
            return daftarMahasiswa;
        }

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

        // POST: api/mahasiswa
        [HttpPost]
        public ActionResult Add([FromBody] Mahasiswa mhs)
```



```

        {
            daftarMahasiswa.Add(mhs);
            return Ok(new { message = "Mahasiswa ditambahkan." });
        }

        // DELETE: api/mahasiswa/{index}
        [HttpDelete("{index}")]
        public ActionResult Delete(int index)
        {
            if (index < 0 || index >= daftarMahasiswa.Count)
                return NotFound();
            daftarMahasiswa.RemoveAt(index);
            return Ok(new { message = "Mahasiswa dihapus." });
        }
    }
}

```

Penjelasan kode:

Kode di atas adalah implementasi **Web API Controller** menggunakan ASP.NET Core yang bernama MahasiswaController. Controller ini menyediakan endpoint RESTful untuk mengelola data mahasiswa yang disimpan dalam list statis daftarMahasiswa. Terdapat empat endpoint utama: GET /api/mahasiswa untuk mengambil seluruh data mahasiswa, GET /api/mahasiswa/{index} untuk mengambil satu mahasiswa berdasarkan indeks, POST /api/mahasiswa untuk menambahkan data mahasiswa baru, dan DELETE /api/mahasiswa/{index} untuk menghapus data mahasiswa berdasarkan indeks. Setiap metode memanfaatkan atribut routing seperti [HttpGet], [HttpPost], dan [HttpDelete] untuk mengatur jenis permintaan HTTP yang diterima. Model Mahasiswa sendiri diasumsikan memiliki properti Nama dan Nim. API ini bersifat sederhana karena data hanya disimpan dalam memori dan tidak menggunakan database.

Kesimpulan laporan TP 9

Pada tugas ini, telah berhasil dibuat sebuah aplikasi Web API sederhana menggunakan .NET 6.0 dan Visual Studio di MacBook Air M1. Proyek ini bertujuan untuk memahami konsep dasar pembuatan RESTful API tanpa menggunakan database, yaitu dengan menyimpan data dalam list statis. API yang dibangun memiliki endpoint untuk mengambil seluruh data mahasiswa, mengambil data berdasarkan indeks, menambahkan data, dan menghapus data mahasiswa. Model Mahasiswa digunakan sebagai representasi data, sedangkan controller

MahasiswaController menangani logika dan routing dari permintaan HTTP. Proyek ini juga telah diuji melalui endpoint Swagger dan terbukti berjalan dengan baik. Seluruh hasil percobaan seperti GET, POST, dan DELETE