

1. Link Github Repository

[MuhammadNurShodiq/tmodul10_1302223054 \(github.com\)](https://MuhammadNurShodiq/tmodul10_1302223054.github.com)

2. Hasil Run

- Get

The screenshot shows a REST API tool interface with the following details:

- Method:** GET
- Endpoint:** /api/Mahasiswa
- Parameters:** No parameters
- Buttons:** Execute (blue), Clear (white), Cancel (red)
- Responses:**
 - Curl:** curl -X 'GET' \ 'http://localhost:5044/api/Mahasiswa' \ -H 'accept: text/plain'
 - Request URL:** http://localhost:5044/api/Mahasiswa
 - Server response:**
 - Code:** 200
 - Response body:**

```
[  
  {  
    "nama": "Muhammad Nur Shodiq",  
    "nim": "1302223054"  
  },  
  {  
    "nama": "Fauzein Mulya Warman",  
    "nim": "1302223127"  
  },  
  {  
    "nama": "Iksan Oktav Risandy",  
    "nim": "1302223042"  
  },  
  {  
    "nama": "Muhammad Dias Adani",  
    "nim": "1302223134"  
  },  
  {  
    "nama": "Muhammad Haulul Azkiyaa",  
    "nim": "1302223007"  
  }]
```

 - Download:** A button to download the response body.
 - Response headers:**

```
content-type: application/json; charset=utf-8  
date: Sun, 28 Apr 2024 20:22:37 GMT  
server: Kestrel  
transfer-encoding: chunked
```
- Responses:**
 - Code:** 200
 - Description:** Success
 - Links:** No links

- Post

POST /api/Mahasiswa

Parameters

No parameters

Request body

application/json

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

Execute Clear

Responses

Curl

```
curl -X 'POST' \  
  'http://localhost:5044/api/Mahasiswa' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "nama": "John Doe",  
    "nim": "1302199999"  
}'
```

Request URL

http://localhost:5044/api/Mahasiswa

Server response

Code	Details
200	Response headers content-length: 0 date: Sun, 28 Apr 2024 20:23:44 GMT server: Kestrel

Responses

Code Description Links

- Get John Doe

GET /api/Mahasiswa/{id}

Parameters

Name Description

id * required
integer(\$int32)
(path) 5

Cancel

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5044/api/Mahasiswa/5' \
-H "accept: text/plain"
```

Request URL

```
http://localhost:5044/api/Mahasiswa/5
```

Server response

Code Details

200 Response body

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

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sun, 28 Apr 2024 20:24:40 GMT
server: Kestrel
transfer-encoding: chunked
```

Responses

Code Description Links

200 Success No links

Media type

text/plain

Controls Accept header.

Example Value | Schema

- Get our nim (index 0)

The screenshot shows a REST API documentation interface for a GET request to `/api/Mahasiswa/{id}`. The `id` parameter is set to 0. The interface includes sections for Parameters, Responses, and a detailed view of the 200 response.

Parameters

Name	Description
<code>id</code> * required integer(\$int32) (path)	0

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5044/api/Mahasiswa/0' \
-H 'accept: text/plain'
```

Request URL

```
http://localhost:5044/api/Mahasiswa/0
```

Server response

Code	Details	Links
200	Response body <pre>{ "nama": "Muhammad Nur Shodiq", "nim": "1302223054" }</pre> Response headers <pre>content-type: application/json; charset=utf-8 date: Sun, 28 Apr 2024 20:26:00 GMT server: Kestrel transfer-encoding: chunked</pre>	Copy Download

Responses

Code	Description	Links
200	Success	No links

Media type: `text/plain`

Controls Accept header.

Example Value | Schema

```
{  
    "nama": "Muhammad Nur Shodiq",  
    "nim": "1302223054"  
}
```

- Delete our nim (index 0)

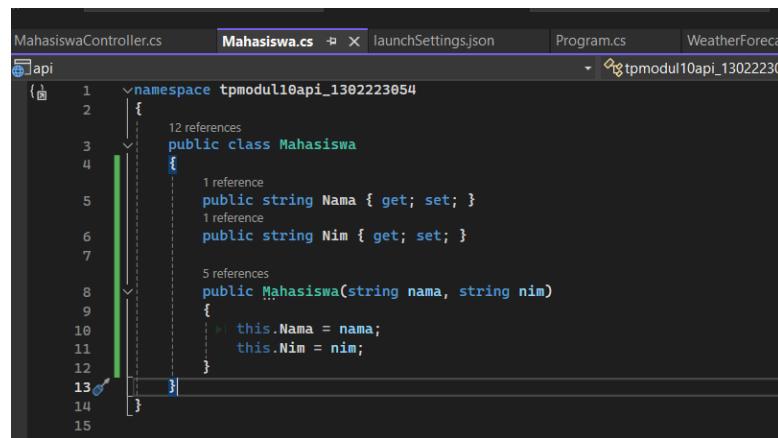
The screenshot shows a REST API endpoint configuration. At the top left is a red "DELETE" button. To its right is the URL path "/api/Mahasiswa/{id}". Below this is a "Parameters" section with a table. A single row is shown for the parameter "id", which is marked as required and has a type of "integer(\$int32)". The value field contains "0". On the far right of the parameters section is a "Cancel" button. Below the parameters is a large blue "Execute" button. Underneath the execute button is a circular "LOADING" indicator. At the bottom of the interface is a "Responses" section with a table. It shows one entry for code 200 with the description "Success". To the right of this table is a "Links" section with the text "No links".

Note: Selalu buffering, seharusnya berhasil karena telah memakai sintaks yang sudah sesuai dan dapat membaca error saat input diluar range index, namun saat memasukkan id 0 yang terjadi adalah buffering yang tidak selesai.

```
// DELETE api/MahasiswaController/5
[HttpDelete("{id}")]
0 references
public void Delete(int id)
{
    dataMahasiswa.RemoveAt(id); X
}
```

3. Code

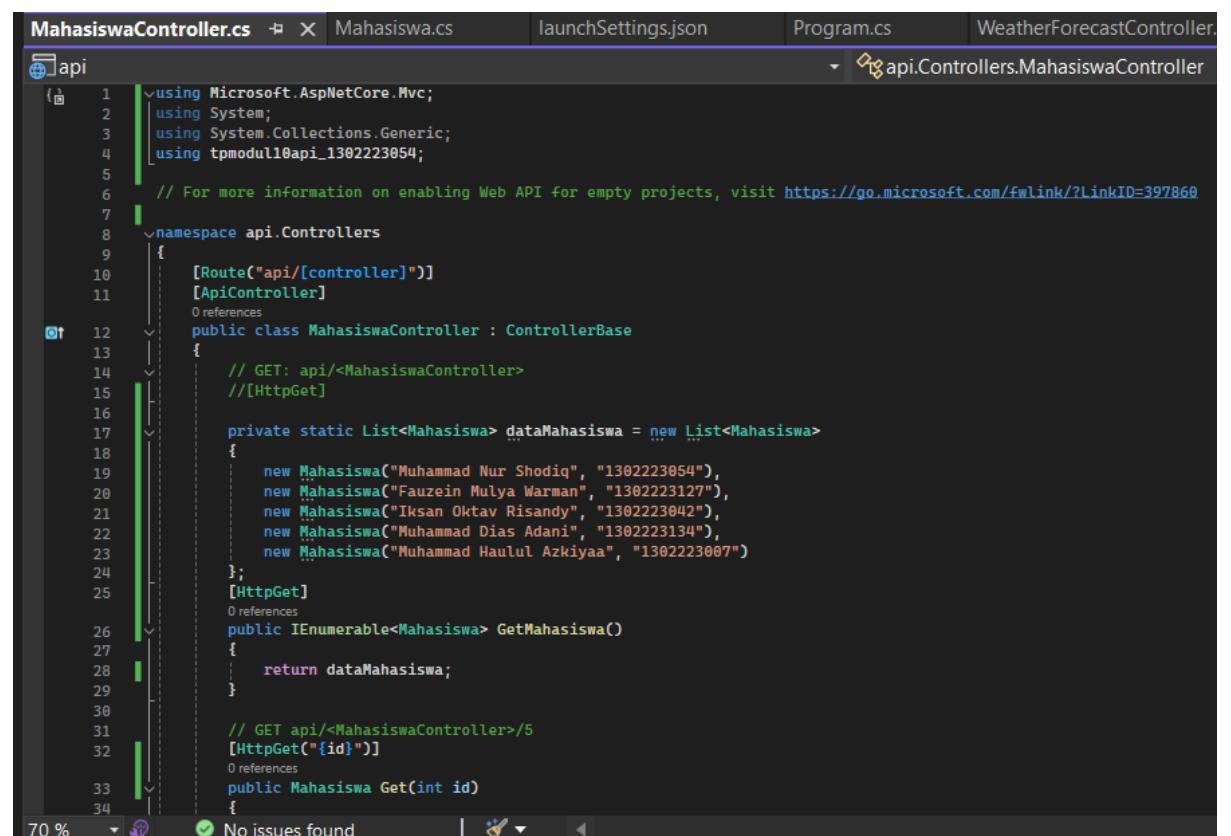
a. Class Mahasiswa.cs



```
namespace tpmodul10api_1302223054
{
    public class Mahasiswa
    {
        public string Nama { get; set; }
        public string Nim { get; set; }

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

b. Controller MahasiswaController.cs



```
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using tpmodul10api_1302223054;

// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397869

namespace api.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class MahasiswaController : ControllerBase
    {
        // GET: api/<MahasiswaController>
        //#[HttpGet]

        private static List<Mahasiswa> dataMahasiswa = new List<Mahasiswa>
        {
            new Mahasiswa("Muhammad Nur Shodiq", "1302223054"),
            new Mahasiswa("Fauzein Mulya Warman", "1302223127"),
            new Mahasiswa("Ikhsan Oktav Risandy", "1302223042"),
            new Mahasiswa("Muhammad Dias Adani", "1302223134"),
            new Mahasiswa("Muhammad Haulul Azkiyaa", "1302223007")
        };
        #[HttpGet]
        public IEnumerable<Mahasiswa> GetMahasiswa()
        {
            return dataMahasiswa;
        }

        // GET api/<MahasiswaController>/5
        #[HttpGet("{id}")]
        public Mahasiswa Get(int id)
        {
    
```

```
32     [HttpGet("{id}")]
33     public Mahasiswa Get(int id)
34     {
35         return dataMahasiswa[id];
36     }
37
38     // POST api/<MahasiswaController>
39     [HttpPost]
40     public void Post([FromBody] Mahasiswa value)
41     {
42         dataMahasiswa.Add(value);
43     }
44
45     // PUT api/<MahasiswaController>/5
46     [HttpPut("{id}")]
47     public void Put(int id, [FromBody] Mahasiswa value)
48     {
49         dataMahasiswa[id] = value;
50     }
51
52     // DELETE api/<MahasiswaController>/5
53     [HttpDelete("{id}")]
54     public void Delete(int id)
55     {
56         dataMahasiswa.RemoveAt(id); X
57     }
58 }
59
60 }
```

4. Penjelasan singkat kode

API yang dibuat memiliki fungsi untuk mengelola data mahasiswa.

Controller menyediakan beberapa fungsi endpoint untuk mengakses data mahasiswa, yaitu:

- GET /api/mahasiswa: Mengambil semua data mahasiswa.
- GET /api/mahasiswa/{id}: Mengambil data mahasiswa dengan id tertentu.
- POST /api/mahasiswa: Menambahkan data mahasiswa baru.
- PUT /api/mahasiswa/{id}: Memperbarui data mahasiswa dengan id tertentu.

- `DELETE /api/mahasiswa/{id}`: Menghapus data mahasiswa dengan id tertentu.

Data mahasiswa disimpan dalam array `dataMahasiswa` yang bersifat statis. Setiap mahasiswa memiliki properti Nama dan Nim.

Ketika pengguna melakukan request ke salah satu endpoint API, controller akan memproses request tersebut dan memberikan respons yang sesuai.

Contohnya, jika pengguna melakukan request `GET /api/mahasiswa`, controller akan mengambil semua data mahasiswa dari array `dataMahasiswa` dan mengembalikannya dalam format JSON.

API ini dapat digunakan oleh aplikasi lain untuk mengakses dan mengelola data mahasiswa.