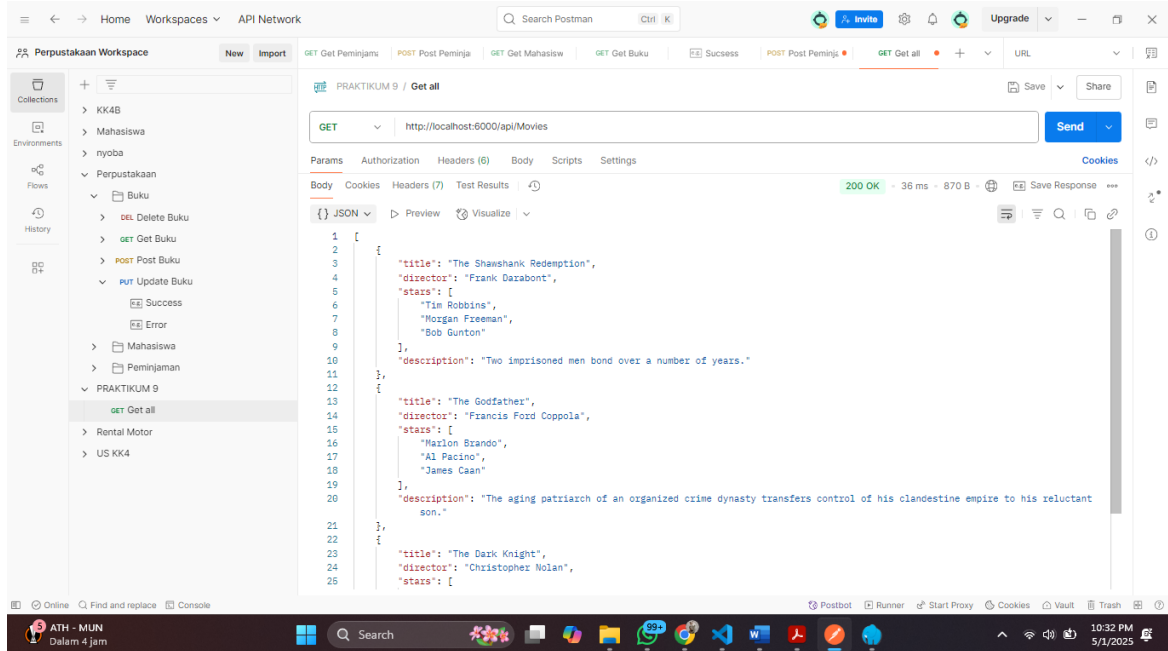


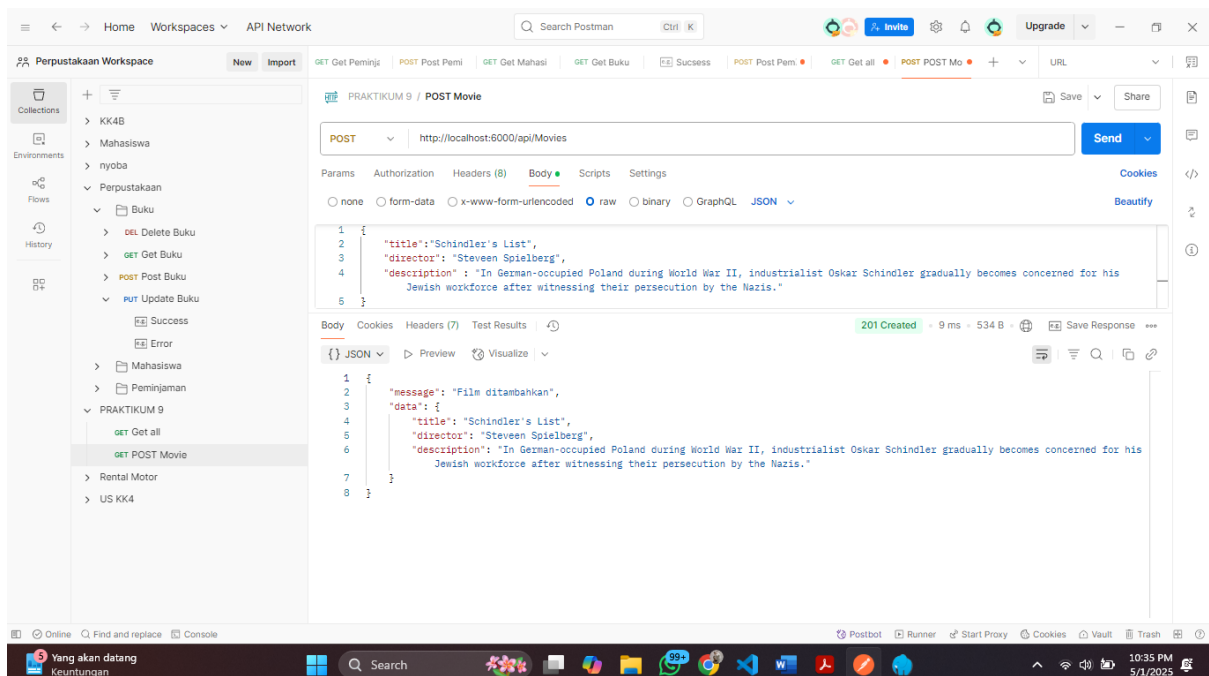
Novita Syahwa Tri Hapsari

SE07-01/ 2311104007

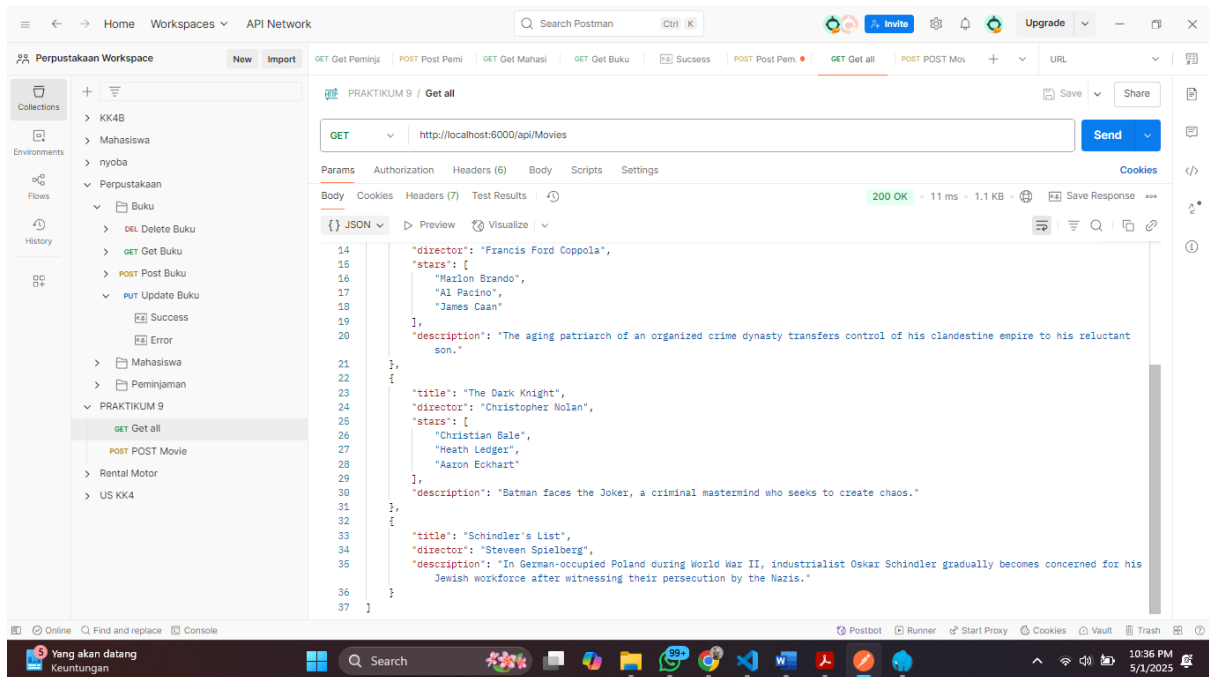
Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



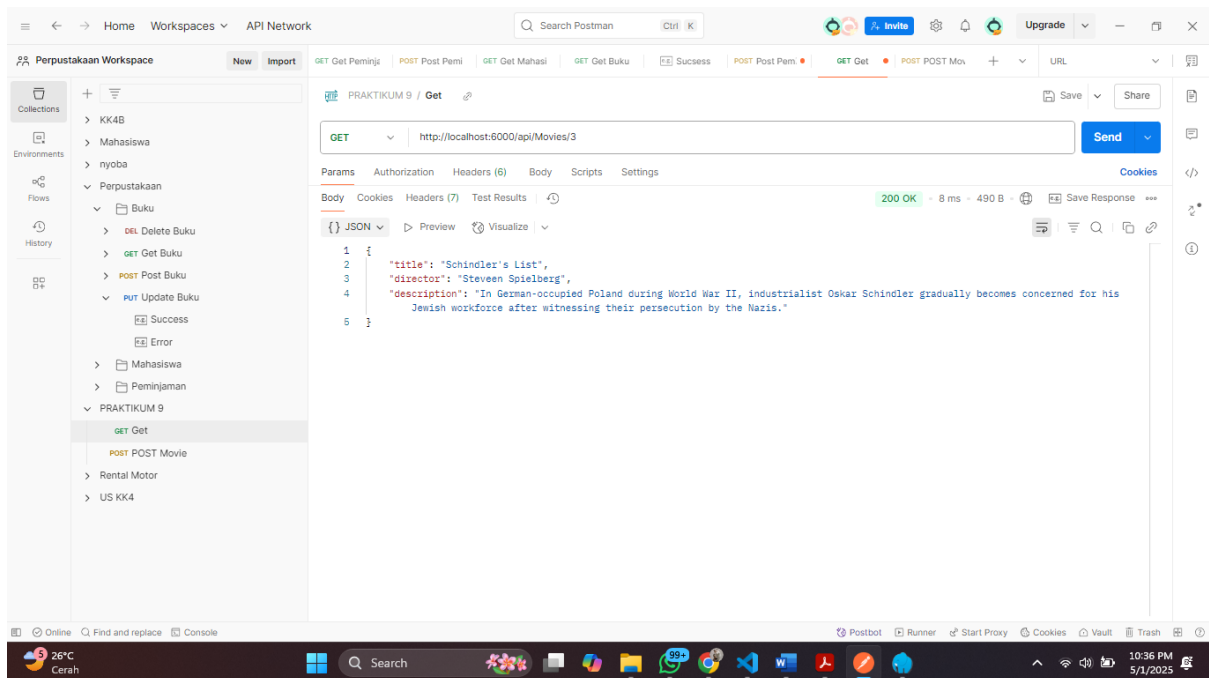
Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



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



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



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

The screenshot shows the Postman interface with a DELETE request to `http://localhost:6000/api/Movies/1`. The response is a 200 OK status with a JSON body:

```
{
  "message": "Film pada index 1 berhasil dihapus",
  "data": {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman",
      "Bob Gunton"
    ],
    "description": "Two imprisoned men bond over a number of years."
  }
}
```

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

The screenshot shows the Postman interface with a GET request to `http://localhost:6000/api/Movies/`. The response is a 200 OK status with a JSON array of movie objects:

```
[
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger",
      "Aaron Eckhart"
    ],
    "description": "Batman faces the Joker, a criminal mastermind who seeks to create chaos."
  },
  {
    "title": "Schindler's List",
    "director": "Steven Spielberg",
    "description": "In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his"
  }
]
```

Code:

```
const express = require('express');
const app = express();
const PORT = 6000;
app.use(express.json());
```

```

let movieList = [
  {
    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
of his clandestine empire to his reluctant son."
  },
  {
    title: "The Dark Knight",
    director: "Christopher Nolan",
    stars: ["Christian Bale", "Heath Ledger", "Aaron Eckhart"],
    description: "Batman faces the Joker, a criminal mastermind who seeks to create
chaos."
  }
];

app.get('/api/Movies', (req, res) => {
  res.json(movieList);
});

app.get('/api/Movies/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index < 0 || index >= movieList.length) {
    return res.status(404).json({ message: 'Film tidak ditemukan' });
  }
  res.json(movieList[index]);
});

app.post('/api/Movies', (req, res) => {
  const { title, director, stars, description } = req.body;
  movieList.push({ title, director, stars, description });
  res.status(201).json({ message: 'Film ditambahkan', data: { title, director, stars,
description } });
});

app.delete('/api/Movies/:index', (req, res) => {
  const index = parseInt(req.params.index);
  if (index < 0 || index >= movieList.length) {
    return res.status(404).json({ message: 'Index tidak ditemukan' });
  }
  const deleted = movieList.splice(index, 1);
  res.json({ message: `Movie pada index ${index} berhasil dihapus`, data: deleted[0]
});
});

app.listen(PORT, () => {
  console.log(`Server berjalan di http://localhost:${PORT}/api/Movies`);
});

```

```
{
  "name": "movie_api",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Penjelasan code:

Code tersebut menggunakan Express.js yang berjalan di port 6000. Program ini membuat REST API untuk mengelola daftar film. Terdapat beberapa fitur: menampilkan semua film (GET /api/Movies), menampilkan film berdasarkan urutan/index (GET /api/Movies/:index), menambahkan film baru (POST /api/Movies), dan menghapus film berdasarkan index (DELETE /api/Movies/:index). Data film disimpan dalam array bernama movieList. Program juga menangani error jika index yang diminta tidak valid, dan menggunakan express.json() agar bisa membaca data JSON dari request. Setelah dijalankan, server akan aktif di <http://localhost:6000/api/Movies>.