

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 .

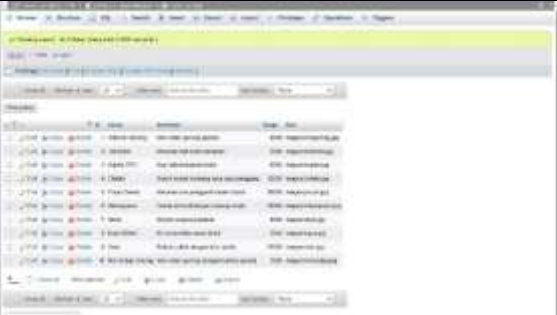
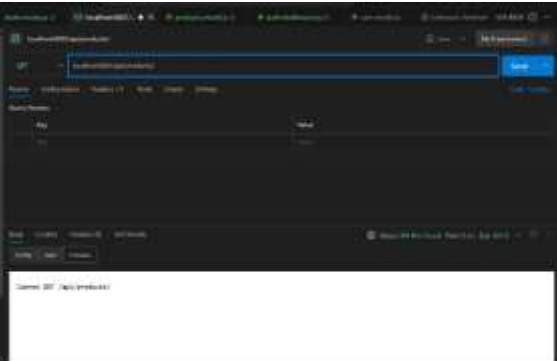

git commit -m "first commit"

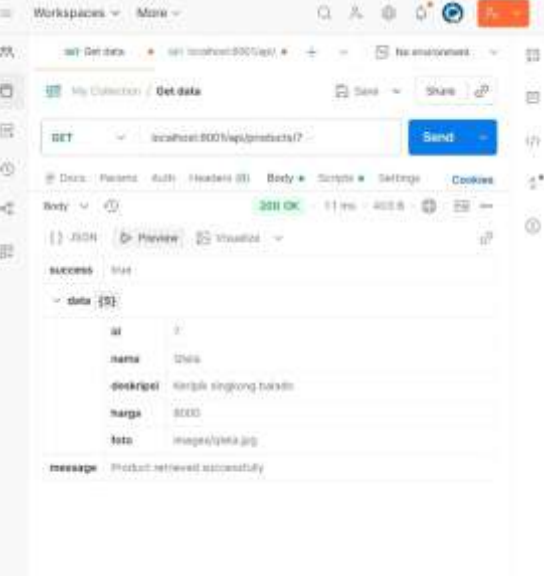
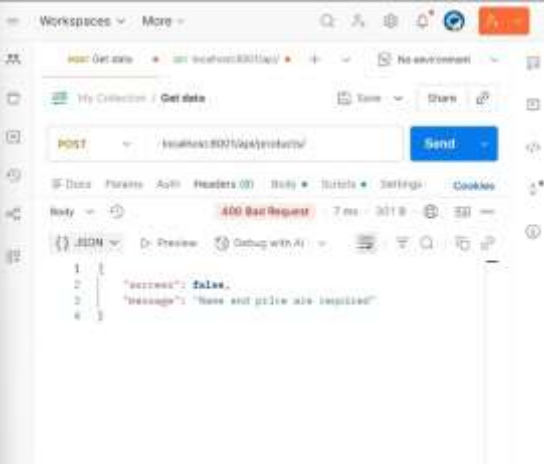

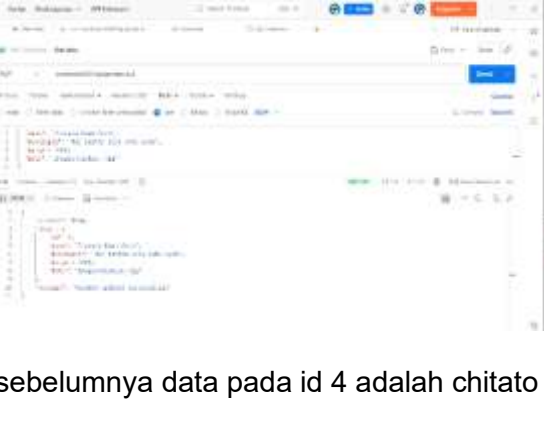
git branch -M main


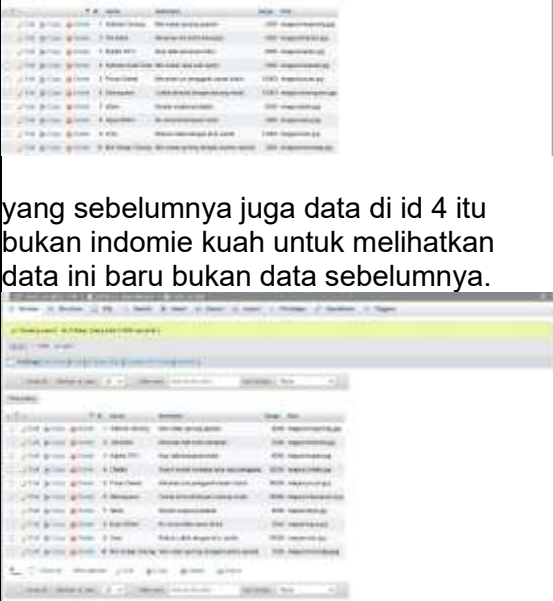
git remote add origin https://github.com/agunghakase/Latihan9.git


git push -u origin main

Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Instalasi dan Konfigurasi		
1.	Membuat Tabel "Produk" dan kolom atau isi yang di butuhkan untuk database dan mengisi 10 produk berdasarkan perintah tugas.		
2.	GET : localhost:8001/api/products/ (gagal)		Masih belum muncul, karena masih ada salah code
3.	GET : localhost:8001/api/products/ (berhasil)		Menampilkan 10 data produk

<p>4.</p> <p>GET : localhost:8001/api/products/ 7</p> <p>Menampilkan data berdasarkan id produk</p>			
<p>5.</p> <p>POST : localhost:8001/api/products (gagal)</p>			<p>Masih gagal karena saya masih belum setting token belum mencoba menambahkan data nya</p>
<p>6.</p> <p>POST : localhost:8001/api/products (berhasil)</p>			
<p>7.</p> <p>PUT : localhost:8001/api/products/ 4</p>		 <p>sebelumnya data pada id 4 adalah chitato</p>	

			
8.	<p>DELETE : localhost:8001/api/product/11</p>	<p>bisa di lihat ini data baru dan data no 11 sudah di hapus</p>  <p>yang sebelumnya juga data di id 4 itu bukan indomie kuah untuk melihatkan data ini baru bukan data sebelumnya.</p>	<p>Pada data sebelumnya sudah ada 11 data karena saya tambah hanya saya lupa meng scrensnot hasil</p>
9.			
B.	Github dan Viscode		

1.	Saya sudah menambahkan file yang di butuhkan seperti - products.controller.js - products.model.js - products.route.js - auth.middleware.js			
2.				
...				