


Js

```
09_API_Swagger > TP_09_2311104022 >  mahasiswaApp.js > ...
1  const express = require('express');
2  const bodyParser = require('body-parser');
3
4  const app = express();
5  const port = 5001; // Bebas sesuai kebutuhan
6
7  // Data mahasiswa statis (static variable), urutan pertama adalah nama
8  let mahasiswaList = [
9    { nama: "Fathir Afa", nim: "1302025001" },
10   { nama: "Stephen Curry", nim: "1302000002" },
11   { nama: "LeBron James", nim: "1302000003" }
12 ];
13
14 app.use(bodyParser.json());
15
16 // GET seluruh mahasiswa
17 app.get('/api/mahasiswa', (req, res) => {
18   res.json(mahasiswaList);
19 });
20
21 // GET mahasiswa by index
22 app.get('/api/mahasiswa/:index', (req, res) => {
23   const idx = parseInt(req.params.index, 10);
24   if (idx >= 0 && idx < mahasiswaList.length) {
25     res.json(mahasiswaList[idx]);
26   } else {
27     res.status(404).json({ message: "Mahasiswa tidak ditemukan" });
28   }
29 });
```

```

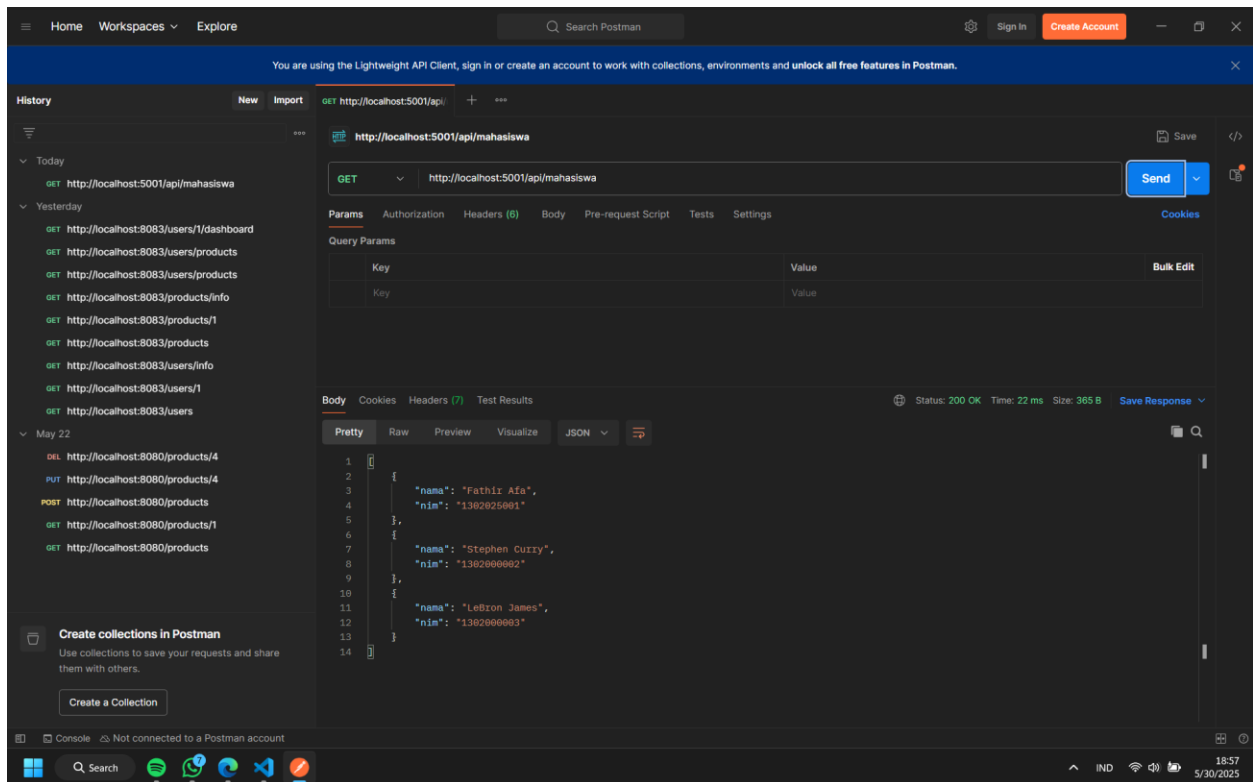
30
31 // POST tambah mahasiswa
32 app.post('/api/mahasiswa', (req, res) => {
33   const { nama, nim } = req.body;
34   const mahasiswaBaru = { nama, nim };
35   mahasiswaList.push(mahasiswaBaru);
36   res.status(201).json(mahasiswaBaru);
37 });
38
39 // DELETE mahasiswa by index
40 app.delete('/api/mahasiswa/:index', (req, res) => {
41   const idx = parseInt(req.params.index, 10);
42   if (idx >= 0 && idx < mahasiswaList.length) {
43     const terhapus = mahasiswaList.splice(idx, 1);
44     res.json(terhapus[0]);
45   } else {
46     res.status(404).json({ message: "Mahasiswa tidak ditemukan" });
47   }
48 });
49
50 app.listen(port, () => {
51   console.log(`Server berjalan di http://localhost:${port}/api/mahasiswa`);
52 });

```

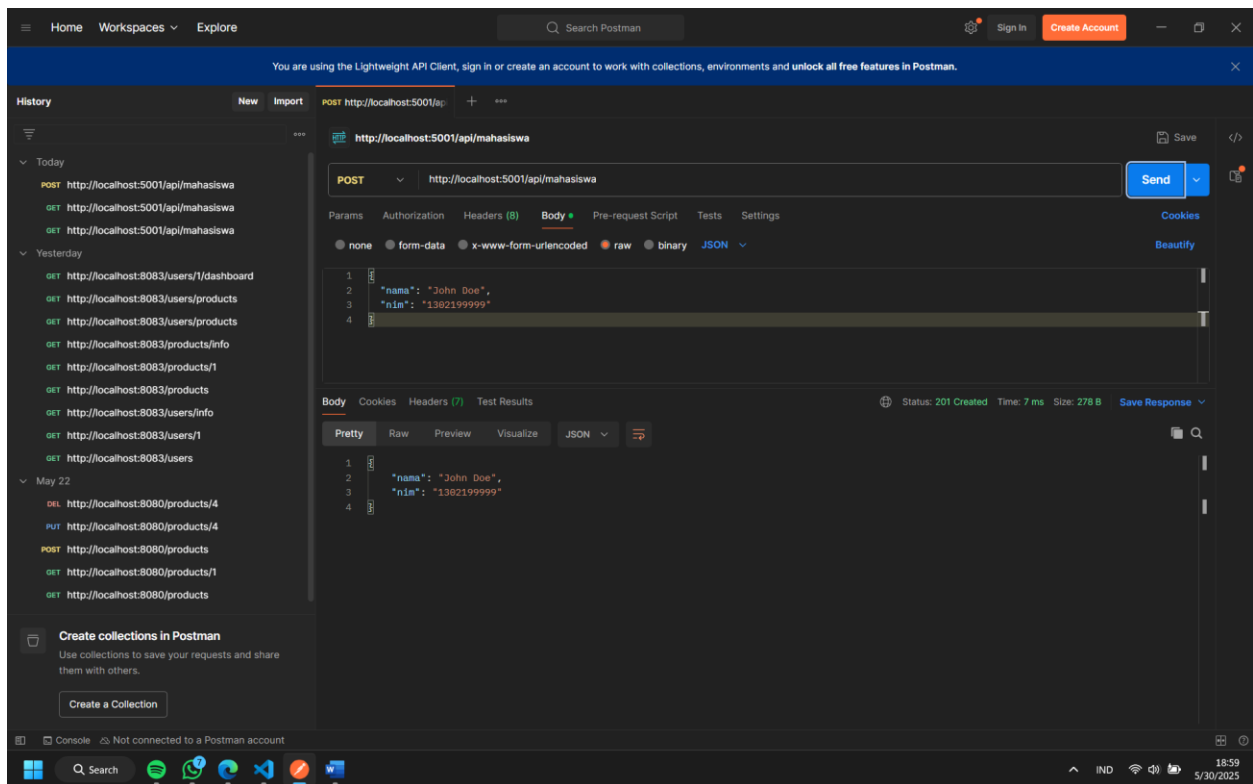
Penjelasan:

File mahasiswaApp.js adalah aplikasi backend sederhana berbasis Node.js menggunakan framework Express yang berfungsi sebagai REST API untuk pengelolaan data mahasiswa tanpa database, di mana data mahasiswa hanya tersimpan di memori dalam bentuk array statis. File ini menyediakan endpoint untuk menampilkan seluruh data mahasiswa, menampilkan data mahasiswa berdasarkan index, menambahkan mahasiswa baru, serta menghapus mahasiswa berdasarkan index, sesuai permintaan HTTP GET, POST, dan DELETE di jalur /api/mahasiswa. Dengan arsitektur ini, aplikasi mudah digunakan untuk keperluan praktikum pemrograman web atau pengujian API sederhana, dan setiap perubahan data hanya berlaku selama server dijalankan (tidak permanen setelah server dimatikan).

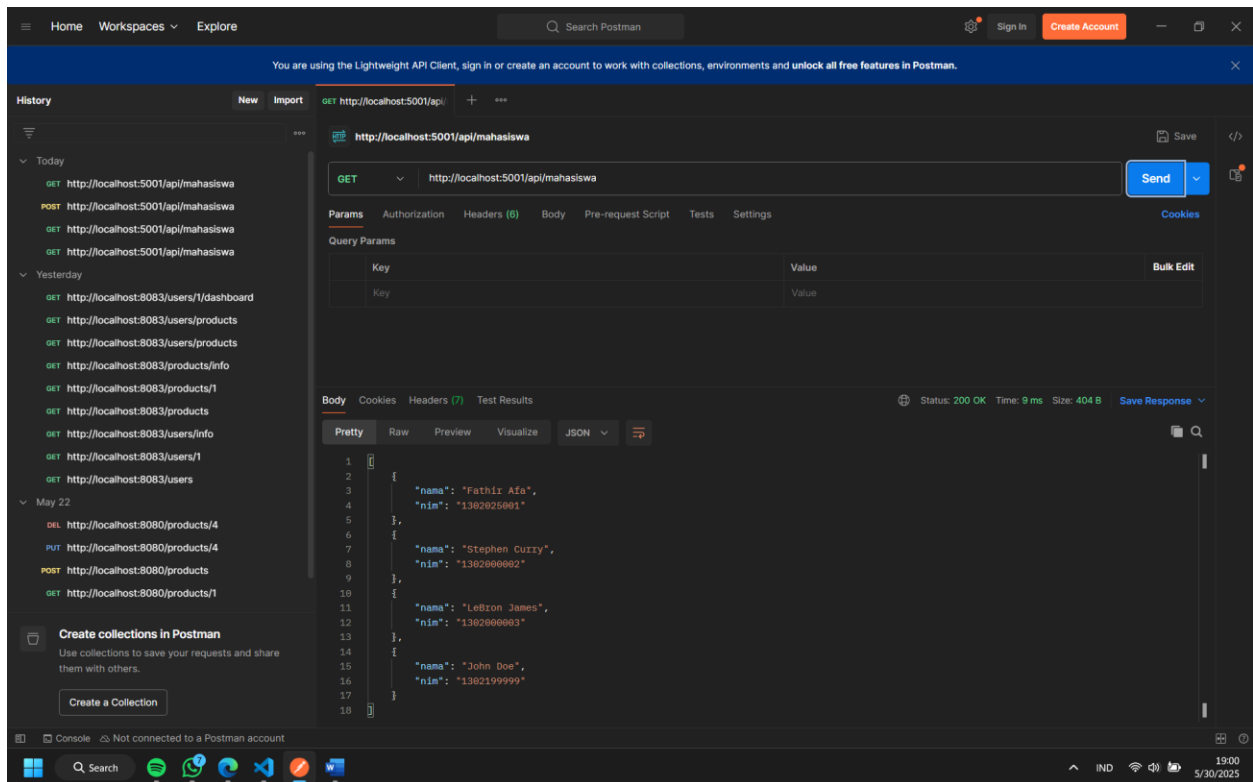
Get - all



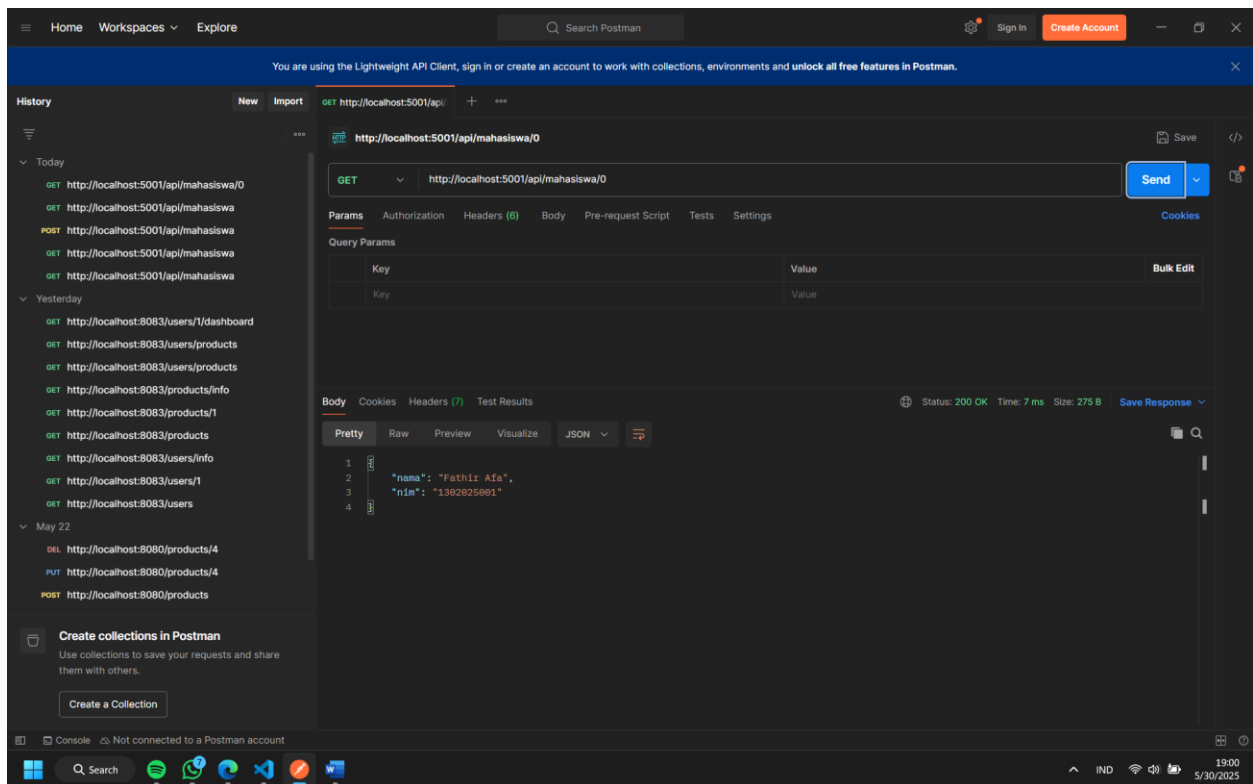
Post - new



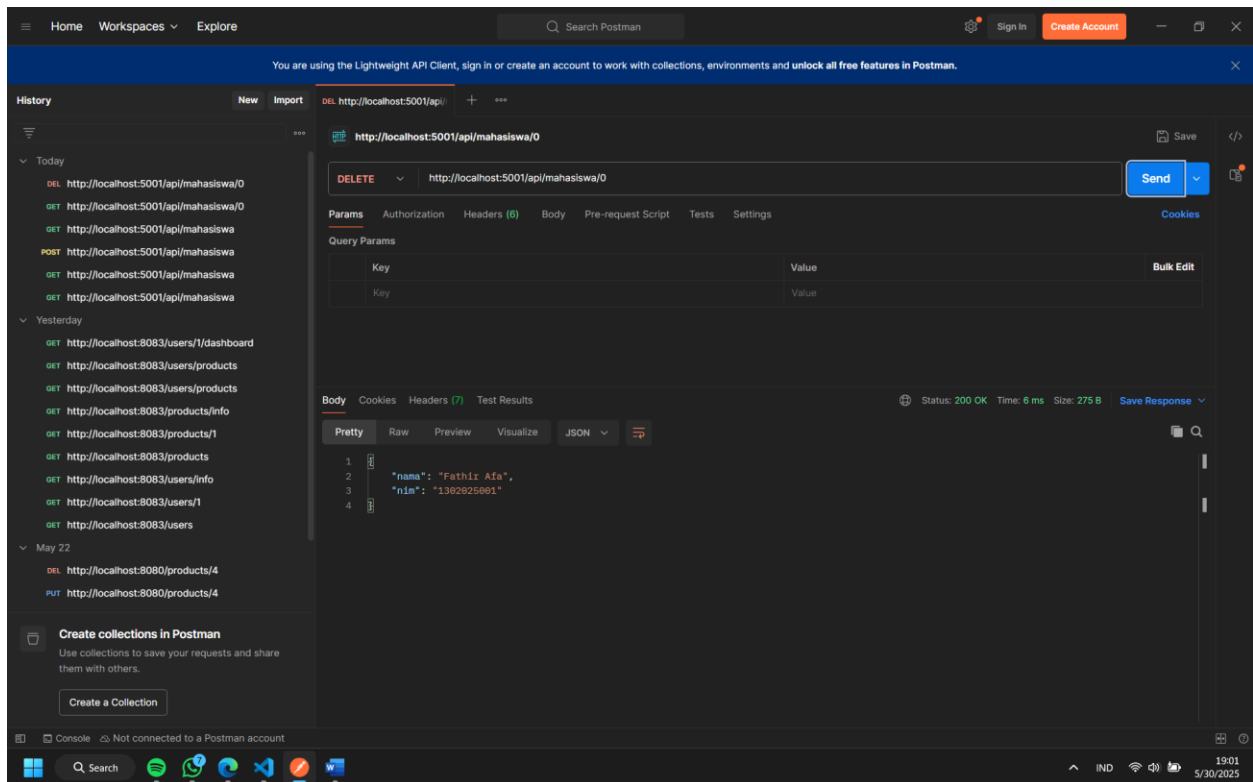
Get - all



Get – id



Delete – id



Get – all

