

SE0701_2311104003_MARTRYATUS SOFIA

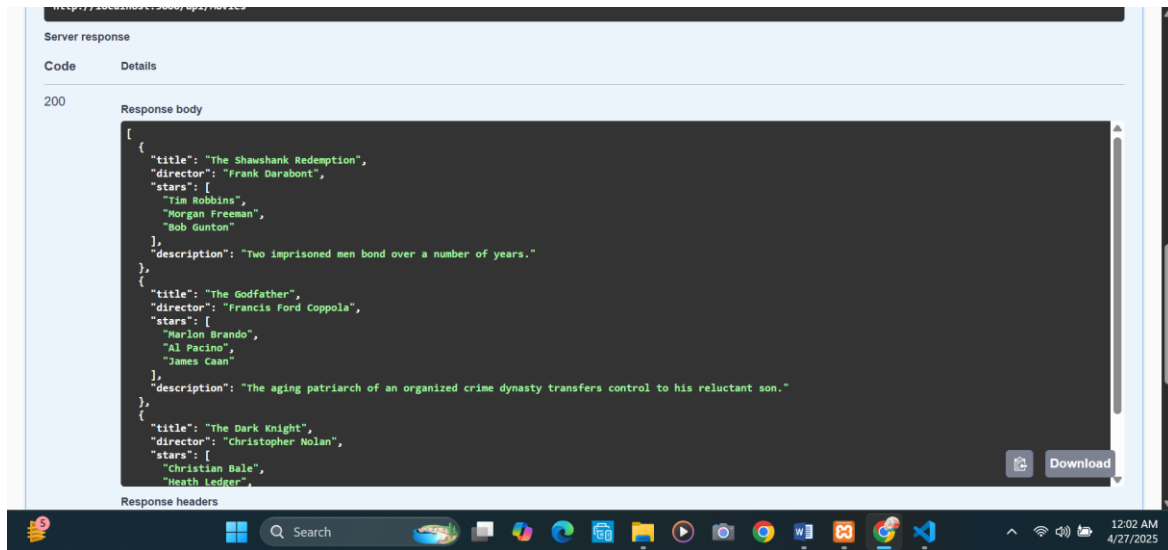
MODUL 9

Link Github :

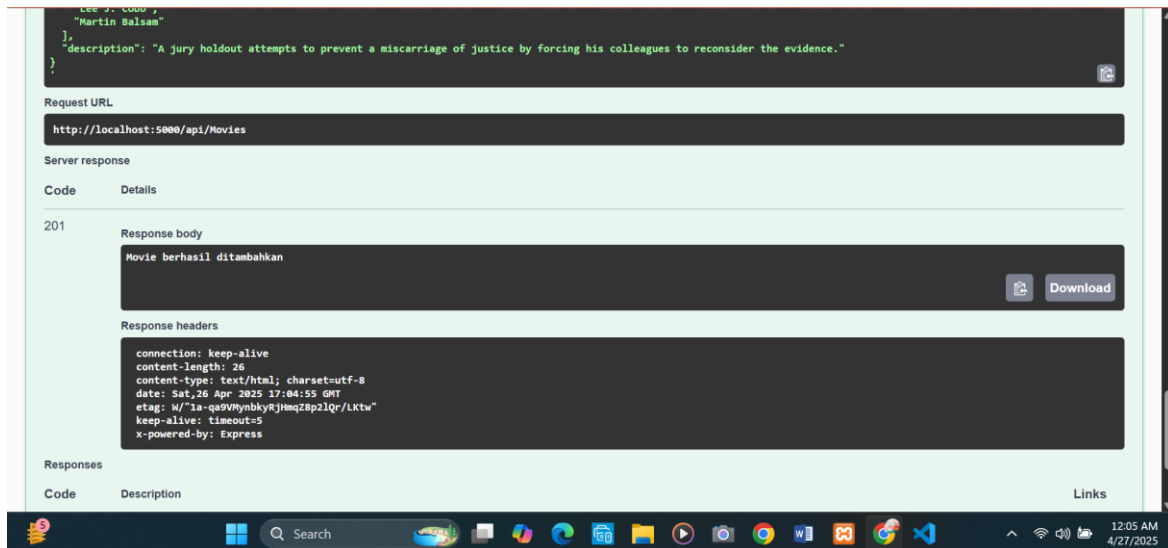
https://github.com/MartryatusSofia/KPL_Martryatus-Sofia_2311104003_SE0701/tree/master/09_API%20Design%20dan%20Construction%20Using%20Swagger

Screenshot hasil running :

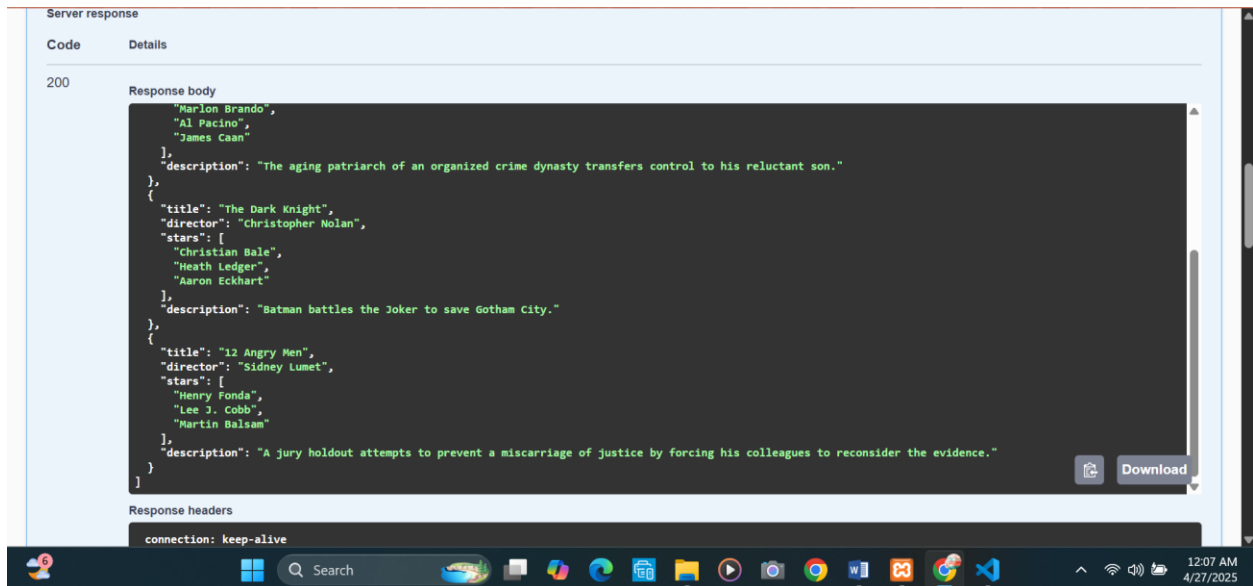
GET /api/Movies



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



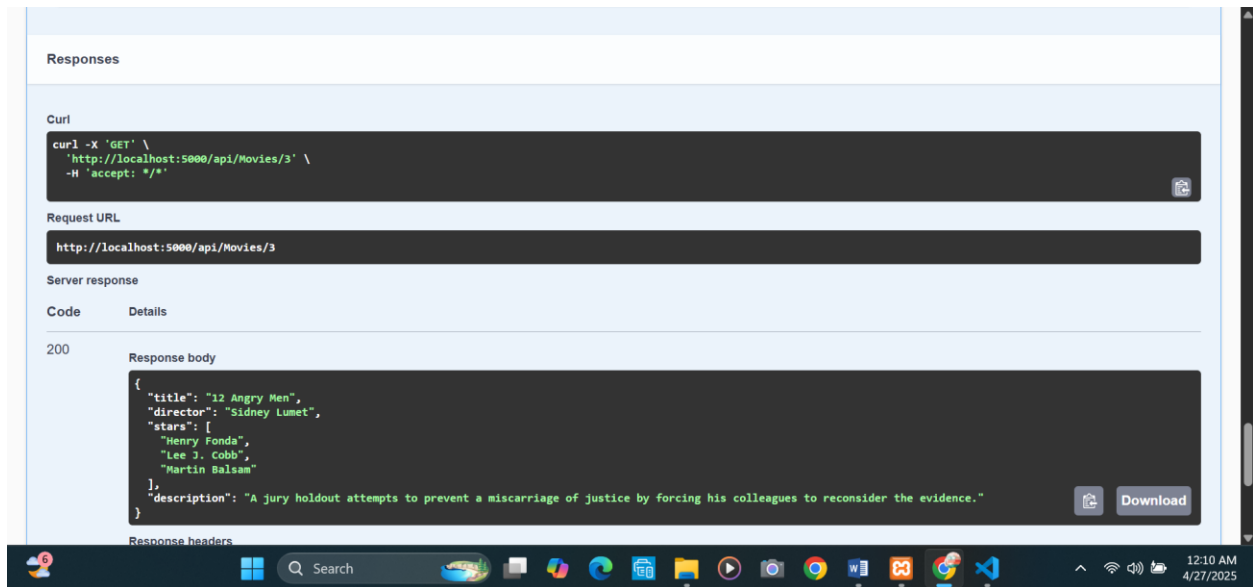
Cek list/array dari semua Movie lagi dengan “GET /api/Movies”



The screenshot shows a REST client interface with a 'Server response' tab selected. The response code is 200. The response body is a JSON array of movie objects. The first movie is 'The Godfather' with stars 'Marlon Brando', 'Al Pacino', and 'James Caan'. The second movie is 'The Dark Knight' with director 'Christopher Nolan' and stars 'Christian Bale', 'Heath Ledger', and 'Aaron Eckhart'. The third movie is '12 Angry Men' with director 'Sidney Lumet' and stars 'Henry Fonda', 'Lee J. Cobb', and 'Martin Balsam'. The response headers show 'connection: keep-alive'.

```
200
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "stars": [
    "Marlon Brando",
    "Al Pacino",
    "James Caan"
  ],
  "description": "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."
},
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger",
    "Aaron Eckhart"
  ],
  "description": "Batman battles the Joker to save Gotham City."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb",
    "Martin Balsam"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence."
}
]
Response headers
connection: keep-alive
```

Mencoba meminta Movie dengan index 3, “GET /api/Movies/3”



The screenshot shows a REST client interface with a 'Responses' tab selected. The 'Curl' section shows the command: `curl -X 'GET' \ 'http://localhost:5000/api/Movies/3' \ -H 'accept: */*'`. The 'Request URL' is `http://localhost:5000/api/Movies/3`. The 'Server response' tab shows a 200 status code. The response body is a JSON object for the movie '12 Angry Men'.

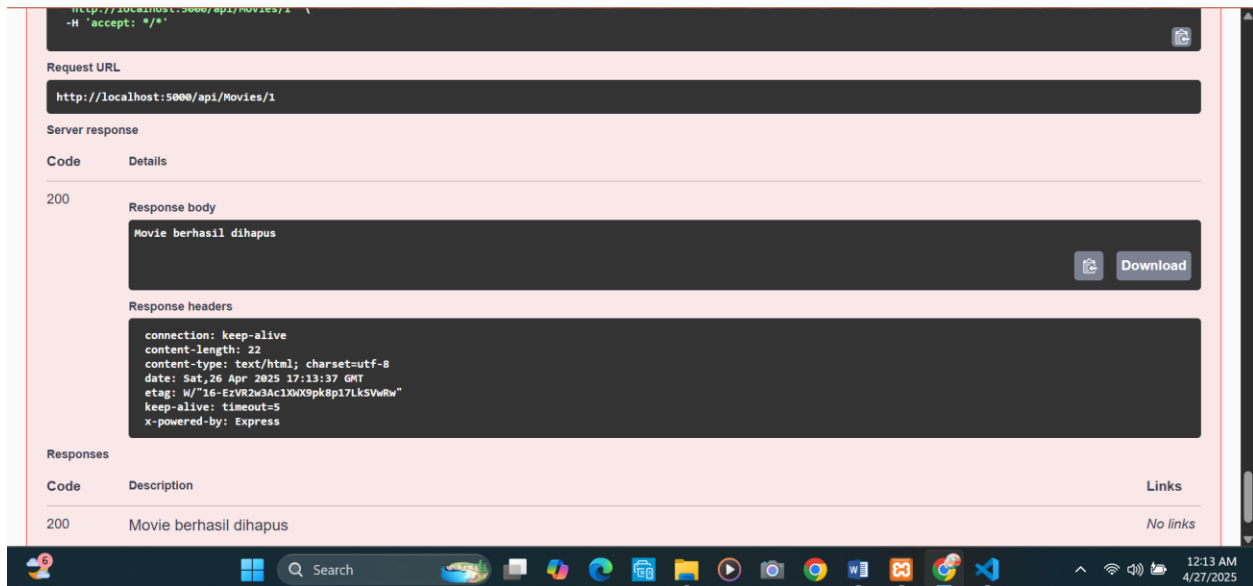
```
Responses

Curl
curl -X 'GET' \
'http://localhost:5000/api/Movies/3' \
-H 'accept: */*'

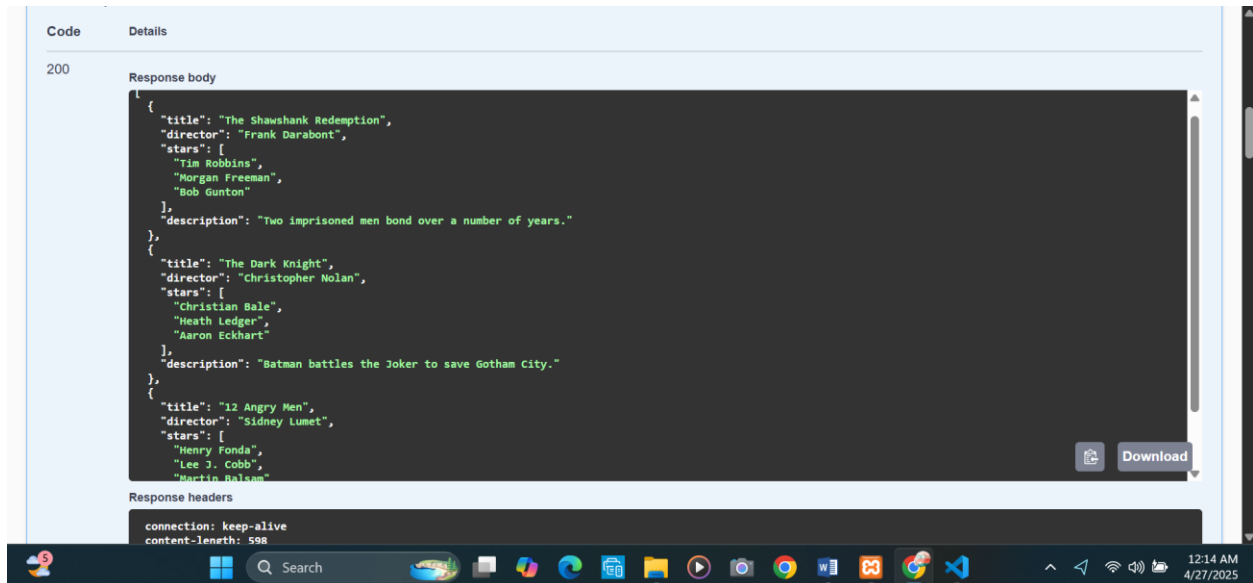
Request URL
http://localhost:5000/api/Movies/3

Server response
200
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb",
    "Martin Balsam"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice by forcing his colleagues to reconsider the evidence."
}
Response headers
```

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

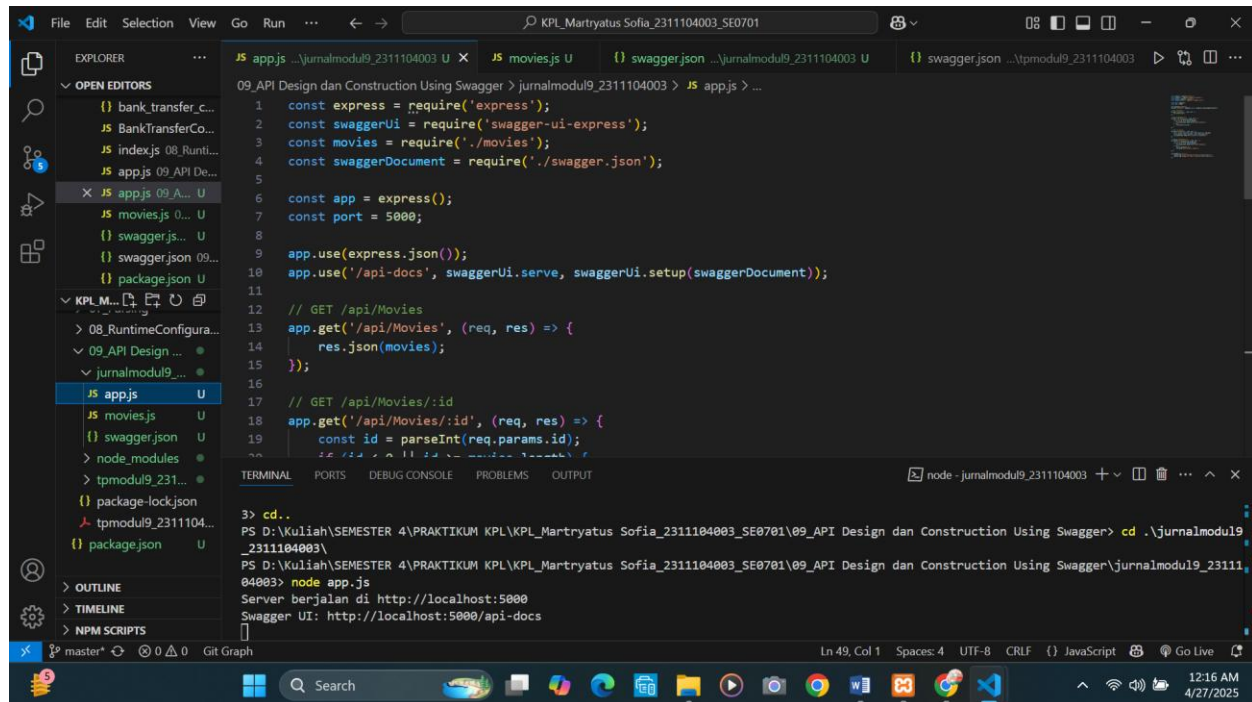


Cek list/array dari semua Movie sekali lagi dengan “GET /api/Movies”, film dengan ranking kedua “Godfather” sudah tidak ada di list:



Penjelasan Code Program

app.js



The screenshot shows a Visual Studio Code editor with the following components:

- Explorer:** Shows the file structure with 'app.js' selected under the 'jurnalmodul9_2311104003' folder.
- Editor:** Displays the code for 'app.js'. The code uses Express.js to create a REST API with two endpoints: `GET /api/Movies` and `GET /api/Movies/:id`. It also integrates Swagger UI for API documentation.
- Terminal:** Shows the command `node app.js` being executed, resulting in the message: `Server berjalan di http://localhost:5000` and `Swagger UI: http://localhost:5000/api-docs`.

membuat REST API sederhana untuk manajemen data film (movies) menggunakan Express.js dan dokumentasinya ditampilkan dengan Swagger UI.

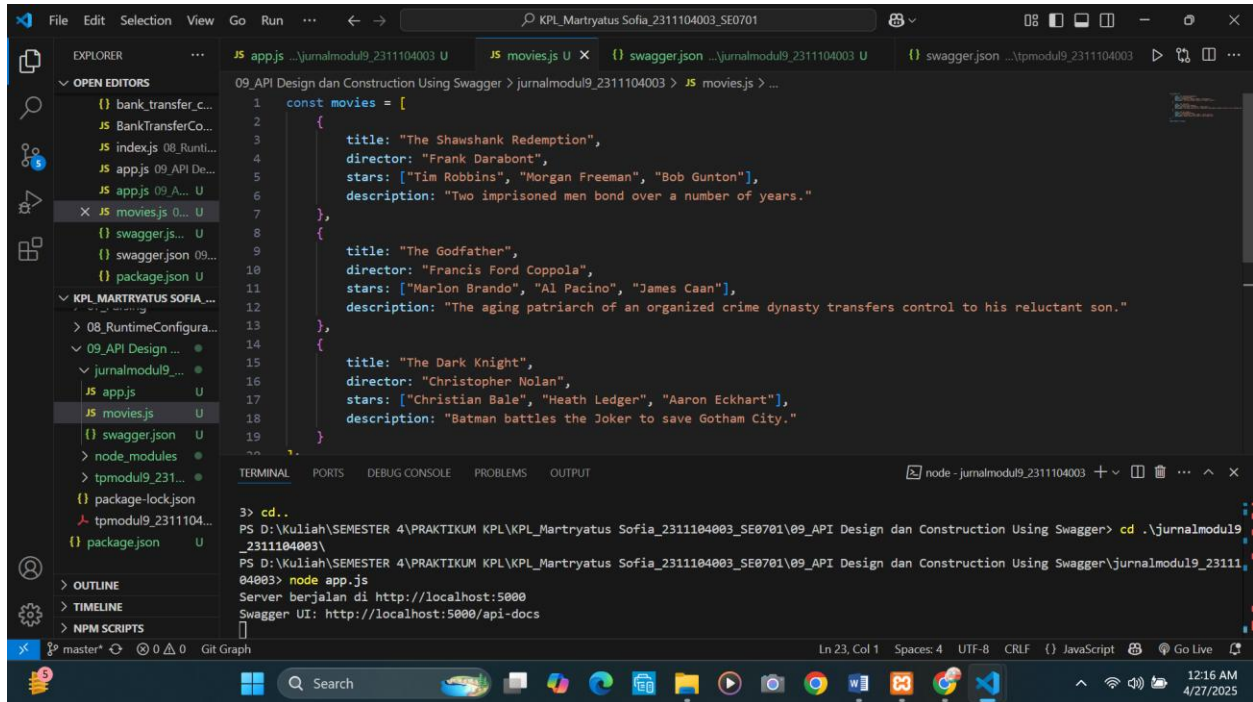
Penjelasan singkatnya:

- Menggunakan Express.js untuk membuat server dan endpoint REST API.
- Swagger UI dipasang di `/api-docs` supaya bisa melihat dokumentasi API secara visual.
- Ada 4 endpoint utama:
 1. `GET /api/Movies` → Ambil semua daftar film.
 2. `GET /api/Movies/:id` → Ambil detail satu film berdasarkan ID.
 3. `POST /api/Movies` → Menambahkan film baru ke daftar.
 4. `DELETE /api/Movies/:id` → Menghapus film berdasarkan ID.

Film-film ini disimpan di file `movies.js`

Swagger dokumennya diambil dari file `swagger.json`

movies.js



The screenshot shows a Visual Studio Code editor with a project named 'KPL_Martryatus Sofia_2311104003'. The Explorer sidebar on the left shows the file structure, including 'app.js', 'movies.js', 'swagger.json', and 'package.json'. The main editor window displays the content of 'movies.js', which defines a constant array named 'movies' containing three movie objects. The terminal window at the bottom shows the command 'cd ..' and the output of 'node app.js', which indicates the server is running on 'http://localhost:5000' and the Swagger UI is available at 'http://localhost:5000/api-docs'.

```
const movies = [  
  {  
    title: "The Shawshank Redemption",  
    director: "Frank Darabont",  
    stars: ["Tim Robbins", "Morgan Freeman", "Bob Gunton"],  
    description: "Two imprisoned men bond over a number of years."  
  },  
  {  
    title: "The Godfather",  
    director: "Francis Ford Coppola",  
    stars: ["Marlon Brando", "Al Pacino", "James Caan"],  
    description: "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."  
  },  
  {  
    title: "The Dark Knight",  
    director: "Christopher Nolan",  
    stars: ["Christian Bale", "Heath Ledger", "Aaron Eckhart"],  
    description: "Batman battles the Joker to save Gotham City."  
  }  
]
```

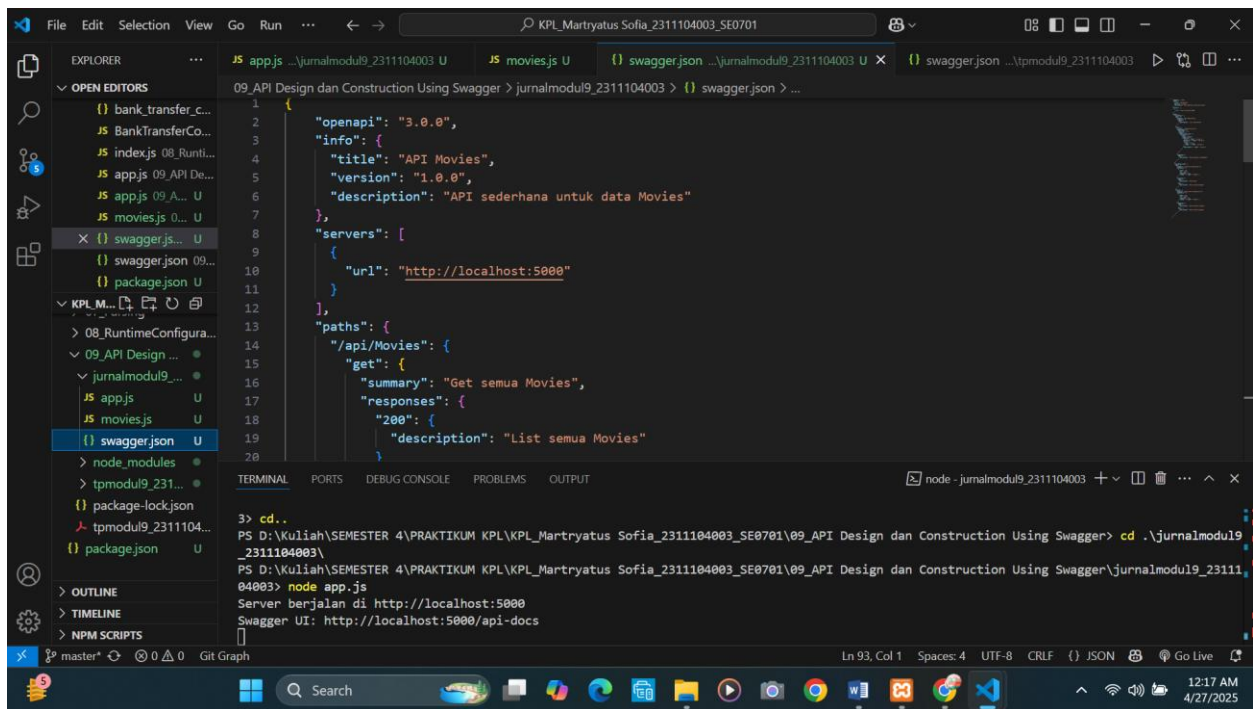
```
3> cd..  
PS D:\Kuliah\SEMESTER 4\PRAKTIKUM KPL\KPL_Martryatus Sofia_2311104003_SE0701\09_API Design dan Construction Using Swagger> cd .\jurnalmodul9_2311104003\  
PS D:\Kuliah\SEMESTER 4\PRAKTIKUM KPL\KPL_Martryatus Sofia_2311104003_SE0701\09_API Design dan Construction Using Swagger\jurnalmodul9_2311104003> node app.js  
Server berjalan di http://localhost:5000  
Swagger UI: http://localhost:5000/api-docs
```

kode yang ini berisi data film yang kamu gunakan untuk API tadi.

Penjelasan simpel:

- Kamu membuat sebuah array movies yang isinya daftar 3 film terkenal.
- Setiap film berbentuk objek dengan properti:
 - title → judul film
 - director → nama sutradara
 - stars → daftar aktor/aktris utama
 - description → deskripsi singkat tentang film
- Lalu kamu export array movies supaya bisa dipakai di file lain (seperti server Express-mu tadi).

swagger.json



kode ini adalah file dokumentasi API dengan format OpenAPI 3.0 untuk Swagger UI.

Penjelasan singkat:

- Kamu mendokumentasikan API Movies buatanmu, supaya bisa tampil rapi di Swagger.
- File ini menjelaskan:
 - Nama API: API Movies
 - Server base URL: `http://localhost:5000`
 - Endpoint `/api/Movies`:
 - GET → ambil semua film.
 - POST → tambah film baru (dengan struktur data: title, director, stars, description).
 - Endpoint `/api/Movies/{id}`:
 - GET → ambil film berdasarkan id.
 - DELETE → hapus film berdasarkan id.
- Setiap endpoint dijelaskan juga response-nya, misal sukses (200, 201) atau error (404).