

1. models/movie.js

```
1 class Mahasiswa {
2   constructor(name, ntm, course, year) {
3     this.name = name;
4     this.ntm = ntm;
5     this.course = course;
6     this.year = year;
7   }
8 }
9
10 module.exports = Mahasiswa;
11
```

Penjelasan :

Kode tersebut mendefinisikan class `Mahasiswa` yang merepresentasikan objek mahasiswa dengan properti `name`, `nim`, `course`, dan `year`. Properti tersebut diatur melalui constructor saat objek dibuat. Class ini kemudian diekspor menggunakan `module.exports` agar bisa digunakan di file lain dalam proyek Node.js.

2. routes/movies.js

```
1 const express = require("express");
2 const router = express.Router();
3
4 const Movie = require("../models/movie");
5 let movies = require("../data");
6
7 // GET all movies
8 router.get("/", (req, res) => {
9   res.json(movies);
10 });
11
12 // GET movie by index
13 router.get("/:id", (req, res) => {
14   const id = parseInt(req.params.id);
15   console.log("GET /api/Movies/${id}");
16   console.log("Total movies: ${movies.length}");
17
18   if (isNaN(id) || id < 0 || id >= movies.length) {
19     console.log("Invalid index accessed");
20     return res.status(404).json({ error: "Not Found" });
21   }
22
23   console.log("Returning movie:", movies[id]);
24   res.json(movies[id]);
25 });
26
27
28
29 // POST: Add one or multiple movies
30 router.post("/", (req, res) => {
31   const data = req.body;
32   const added = [];
33
34   if (Array.isArray(data)) {
35     data.forEach(( { title, director, stars, description } ) => {
36       const movie = new Movie(title, director, stars, description);
37       movies.push(movie);
38
39       router.post("/", (req, res) => {
40         data.forEach(( { title, director, stars, description } ) => {
41           movies.push(movie);
42           added.push(movie);
43         });
44         res.status(201).json(added);
45       } else {
46         const { title, director, stars, description } = data;
47         const movie = new Movie(title, director, stars, description);
48         movies.push(movie);
49         res.status(201).json(movie);
50       }
51     });
52   }
53
54   // DELETE movie by index
55   router.delete("/:id", (req, res) => {
56     const id = parseInt(req.params.id);
57     if (id < 0 || id >= movies.length) return res.status(404).json({ error: "Not Found" });
58     movies.splice(id, 1);
59     res.sendStatus(204);
60   });
61
62   module.exports = router;
63 }
```

Penjelasan :

Kode tersebut merupakan router Express yang digunakan untuk mengelola data film dalam aplikasi Node.js. Data film diambil dari file `data.js` dan dibentuk menggunakan class `Movie`. Router ini menyediakan beberapa endpoint, yaitu `GET /` untuk menampilkan semua film, `GET /:id` untuk menampilkan film berdasarkan indeks, `POST /` untuk menambahkan satu atau beberapa film baru, dan `DELETE /:id` untuk menghapus film berdasarkan indeks. Semua data diproses dalam format JSON dan router ini diekspor agar dapat digunakan di file utama server.

3. data.js

```
1 const Movie = require("../models/movie");
2
3 let movies = [
4   new Movie(
5     "Doraemon: Nobita's New Dinosaur",
6     "Kazuaki Imai",
7     ["Nobita", "Doraemon"],
8     "Nobita finds twin dinosaurs and takes them to the Dinosaur Era with Doraemon's help."
9   ),
10  new Movie(
11    "Shaun the Sheep Movie",
12    "Mark Burton",
13    ["Shaun", "Bitzer"],
14    "Shaun takes a trip to the big city to rescue his farmer and gets into lots of trouble."
15  ),
16  new Movie(
17    "Masha and the Bear: Holiday on Ice",
18    "Oleg Kuzovkov",
19    ["Masha", "The Bear"],
20    "Masha brings chaos to a winter ice show while learning about friendship and fun."
21  ),
22 ];
23
24 module.exports = movies;
25
```

Penjelasan :

Kode tersebut membuat daftar tiga film dalam bentuk array menggunakan class `Movie`, lengkap dengan judul, sutradara, karakter, dan deskripsi. Data film ini disimpan dalam variabel `movies` dan diekspor agar bisa digunakan di file lain dalam aplikasi Node.js.

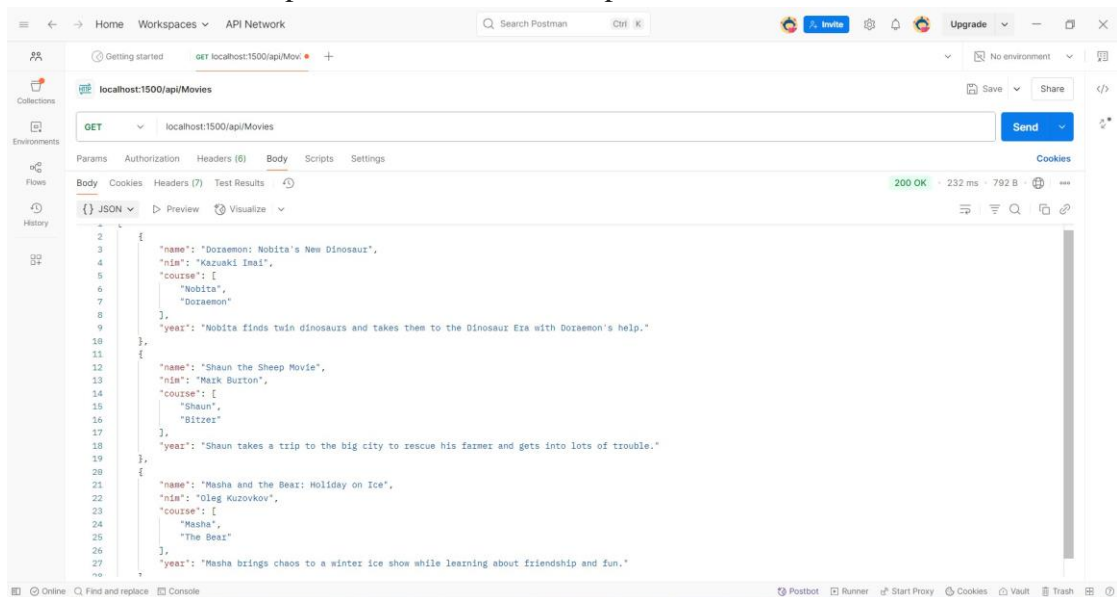
4. index.js

```
1 const express = require("express");
2 const app = express();
3 const movieRoutes = require("../routes/movies");
4
5 app.use(express.json());
6 app.use("/api/Movies", movieRoutes);
7
8 const PORT = 1500;
9 app.listen(PORT, () => {
10   console.log(`Server running at http://localhost:${PORT}/api/Movies`);
11 });
12
```

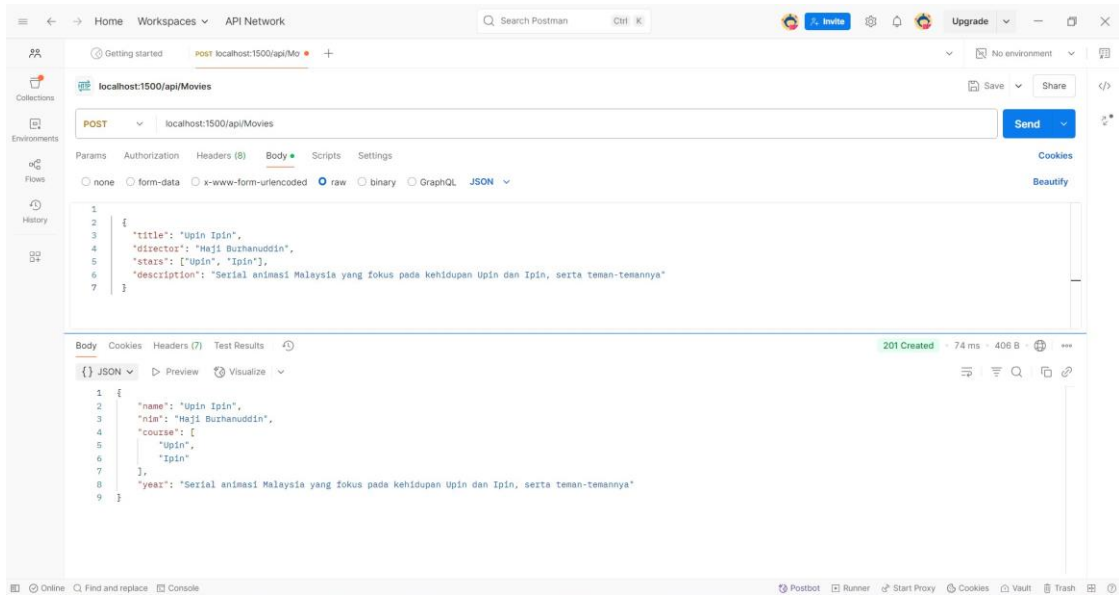
Penjelasan :

Kode tersebut membuat server Express dan menghubungkan rute film dari file `routes/movies`. Middleware `express.json()` digunakan untuk memproses data JSON dari request. Server berjalan pada port 1500 dan melayani endpoint utama di `/api/Movies`.

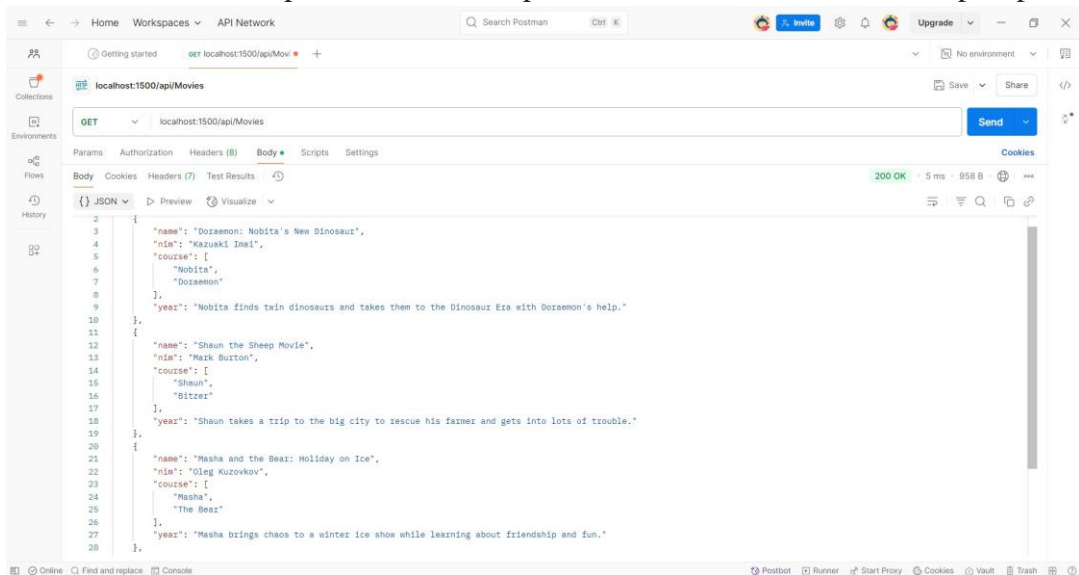
5. GET localhost:1500/api/Movies, untuk menampilkan 3 list Movies

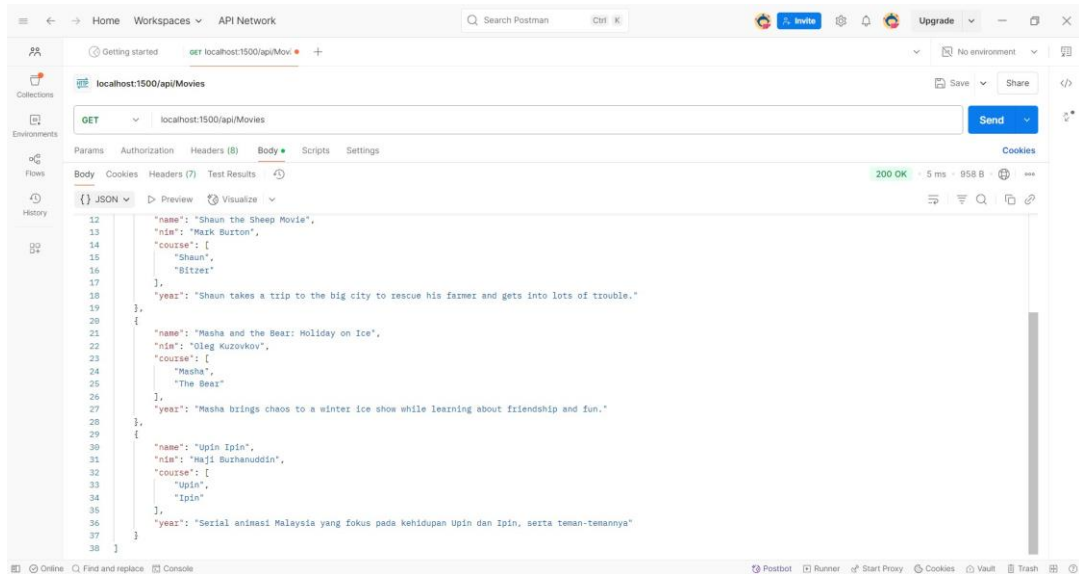


6. POST localhost:1500/api/Movies, untuk menambahkan list upin ipin

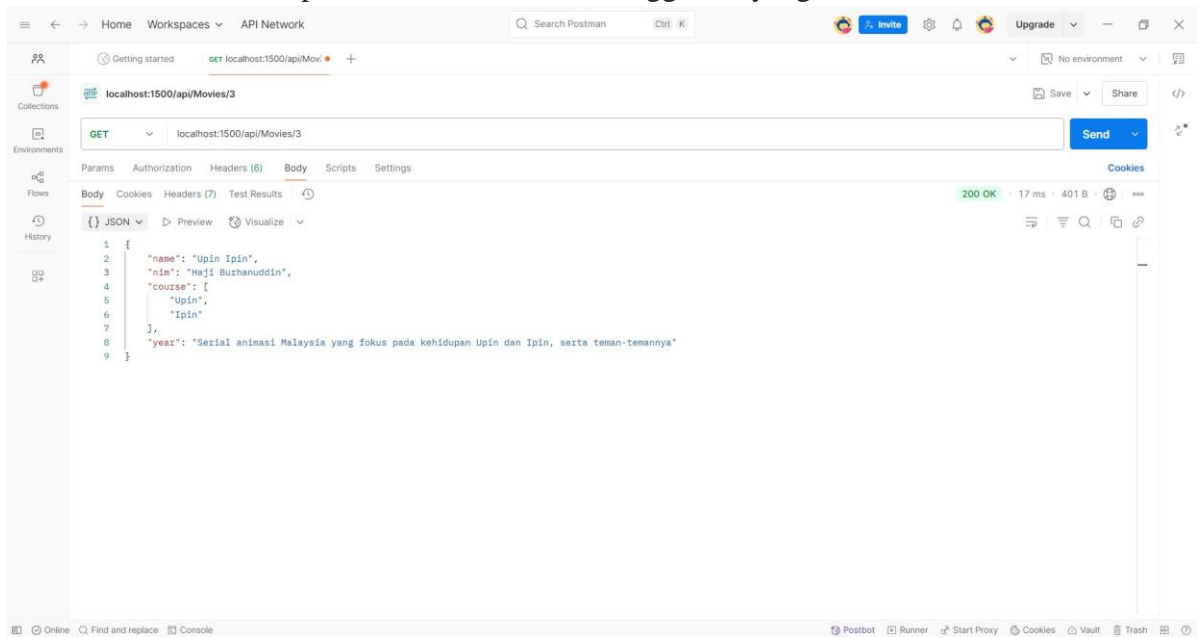


7. GET localhost:1500/api/Movies, menampilkan list dan ada tambahan film upin ipin

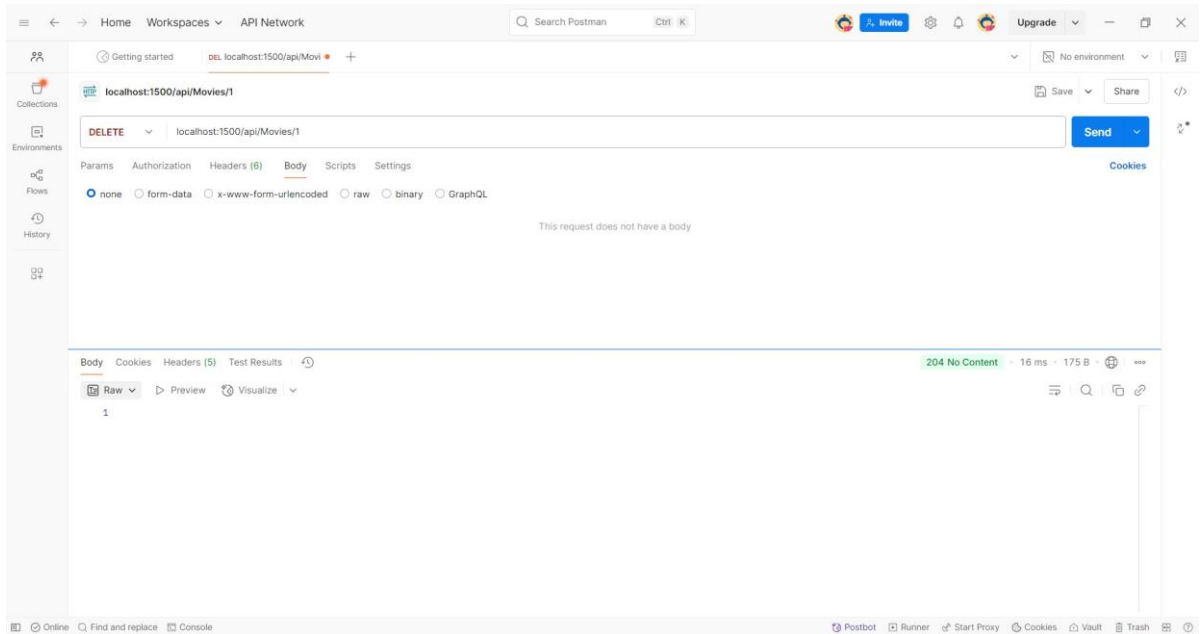




8. POST localhost:1500/api/Movies/3, untuk memanggil list yang baru ditambahkan



9. DELETE localhost:1500/api/Movies/1, untuk menghapus list dengan index 1



10. GET `localhost:1500/api/Movies`, memanggil list dan index 1 sudah tidak ada

