

Nama : Candra Dinata

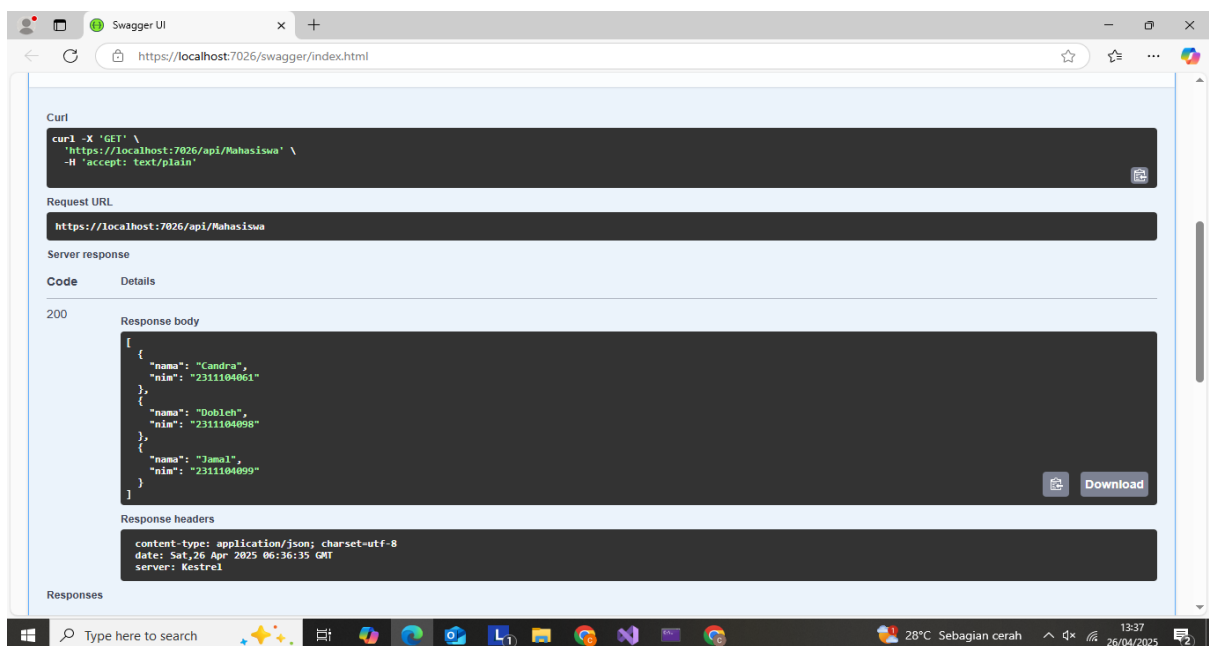
Nim : 2311104061

Kelas : SE0702

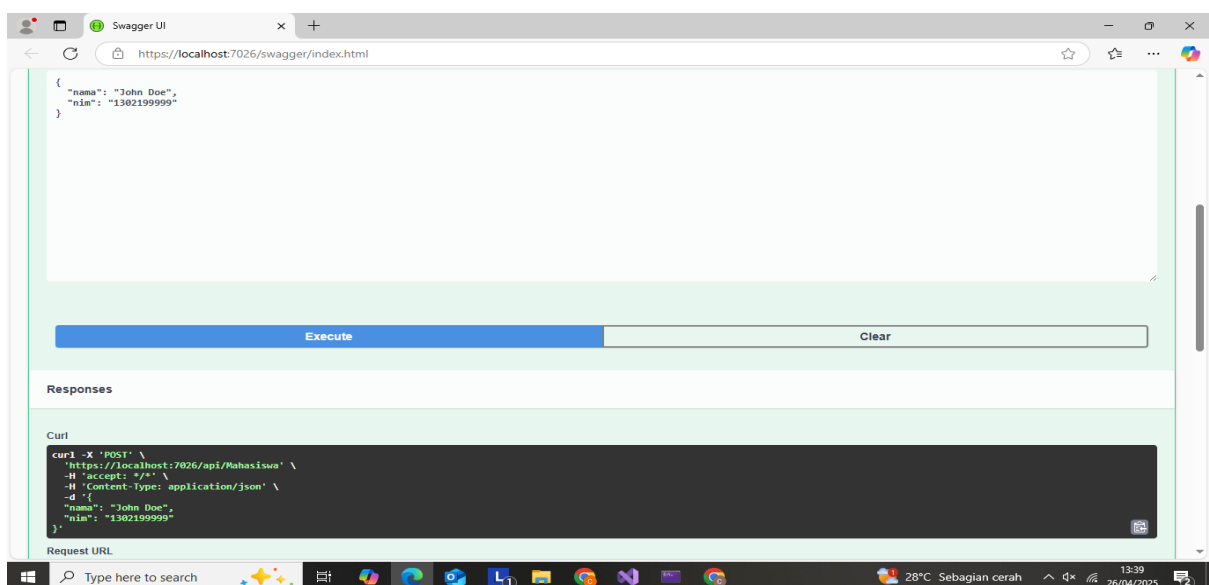
https://github.com/CandraDinataa/KPL_CandraDinata_2311104061_SE0702/tree/9e4e58fb6af29f9b78afd6e30c69a6c22961b108/09_API_Design_dan_Construction_Using_Swagger/TPMODUL9_2311104061

4.

a.



b.



c.

Swagger UI interface showing a GET request to `https://localhost:7026/api/Mahasiswa`. The request is executed using curl. The response is a 200 status code with a JSON array of student data.

```
curl -X 'GET' \
  'https://localhost:7026/api/Mahasiswa' \
  -H 'accept: text/plain'
```

Request URL: `https://localhost:7026/api/Mahasiswa`

Server response

Code: 200

Response body

```
{
  "nama": "Candra",
  "nim": "2311104061"
},
{
  "nama": "Dobleh",
  "nim": "2311104098"
},
{
  "nama": "Jamal",
  "nim": "2311104099"
},
{
  "nama": "John Doe",
  "nim": "1302199999"
}
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 26 Apr 2025 06:58:05 GMT
server: Kestrel
```

Responses

d.

Swagger UI interface showing a GET request to `https://localhost:7026/api/Mahasiswa/0`. The request is executed using curl. The response is a 200 status code with a JSON object of student data.

index * required

integer(\$int32)

(path)

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7026/api/Mahasiswa/0' \
  -H 'accept: text/plain'
```

Request URL: `https://localhost:7026/api/Mahasiswa/0`

Server response

Code: 200

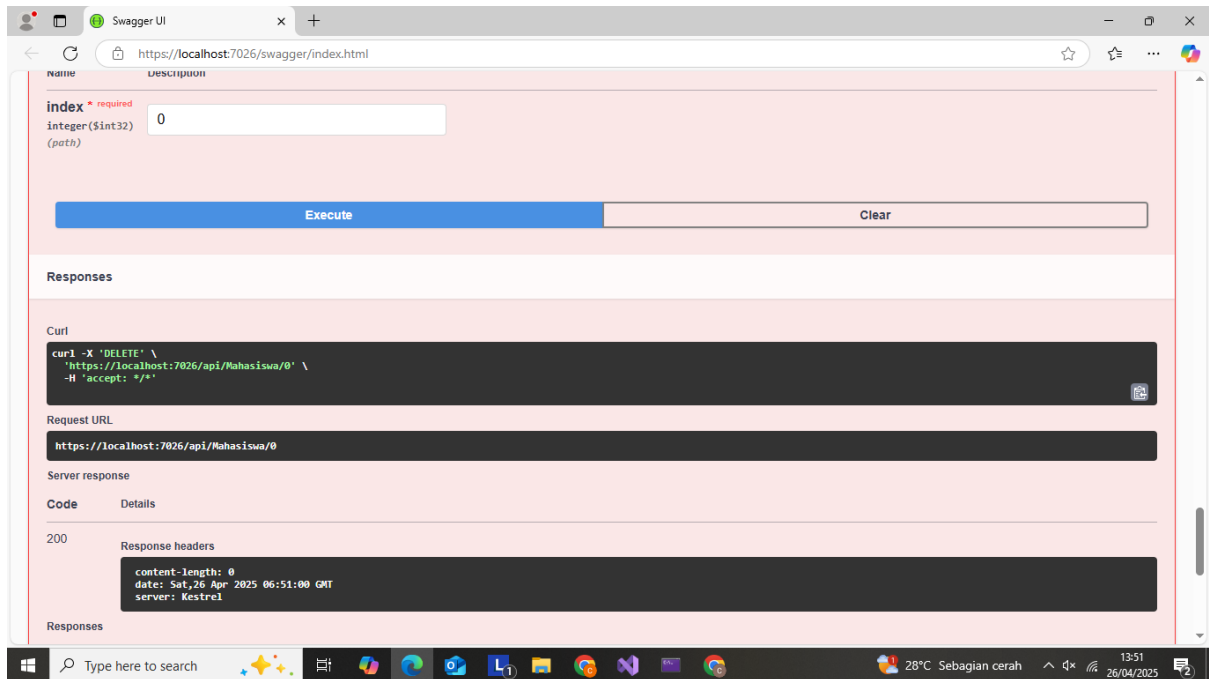
Response body

```
{
  "nama": "Candra",
  "nim": "2311104061"
}
```

Response headers

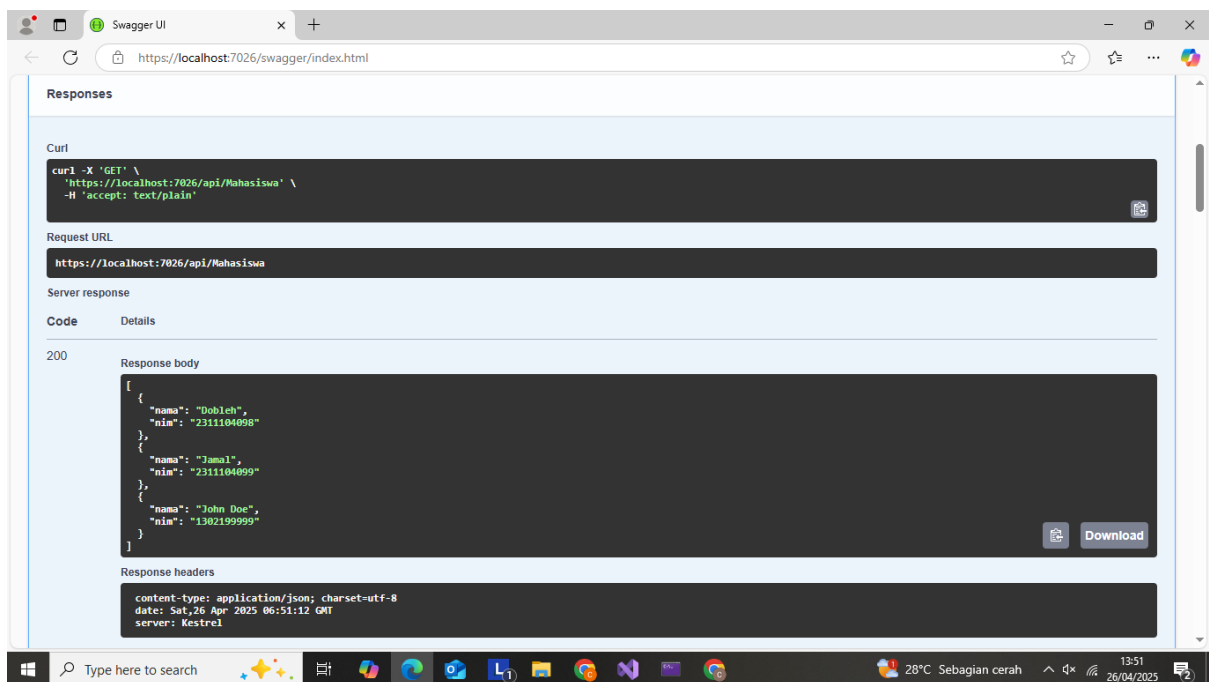
```
content-type: application/json; charset=utf-8
date: Sat, 26 Apr 2025 06:58:40 GMT
```

e.



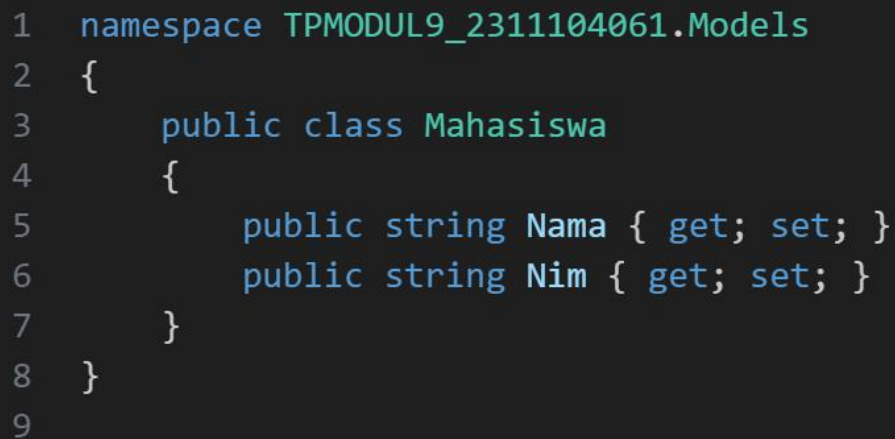
A screenshot of the Swagger UI interface in a web browser. The browser tab is titled "Swagger UI" and the address bar shows "https://localhost:7026/swagger/index.html". The main content area displays an API endpoint with a red background. The endpoint is labeled "index" with a red asterisk indicating it is required. The parameter is "integer(\$int32)" with a value of "0" entered in a text box. Below the parameter is a "path" label. There are two buttons: "Execute" (blue) and "Clear" (white). The "Responses" section shows a 200 status code with response headers: "content-length: 0", "date: Sat, 26 Apr 2025 06:51:00 GMT", and "server: Kestrel". The "Curl" section shows a DELETE request: "curl -X 'DELETE' \ 'https://localhost:7026/api/Mahasiswa/0' \ -H 'accept: */*'". The "Request URL" section shows "https://localhost:7026/api/Mahasiswa/0". The "Server response" section shows the 200 status code and headers. The "Responses" section shows the 200 status code and headers.

f.



A screenshot of the Swagger UI interface in a web browser. The browser tab is titled "Swagger UI" and the address bar shows "https://localhost:7026/swagger/index.html". The main content area displays an API endpoint with a blue background. The endpoint is labeled "index" with a red asterisk indicating it is required. The parameter is "integer(\$int32)" with a value of "0" entered in a text box. Below the parameter is a "path" label. There are two buttons: "Execute" (blue) and "Clear" (white). The "Responses" section shows a 200 status code with response headers: "content-type: application/json; charset=utf-8", "date: Sat, 26 Apr 2025 06:51:12 GMT", and "server: Kestrel". The "Curl" section shows a GET request: "curl -X 'GET' \ 'https://localhost:7026/api/Mahasiswa' \ -H 'accept: text/plain'". The "Request URL" section shows "https://localhost:7026/api/Mahasiswa". The "Server response" section shows the 200 status code and headers. The "Responses" section shows the 200 status code and headers. The "Response body" section shows a JSON array of three objects: [{"nama": "Doblich", "nim": "2311104098"}, {"nama": "Jamaal", "nim": "2311104099"}, {"nama": "John Doe", "nim": "1302199999"}]. The "Response headers" section shows the headers.

kode implementasi



```
1 namespace TPMODUL9_2311104061.Models
2 {
3     public class Mahasiswa
4     {
5         public string Nama { get; set; }
6         public string Nim { get; set; }
7     }
8 }
9
```

Kode tersebut mendefinisikan sebuah class bernama Mahasiswa di dalam namespace tpmodul9_2311104061.Models. Di dalam class Mahasiswa, terdapat dua properti publik yaitu Nama dan Nim, yang masing-masing bertipe data string. Properti Nama digunakan untuk menyimpan nama mahasiswa, sedangkan Nim digunakan untuk menyimpan Nomor Induk Mahasiswa (NIM). Dengan modifier public, kedua properti ini dapat diakses dan dimodifikasi dari luar class. Penulisan class ini mengikuti konsep POCO (Plain Old CLR Object) di C#, yang umum digunakan untuk merepresentasikan data sederhana dalam aplikasi berbasis .NET, khususnya dalam pengembangan Web API. Class ini nantinya akan digunakan untuk membuat objek-objek mahasiswa yang akan dikelola melalui API.

```
1  using Microsoft.AspNetCore.Mvc;
2  using TPMODUL9_2311104061.Models;
3  using System.Collections.Generic;
4
5  namespace tpmodul9_2311104061.Controllers
6  {
7      [ApiController]
8      [Route("api/[controller]")]
9      public class MahasiswaController : ControllerBase
10     {
11         private static List<Mahasiswa> daftarMahasiswa = new List<Mahasiswa>
12         {
13             new Mahasiswa { Nama = "Candra", Nim = "2311104061" },
14             new Mahasiswa { Nama = "Dobleh", Nim = "2311104098" },
15             new Mahasiswa { Nama = "Jama1", Nim = "2311104099" }
16         };
17
18         [HttpGet]
19         public ActionResult<List<Mahasiswa>> GetAll()
20         {
21             return daftarMahasiswa;
22         }
23
24         [HttpGet("{index}")]
25         public ActionResult<Mahasiswa> GetByIndex(int index)
26         {
27             if (index < 0 || index >= daftarMahasiswa.Count)
28                 return NotFound();
29             return daftarMahasiswa[index];
30         }
31
32         [HttpPost]
33         public ActionResult AddMahasiswa([FromBody] Mahasiswa mhs)
34         {
35             daftarMahasiswa.Add(mhs);
36             return Ok();
37         }
38
39         [HttpDelete("{index}")]
40         public ActionResult DeleteMahasiswa(int index)
41         {
42             if (index < 0 || index >= daftarMahasiswa.Count)
43                 return NotFound();
44             daftarMahasiswa.RemoveAt(index);
45             return Ok();
46         }
47     }
48 }
49
```

Kode tersebut mendefinisikan sebuah controller Web API bernama MahasiswaController di dalam namespace tpmodul9_2311104061.Controllers. Controller ini menggunakan atribut [ApiController] dan [Route("api/[controller]")], yang berarti semua endpoint di controller ini akan tersedia melalui URL api/mahasiswa. Di dalamnya, terdapat sebuah list statis daftarMahasiswa yang berisi beberapa data mahasiswa awal, masing-masing berupa objek dari class Mahasiswa yang memiliki properti Nama dan Nim. Controller ini menyediakan empat endpoint HTTP: pertama, [HttpGet] untuk mengambil seluruh daftar mahasiswa; kedua, [HttpGet("{index}")] untuk mengambil data mahasiswa berdasarkan indeks tertentu; ketiga, [HttpPost] untuk menambahkan mahasiswa baru ke dalam list melalui data yang dikirim di body request; dan keempat, [HttpDelete("{index}")] untuk menghapus data mahasiswa berdasarkan indeks. Setiap metode dilengkapi dengan validasi sederhana untuk mengecek apakah indeks yang diminta valid, dan akan mengembalikan NotFound jika tidak valid. Secara keseluruhan, kode ini membentuk API CRUD sederhana tanpa database menggunakan penyimpanan berbasis list static.