

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 Enpoints

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

1. Buat proyek di Github dengan nama **Latihan9**

git init

git add

.

git commit -m "first commit" git branch -M main git remote add

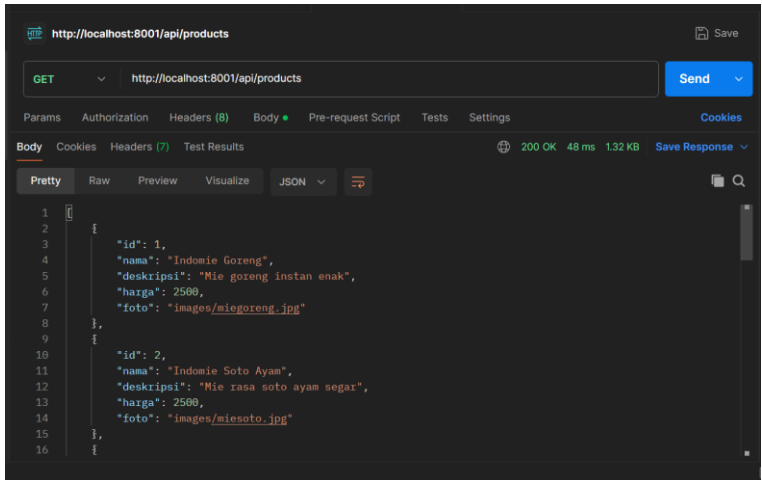
origin <https://github.com/agunghakase/Latihan9.git> git push -u

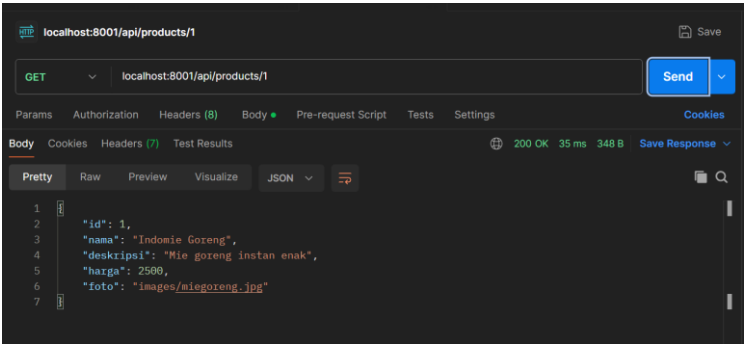
origin main

Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Instalasi dan Konfigurasi		
1.			
2.			
B.	code		

1.	products.controller	<pre> controllers > products.controller.js > ... 1 const Product = require("../models/products.model"); 2 3 exports.getProducts = (req, res) => { 4 Product.getAll((err, results) => { 5 if (err) res.send(err); 6 else res.json(results); 7 }); 8 }; 9 10 exports.getProductById = (req, res) => { 11 Product.getById(req.params.id, (err, results) => { 12 if (err) res.send(err); 13 else res.json(results[0]); 14 }); 15 }; 16 17 exports.createProduct = (req, res) => { 18 Product.create(req.body, (err, results) => { 19 if (err) res.send(err); 20 else res.json({ 21 message: "Produk berhasil ditambahkan!", 22 data: req.body, 23 }); 24 }); 25 }; 26 27 exports.updateProduct = (req, res) => { 28 Product.update(req.params.id, req.body, (err, results) => { 29 if (err) res.send(err); 30 else res.json({ message: "Produk berhasil diperbarui!" }); 31 }); 32 }; 33 34 exports.deleteProduct = (req, res) => { 35 Product.remove(req.params.id, (err, results) => { 36 if (err) res.send(err); 37 else res.json({ message: "Produk berhasil dihapus!" }); 38 }); 39 }; 40 </pre>	
	products.model	<pre> 35 36 } 37 qr.dnsl("DELETE FROM bloqncfz MHEBE IQ = 3" [IQ] LgnJf): 38 exbolfz.LgWOLG = (IQ LgnJf) => { 39 40 } 41 qr.dnsl("UPDATE bloqncfz ZEL 3 MHEBE IQ = 3" [qaf9 IQ] LgnJf): 42 exbolfz.nbqaf6 = (IQ qaf9 LgnJf) => { 43 44 } 45 qr.dnsl("INSERT INTO bloqncfz ZEL 3" qaf9 LgnJf): 46 exbolfz.cLgaf6 = (qaf9 LgnJf) => { 47 48 } 49 qr.dnsl("DELETE * FROM bloqncfz MHEBE IQ = 3" [IQ] LgnJf): 50 exbolfz.BefblyIQ = (IQ LgnJf) => { 51 52 } 53 qr.dnsl("DELETE * FROM bloqncfz" LgnJf): 54 exbolfz.BefvJJ = (LgnJf) => { 55 56 } 57 couzf qr = LEdnJL("'\coufz\qaf9p926"): 58 wogfz > bloqncfz:wogfz > "" </pre>	

	products.routes	<pre> routes > products.routes.js > ... 1 const express = require("express"); 2 const { 3 getProducts, 4 getProductById, 5 createProduct, 6 updateProduct, 7 deleteProduct, 8 } = require("../controllers/products.controller"); 9 10 const { authBearer } = require("../middlewares/auth.middleware"); 11 12 const router = express.Router(); 13 14 router.get("/", getProducts); 15 router.get("/:id", getProductById); 16 router.post("/", authBearer, createProduct); 17 router.put("/:id", authBearer, updateProduct); 18 router.delete("/:id", authBearer, deleteProduct); 19 20 module.exports = router; 21 </pre>	
	auth.middleware	<pre> middlewares > auth.middleware.js > ... 1 exports.authBearer = (req, res, next) => { 2 const auth = req.headers["authorization"]; 3 4 if (!auth !auth.startsWith("Bearer ")) { 5 return res.status(401).json({ message: "Token tidak ditemukan" }); 6 } 7 8 const token = auth.split(" ")[1]; 9 10 if (token !== "12345TOKENRAHASIA") { 11 return res.status(403).json({ message: "Token tidak valid" }); 12 } 13 14 next(); 15 } 16 </pre>	
	<p>GET</p> <p>http://localhost:8001/api/products</p>	 <pre> 1 { 2 "id": 1, 3 "nama": "Indomie Goreng", 4 "deskripsi": "Mie goreng instan enak", 5 "harga": 2500, 6 "foto": "images/miegoreng.jpg" 7 }, 8 { 9 "id": 2, 10 "nama": "Indomie Soto Ayam", 11 "deskripsi": "Mie rasa soto ayam segar", 12 "harga": 2500, 13 "foto": "images/miesoto.jpg" 14 }, 15], 16 {} </pre>	

	<p>GET</p> <p>http://localhost:8001/api/products/1</p>		
	<p>POST</p> <p>http://localhost:8001/api/products</p>		
B.	Github dan Viscode		
1.			
2.			
...			