

SE0701_2311104003_MARTRYATUS SOFIA

MODUL 9

Link Github :

https://github.com/MartryatusSofia/KPL_Martryatus-Sofia_2311104003_SE0701/tree/master/09_API%20Design%20dan%20Construction%20Using%20Swagger

Screenshot hasil running :

GET - "GET /api/mahasiswa"

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

- Curl:** `curl -X 'GET' \ 'http://localhost:5000/api/mahasiswa' \ -H 'accept: */*'`
- Request URL:** `http://localhost:5000/api/mahasiswa`
- Server response:** **Code** 200, **Details**
- Response body:**

```
[
  {
    "nama": "Martryatus Sofia",
    "nim": "2311104003"
  },
  {
    "nama": "Sinta Alimah",
    "nim": "23111040100"
  },
  {
    "nama": "Banyu Bim",
    "nim": "23111040101"
  }
]
```
- Response headers:** (empty)

POST - "POST /api/mahasiswa"

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

- No parameters**
- Request body** required: `{ "nama": "John Doe", "nim": "1302199999" }`
- Content-Type:** `application/json`
- Buttons:** **Execute** and **Clear**

GET setelah data ditambahkan - “GET /api/mahasiswa”

Curl

```
curl -X 'GET' \
  'http://localhost:5000/api/mahasiswa' \
  -H 'accept: */*'
```

Request URL

http://localhost:5000/api/mahasiswa

Server response

Code	Details
200	<p>Response body</p> <pre>{ { "nama": "Martryatus Sofia", "nim": "2311104003" }, { "nama": "Sinta Alimah", "nim": "23111040100" }, { "nama": "Banyu Rim", "nim": "23111040101" }, { "nama": "John Doe", "nim": "1302199999" } }</pre> <p>Download</p>

GET berdasarkan index 0 - “GET /api/mahasiswa/0”

Curl

```
curl -X 'GET' \
  'http://localhost:5000/api/mahasiswa/0' \
  -H 'accept: */*'
```

Request URL

http://localhost:5000/api/mahasiswa/0

Server response

Code	Details
200	<p>Response body</p> <pre>{ "nama": "Martryatus Sofia", "nim": "2311104003" }</pre> <p>Download</p> <p>Response headers</p> <pre>connection: keep-alive content-length: 46 content-type: application/json; charset=utf-8 date: Sat, 26 Apr 2025 14:26:11 GMT etag: W/"2e-qWtNxiwT8ebjySGjQ9kQp4H8v4" keep-alive: timeout=5 x-powered-by: Express</pre>

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

Curl

```
curl -X 'DELETE' \
'http://localhost:5000/api/mahasiswa/0' \
-H 'accept: */*' \
```

Request URL

```
http://localhost:5000/api/mahasiswa/0
```

Server response

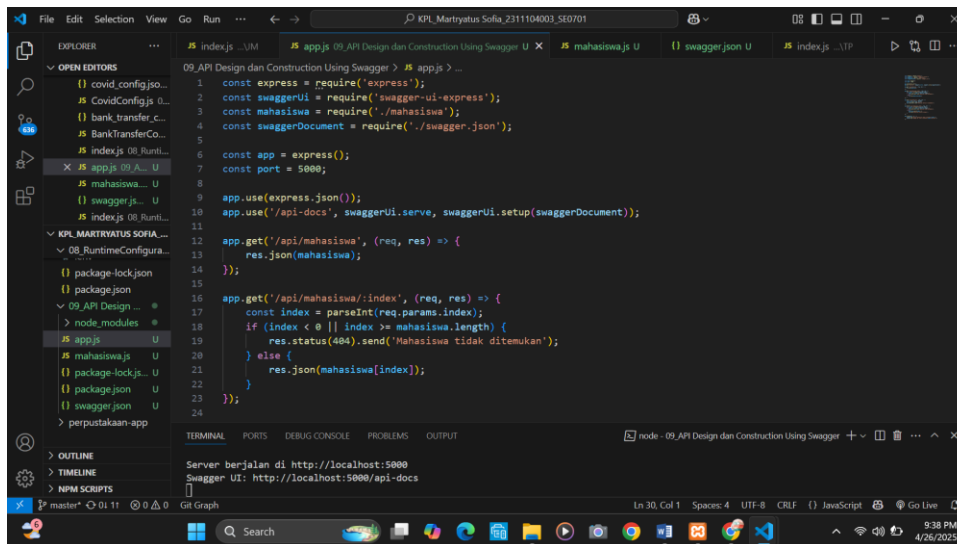
Code	Details
200	<p>Response body</p> <p>Mahasiswa berhasil dihapus</p> <p>Response headers</p> <pre>connection: keep-alive content-length: 26 content-type: text/html; charset=utf-8 date: Sat, 26 Apr 2025 14:35:01 GMT etag: W/"1a-5y4xEyBR1GZ7McIuLz1/2IHDCQc" keep-alive: timeout=5 x-powered-by: Express</pre>

Cek list/array dari semua mahasiswa sekali lagi dengan “GET /api/mahasiswa”,

Code	Details	
200	<p>Response body</p> <pre>[{ "nama": "Sinta Alimah", "nim": "23111040100" }, { "nama": "Banyu", "nim": "23111040101" }, { "nama": "John Doe", "nim": "1302199999" }]</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 121 content-type: application/json; charset=utf-8 date: Sat, 26 Apr 2025 14:36:12 GMT etag: W/"79-gynsaofAAaTVfEMRktK6533RM1k" keep-alive: timeout=5 x-powered-by: Express</pre>	
Responses		
Code	Description	Links
200	List mahasiswa	No links

Penjelasan Code:

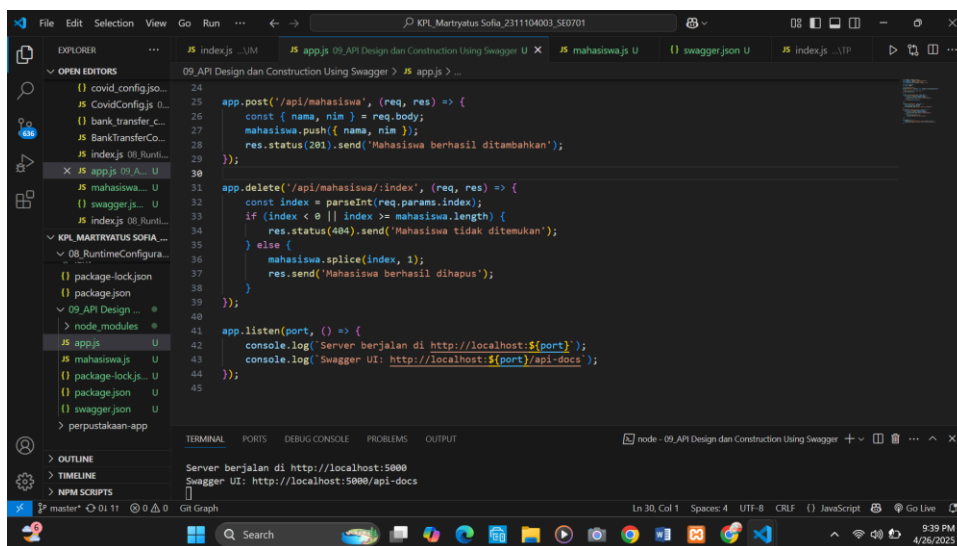
app.js



```
1 const express = require('express');
2 const swaggerUi = require('swagger-ui-express');
3 const mahasiswa = require('./mahasiswa');
4 const swaggerDocument = require('./swagger.json');
5
6 const app = express();
7 const port = 5000;
8
9 app.use(express.json());
10 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
11
12 app.get('/api/mahasiswa', (req, res) => {
13   res.json(mahasiswa);
14 });
15
16 app.get('/api/mahasiswa/:index', (req, res) => {
17   const index = parseInt(req.params.index);
18   if (index < 0 || index >= mahasiswa.length) {
19     res.status(404).send('Mahasiswa tidak ditemukan');
20   } else {
21     res.json(mahasiswa[index]);
22   }
23 });
24
```

TERMINAL

```
Server berjalan di http://localhost:5000
Swagger UI: http://localhost:5000/api-docs
```



```
24
25 app.post('/api/mahasiswa', (req, res) => {
26   const { nama, nim } = req.body;
27   mahasiswa.push({ nama, nim });
28   res.status(201).send('Mahasiswa berhasil ditambahkan');
29 });
30
31 app.delete('/api/mahasiswa/:index', (req, res) => {
32   const index = parseInt(req.params.index);
33   if (index < 0 || index >= mahasiswa.length) {
34     res.status(404).send('Mahasiswa tidak ditemukan');
35   } else {
36     mahasiswa.splice(index, 1);
37     res.send('Mahasiswa berhasil dihapus');
38   }
39 });
40
41 app.listen(port, () => {
42   console.log('Server berjalan di http://localhost:${port}');
43   console.log('Swagger UI: http://localhost:${port}/api-docs');
44 });
45
```

TERMINAL

```
Server berjalan di http://localhost:5000
Swagger UI: http://localhost:5000/api-docs
```

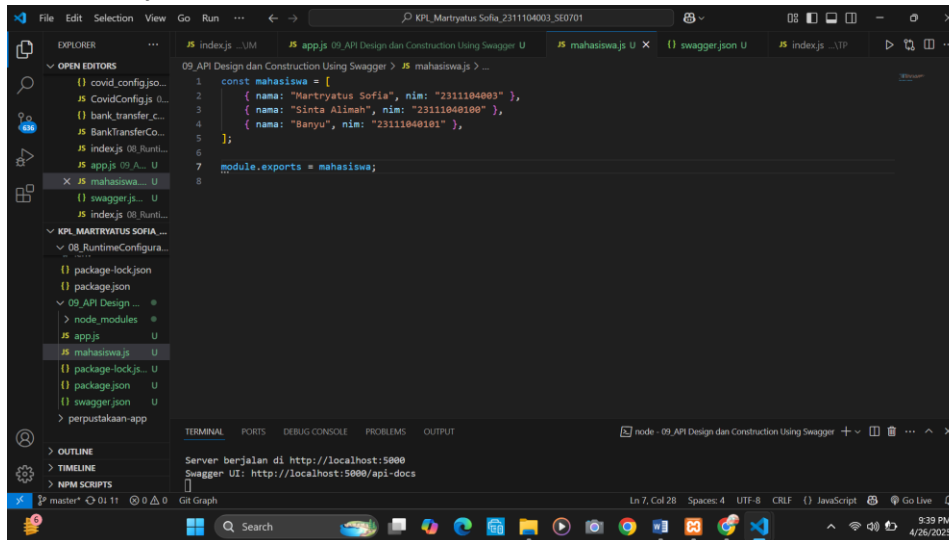
Kode ini membuat Web API sederhana dengan Node.js dan Express, yang bisa melakukan operasi pada data Mahasiswa.

Fitur-fiturnya:

1. Import library:
 - 1) express: untuk membuat server.
 - 2) swagger-ui-express: untuk membuat dokumentasi API di Swagger.
 - 3) mahasiswa.js: file berisi list data mahasiswa awal.
 - 4) swagger.json: file konfigurasi dokumentasi Swagger.
2. Setup server:
 - 1) Membuat server di port 5000.
 - 2) Mengaktifkan Swagger UI di /api-docs.
3. Endpoint API:
 - 1) GET /api/mahasiswa: Mengambil semua data mahasiswa.

- 2) GET /api/mahasiswa/:index: Mengambil 1 mahasiswa berdasarkan index.
- 3) POST /api/mahasiswa: Menambahkan mahasiswa baru ke dalam list.
- 4) DELETE /api/mahasiswa/:index: Menghapus mahasiswa berdasarkan index.
4. Menjalankan server:
 - 1) Menampilkan URL server dan URL Swagger saat server dinyalakan.

mahasiswa.js



```
1 const mahasiswa = [
2   { nama: "Martryatus Sofia", nim: "2311104003" },
3   { nama: "Sinta Alimah", nim: "23111040100" },
4   { nama: "Banyu", nim: "23111040101" },
5 ];
6
7 module.exports = mahasiswa;
```

Terminal output:

```
Server berjalan di http://localhost:5000
Swagger UI: http://localhost:5000/api-docs
```

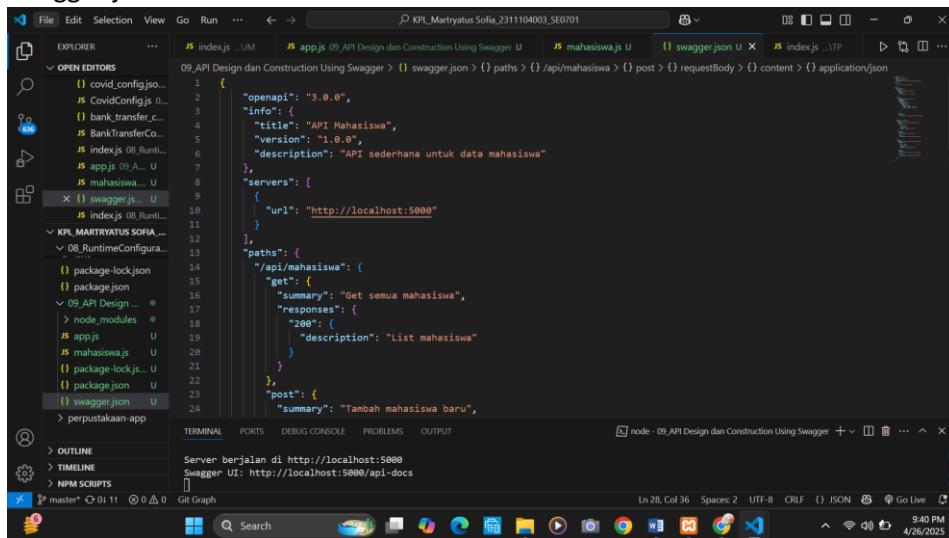
Kode ini membuat array bernama mahasiswa yang berisi data 3 orang mahasiswa.

Setiap mahasiswa punya dua properti:

- a) nama: nama mahasiswa
- b) nim: nomor induk mahasiswa

Setelah itu, array mahasiswa ini di-export supaya bisa dipakai di file lain (seperti di app.js).

swagger.js



```
1 {
2   "openapi": "3.0.0",
3   "info": {
4     "title": "API Mahasiswa",
5     "version": "1.0.0",
6     "description": "API sederhana untuk data mahasiswa"
7   },
8   "servers": [
9     {
10      "url": "http://localhost:5000"
11    }
12  ],
13  "paths": {
14    "/api/mahasiswa": {
15      "get": {
16        "summary": "Get semua mahasiswa",
17        "responses": {
18          "200": {
19            "description": "List mahasiswa"
20          }
21        }
22      },
23      "post": {
24        "summary": "Tambah mahasiswa baru",
```

Terminal output:

```
Server berjalan di http://localhost:5000
Swagger UI: http://localhost:5000/api-docs
```

File ini mendeskripsikan semua endpoint, method, input, dan output untuk API Mahasiswa dalam format Swagger/OpenAPI.