

Praktikum 9 - Matakuliah Pilihan 1 (Web)

Program Studi: Teknik Informatika

Lakukan praktikum dibawah ini, dan buat screenshot untuk pembuktian mengerjakan setiap poin dengan mengisi tabel dibawah, kemudian tunjukan hasil akhir dari men-share repository github yang telah dibuat.

A. Menyediakan API Endpoints

1. Lanjutkan Project Praktikum 8, dengan menggunakan file yang sama (copy)
2. Tambahkan pada database, sebuah tabel produk
Isi tabel produk seperti pada tabel berikut: (Buat 10 item)

id	nama	deskripsi	harga	foto
1.	Indomie Goreng		2500	images/miegoreng.jpg
2.
10.

3. Seperti pada perintah praktikum 8 buatkan beberapa endpoints
GET : localhost:8001/api/products/
GET :
localhost:8001/api/products/:id POST
: localhost:8001/api/products/
PUT :
localhost:8001/api/products/:id DELETE:
localhost:8001/api/products/:id
4. Pastikan memiliki file dengan struktur sebagai berikut.
controllers/[products.controller.js](#)
routes/[products.routes.js](#)
models/[products.model.js](#)
5. Test API dengan menggunakan **POSTMAN**

B. Menambahkan Proteksi API (Simple - Bearer Method)

1. Buat sebuah folder bernama middlewares, kemudian buat didalamnya sebuah file dengan nama [auth.middleware.js](#) dengan kode program seperti dibawah ini

```
export const authBearer = (req, res, next) => {

    const authHeader = req.headers.authorization;

    // Tidak ada Authorization
    if (!authHeader) {
        return res.status(401).json({ message: "No authorization header" });
    }

    // Harus Bearer
    if (!authHeader.startsWith("Bearer ")) {
        return res.status(401).json({ message: "Bearer token required" });
    }

    // Ambil token
    const token = authHeader.split(" ")[1];

    // Token yang benar (misal hardcode)
    const VALID_TOKEN = "12345TOKENRAHASIA";

    if (token !== VALID_TOKEN) {
        return res.status(403).json({ message: "Invalid token" });
    }

    next();
};
```

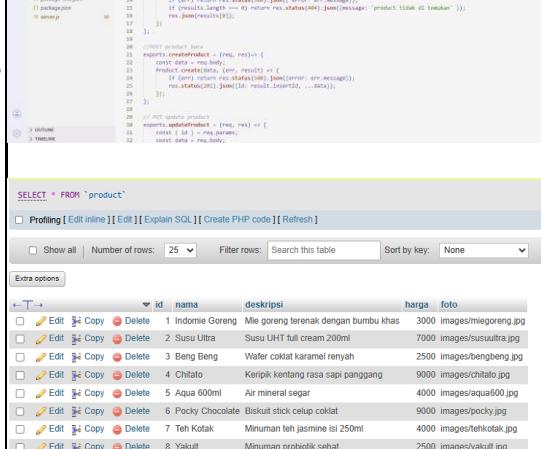
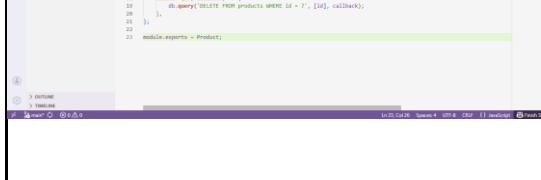
2. Pada file [product.routes.js](#) panggil [auth.middleware.js](#)
import { authBearer } from "../middleware/auth.middleware.js"
3. Lalu tambahkan authBearer ke endpoints yang perlu di proteksi
router.post("/", authBearer, createProduct);
router.put("/:id", authBearer, updateProduct);
router.delete("/:id", authBearer, deleteProduct);
4. Gunakan POSTMAN untuk akses 3 endpoints ini dengan menambahkan bearer
12345TOKENRAHASIA

F. Github + Visual Code

- Buat proyek di Github dengan nama **Latihan9**

```
git init
git add .
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/agunghakase/Latihan9.git
git push -u origin main
```

Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Installasi dan Konfigurasi		
1.	<p>1. Lanjutkan Project Praktikum 8, dengan menggunakan file yang sama (copy)</p> <p>2. Tambahkan pada database, sebuah tabel produk Isi tabel produk seperti pada tabel berikut: (Buat 10 item)</p> <p>3. Seperti pada perintah praktikum 8 buatkan beberapa endpoints GET : localhost:8001/api/products / GET : localhost:8001/api/products/:id POST : localhost:8001/api/products / PUT : localhost:8001/api/products/:id DELETE: localhost:8001/api/products/:id</p> <p>4. Pastikan memiliki file dengan struktur sebagai berikut.</p> <p>controllers/products.controller.js routes/products.routes.js models/products.model.js</p> <p>5. Test API dengan menggunakan POSTMAN</p>	  	

	<p>The code editor shows the file product.routes.js with the following content:</p> <pre> const express = require('express'); const router = express.Router(); const authMiddleware = require('../middlewares/auth.middleware'); router.get('/', authMiddleware, getProducts); router.get('/:id', authMiddleware, getProduct); router.post('/', authMiddleware, createProduct); router.delete('/:id', authMiddleware, deleteProduct); module.exports = router; </pre> <p>The browser screenshot shows a successful GET request to http://localhost:8001/api/products/5. The response body is a JSON object:</p> <pre> { "id": 5, "name": "Teh Susu Mentega", "description": "Teh Susu Mentega", "harga": 10000, "foto": "images/teh.jpg" } </pre>
2.	<p>1. Buat sebuah folder bernama middlewares, kemudian buat didalamnya sebuah file dengan nama auth.middleware.js dengan kode program seperti dibawah ini</p> <p>2. Pada file product.routes.js panggil auth.middleware.js import { authBearer } from</p> <p>The code editor shows the file auth.middleware.js with the following content:</p> <pre> const jwt = require('jsonwebtoken'); const authMiddleware = (req, res, next) => { const authHeader = req.headers.authorization; if (!authHeader) { return res.status(401).json({ message: 'No authorization header' }); } const token = authHeader.split(' ')[1]; const decodedToken = jwt.verify(token, process.env.JWT_SECRET); if (!decodedToken) { return res.status(401).json({ message: 'Invalid token' }); } next(); }; module.exports = authMiddleware; </pre> <p>The browser screenshot shows a successful POST request to http://localhost:8001/api/products/. The response body is a JSON object:</p> <pre> { "id": 6, "name": "Susu Hazz", "description": "Susu Hazz mengandung protein tinggi", "harga": 10000, "foto": "images/susu_hazz.jpg" } </pre>

	<pre>"/middleware/auth.middleware.js"</pre> <p>3. Lalu tambahkan authBearer ke endpoints yang perlu di proteksi</p> <pre>router.post("/", authBearer, createProduct); router.put("/:id", authBearer, updateProduct); router.delete("/:id", authBearer, deleteProduct);</pre> <p>4. Gunakan POSTMAN untuk akses 3 endpoints ini dengan menambahkan bearer 12345TOKENRAHASIA</p>		
B.	Github dan Viscode		
1.	<p>1. Buat proyek di Github dengan nama Latihan9</p> <pre>git init git add . git commit -m "first commit" git branch -M main git remote add origin https://github.com/anastasyafdg/Latihan9.git git push -u origin main</pre>		