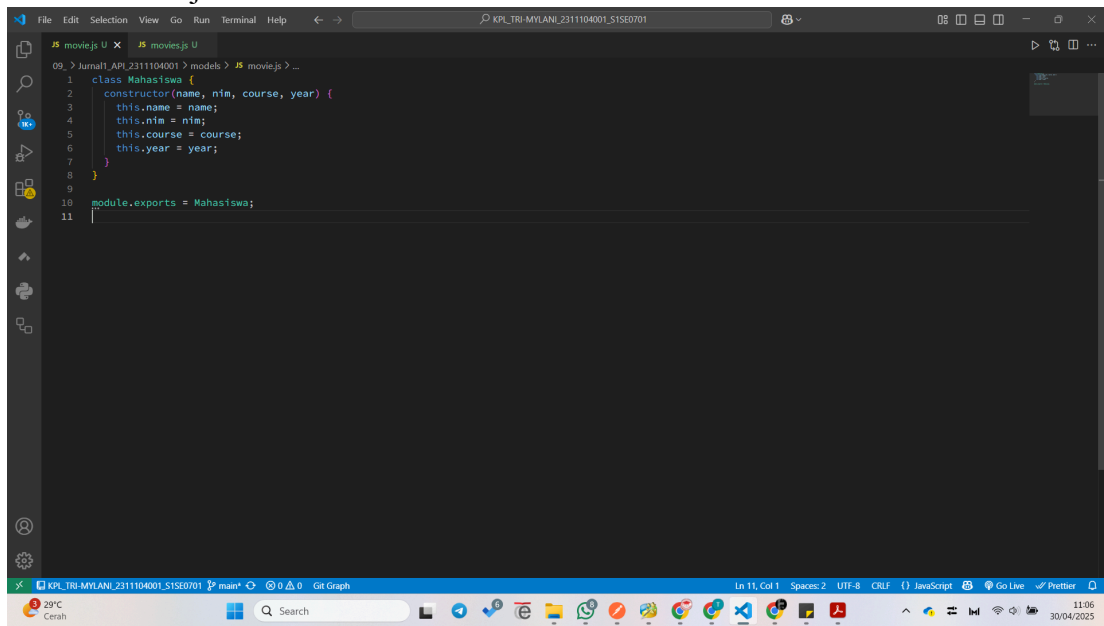


1. Membuat folder Jurnal2-API-2311104001
2. `npm init -y`
npm init digunakan untuk membuat file `package.json` — file konfigurasi utama dalam proyek Node.js yang menyimpan metadata proyek dan dependensi. `-y` (atau `--yes`) secara otomatis menjawab semua pertanyaan dengan nilai default, sehingga prosesnya cepat dan tanpa interaksi manual.
3. `npm install express`
Perintah `npm install express` digunakan untuk menginstal Express.js, yaitu framework web minimalis dan fleksibel untuk Node.js. Fungsi Utama express yakni untuk memudahkan pembuatan API atau aplikasi web/server dengan sintaks yang lebih sederhana dibandingkan menggunakan Node.js murni.
4. `models/movie.js`

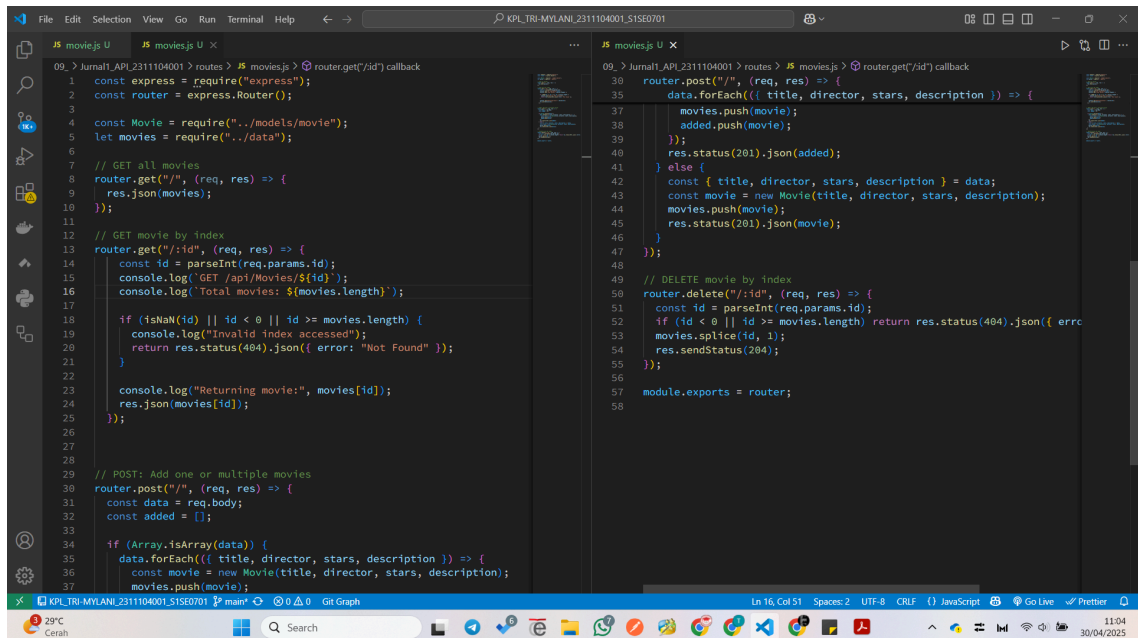


```
1 class Mahasiswa {
2   constructor(name, nim, course, year) {
3     this.name = name;
4     this.nim = nim;
5     this.course = course;
6     this.year = year;
7   }
8 }
9
10 module.exports = Mahasiswa;
```

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.

5. `routes/movies.js`



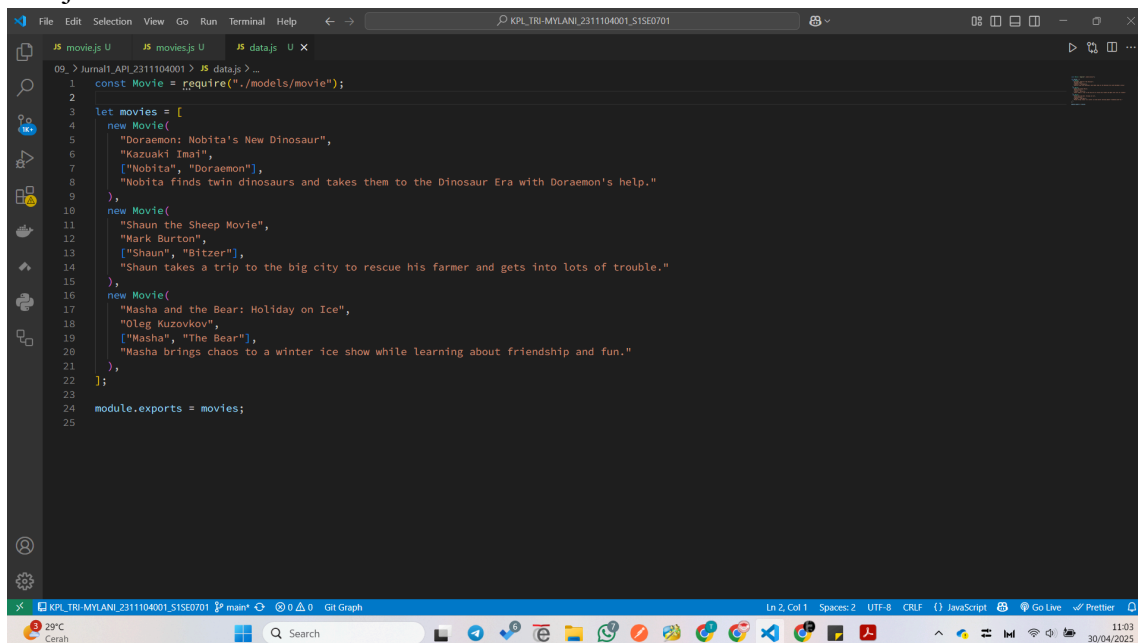
```
09. > Jurnal_API_2311104001 > routes > JS movies.js > router.get('/:id') callback
1  const express = require("express");
2  const router = express.Router();
3
4  const Movie = require("../models/movie");
5  let movies = require("../data");
6
7  // GET all movies
8  router.get("/", (req, res) => {
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   } else {
40     const { title, director, stars, description } = data;
41     const movie = new Movie(title, director, stars, description);
42     movies.push(movie);
43     res.status(201).json(movie);
44   }
45
46   // DELETE movie by index
47   router.delete("/:id", (req, res) => {
48     const id = parseInt(req.params.id);
49     if (id < 0 || id >= movies.length) return res.status(404).json({ error: "Not Found" });
50     movies.splice(id, 1);
51     res.sendStatus(204);
52   });
53
54   module.exports = router;
55 }
56
57
58
```

```
09. > Jurnal_API_2311104001 > JS 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 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.

6. data.js

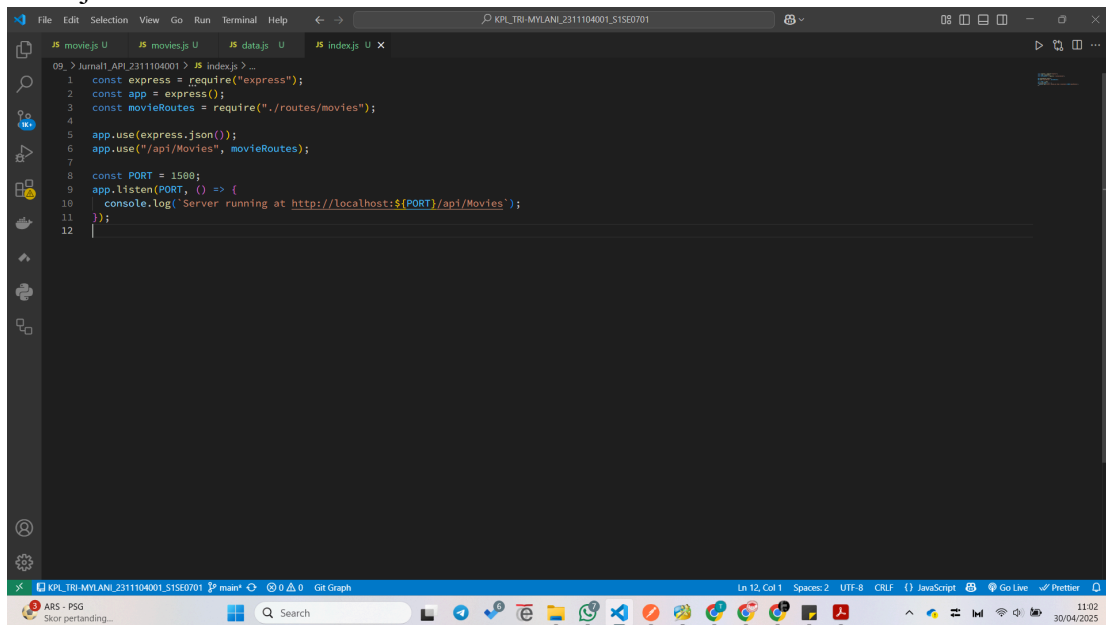


```
09. > Jurnal_API_2311104001 > JS data.js
1  const Movie = require("../models/movie");
2
3  let movies = [
4    new Movie(
5      "Doraemon: Nobita's New Dinosaur",
6      "Kazuaki Imai",
7      ["Nobita", "Doraemon"],
8      "Nobita finds twin dinosaurs and takes them to the Dinosaur Era with Doraemon's 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.

7. index.js

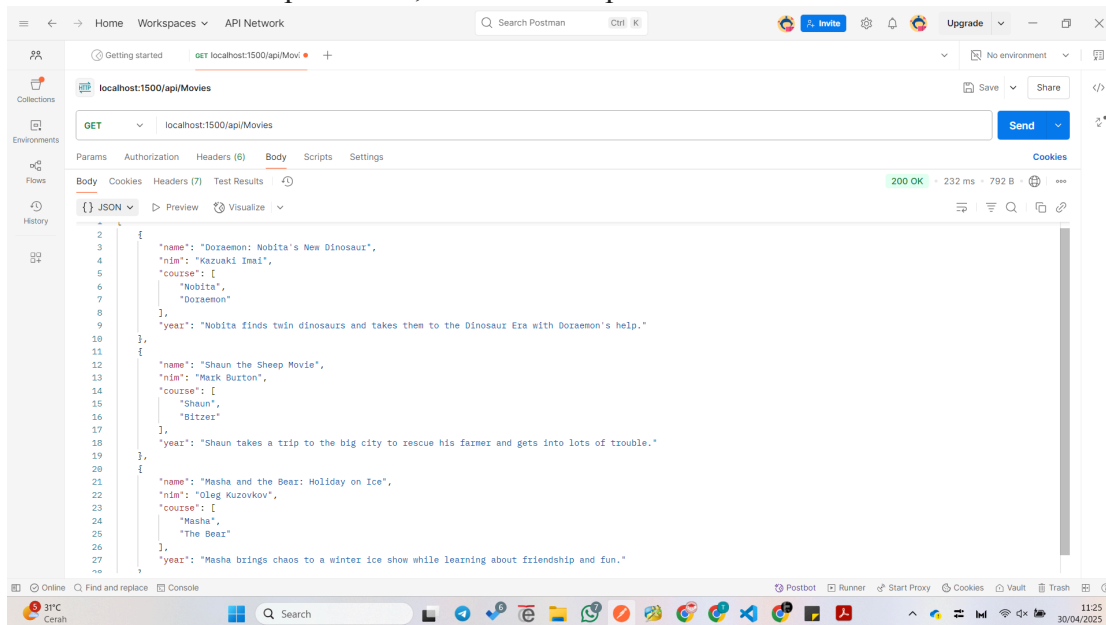


```
09. > Jurnal_API_2311104001 > JS index.js > ...
1  const express = require("express");
2  const app = express();
3  const movieRoutes = require("./routes/movies");
4
5  app.use(express.json());
6  app.use("/api/Movies", movieRoutes);
7
8  const PORT = 1500;
9  app.listen(PORT, () => {
10    console.log(`Server running at http://localhost:${PORT}/api/Movies`);
11  });
12
```

Penjelasan :

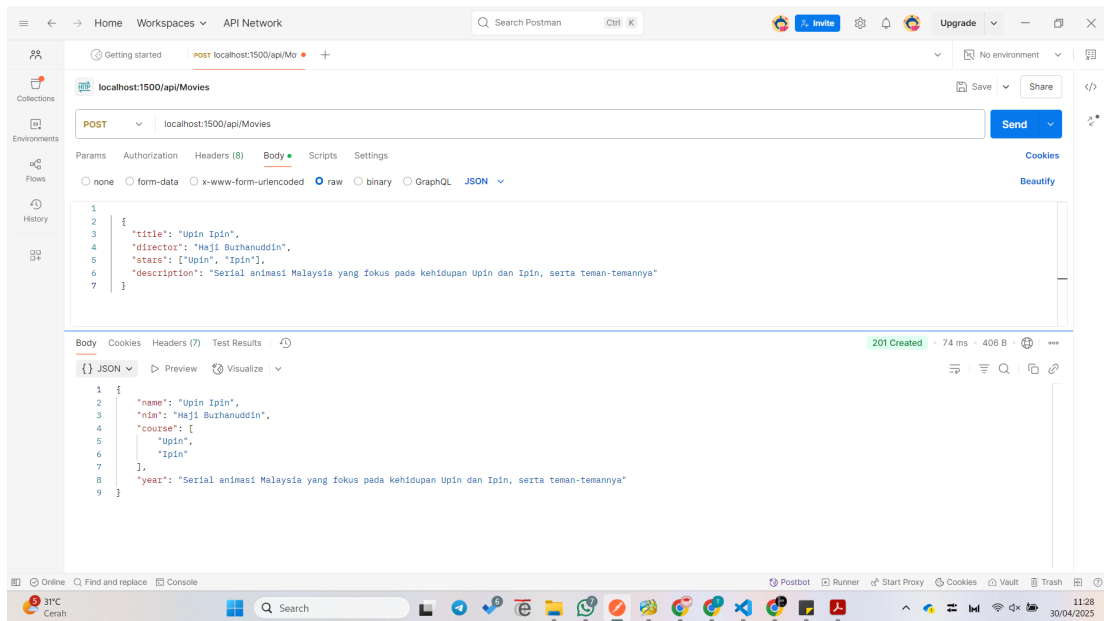
Kode tersebut membuat server Express dan menghubungkan rute film dari file `routes/movies`. Middleware `express.json()` digunakan untuk memproses data JSON dari request. Server berjalan pada port 1500 dan melayani endpoint utama di `/api/Movies`.

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

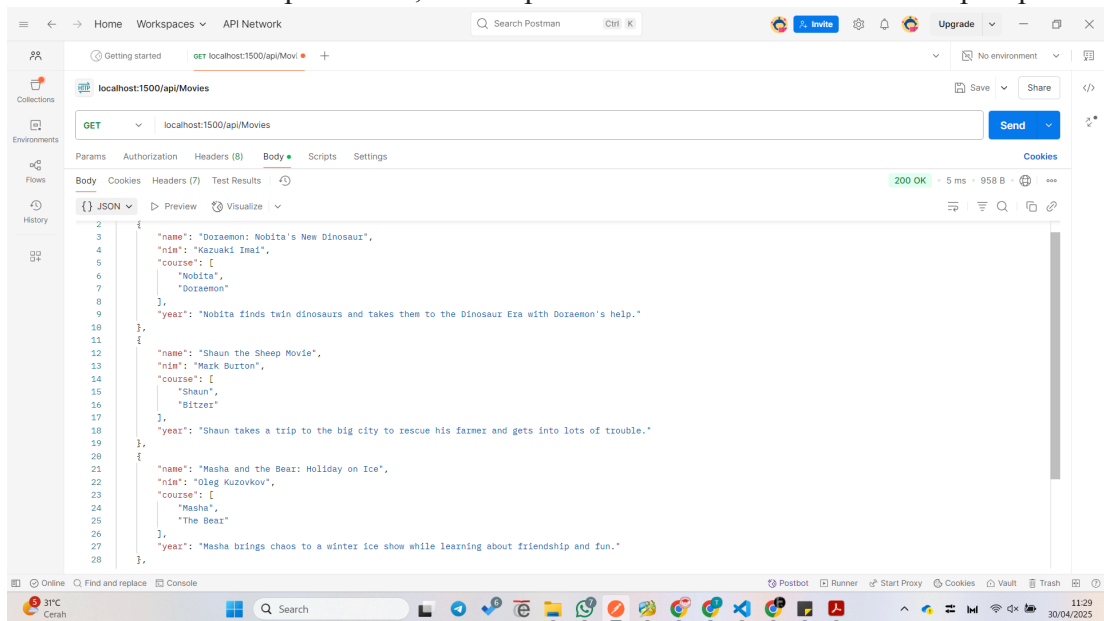


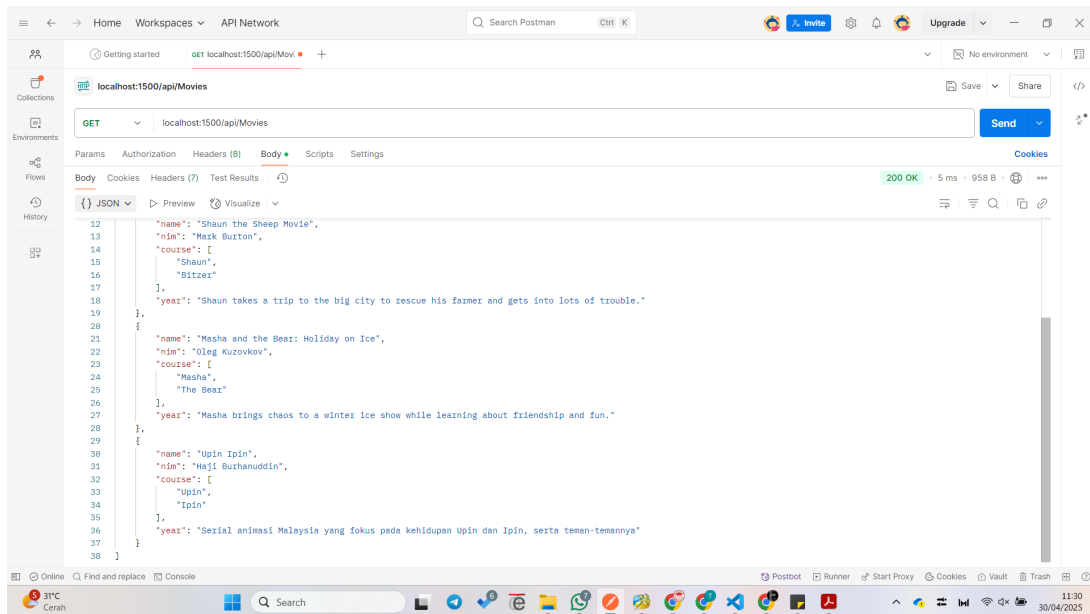
```
1  {
2    "name": "Doraemon: Nobita's New Dinosaur",
3    "nim": "Kazusaki Imai",
4    "course": [
5      "Nobita",
6      "Doraemon"
7    ],
8    "year": "Nobita finds twin dinosaurs and takes them to the Dinosaur Era with Doraemon's help."
9  },
10 },
11 {
12   "name": "Shaun the Sheep Movie",
13   "nim": "Mark Burton",
14   "course": [
15     "Shaun",
16     "Bitzer"
17   ],
18   "year": "Shaun takes a trip to the big city to rescue his farmer and gets into lots of trouble."
19 },
20 {
21   "name": "Masha and the Bear: Holiday on Ice",
22   "nim": "Oleg Kuzovkov",
23   "course": [
24     "Masha",
25     "The Bear"
26   ],
27   "year": "Masha brings chaos to a winter ice show while learning about friendship and fun."
28 },
29 ]
```

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

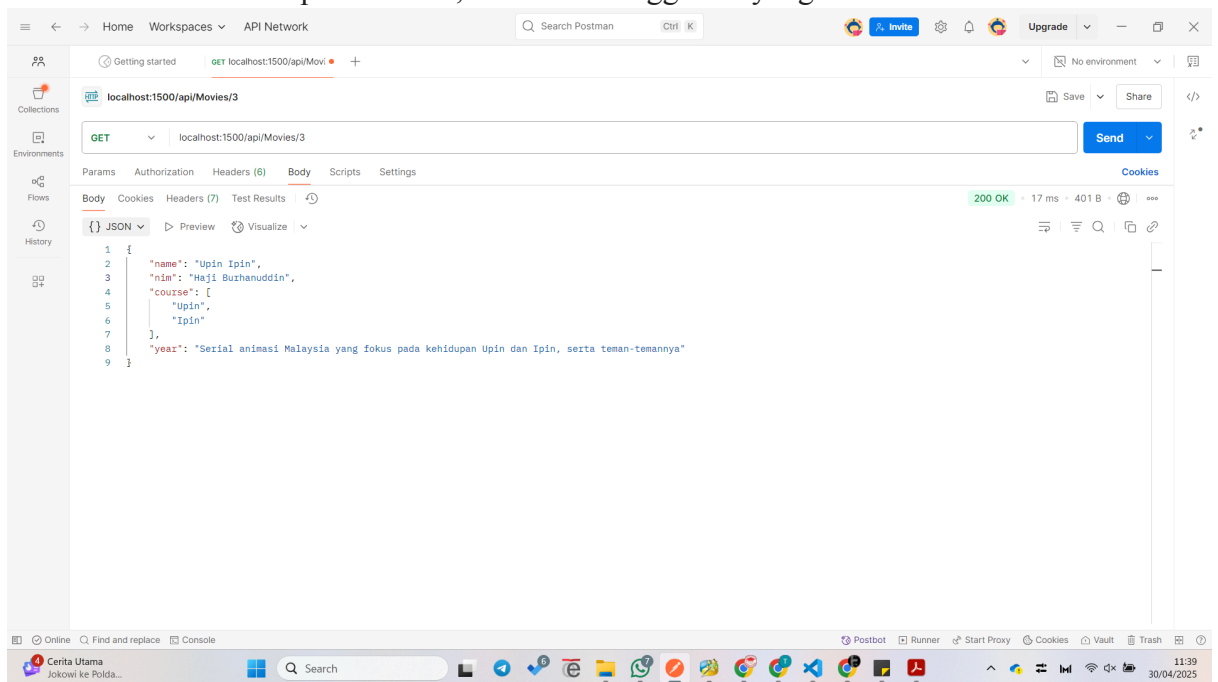


10. GET localhost:1500/api/Movies, menampilkan list dan ada tambahan film upin ipin

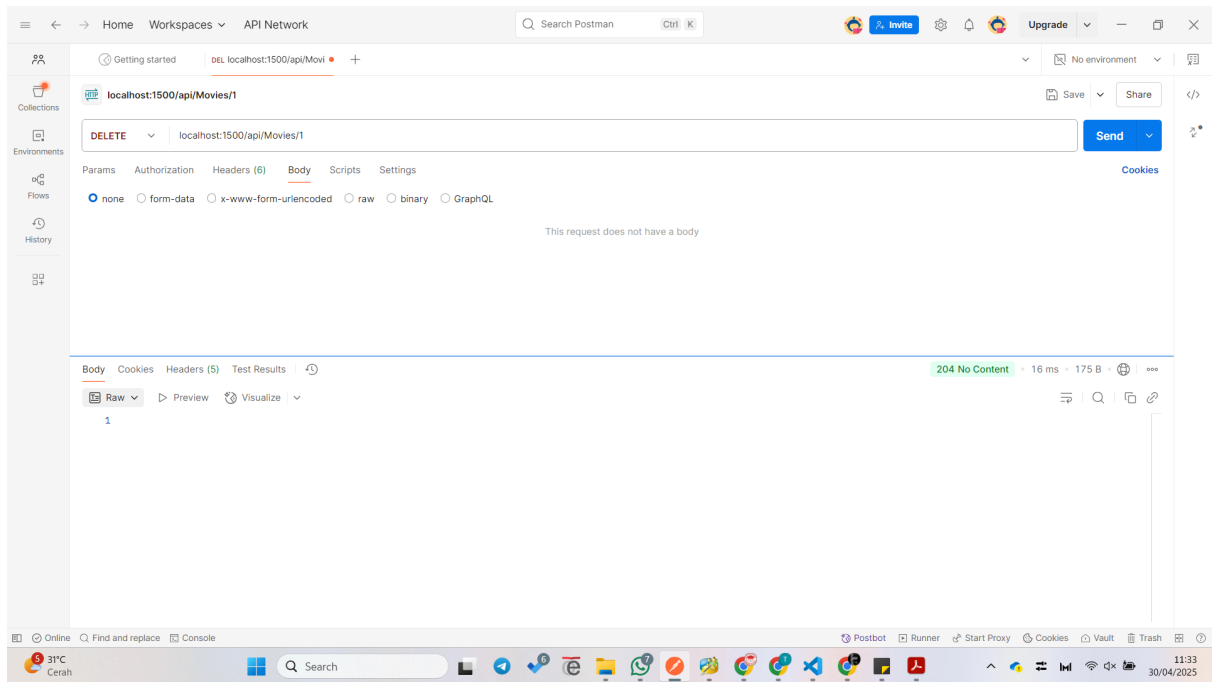




11. POST localhost:1500/api/Movies/3, untuk memanggil list yang baru ditambahkan



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



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

