

TP9_2311104002_Shilfi Habibah_SE0701

I. Link github

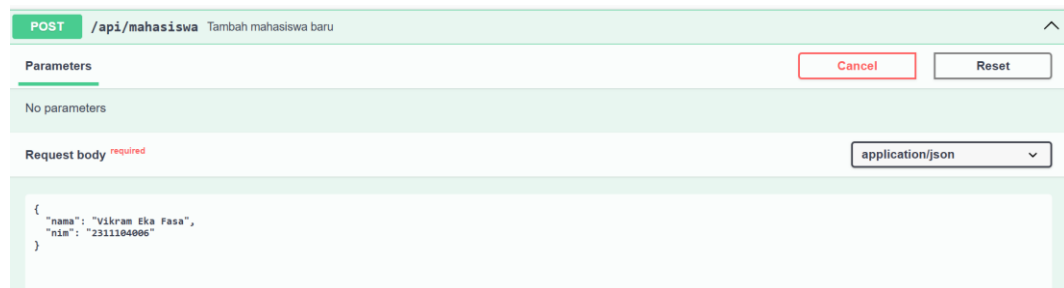
https://github.com/shilfihabibah/KPL_Shilfi-Habibah_2311104002_SE07-01/tree/master/09_API%20Design%20dan%20Construction%20Using%20Swagger/TP9_2311104002

II. Screenshot hasil running

- a. Mencoba “GET /api/mahasiswa” saat baru dijalankan (mengeluarkan list nama mahasiswa dan nim anggota kelompok)



- b. Menambahkan mahasiswa => Nama: “Vikram Eka Fasa” dan NIM: “2311104006” dengan “POST/api/mahasiswa”



Curl

```
curl -X 'POST' \
  http://localhost:5000/api/mahasiswa' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "nama": "Vikram Eka Fasa",
    "nim": "2311104006"
  }'
```

Request URL

http://localhost:5000/api/mahasiswa

Server response

Code	Details
201	<p>Response body</p> <p>Mahasiswa berhasil ditambahkan</p> <p>Response headers</p> <pre>connection: keep-alive content-length: 30 content-type: text/html; charset=utf-8 date: Sun, 27 Apr 2025 13:41:25 GMT etag: W/"1e-BnoxVRu+1YU0UwVrC7BxYVw9xg" keep-alive: timeout=5 x-powered-by: Express</pre>

- c. Cek list/array dari semua mahasiswa lagi dengan “GET /api/mahasiswa”, pastikan mahasiswa yang baru ditambahkan sebelumnya ada di list mahasiswa:

Curl

```
curl -X 'GET' \
  http://localhost:5000/api/mahasiswa' \
  -H 'accept: */*' \
```

Request URL

http://localhost:5000/api/mahasiswa

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "nama": "Shilfi Habibah", "nim": "2311104002" }, { "nama": "Hannum Gusmiarni", "nim": "2311104015" }, { "nama": "Jihan Kencana", "nim": "2311104008" }, { "nama": "Vikram Eka Fasa", "nim": "2311104006" }]</pre> <p>Response headers</p>

- d. Mencoba meminta mahasiswa dengan index 0, “GET /api/mahasiswa/0” yang seharusnya mengeluarkan nama dan nim anda:

index * required
integer
(path)

0

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:5000/api/mahasiswa/0' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5000/api/mahasiswa/0
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "nama": "Shilfi Habibah", "nim": "2311184002" }</pre> <p>Download</p>

- e. Menghapus objek mahasiswa dengan index ke-0 dengan “DELETE /api/mahasiswa/0”

index * required
integer
(path)

0

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://localhost:5000/api/mahasiswa/0' \
-H 'accept: */*'
```

Request URL

```
http://localhost:5000/api/mahasiswa/0
```

Server response

Code	Details
200	<p>Response body</p> <pre>Mahasiswa berhasil dihapus</pre> <p>Download</p>

- f. Cek list/array dari semua mahasiswa sekali lagi dengan “GET /api/mahasiswa”, pastikan nama anda sudah tidak muncul di list tersebut:

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:5000/api/mahasiswa' \
  -H 'accept: */*'
```

Request URL

```
http://localhost:5000/api/mahasiswa
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "nama": "Hannum Gusniarni", "nim": "2311104015" }, { "nama": "Jihan Kencana", "nim": "2311104008" }, { "nama": "Vikram Eka Fasa", "nim": "2311104006" } }</pre> <p>Download</p>

III. Codingan

Code :

app.js

```
1  const express = require('express');
2  const swaggerUi = require('swagger-ui-express');
3  const mahasiswa = require('./mahasiswa');
4  const swaggerDocument = require('./swagger.json');
5
6  const app = express();
7  const port = 5000;
8
9  app.use(express.json());
10 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
11
12 app.get('/api/mahasiswa', (req, res) => {
13   res.json(mahasiswa);
14 });
15
16 app.get('/api/mahasiswa/:index', (req, res) => {
17   const index = parseInt(req.params.index);
18   if (index < 0 || index >= mahasiswa.length) {
19     res.status(404).send('Mahasiswa tidak ditemukan');
20   } else {
21     res.json(mahasiswa[index]);
22   }
23 });
24
25 app.post('/api/mahasiswa', (req, res) => {
26   const { nama, nim } = req.body;
27   mahasiswa.push({ nama, nim });
28   res.status(201).send('Mahasiswa berhasil ditambahkan');
29 });
30
31 app.delete('/api/mahasiswa/:index', (req, res) => {
32   const index = parseInt(req.params.index);
33   if (index < 0 || index >= mahasiswa.length) {
34     res.status(404).send('Mahasiswa tidak ditemukan');
35   } else {
36     mahasiswa.splice(index, 1);
37     res.send('Mahasiswa berhasil dihapus');
38   }
39 });
40
41 app.listen(port, () => {
42   console.log(`Server berjalan di http://localhost:${port}`);
43   console.log(`Swagger UI: http://localhost:${port}/api-docs`);
44 });
```

```
1  woqni6'exbofz = wapsz7ms?
2
3  ]?
4  { usms: "j1pys kencas", utw: "331104008" },
5  { usms: "Hysuun cenzalut", utw: "331104012" },
6  { usms: "2p11t Hapirp", utw: "331104005" },
7  ^ couzf wapsz7ms = [
```

```

1  {
2    "openapi": "3.0.0",
3    "info": {
4      "title": "API Mahasiswa",
5      "version": "1.0.0",
6      "description": "API sederhana untuk data mahasiswa"
7    },
8    "servers": [
9      {
10       "url": "http://localhost:5000"
11     }
12   ],
13   "paths": {
14     "/api/mahasiswa": {
15       "get": {
16         "summary": "Get semua mahasiswa",
17         "responses": {
18           "200": {
19             "description": "List mahasiswa"
20           }
21         }
22       },
23       "post": {
24         "summary": "Tambah mahasiswa baru",
25         "requestBody": {
26           "required": true,
27           "content": {
28             "application/json": {
29               "schema": {

```

```

30         "type": "object",
31         "properties": {
32             "nama": { "type": "string" },
33             "nim": { "type": "string" }
34         }
35     },
36 }
37 },
38 },
39 "responses": {
40     "201": {
41         "description": "Mahasiswa berhasil ditambahkan"
42     }
43 }
44 },
45 },
46 "/api/mahasiswa/{index}": {
47     "get": {
48         "summary": "Get mahasiswa berdasarkan index",
49         "parameters": [
50             {
51                 "name": "index",
52                 "in": "path",
53                 "required": true,

```

```

54         "schema": { "type": "integer" }
55     }
56 ],
57 "responses": {
58     "200": {
59         "description": "Data mahasiswa"
60     },
61     "404": {
62         "description": "Mahasiswa tidak ditemukan"
63     }
64 }
65 },
66 "delete": {
67     "summary": "Hapus mahasiswa berdasarkan index",
68     "parameters": [
69         {
70             "name": "index",
71             "in": "path",
72             "required": true,
73             "schema": { "type": "integer" }
74         }
75     ],
76     "responses": {
77         "200": {

```

```

78             "description": "Mahasiswa berhasil dihapus"
79         },
80         "404": {
81             "description": "Mahasiswa tidak ditemukan"
82         }
83     }
84 }
85 },
86 ],
87 }

```

Penjelasan:

Kode ini adalah implementasi API sederhana untuk mengelola data mahasiswa menggunakan Node.js dan framework Express.js. Berikut penjelasannya:

- Mengimpor Modul: Menggunakan require untuk memasukkan library express, swagger-ui-express (untuk dokumentasi API), file data mahasiswa.js, dan file konfigurasi Swagger swagger.json.
- Membuat Aplikasi Express: `const app = express();` membuat instance aplikasi Express.
- Menentukan Port: `const port = 5000;` menetapkan port server akan berjalan.
- Middleware:
 - `app.use(express.json());` mengaktifkan middleware untuk mengurai body permintaan JSON.
 - `app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));` mengaktifkan Swagger UI untuk menampilkan dokumentasi API di /api-docs.
- Endpoint API:
 - GET /api/mahasiswa: Mengambil semua data mahasiswa dan mengirimkannya sebagai JSON.
 - GET /api/mahasiswa/:index: Mengambil data mahasiswa berdasarkan index (posisi dalam array). Mengirimkan error 404 jika index tidak valid.
 - POST /api/mahasiswa: Menerima data nama dan NIM dari body permintaan JSON, menambahkan mahasiswa baru ke array mahasiswa, dan mengirimkan respons sukses (201).
 - DELETE /api/mahasiswa/:index: Menghapus data mahasiswa berdasarkan index. Mengirimkan error 404 jika index tidak valid.
- Menjalankan Server: `app.listen(port, ...)` memulai server dan menampilkan pesan di konsol tentang alamat server dan Swagger UI.
- Data Mahasiswa: Array mahasiswa berisi data awal mahasiswa yang diekspor agar bisa digunakan di file lain.