

## 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.


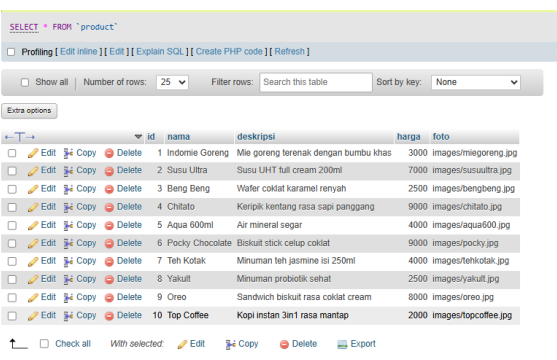
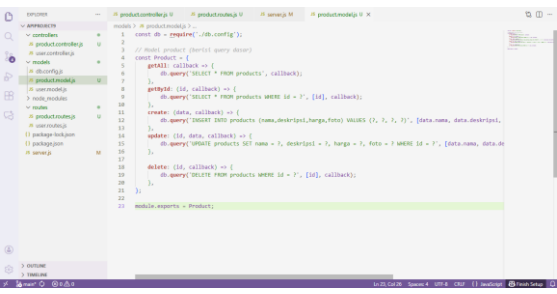
git commit -m "first commit"

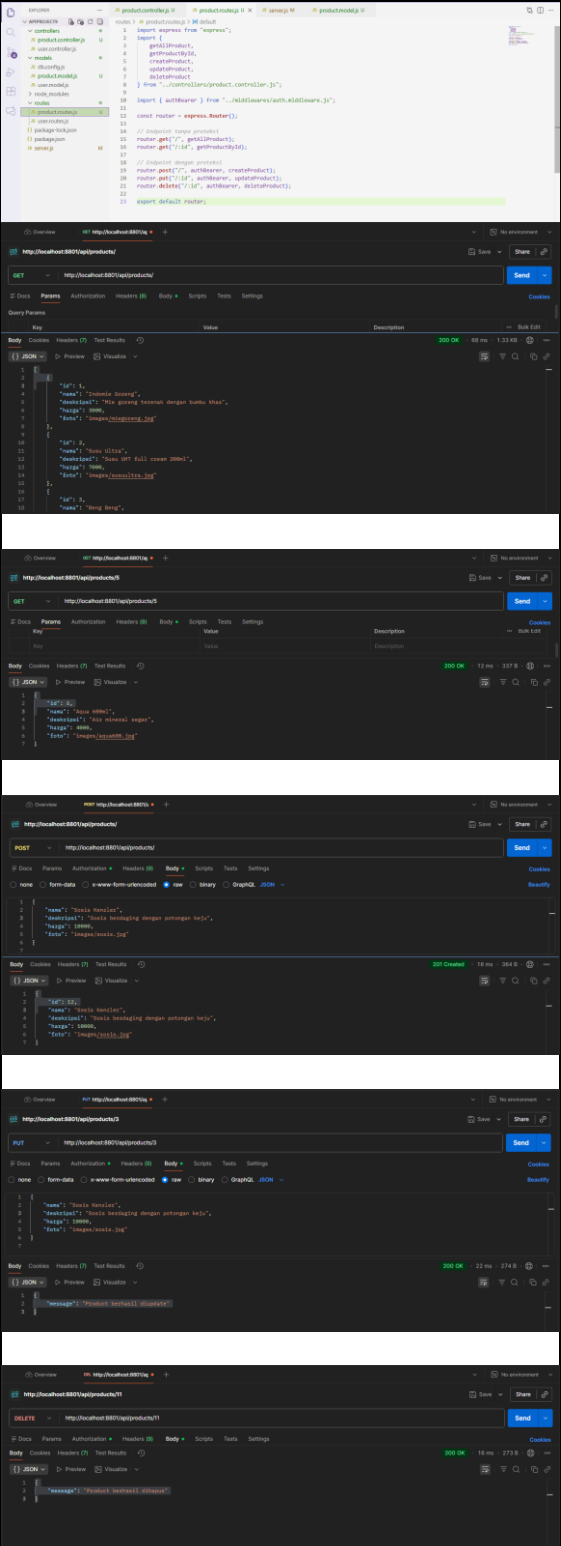
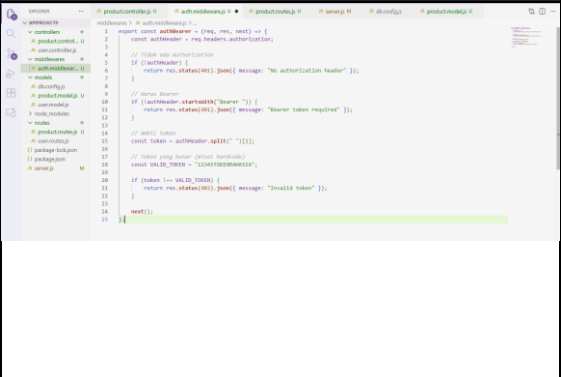
git branch -M main

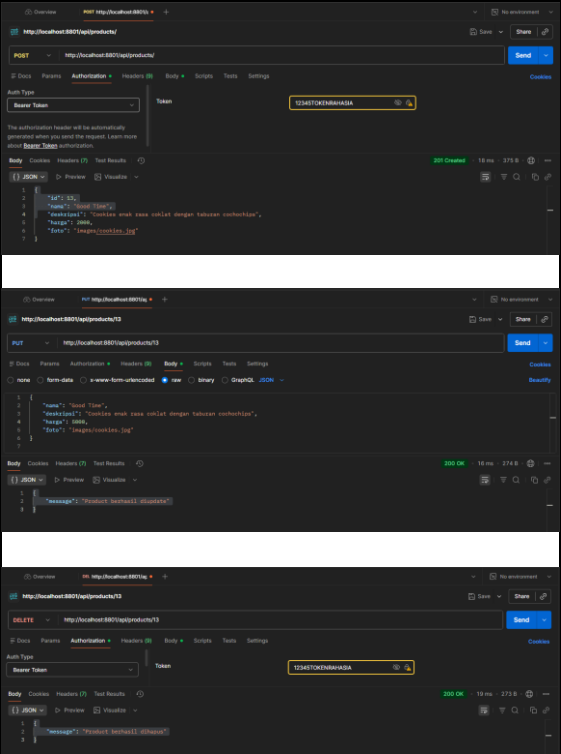
git remote add origin https://github.com/agunghakase/Latihan9.git

git push -u origin main

## Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Instalasi dan Konfigurasi		
1.	<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 buat 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. controllers/products.controller.js routes/products.routes.js models/products.model.js</p> <p>5. Test API dengan menggunakan POSTMAN</p>	  	

		 <pre> // product.routes.js const express = require('express'); const router = express.Router();  // Import controllers const {   getProductById,   createProduct,   updateProduct,   deleteProduct, } = require('../controllers/product.controller');  // Routes // GET /api/products/:id router.get('/:id', getProductById);  // POST /api/products router.post('/', createProduct);  // PUT /api/products/:id router.put('/:id', updateProduct);  // DELETE /api/products/:id router.delete('/:id', deleteProduct);  export default router; </pre> <p>Postman GET /http://localhost:8080/api/products/</p> <pre> {   "id": 1,   "name": "Produk Baru",   "description": "Produk baru yang sedang diuji",   "price": 10000,   "category": "Elektronik",   "image": "image/produk.jpg" } </pre> <p>Postman GET /http://localhost:8080/api/products/5</p> <pre> {   "id": 5,   "name": "Produk Baru",   "description": "Produk baru yang sedang diuji",   "price": 10000,   "category": "Elektronik",   "image": "image/produk.jpg" } </pre> <p>Postman POST /http://localhost:8080/api/products/</p> <pre> {   "name": "Produk Baru",   "description": "Produk baru yang sedang diuji",   "price": 10000,   "category": "Elektronik",   "image": "image/produk.jpg" } </pre> <p>Postman PUT /http://localhost:8080/api/products/3</p> <pre> {   "id": 3,   "name": "Produk Baru",   "description": "Produk baru yang sedang diuji",   "price": 10000,   "category": "Elektronik",   "image": "image/produk.jpg" } </pre> <p>Postman DELETE /http://localhost:8080/api/products/1</p> <pre> {   "message": "Produk berhasil dihapus" } </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>	 <pre> // auth.middleware.js const jwt = require('jsonwebtoken');  const authBearer = (req, res, next) =&gt; {   const authHeader = req.headers.authorization;    // Tidak ada authorization   if (!authHeader) {     return res.status(401).json({ message: 'No authorization header' });   }    // Token Bearer   if (authHeader.startsWith('bearer ')) {     const token = authHeader.split(' ')[1];      // Token yang benar (tidak terverifikasi)     const validToken = 'tokenValidToken';      if (token !== validToken) {       return res.status(401).json({ message: 'Token tidak valid' });     }      next();   } }  module.exports = authBearer; </pre> <pre> // product.routes.js const express = require('express'); const router = express.Router();  // Import controllers const {   getProductById,   createProduct,   updateProduct,   deleteProduct, } = require('../controllers/product.controller');  // Import middlewares const { authBearer } = require('../middlewares/auth.middleware');  // Routes // GET /api/products/:id router.get('/:id', authBearer, getProductById);  // POST /api/products router.post('/', authBearer, createProduct);  // PUT /api/products/:id router.put('/:id', authBearer, updateProduct);  // DELETE /api/products/:id router.delete('/:id', authBearer, deleteProduct);  export default router; </pre>	

	<p>"../middleware/auth.middleware.js"</p> <p>3. Lalu tambahkan authBearer ke endpoints yang perlu di proteksi  router.post("/", authBearer, createProduct);  router.put("/:id", authBearer, updateProduct);  router.delete("/:id", authBearer, deleteProduct);</p> <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> <p>git init  git add .  git commit -m "first commit"  git branch -M main  git remote add origin <a href="https://github.com/anastasyafdg/Latihan9.git">https://github.com/anastasyafdg/Latihan9.git</a>  git push -u origin main</p>	