

## 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/miegor eng.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
```

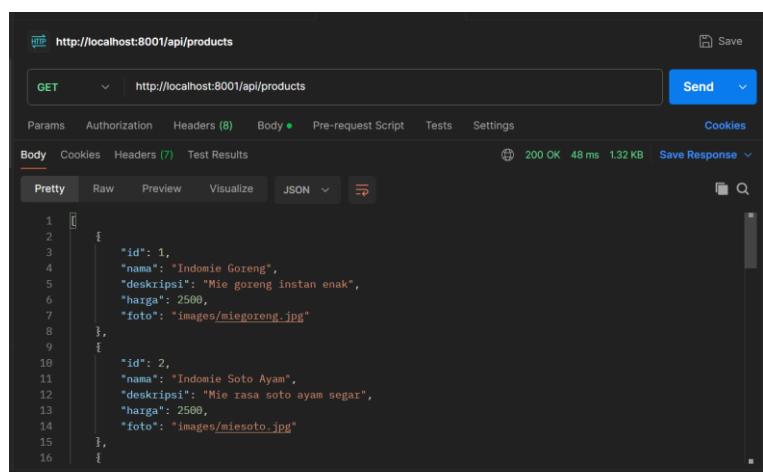
```
git add
```

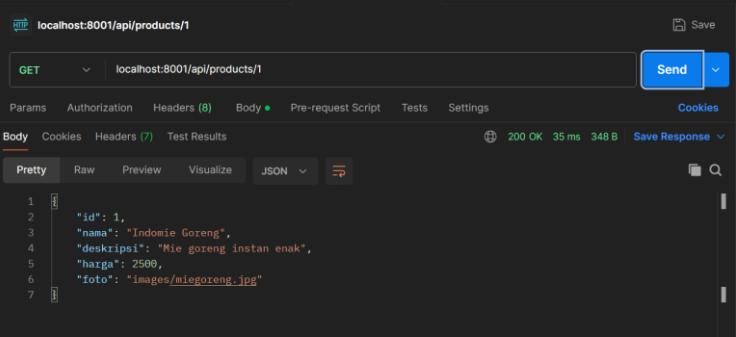
```
git commit -m "first commit" git branch -M main git remote add origin https://github.com/agunghakase/Latihan9.git git push -u origin main
```

## Hasil Pengeraan

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

1.	products.controller	<pre> controllers &gt; products.controller.js &gt; ... 1  const Product = require("../models/products.model"); 2 3  exports.getProducts = (req, res) =&gt; { 4    Product.getAll((err, results) =&gt; { 5      if (err) res.send(err); 6      else res.json(results); 7    }); 8  }; 9 10 exports.getProductById = (req, res) =&gt; { 11   Product.getById(req.params.id, (err, results) =&gt; { 12     if (err) res.send(err); 13     else res.json(results[0]); 14   }); 15 }; 16 17 exports.createProduct = (req, res) =&gt; { 18   Product.create(req.body, (err, results) =&gt; { 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) =&gt; { 28   Product.update(req.params.id, req.body, (err, results) =&gt; { 29     if (err) res.send(err); 30     else res.json({ message: "Produk berhasil diperbarui!" }); 31   }); 32 }; 33 34 exports.deleteProduct = (req, res) =&gt; { 35   Product.remove(req.params.id, (err, results) =&gt; { 36     if (err) res.send(err); 37     else res.json({ message: "Produk berhasil dihapus!" }); 38   }); 39 }; 40 </pre>	
	products.model	<pre> 55 56  }: 57     qd.dnul(`DELETE FROM products WHERE id = \${`[\${`q`}, \${`resut`}]`}`); 58  exports.delete = (q, resut) =&gt; { 59 60  }: 61     qd.dnul(`UPDATE products SET \${`MHEBE`}` = \${`[\${`q`}, \${`q`}, \${`resut`}]`}); 62  exports.update = (q, qrt, resut) =&gt; { 63 64  }: 65     qd.dnul(`INSERT INTO products SET \${`q`}` = \${`[\${`q`}, \${`resut`}]`}); 66  exports.create = (qrt, resut) =&gt; { 67 68  }: 69     qd.dnul(`SELECT * FROM products WHERE id = \${`[\${`q`}]`}`); 70  exports.get = (q) =&gt; { 71 72  }: 73     qd.dnul(`SELECT * FROM products WHERE \${`q`}` = \${`[\${`q`}, \${`resut`}]`}); 74  exports.getBy = (q, resut) =&gt; { 75 76  }: 77     const qr = `DELETE FROM \${`'\${`compt`}\\${`qf`}`}`; 78  exports.deleteCompt = qr; </pre>	

	products.routes	<pre> routes &gt; products.routes.js &gt; ... 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 &gt; auth.middleware.js &gt; ... 1  exports.authBearer = (req, res, next) =&gt; { 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>	
	GET <a href="http://localhost:8001/api/products">http://localhost:8001/api/products</a>	 <pre> [   {     "id": 1,     "nama": "Indomie Goreng",     "deskripsi": "Mie goreng instan enak",     "harga": 2500,     "foto": "images/miegoreng.jpg"   },   {     "id": 2,     "nama": "Indomie Soto Ayam",     "deskripsi": "Mie rasa soto ayam segar",     "harga": 2500,     "foto": "images/miesoto.jpg"   } ] </pre>	

	<p><b>GET</b>  <a href="http://localhost:8001/api/products/1">http://localhost:8001/api/products/1</a></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 </pre>	
	<p><b>POST</b>  <a href="http://localhost:8001/api/products">http://localhost:8001/api/products</a></p>		
B.	Github dan Viscode		
1.			
2.			
...			