

1. models/mahasiswa.js

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

Penjelasan :

Kode tersebut mendefinisikan kelas Mahasiswa dalam JavaScript/Node.js yang digunakan untuk merepresentasikan data seorang mahasiswa.

2. routes/mahasiswa.js

```
1 const express = require("express");
2 const router = express.Router();
3
4 const Mahasiswa = require("../models/mahasiswa");
5
6 // Simulasi data mahasiswa
7 let mahasiswaList = [
8   new Mahasiswa("Budi", "120220001", "Math", "AI", 2022),
9   new Mahasiswa("Seri", "120220002", "SE", "WebProg", 2023),
10  new Mahasiswa("Dewi", "120220003", "DBMS", "Cloud", 2021),
11 ];
12
13 // GET all mahasiswa
14 router.get("/", (req, res) => {
15   res.json(mahasiswaList);
16 });
17
18 // GET mahasiswa by index
19 router.get("/:id", (req, res) => {
20   const id = parseInt(req.params.id);
21   console.log(`GET /api/mahasiswa/${id}`);
22   console.log(`Total mahasiswa: ${mahasiswaList.length}`);
23
24   if (isNaN(id) || id < 0 || id >= mahasiswaList.length) {
25     console.log("Invalid index accessed");
26     return res.status(404).json({ error: "Not Found" });
27   }
28
29   console.log(`Returning mahasiswa:`, mahasiswaList[id]);
30   res.json(mahasiswaList[id]);
31 });
32
33 // POST: Add one or multiple mahasiswa
34 router.post("/", (req, res) => {
35   const data = req.body;
36   const added = [];
37
38   if (Array.isArray(data)) {
39     data.forEach(( { name, nlm, course, year } ) => {
40       const mhs = new Mahasiswa(name, nlm, course, year);
41       mahasiswaList.push(mhs);
42       added.push(mhs);
43     });
44     res.status(201).json(added);
45   } else {
46     const { name, nlm, course, year } = data;
47     const mhs = new Mahasiswa(name, nlm, course, year);
48     mahasiswaList.push(mhs);
49     res.status(201).json(mhs);
50   }
51 });
52
53 // DELETE mahasiswa by index
54 router.delete("/:id", (req, res) => {
55   const id = parseInt(req.params.id);
56   if (isNaN(id) || id < 0 || id >= mahasiswaList.length) {
57     return res.status(404).json({ error: "Not Found" });
58   }
59   mahasiswaList.splice(id, 1);
60   res.sendStatus(204);
61 });
62
63 module.exports = router;
64
```

Penjelasan :

Kode tersebut adalah router Express untuk mengelola data mahasiswa secara

sederhana. Data mahasiswa disimpan dalam array dan dapat diakses melalui beberapa endpoint: `GET /` untuk melihat semua mahasiswa, `GET /:id` untuk melihat mahasiswa berdasarkan indeks, `POST /` untuk menambah satu atau beberapa mahasiswa, dan `DELETE /:id` untuk menghapus mahasiswa berdasarkan indeks. Semua data menggunakan class `Mahasiswa` dari file model, dan router ini diekspor agar bisa digunakan di file utama aplikasi.

3. data.js

```
1  const Mahasiswa = require("../models/mahasiswa");
2
3  let mahasiswaList = [
4    new Mahasiswa(
5      "Budi Santoso",
6      "120220001",
7      ["Math", "AI"],
8      2022
9    ),
10   new Mahasiswa(
11     "Seri Putri",
12     "120220002",
13     ["SE", "Web Programming"],
14     2023
15   ),
16   new Mahasiswa(
17     "Dewi Lestari",
18     "120220003",
19     ["DBMS", "Cloud Computing"],
20     2021
21   )
22 ];
23
24 module.exports = mahasiswaList;
25
```

Penjelasan :

Kode tersebut membuat daftar tiga mahasiswa menggunakan class `Mahasiswa`, lengkap dengan nama, NIM, mata kuliah, dan tahun masuk. Data ini disimpan dalam array `mahasiswaList` dan diekspor agar bisa digunakan di file lain dalam aplikasi Node.js.

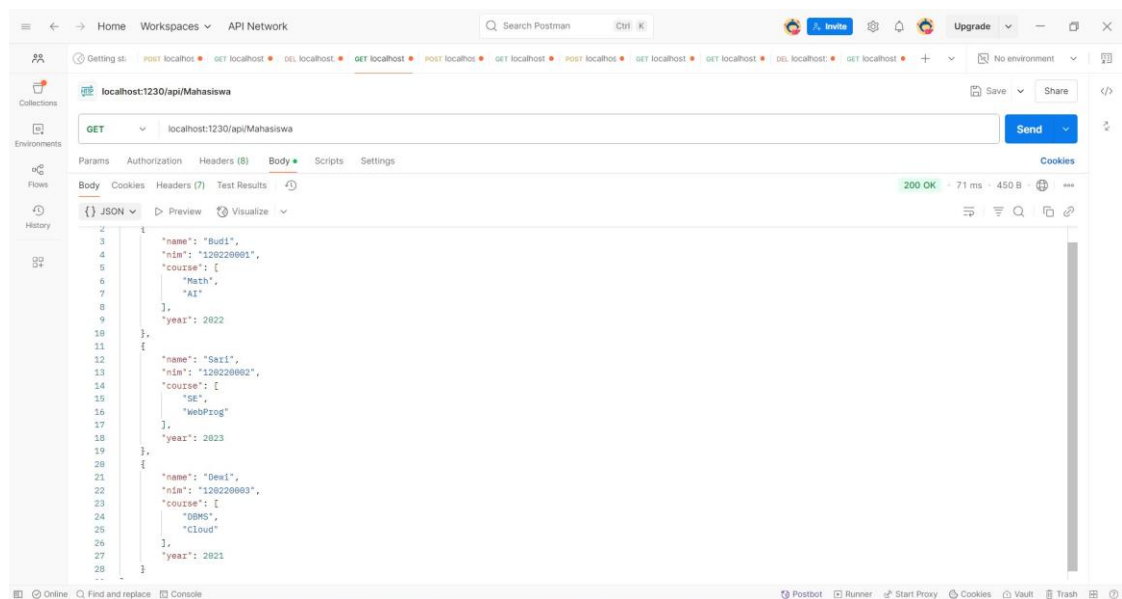
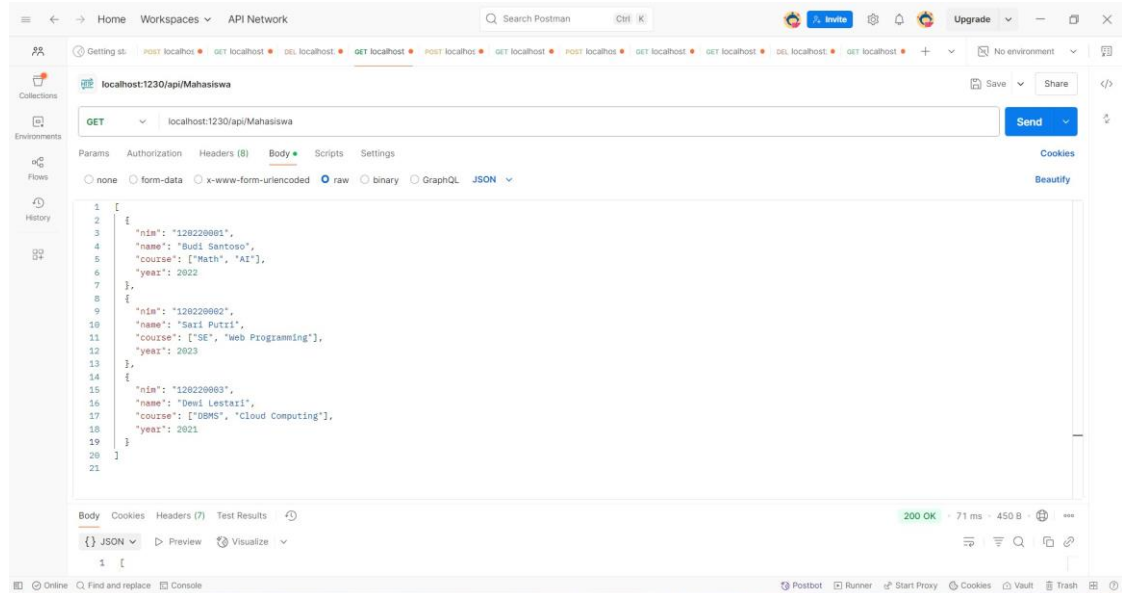
4. index.js

```
1  const express = require("express");
2  const app = express();
3  const mahasiswaRoutes = require("./routes/mahasiswa");
4
5  app.use(express.json());
6  app.use("/api/mahasiswa", mahasiswaRoutes);
7
8  const PORT = 1230;
9  app.listen(PORT, () => {
10    console.log(`Server running at http://localhost:${PORT}/api/mahasiswa`);
11  });
12
```

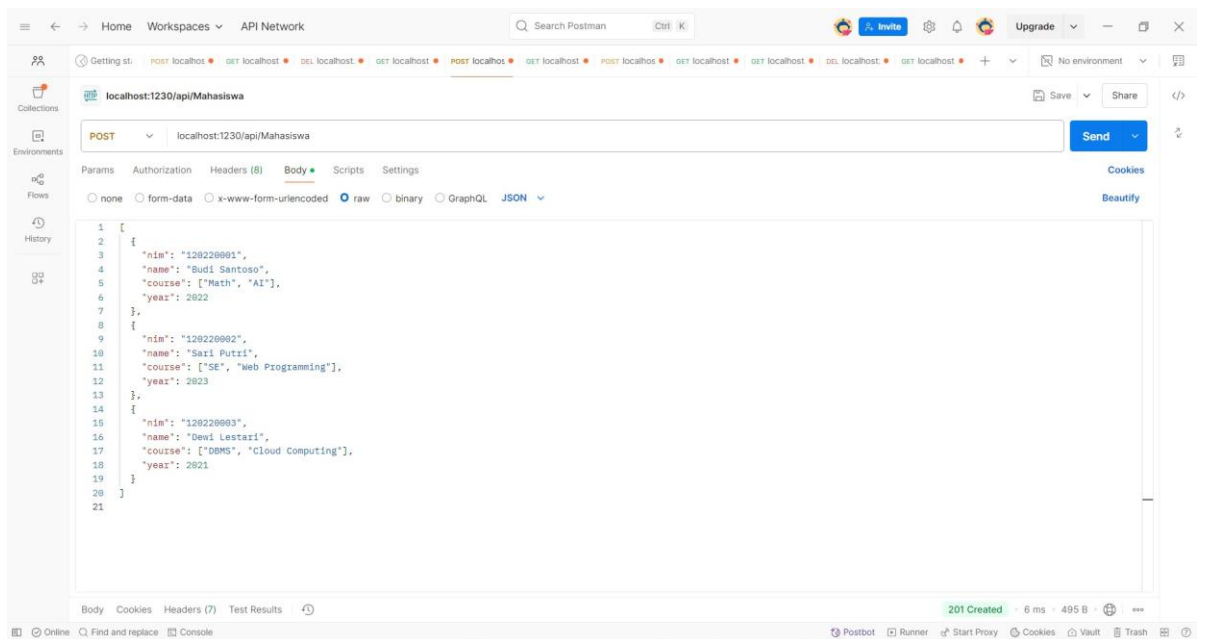
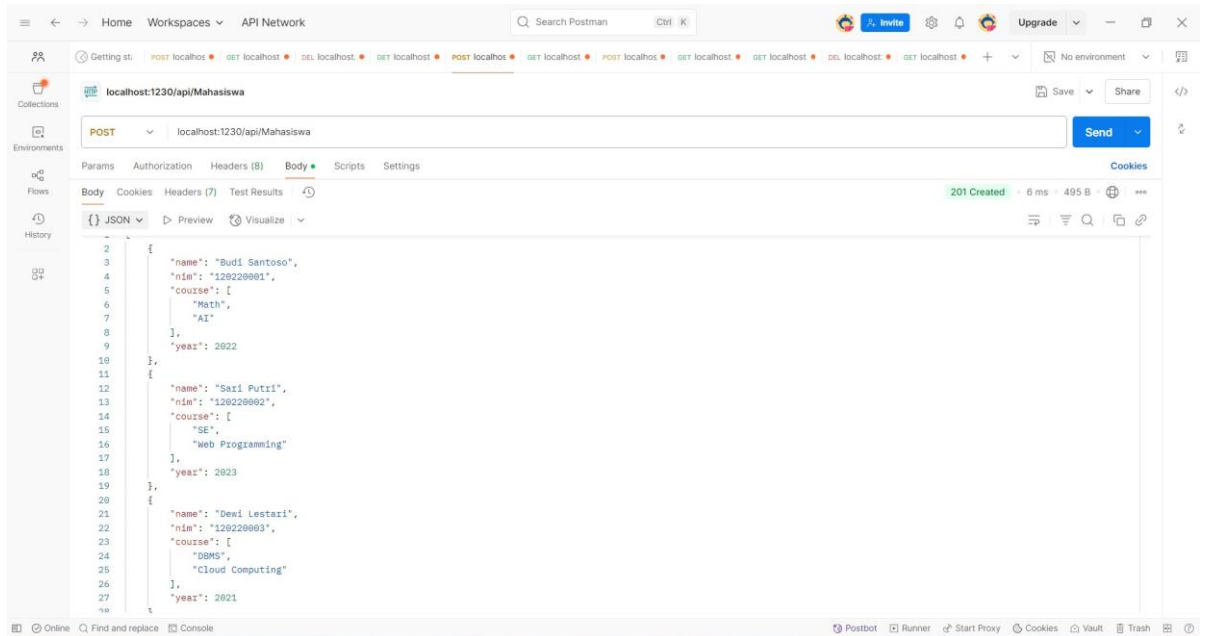
Penjelasan :

Kode tersebut membuat server Express dan menghubungkan rute mahasiswa dari file `routes/mahasiswa`. Middleware `express.json()` digunakan untuk membaca data JSON dari request. Server dijalankan pada port 1230 dan akan merespons permintaan ke endpoint `/api/mahasiswa`.

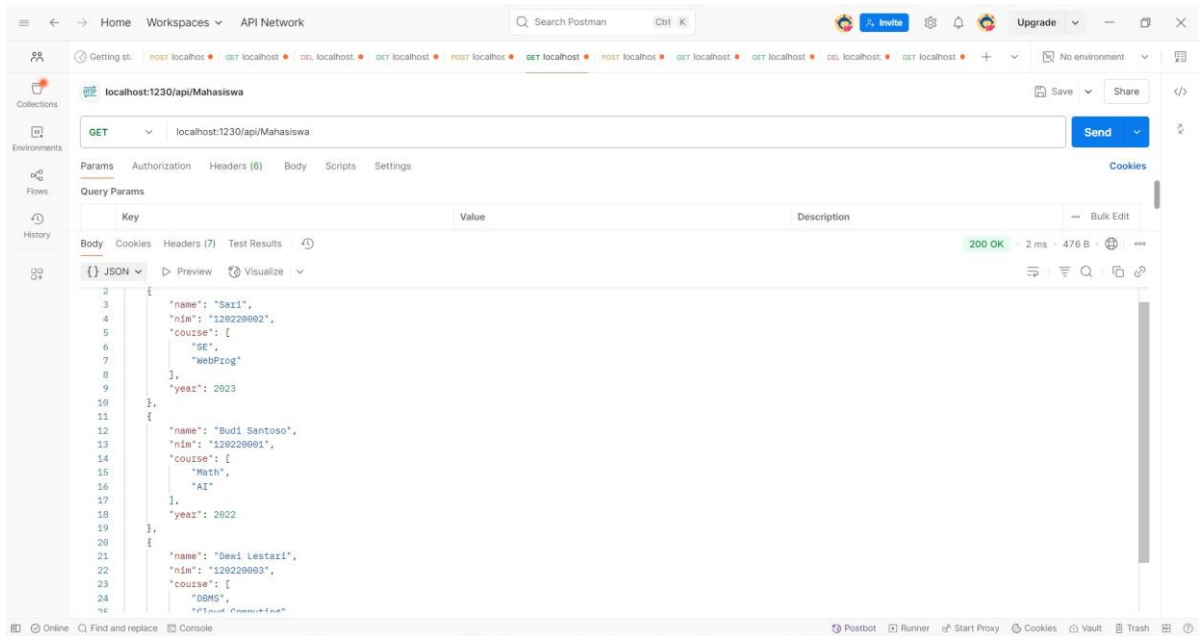
5. GET localhost:1230/api/Mahasiswa, untuk menampilkan 3 list



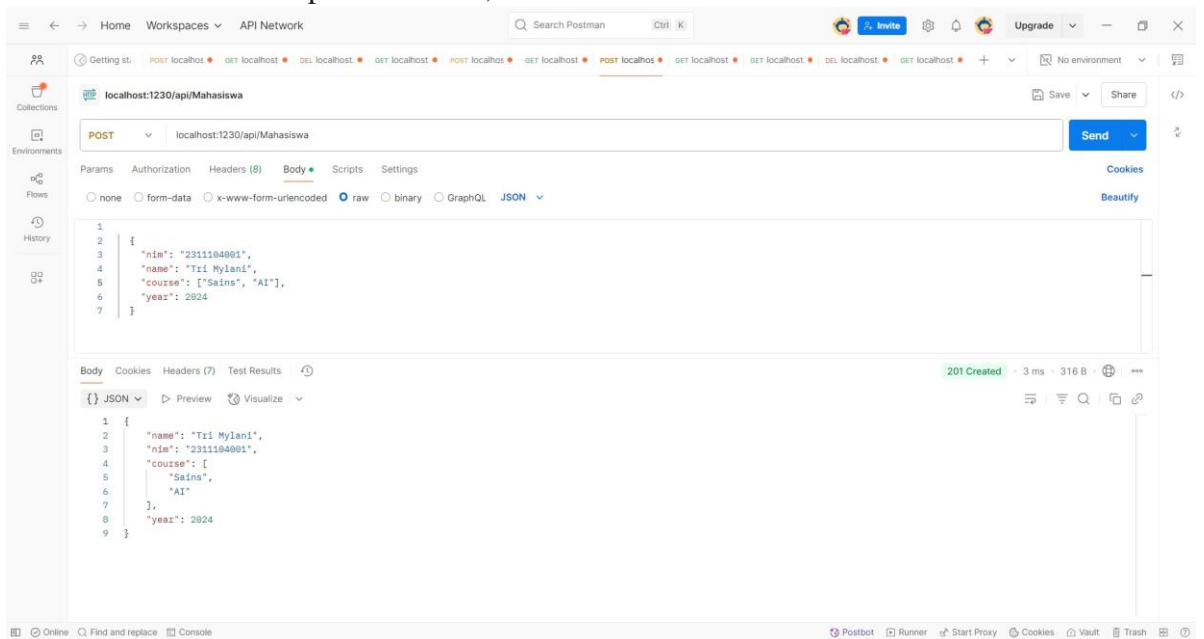
6. POST localhost:1230/api/Mahasiswa, untuk menambahkan list



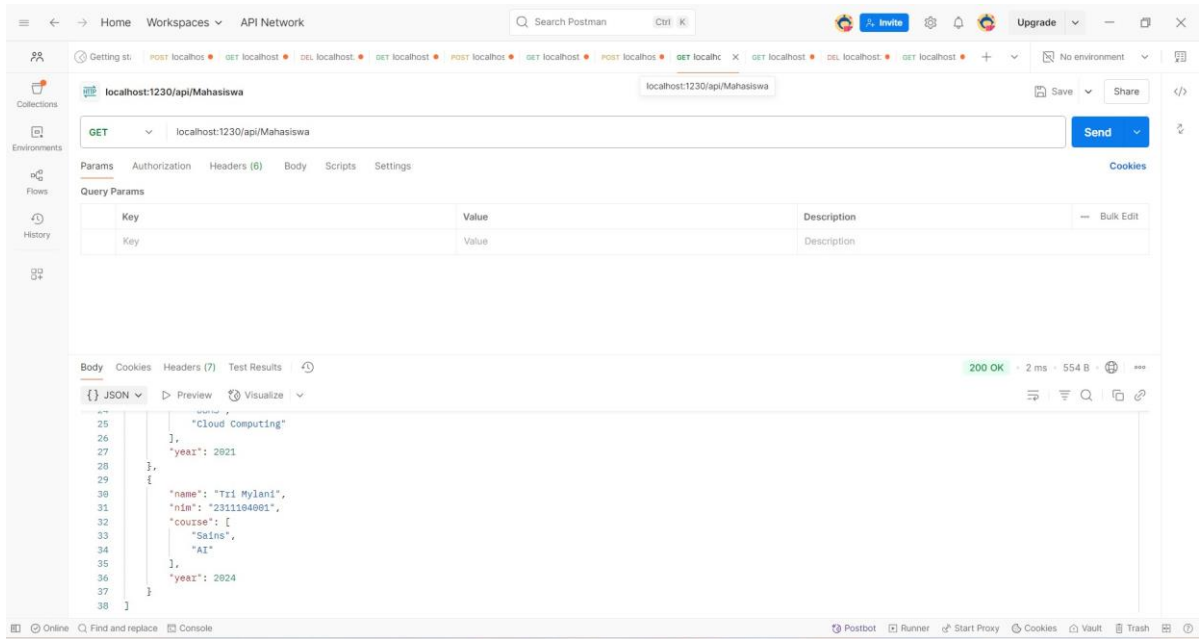
7. GET localhost:1230/api/Mahasiswa, menampilkan list



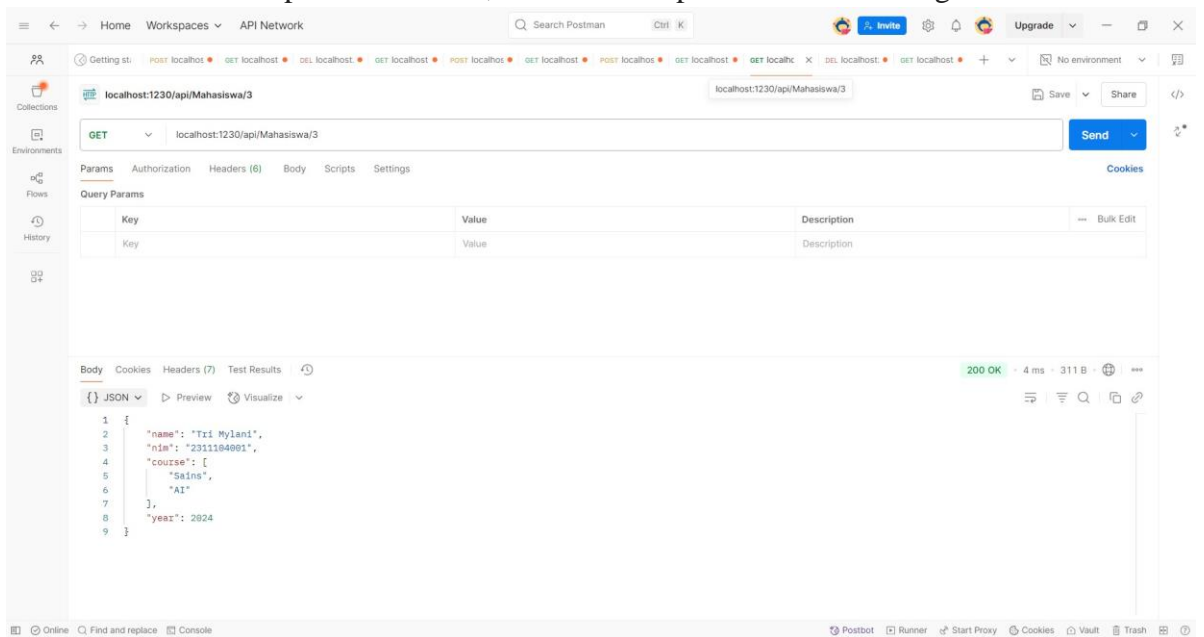
8. POST localhost:1230/api/Mahasiswa, untuk menambahkan list



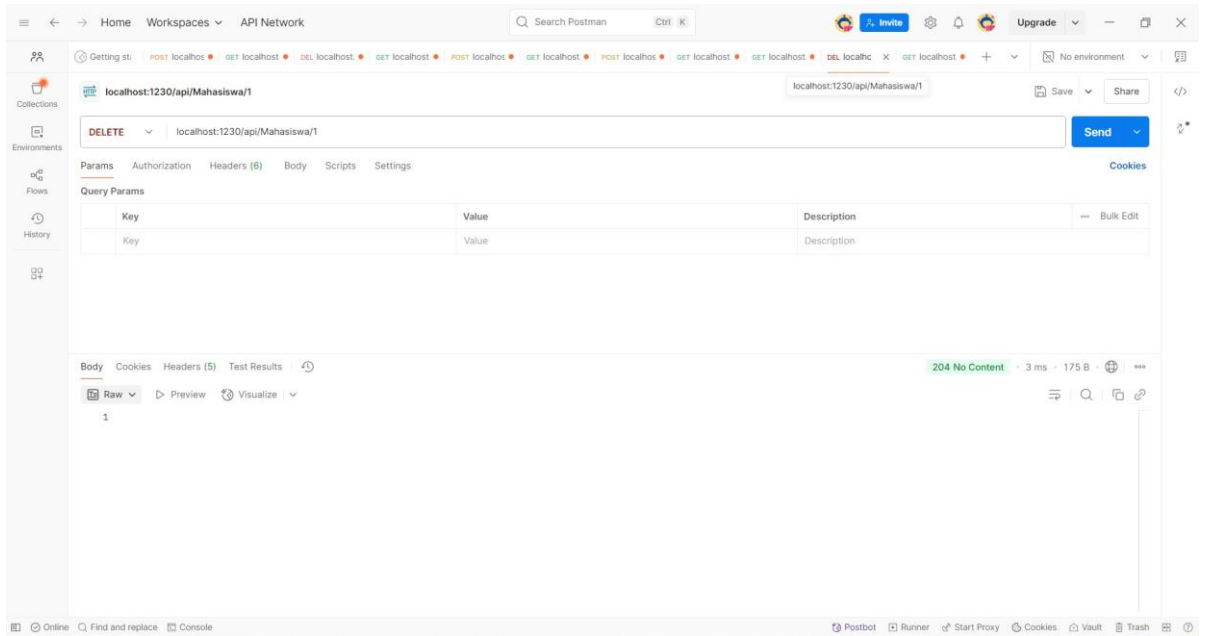
9. GET localhost:1230/api/Mahasiswa, menampilkan semua list



10. GET localhost:1230/api/Mahasiswa/3, untuk menampilkan list nama dengan index 3



11. DELETE localhost:1230/api/Mahasiswa/1, untuk menghapus list dengan index 1



12. GET `localhost:1230/api/Mahasiswa`, memanggil list dan index 1 sudah tidak ada

