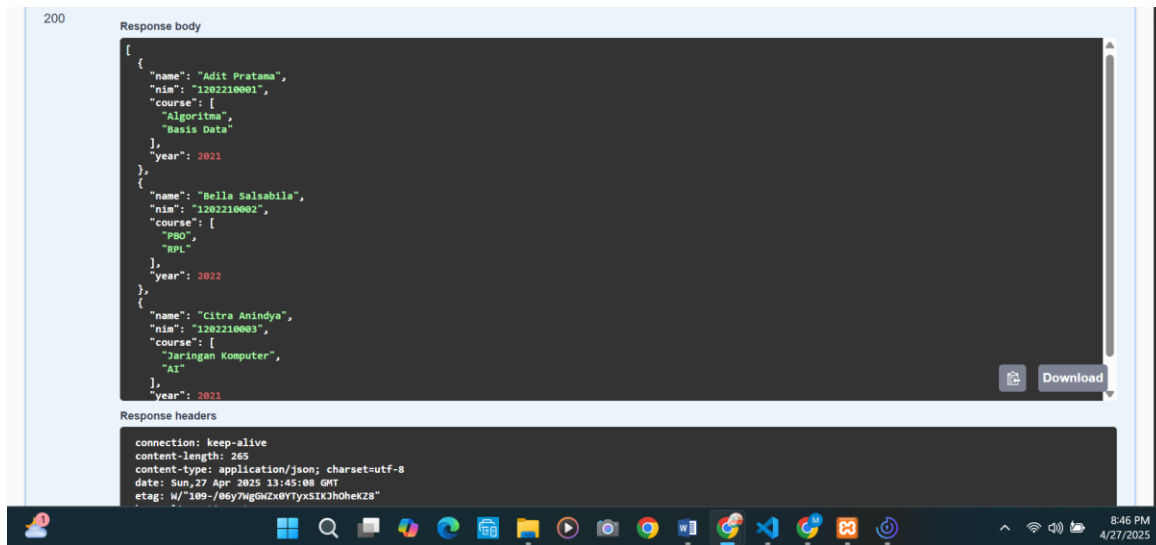


Link Github :

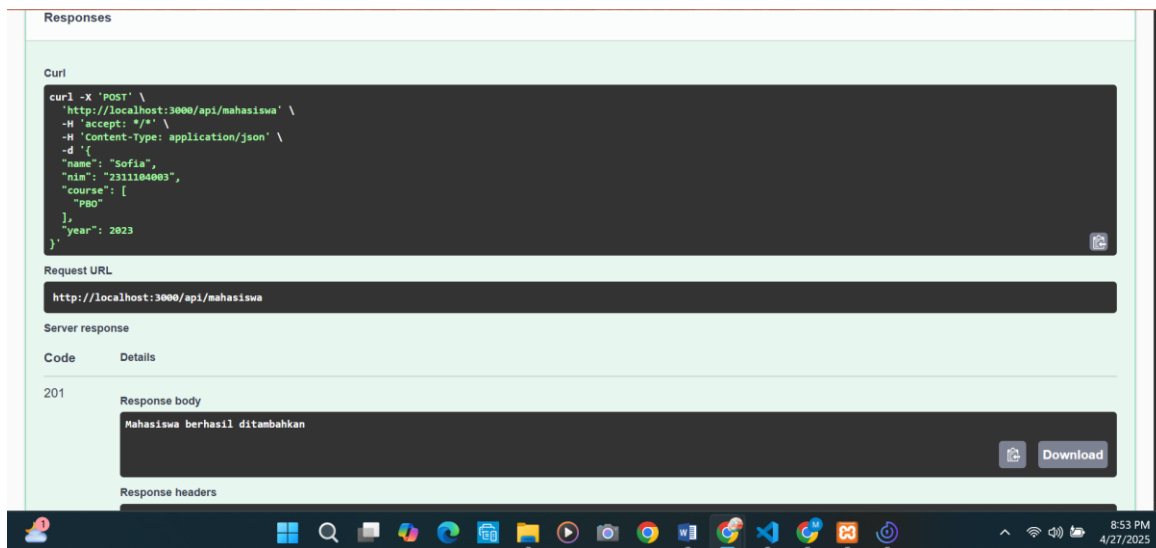
https://github.com/MartryatusSofia/KPL_Martryatus-Sofia_2311104003_SE0701/tree/master/09_API%20Design%20dan%20Construction%20Using%20Swagger

Screenshot hasil running :

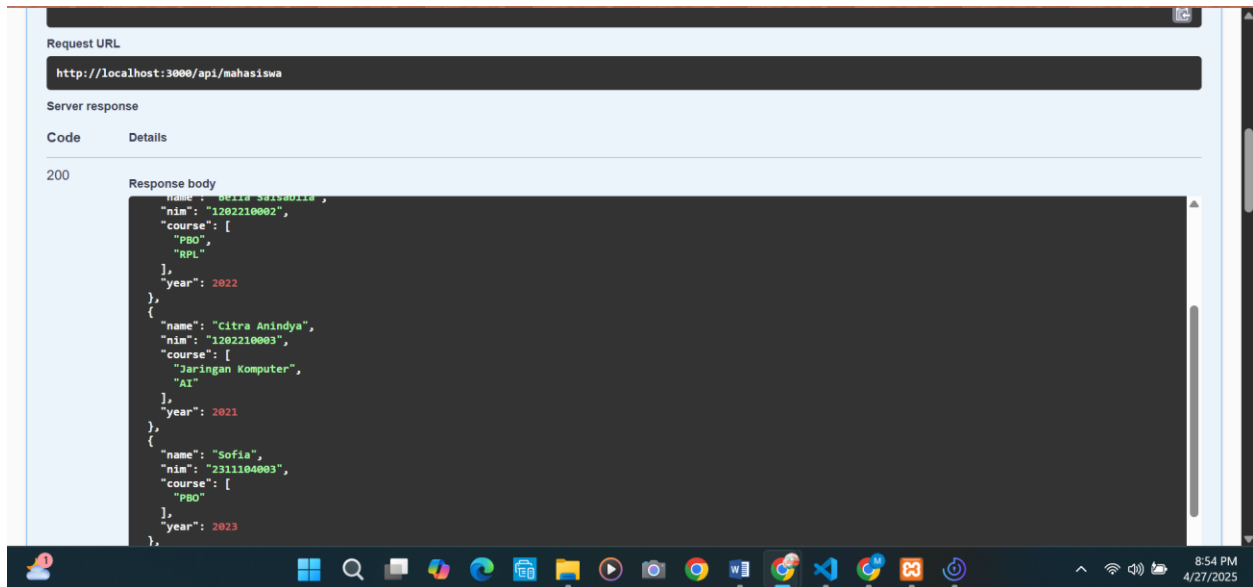
“GET /api/Mahasiswa”



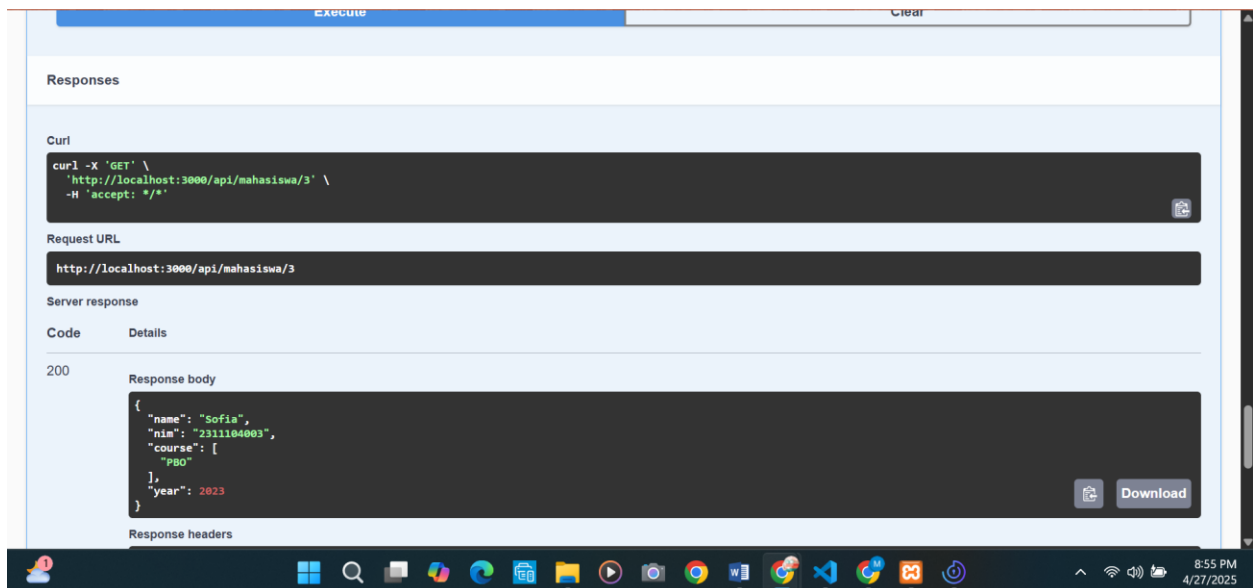
Menambahkan Mahasiswa baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Mahasiswa”



Cek list/array dari semua Mahasiswa lagi dengan “GET /api/Mahasiswa”, pastikan Mahasiswa yang baru ditambahkan sebelumnya sudah ada



Mencoba meminta Mahasiswa dengan index 3, “GET /api/Mahasiswa/3” yang seharusnya mengembalikan Mahasiswa yang baru saja ditambah



Menghapus objek Mahasiswa dengan index ke-1 dengan “DELETE /api/Mahasiswa/1”

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

- Curl:**

```
curl -X 'DELETE' \
'http://localhost:3000/api/mahasiswa/1' \
-H 'accept: */*'
```
- Request URL:** `http://localhost:3000/api/mahasiswa/1`
- Server response:**
 - Code:** 200
 - Response body:** `Mahasiswa berhasil dihapus`
 - Response headers:**

```
connection: keep-alive
content-length: 26
content-type: text/html; charset=utf-8
date: Sun, 27 Apr 2025 13:55:59 GMT
etag: W/"1a-5y4xEy8R1GZ7McIuLz1/2iHDCqc"
keep-alive: timeout=5
x-powered-by: Express
```

The Windows taskbar at the bottom shows the time as 8:56 PM on 4/27/2025.

Cek list/array dari semua Mahasiswa sekali lagi dengan “GET /api/Mahasiswa.”

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

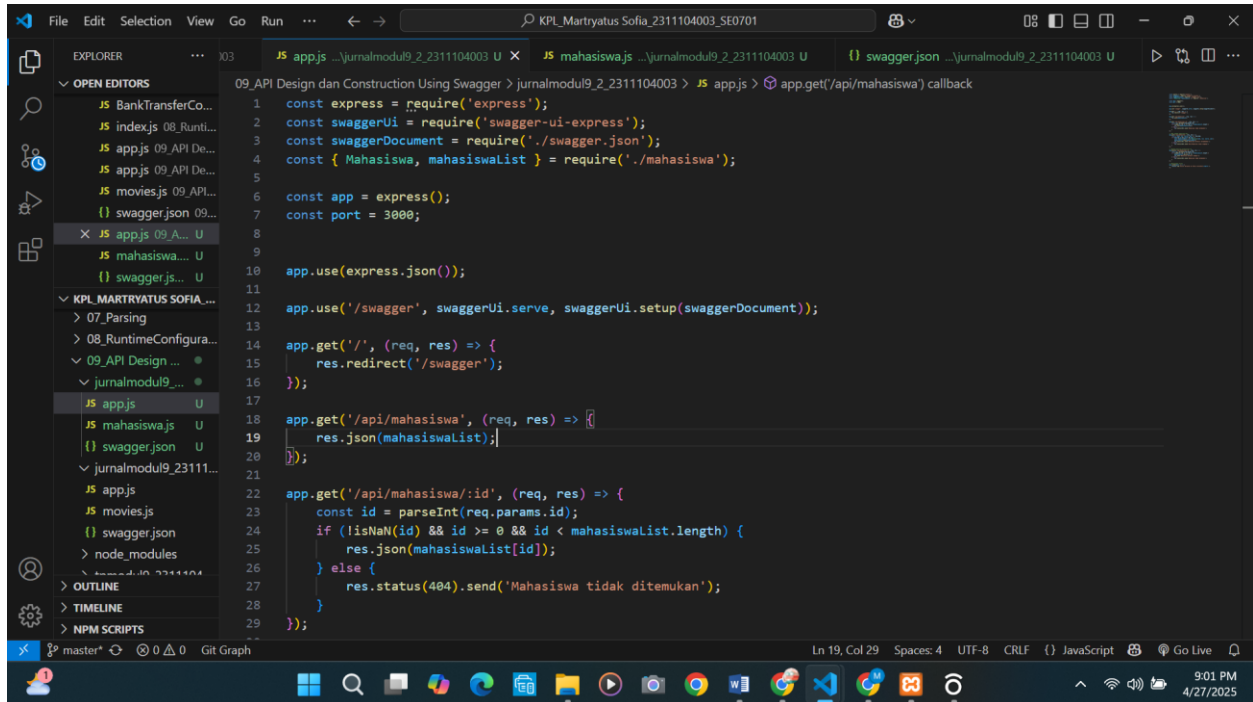
- Request URL:** `http://localhost:3000/api/mahasiswa`
- Server response:**
 - Code:** 200
 - Response body:**

```
[
  {
    "name": "Adit Pratama",
    "nim": "1202210001",
    "course": [
      "Algoritma",
      "Basis Data"
    ],
    "year": 2021
  },
  {
    "name": "Citra Anindya",
    "nim": "1202210003",
    "course": [
      "Jaringan Komputer",
      "AI"
    ],
    "year": 2021
  },
  {
    "name": "Sofia",
    "nim": "2311104003",
    "course": [
      "PBO"
    ],
    "year": 2023
  }
]
```

The Windows taskbar at the bottom shows the time as 8:57 PM on 4/27/2025.

Penjelasan Code Program

app.js



```
1 const express = require('express');
2 const swaggerUi = require('swagger-ui-express');
3 const swaggerDocument = require('./swagger.json');
4 const { Mahasiswa, mahasiswaList } = require('./mahasiswa');
5
6 const app = express();
7 const port = 3000;
8
9
10 app.use(express.json());
11
12 app.use('/swagger', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
13
14 app.get('/', (req, res) => {
15   res.redirect('/swagger');
16 });
17
18 app.get('/api/mahasiswa', (req, res) => {
19   res.json(mahasiswaList);
20 });
21
22 app.get('/api/mahasiswa/:id', (req, res) => {
23   const id = parseInt(req.params.id);
24   if (!isNaN(id) && id >= 0 && id < mahasiswaList.length) {
25     res.json(mahasiswaList[id]);
26   } else {
27     res.status(404).send('Mahasiswa tidak ditemukan');
28   }
29 });
```

File utama untuk menjalankan server Express. Definisikan semua route API (GET, POST, DELETE) untuk mahasiswa. Integrasi dengan Swagger UI supaya API bisa dites dari browser.

GET /api/mahasiswa → Ambil semua data mahasiswa.

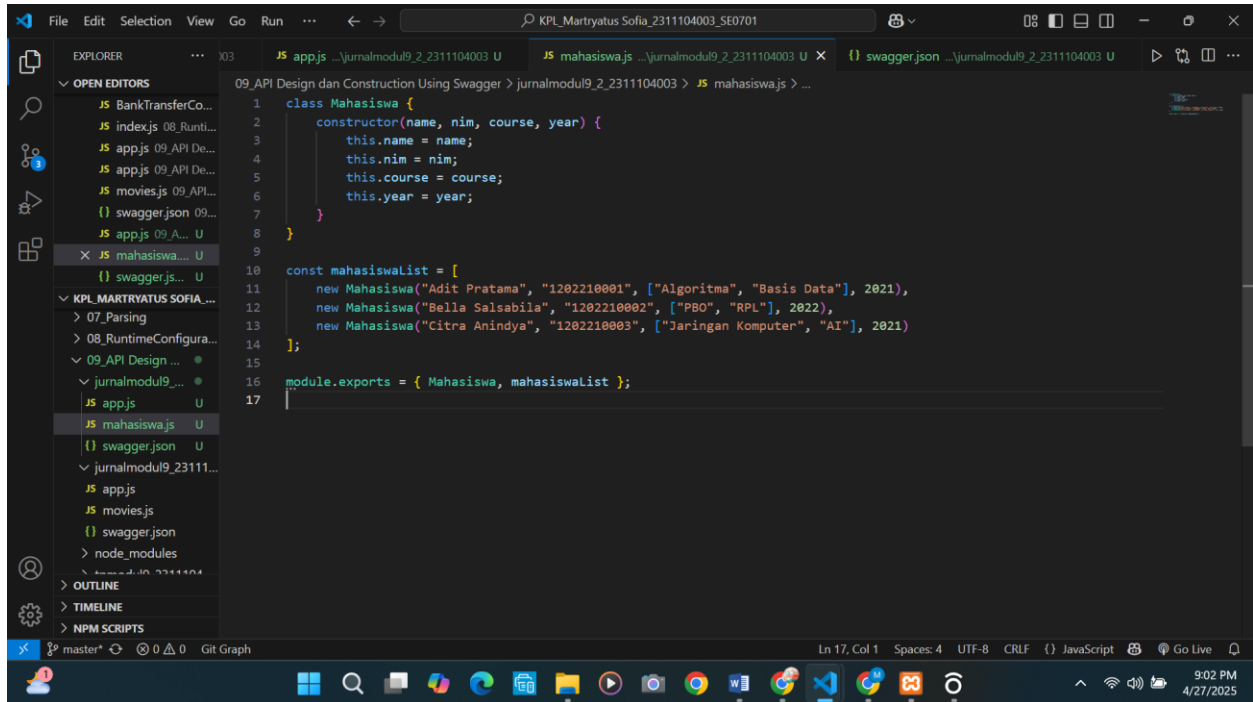
GET /api/mahasiswa/:id → Ambil 1 data mahasiswa berdasarkan index.

POST /api/mahasiswa → Menambah mahasiswa baru.

DELETE /api/mahasiswa/:id → Menghapus mahasiswa berdasarkan index.

Redirect / → ke /swagger otomatis.

Mahasiswa.js



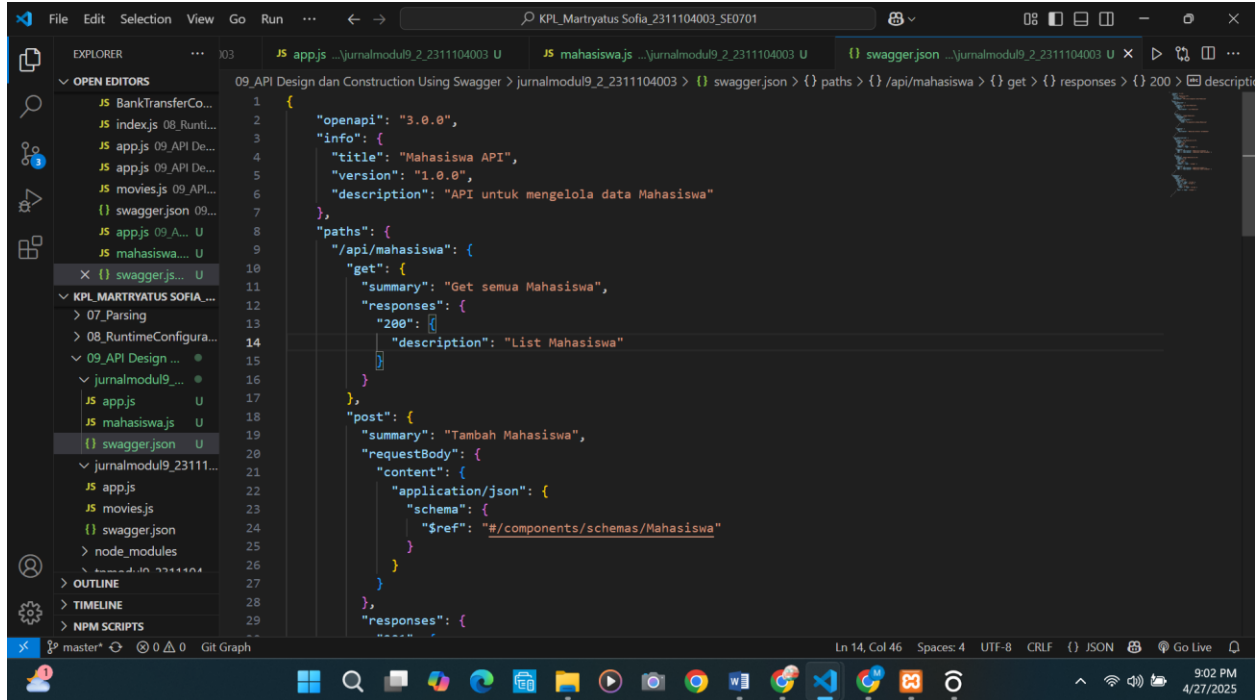
```
1 class Mahasiswa {
2   constructor(name, nim, course, year) {
3     this.name = name;
4     this.nim = nim;
5     this.course = course;
6     this.year = year;
7   }
8 }
9
10 const mahasiswaList = [
11   new Mahasiswa("Adit Pratama", "1202210001", ["Algoritma", "Basis Data"], 2021),
12   new Mahasiswa("Bella Salsabila", "1202210002", ["PBO", "RPL"], 2022),
13   new Mahasiswa("Citra Anindya", "1202210003", ["Jaringan Komputer", "AI"], 2021)
14 ];
15
16 module.exports = { Mahasiswa, mahasiswaList };
17
```

Fungsi ini Menyimpan class Mahasiswa dan list data mahasiswa (statis).

class Mahasiswa → Blueprint objek mahasiswa (punya name, nim, course, year).

mahasiswaList → Isi awal (3 mahasiswa contoh dari anggota kelompok).

Swagger.json



File konfigurasi untuk Swagger documentation. Menjelaskan ke Swagger endpoint apa saja yang ada di API-mu.

/api/mahasiswa → Untuk GET semua mahasiswa dan POST mahasiswa baru.

/api/mahasiswa/{id} → Untuk GET dan DELETE mahasiswa berdasarkan ID.

Definisi schema Mahasiswa: terdiri dari name, nim, course, dan year.