LAPORAN PRAKTIKUM TP MODUL 9

NIM / Nama: 2311104072 – Jauhar Fajar Zuhair

Bagian I: Pendahuluan

Tugas Pendahuluan Modul 9 ini berfokus pada pembuatan sebuah Web API sederhana menggunakan ASP.NET Core. API ini akan mengelola data Mahasiswa (Nama dan NIM) yang disimpan dalam sebuah list di memori. API ini menyediakan operasi dasar CRUD (Create, Read, Delete) melalui endpoint HTTP.

Bagian II: Model Data `Mahasiswa.cs`

Kelas ini merepresentasikan entitas Mahasiswa yang akan dikelola oleh API.

Isi `Mahasiswa.cs`:

```
namespace tpmodul9_2311104072
{
    public class Mahasiswa
    {
        public string Nama { get; set; }
        public string Nim { get; set; }

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

Penjelasan:

- Kelas `Mahasiswa` memiliki dua properti publik: `Nama` (string) dan `Nim` (string).
- Konstruktor `Mahasiswa(string nama, string nim)` digunakan untuk menginisialisasi objek Mahasiswa baru dengan nama dan NIM yang diberikan.

Bagian III: Controller `MahasiswaController.cs`

Controller ini menangani request HTTP yang masuk dan mengelola data Mahasiswa.

Deklarasi Controller dan Inisialisasi Data:

```
using Microsoft.AspNetCore.Mvc; // Diperlukan untuk [ApiController],
ControllerBase, IActionResult, dll.
using System.Collections.Generic; // Diperlukan untuk List<>
using System.Linq; // Diperlukan jika menggunakan LINQ (tidak dalam contoh
ini)

namespace tpmodul9_2311104072.Controllers
{
    [ApiController]
    [Route("[controller]")] // Route otomatis menjadi /Mahasiswa
    public class MahasiswaController : ControllerBase
    {
        // Data disimpan sementara di memori (static agar bertahan antar
request)
        private static List<Mahasiswa> _mahasiswa = new List<Mahasiswa>
        {
            new Mahasiswa("jauharfz", "2311104072"),
            new Mahasiswa("anggap saja anggota", "23111040XX")
        };

        // ... Action Methods (Endpoints) ...
    }
}
```

Penjelasan:

- `[ApiController]` : Menandakan bahwa kelas ini adalah sebuah API controller, mengaktifkan beberapa fitur konvensi API.
- `[Route("[controller]")]` : Menentukan route dasar untuk controller ini. `[controller]` akan diganti dengan nama controller tanpa suffix "Controller", sehingga route menjadi `/Mahasiswa`.
- `_mahasiswa`: Sebuah `List<Mahasiswa>` statis digunakan sebagai penyimpanan data sementara di memori. Data awal (seed data) ditambahkan saat inisialisasi.

Bagian IV: Endpoint API

Controller ini mendefinisikan beberapa action method yang memetakan ke endpoint HTTP untuk operasi CRUD.

1. Get All Mahasiswa (`GET /Mahasiswa`)

```
[HttpGet] // Menangani HTTP GET request ke route dasar (/Mahasiswa)
public IActionResult GetAll()
{
    // Mengembalikan seluruh list mahasiswa dengan status 200 OK
    return Ok(_mahasiswa);
}
```

Penjelasan: Mengembalikan semua data mahasiswa yang ada di dalam list `_mahasiswa` dalam format JSON dengan status HTTP 200 (OK).

2. Get Mahasiswa by Index (`GET /Mahasiswa/{index}`)

```
[HttpGet("{index}")] // Menangani HTTP GET ke /Mahasiswa/angka (misal:
/Mahasiswa/0)
public IActionResult GetByIndex(int index)
{
    // Validasi index
    if (index < 0 || index >= _mahasiswa.Count)
    {
        // Jika index tidak valid, kembalikan status 404 Not Found
        return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");
    }
    // Jika index valid, kembalikan data mahasiswa pada index tersebut dengan
status 200 OK
    return Ok(_mahasiswa[index]);
}
```

Penjelasan: Mengembalikan data mahasiswa pada `index` tertentu. Jika `index` di luar batas list, akan mengembalikan status HTTP 404 (Not Found).

3. Add New Mahasiswa (`POST /Mahasiswa`)

```
[HttpPost] // Menangani HTTP POST ke route dasar (/Mahasiswa)
public IActionResult Post([FromBody] Mahasiswa mahasiswa) // Data mahasiswa
diambil dari body request
{
    // Menambahkan mahasiswa baru ke dalam list
    _mahasiswa.Add(mahasiswa);
    // Mengembalikan status 201 Created dengan lokasi resource baru dan data
yang baru ditambahkan
    // GetByIndex adalah nama action method untuk mendapatkan resource
individual
    return CreatedAtAction(nameof(GetByIndex), new { index = _mahasiswa.Count
```

```
- 1 }, mahasiswa);
}
```

Penjelasan: Menerima data mahasiswa baru dari *body* request (dalam format JSON), menambahkannya ke list `_mahasiswa`, dan mengembalikan status HTTP 201 (Created) beserta data mahasiswa yang baru ditambahkan dan *header* `Location` yang menunjuk ke URI mahasiswa baru tersebut.

4. Delete Mahasiswa by Index (`DELETE /Mahasiswa/{index}`)

```
[HttpDelete("{index}")] // Menangani HTTP DELETE ke /Mahasiswa/angka
public IActionResult Delete(int index)
{
    // Validasi index
    if (index < 0 || index >= _mahasiswa.Count)
    {
        // Jika index tidak valid, kembalikan status 404 Not Found
        return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");
    }
    // Hapus mahasiswa dari list pada index yang diberikan
    _mahasiswa.RemoveAt(index);
    // Mengembalikan status 204 No Content (sukses, tidak ada body response)
    return NoContent();
}
```

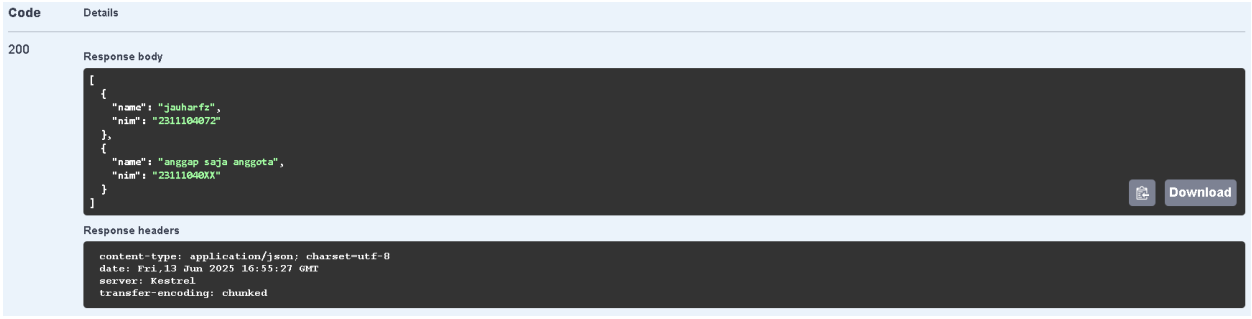
Penjelasan: Menghapus data mahasiswa pada `index` yang diberikan. Jika `index` valid dan penghapusan berhasil, mengembalikan status HTTP 204 (No Content). Jika `index` tidak valid, mengembalikan status 404 (Not Found).

Bagian V: Menjalankan dan Menguji API

1. **Menjalankan Proyek:** Buka terminal di direktori proyek dan jalankan perintah `dotnet run`, atau tekan tombol Run (biasanya F5) di Visual Studio.
2. **Mengakses API:** Setelah proyek berjalan, API dapat diakses melalui browser atau alat pengujian API seperti Postman atau Swagger UI.
3. **Swagger UI:** Jika OpenAPI/Swagger diaktifkan saat pembuatan proyek (default), Anda dapat membuka URL yang ditampilkan di terminal di browser. Swagger UI menyediakan antarmuka grafis untuk melihat semua endpoint yang tersedia dan mengujinya secara langsung.

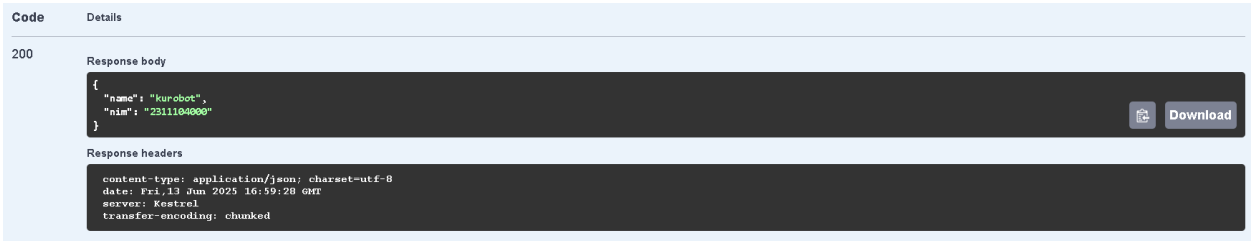
Bagian VI: Contoh Hasil Pengujian (Menggunakan Swagger UI / Postman)

- **Request:** `GET /Mahasiswa`**Response (200 OK):**

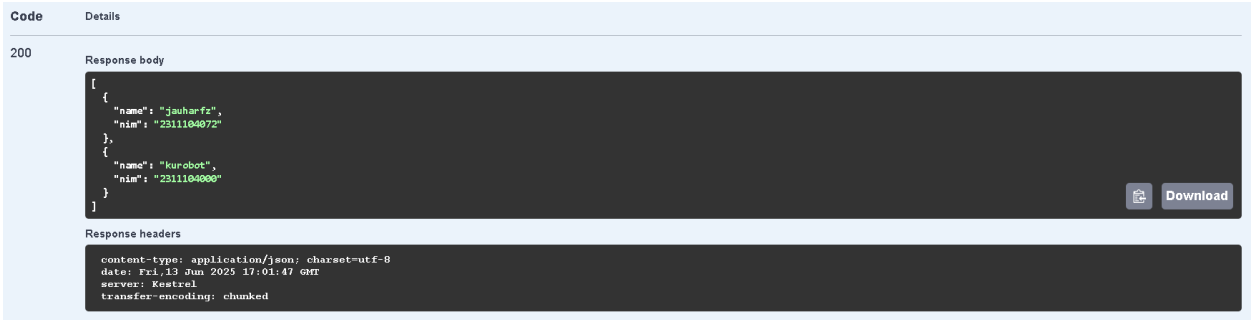


- **Request:** `POST /Mahasiswa` dengan Request Body:

- **Request:** `GET /Mahasiswa/2` (setelah POST di atas) **Response (200 OK):**



- **Request:** `DELETE /Mahasiswa/1`**Response (204 No Content):** (Tidak ada body response)
- **Request:** `GET /Mahasiswa` (setelah DELETE) **Response (200 OK):**



Laporan ini mencakup pembuatan model, controller, dan endpoint dasar untuk sebuah Web API sederhana menggunakan ASP.NET Core.