

Nama:Faishal Arif Setiawan

Nim:2311104066

Kelas:SE-07-02

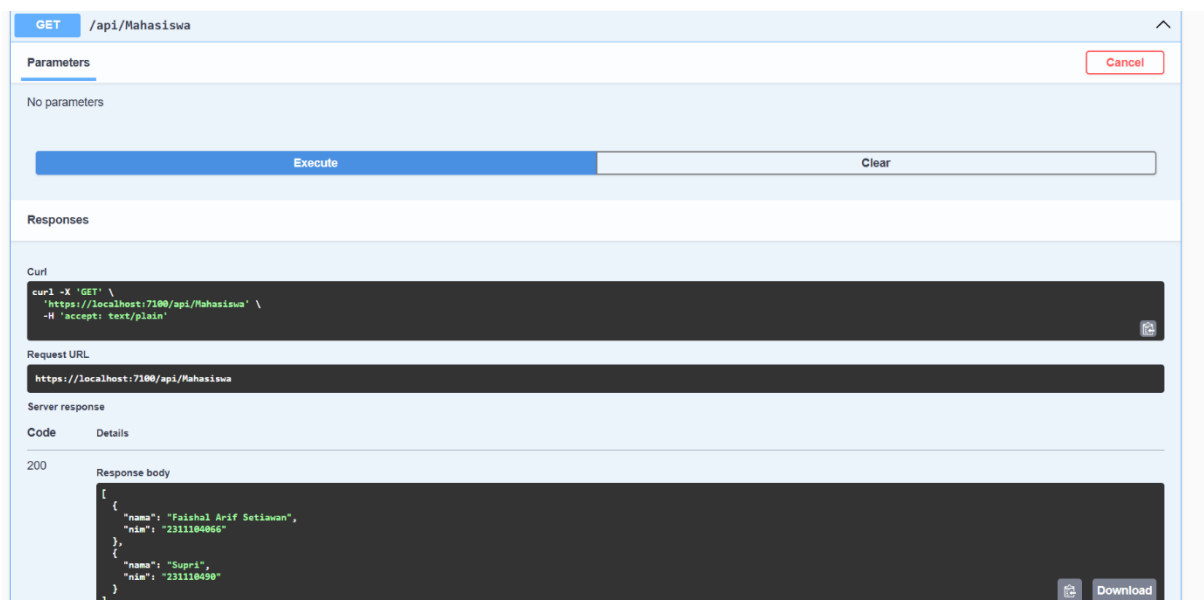
TP Modul 9

I.Link Github

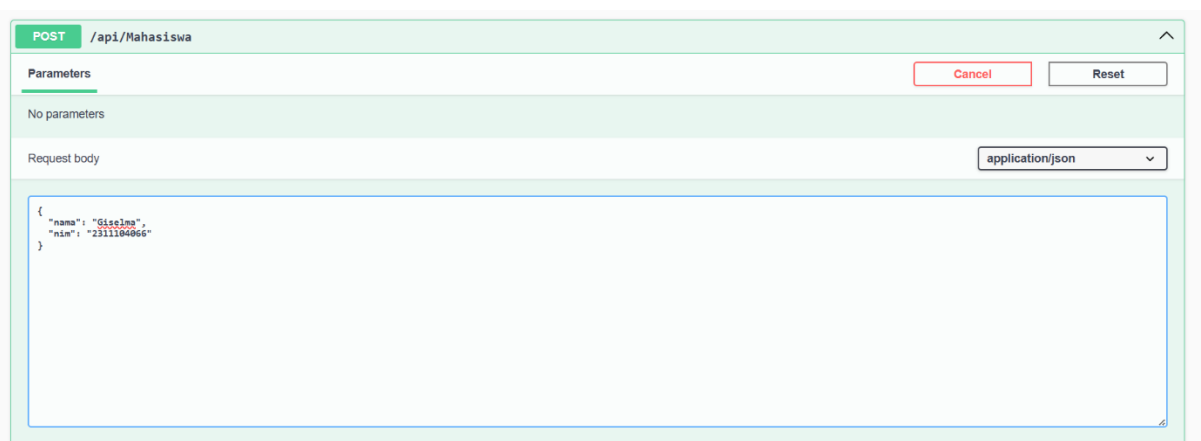
https://github.com/faishalstwn/KPL_FaishalArifSetiawan_2311104066_SE_07_02

II.Demonstrasi Program

A. Mencoba "GET /api/mahasiswa" saat baru dijalankan



B. Menambahkan mahasiswa



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

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

- Parameters:** No parameters are listed.
- Execute:** A blue button to execute the request.
- Clear:** A button to clear the request.
- Responses:**
 - Curl:**

```
curl -X 'GET' \
'https://localhost:7100/api/mahasiswa' \
-H 'accept: text/plain'
```
 - Request URL:** `https://localhost:7100/api/mahasiswa`
 - Server response:**
 - Code:** 200
 - Details:** Response body

The response body is a JSON array of three student objects:

```
{
  "name": "Faishal Arif Setiawan",
  "nim": "2311104066"
},
{
  "name": "Supri",
  "nim": "2311104090"
},
{
  "name": "Gisela",
  "nim": "2311104066"
}
```

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

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

- Parameters:** A table with two columns: Name and Description.
- index:** A required parameter of type integer(\$int32) with a value of 1.
- Execute:** A blue button to execute the request.
- Clear:** A button to clear the request.
- Responses:**
 - Curl:**

```
curl -X 'GET' \
'https://localhost:7100/api/mahasiswa/1' \
-H 'accept: text/plain'
```
 - Request URL:** `https://localhost:7100/api/mahasiswa/1`
 - Server response:**
 - Code:** 200
 - Details:** Response body

The response body is a JSON object for the student at index 1:

```
{
  "name": "Supri",
  "nim": "2311104090"
}
```

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

DELETE /api/Mahasiswa/{index}

Parameters

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

Execute Clear

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

GET /api/Mahasiswa

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7100/api/Mahasiswa' \
  -H 'accept: text/plain'
```

Request URL

https://localhost:7100/api/Mahasiswa

Server response

Code	Details
200	<pre>{ "name": "Supri", "nim": "231110490" }, { "name": "Giselma", "nim": "2311104866" } }</pre>

Download

III.Source Code

Mahasiswa Controller.cs

```

1  using Microsoft.AspNetCore.Mvc;
2  using Tp9.Models;
3
4  namespace Tp9.Controllers
5  {
6      [ApiController]
7      [Route("api/[controller]")]
8      public class MahasiswaController : ControllerBase
9      {
10
11          private static readonly List<Mahasiswa> Mahasiswas = new()
12          {
13              new Mahasiswa { Nama = "Faishal Arif Setiawan", NIM = "2311104066" },
14              new Mahasiswa { Nama = "Supri", NIM = "2311104090" }
15          };
16
17          // GET: api/mahasiswa
18          [HttpGet]
19          public ActionResult<IEnumerable<Mahasiswa>> GetAll()
20          {
21              return Ok(Mahasiswas);
22          }
23
24          // GET: api/mahasiswa/{index}
25          [HttpGet("{index}")]
26          public ActionResult<Mahasiswa> GetByIndex(int index)
27          {
28              if (index < 0 || index >= Mahasiswas.Count)
29                  return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");
30              return Ok(Mahasiswas[index]);
31          }
32
33          // POST: api/mahasiswa
34          [HttpPost]
35          public ActionResult Add([FromBody] Mahasiswa newMhs)
36          {
37              if (string.IsNullOrEmpty(newMhs.Nama) || string.IsNullOrEmpty(newMhs.NIM))
38                  return BadRequest("Nama dan NIM wajib diisi.");
39
40              Mahasiswas.Add(newMhs);
41              return Ok($"Mahasiswa '{newMhs.Nama}' berhasil ditambahkan.");
42          }
43
44          // DELETE: api/mahasiswa/{index}
45          [HttpDelete("{index}")]
46          public ActionResult Delete(int index)
47          {
48              if (index < 0 || index >= Mahasiswas.Count)
49                  return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");
50
51              var removed = Mahasiswas[index];
52              Mahasiswas.RemoveAt(index);
53              return Ok($"Mahasiswa '{removed.Nama}' berhasil dihapus.");
54          }
55      }
56

```

```

namespace Tp9.Controllers
{
    [ApiController]
    [Route("api/[controller]")]

```

- **[ApiController]**: Memberitahu ASP.NET Core bahwa ini adalah controller API. Atribut ini mengaktifkan fitur-fitur seperti validasi model otomatis.
- **[Route("api/[controller]")]**: Routing dasar controller. [controller] akan otomatis diganti dengan nama controller (tanpa "Controller"), jadi endpoint-nya menjadi /api/mahasiswa.

Static List

```

private static readonly List<Mahasiswa> Mahasiswas = new()
{
    new Mahasiswa { Nama = "Faishal Arif Setiawan", NIM = "2311104066" },
    new Mahasiswa { Nama = "Supri", NIM = "2311104090" }
};

```

Membuat list statis Mahasiswas sebagai penyimpanan sementara (in-memory).

GET Semua Mahasiswa

```
[HttpGet]
0 references
public ActionResult<IEnumerable<Mahasiswa>> GetAll()
{
    return Ok(Mahasiswas);
}
```

[HttpGet]: Akan merespon permintaan GET ke api/mahasiswa.

GET Mahasiswa Berdasarkan Index

```
[HttpGet("{index}")]
0 references
public ActionResult<Mahasiswa> GetByIndex(int index)
{
    if (index < 0 || index >= Mahasiswas.Count)
        return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");
    return Ok(Mahasiswas[index]);
}
```

- **[HttpGet("{index}")]**: Menangani GET ke api/mahasiswa/0, api/mahasiswa/1, dst.
- Mengecek apakah index valid (tidak melebihi panjang list).
- Jika valid, mengembalikan mahasiswa pada index tersebut. Jika tidak, mengembalikan 404 Not Found.

POST Tambah Mahasiswa

```
[HttpPost]
0 references
public ActionResult Add([FromBody] Mahasiswa newMhs)
{
    if (string.IsNullOrEmpty(newMhs>Nama) || string.IsNullOrEmpty(newMhs.NIM))
        return BadRequest("Nama dan NIM wajib diisi.");

    Mahasiswas.Add(newMhs);
    return Ok($"Mahasiswa '{newMhs>Nama}' berhasil ditambahkan.");
}
```

- **[HttpPost]**: Menangani permintaan POST ke api/mahasiswa.
- **[FromBody]**: Data Mahasiswa diambil dari body JSON request.
- Validasi: jika Nama atau NIM kosong, mengembalikan 400 Bad Request.
- Menambahkan data baru ke list dan mengembalikan pesan sukses.

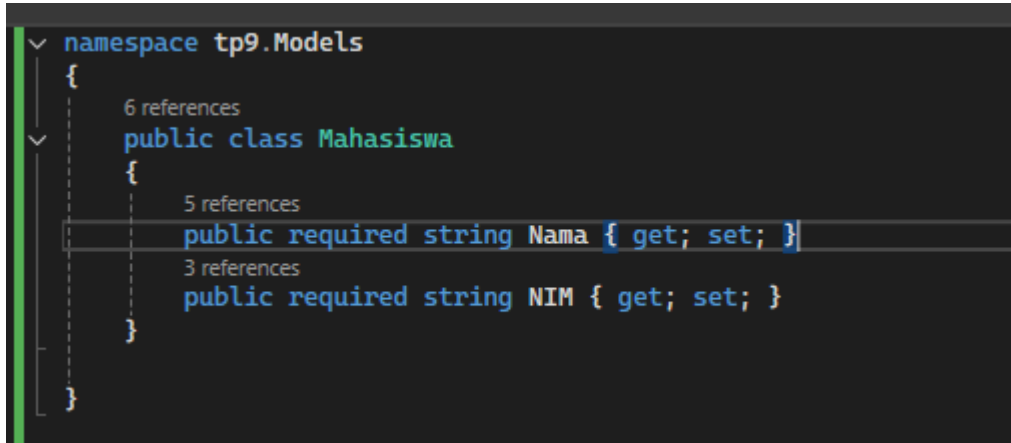
DELETE Mahasiswa Berdasarkan Index

```
[HttpDelete("{index}")]
0 references
public ActionResult Delete(int index)
{
    if (index < 0 || index >= Mahasiswas.Count)
        return NotFound($"Mahasiswa dengan index {index} tidak ditemukan.");

    var removed = Mahasiswas[index];
    Mahasiswas.RemoveAt(index);
    return Ok($"Mahasiswa '{removed>Nama}' berhasil dihapus.");
}
```

- **[HttpDelete("{index}")]**: Menangani DELETE ke api/mahasiswa/1, dll.
- Mengecek apakah index valid.
- Menghapus mahasiswa pada index tersebut dan mengembalikan pesan sukses.
- Jika tidak ditemukan, mengembalikan 404 Not Found.

Mahasiswa.Cs



```
namespace tp9.Models
{
    6 references
    public class Mahasiswa
    {
        5 references
        public required string Nama { get; set; }
        3 references
        public required string NIM { get; set; }
    }
}
```

Mendefinisikan sebuah class bernama Mahasiswa.