

**NIM : 3312411077**

**Nama : Anisa Frity Amelia**

**Kelas : IF3A Web Pagi**

**Link Github : <https://github.com/anisafrityamelia/praktikum8.git>**

## **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**

## C. Github + Visual Code

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

git init

**git add .**

git commit -m "first commit"

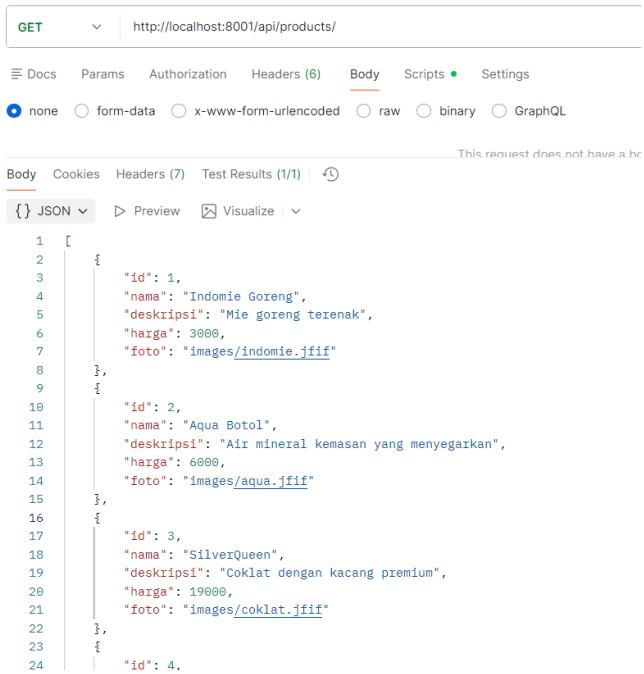
git branch -M main

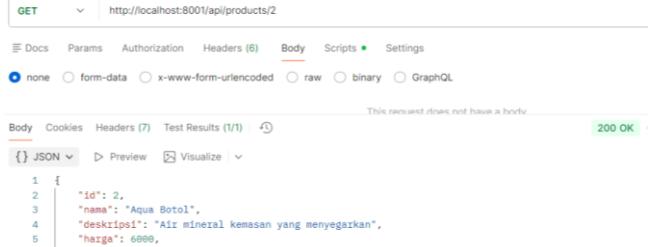
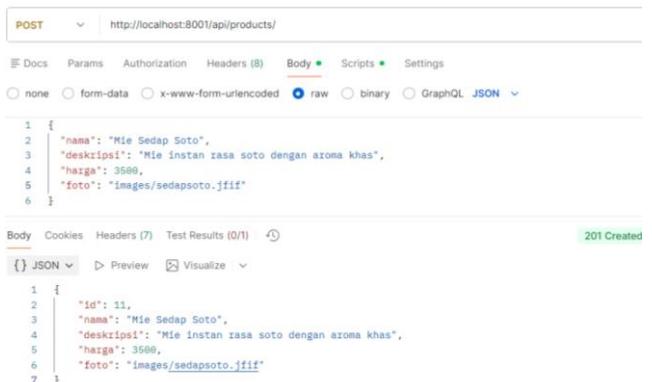
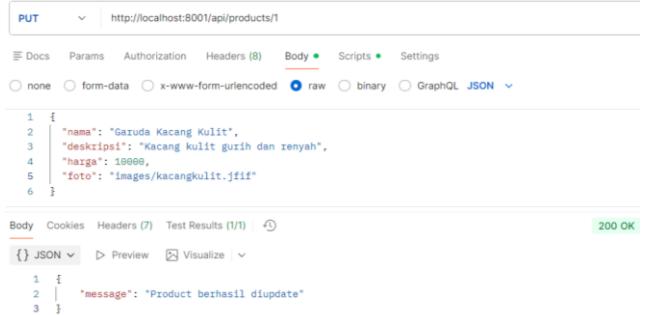
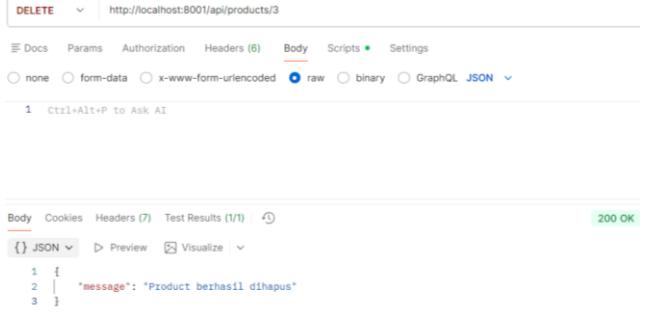
git remote add origin <https://github.com/agunghakase/Latihan9.git>

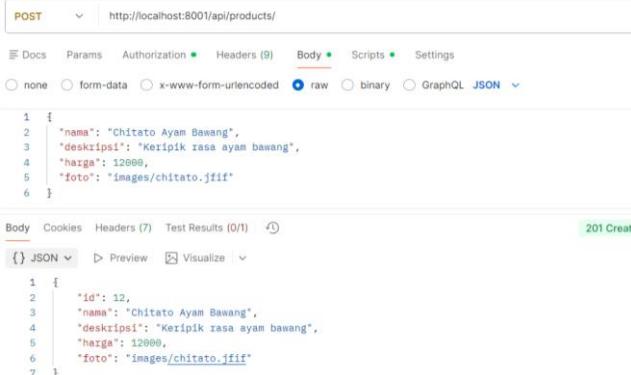
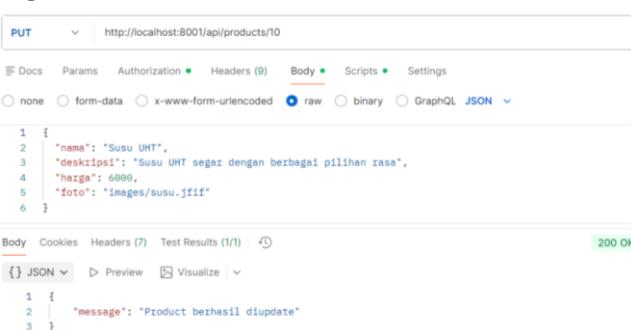
git push -u origin main

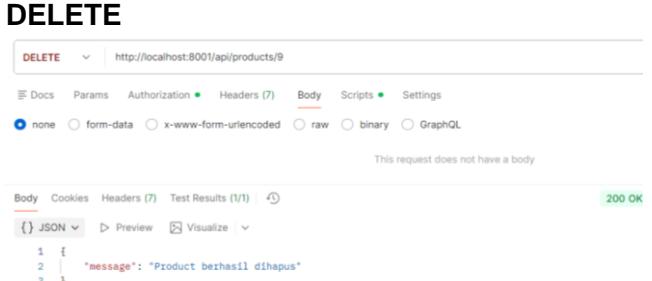
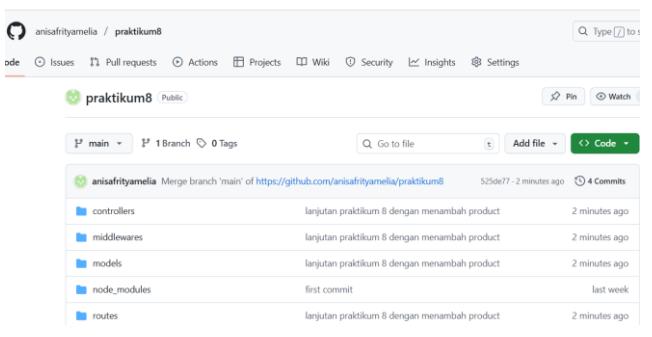
## Hasil Penggeraan

No	Instruksi	Screenshot	Kendala/ Saran																																																																		
A	<b>Menyediakan API Endpoints</b>																																																																				
1.	Pada database, tambahkan sebuah tabel products	<table border="1"> <thead> <tr> <th></th> <th>id</th> <th>nama</th> <th>deskripsi</th> <th>harga</th> <th>foto</th> </tr> </thead> <tbody> <tr><td>Ubah</td><td>1</td><td>Indomie Goreng</td><td>Mie goreng terenak</td><td>3000</td><td>images/indomie.jfif</td></tr> <tr><td>Ubah</td><td>2</td><td>Aqua Botol</td><td>Air mineral kemasan yang menyegarkan</td><td>6000</td><td>images/aqua.jfif</td></tr> <tr><td>Ubah</td><td>3</td><td>SilverQueen</td><td>Coklat dengan kacang premium</td><td>19000</td><td>images/coklat.jfif</td></tr> <tr><td>Ubah</td><td>4</td><td>Oreo</td><td>Oreo dengan varian rasa cocok untuk teman nonton</td><td>9000</td><td>images/oreo.jfif</td></tr> <tr><td>Ubah</td><td>5</td><td>Teh Pucuk</td><td>Minuman teh siap minum dengan aroma melati</td><td>4000</td><td>images/pucuk.jfif</td></tr> <tr><td>Ubah</td><td>6</td><td>Sunsil</td><td>Shampo anti ketombe</td><td>8000</td><td>images/shampo.jfif</td></tr> <tr><td>Ubah</td><td>7</td><td>Freshcare</td><td>Minyak angin aromatherapy</td><td>12000</td><td>images/freshcare.jfif</td></tr> <tr><td>Ubah</td><td>8</td><td>Kopi Kapal Api</td><td>Kopi hitam dengan aroma khas</td><td>2000</td><td>images/kopi.jfif</td></tr> <tr><td>Ubah</td><td>9</td><td>Sari Roti Tawar</td><td>Roti tawar lembut, cocok untuk sarapan</td><td>15000</td><td>images/roti.jfif</td></tr> <tr><td>Ubah</td><td>10</td><td>Roma Malistik</td><td>Biskuit malistik gurih dengan taburan gula</td><td>6000</td><td>images/malistik.jfif</td></tr> </tbody> </table>		id	nama	deskripsi	harga	foto	Ubah	1	Indomie Goreng	Mie goreng terenak	3000	images/indomie.jfif	Ubah	2	Aqua Botol	Air mineral kemasan yang menyegarkan	6000	images/aqua.jfif	Ubah	3	SilverQueen	Coklat dengan kacang premium	19000	images/coklat.jfif	Ubah	4	Oreo	Oreo dengan varian rasa cocok untuk teman nonton	9000	images/oreo.jfif	Ubah	5	Teh Pucuk	Minuman teh siap minum dengan aroma melati	4000	images/pucuk.jfif	Ubah	6	Sunsil	Shampo anti ketombe	8000	images/shampo.jfif	Ubah	7	Freshcare	Minyak angin aromatherapy	12000	images/freshcare.jfif	Ubah	8	Kopi Kapal Api	Kopi hitam dengan aroma khas	2000	images/kopi.jfif	Ubah	9	Sari Roti Tawar	Roti tawar lembut, cocok untuk sarapan	15000	images/roti.jfif	Ubah	10	Roma Malistik	Biskuit malistik gurih dengan taburan gula	6000	images/malistik.jfif	
	id	nama	deskripsi	harga	foto																																																																
Ubah	1	Indomie Goreng	Mie goreng terenak	3000	images/indomie.jfif																																																																
Ubah	2	Aqua Botol	Air mineral kemasan yang menyegarkan	6000	images/aqua.jfif																																																																
Ubah	3	SilverQueen	Coklat dengan kacang premium	19000	images/coklat.jfif																																																																
Ubah	4	Oreo	Oreo dengan varian rasa cocok untuk teman nonton	9000	images/oreo.jfif																																																																
Ubah	5	Teh Pucuk	Minuman teh siap minum dengan aroma melati	4000	images/pucuk.jfif																																																																
Ubah	6	Sunsil	Shampo anti ketombe	8000	images/shampo.jfif																																																																
Ubah	7	Freshcare	Minyak angin aromatherapy	12000	images/freshcare.jfif																																																																
Ubah	8	Kopi Kapal Api	Kopi hitam dengan aroma khas	2000	images/kopi.jfif																																																																
Ubah	9	Sari Roti Tawar	Roti tawar lembut, cocok untuk sarapan	15000	images/roti.jfif																																																																
Ubah	10	Roma Malistik	Biskuit malistik gurih dengan taburan gula	6000	images/malistik.jfif																																																																
2.	Di dalam folder routes buat file dengan nama product.routes.js dan ketik kode program seperti gambar disamping	<pre>API project8 &gt; routes &gt; JS product.routes.js &gt; ... 1 const express = require('express'); 2 const router = express.Router(); 3 const productController = require('../controllers/product.controller'); 4 5 router.get('/', productController.getAllProducts); 6 router.get('/:id', productController.getProductById); 7 router.post('/', productController.createProduct); 8 router.put('/:id', productController.updateProduct); 9 router.delete('/:id', productController.deleteProduct); 10 11 module.exports = router;</pre>																																																																			
3.	Buat file dalam folder controllers dengan nama product.controller.js dan ketik kode program seperti gambar disamping	<pre>project8 &gt; controllers &gt; JS product.controller.js &gt; ... const Product = require('../models/product.model');  exports.getAllProducts = (req, res) =&gt; {   Product.getAll((err, results) =&gt; {     if (err) return res.status(500).json({ error: err.message });     res.json(results);   }); };  exports.getProductById = (req, res) =&gt; {   const {id} = req.params;   Product.getById(id, (err, results) =&gt; {     if (err) return res.status(500).json({ error: err.message });     if (results.length == 0) return res.status(404).json({ message: 'Product tidak ditemukan' });     res.json(results[0]);   }); };  exports.createProduct = (req, res) =&gt; {   const data = req.body;   Product.create(data, (err, result) =&gt; {     if (err) return res.status(500).json({ error: err.message });     res.status(201).json({ id: result.insertId, ...data });   }); };</pre>																																																																			

		<pre> exports.updateProduct = (req, res) =&gt; {   const {id} = req.params;   const data = req.body;   Product.update(id, data, (err, result) =&gt; {     if (err) return res.status(500).json({ error: err.message });     if (result.affectedRows === 0) return res.status(404).json({ message: 'Product tidak ditemukan' });     res.json({ message: 'Product berhasil diupdate' });   }); };  exports.deleteProduct = (req, res) =&gt; {   const {id} = req.params;   Product.delete(id, (err, result) =&gt; {     if (err) return res.status(500).json({ error: err.message });     if (result.affectedRows === 0) return res.status(404).json({ message: 'Product tidak ditemukan' });     res.json({ message: 'Product berhasil dihapus' }); }); </pre>	
4.	Update file server.js dengan menambahkan kode seperti gambar disamping	<pre> project8 &gt; JS server.js &gt; ... 1  const express = require('express'); 2  const app = express(); 3  const PORT = 8001; 4 5  app.use(express.json()); 6 7  app.get('/', (req, res) =&gt; { 8    res.send('Hello, World'); 9  }); 10 11 // Routes 12 const userRoutes = require('./routes/user.routes'); 13 app.use('/api/users', userRoutes); 14 15 const productRoutes = require('./routes/product.routes'); 16 app.use('/api/products', productRoutes); 17 18 app.listen(PORT, () =&gt; { 19   console.log(`Server berjalan di \${PORT}`); 20 }); </pre>	
5.	Buat file untuk model product di dalam folder models dengan nama product.model.js	<pre> project8 &gt; JS product.model.js &gt; M Product 1  const db = require('../db.config'); 2 3  const Product = { 4    getAll: callback =&gt; { 5      db.query('SELECT * FROM products', callback); 6    }, 7 8    getById: (id, callback) =&gt; { 9      db.query('SELECT * FROM products WHERE id = ?', [id], callback); 10   }, 11 12    create: (data, callback) =&gt; { 13      db.query('INSERT INTO products (nama, deskripsi, harga, foto) VALUES (?, ?, ?, ?)', [data.nama, data.deskripsi, data.harga, data.foto], callback); 14    }, 15 16    update: (id, data, callback) =&gt; { 17      db.query('UPDATE products SET nama = ?, deskripsi = ?, harga = ?, foto = ? WHERE id = ?', [data.nama, data.deskripsi, data.harga, data.foto, id], callback); 18    }, 19 20    delete: (id, callback) =&gt; { 21      db.query('DELETE FROM products WHERE id = ?', [id], callback); 22    } 23  }; 24 25 module.exports = Product; </pre>	
6.	Jalankan perintah node server.js dan gunakan postman untuk mendapatkan data getAll products dengan mengunjungi endpoints /api/products	 <pre> {   "id": 1,   "nama": "Indomie Goreng",   "deskripsi": "Mie goreng terenak",   "harga": 3000,   "foto": "images/indomie.jfif" }, {   "id": 2,   "nama": "Aqua Botol",   "deskripsi": "Air mineral kemasan yang menyegarkan",   "harga": 6000,   "foto": "images/aqua.jfif" }, {   "id": 3,   "nama": "SilverQueen",   "deskripsi": "Coklat dengan kacang premium",   "harga": 19000,   "foto": "images/coklat.jfif" }, {   "id": 4, } </pre>	

7.	Dengan menggunakan postman lakukan pengujian pada endpoint /api/products/2 (method: GET)		
8.	Dengan menggunakan postman lakukan pengujian pada endpoint /api/products (method: POST)		
9.	Dengan menggunakan postman lakukan pengujian pada endpoint /api/products/1 (method: PUT)		
10.	Dengan menggunakan postman lakukan pengujian pada endpoint /api/products/3 (method: DELETE)		
B	Menambahkan Proteksi API (Simple - Bearer Method)		

1.	<p>Buat sebuah folder bernama middlewares, kemudian di dalamnya buat file dengan nama auth.middleware.js dan ketik kode program seperti gambar disamping</p>	<pre>!project8 &gt; middlewares &gt; <b>JS</b> auth.middleware.js &gt; authBearer 1 const authBearer = (req, res, next) =&gt; { 2   const authHeader = req.headers.authorization; 3 4   if (!authHeader) { 5     return res.status(401).json({ message: "No authorization header" }); 6   } 7 8   if (!authHeader.startsWith("Bearer ")) { 9     return res.status(401).json({ message: "Bearer token required" }); 10  } 11 12  const token = authHeader.split(" ")[1]; 13 14  const VALID_TOKEN = "12345TOKENRAHASIA"; 15 16  if (token !== VALID_TOKEN) { 17    return res.status(403).json({ message: "Invalid token" }); 18  } 19 20  next(); 21 } 22 23 module.exports = { authBearer };</pre>	
2.	<p>Pada file product.routes.js panggil auth.middleware.js, lalu tambahkan authBearer ke endpoints yang perlu di proteksi, tambahkan kode program seperti pada gambar disamping</p>	<pre>!project8 &gt; routes &gt; <b>JS</b> product.routes.js &gt; ... 1 const express = require('express'); 2 const router = express.Router(); 3 const productController = require('../controllers/product.controller'); 4 const { authBearer } = require('../middlewares/auth.middleware'); 5 6 router.get('/', productController.getAllProducts); 7 router.get('/:id', productController.getProductById); 8 router.post('/', authBearer, productController.createProduct); 9 router.put('/:id', authBearer, productController.updateProduct); 10 router.delete('/:id', authBearer, productController.deleteProduct); 11 12 module.exports = router;</pre>	
3.	<p>Gunakan postman untuk akses 3 endpoints (POST, PUT, DELETE) dengan menambahkan bearer <b>12345TOKENRAHASIA</b></p>	<p><b>POST</b></p>  <p><b>PUT</b></p> 	

			
C	<b>Github + Visual Code</b>		
1.	Membuat repository baru di Github dan commit proyek ke dalam Github		
2.	Link Github	<a href="https://github.com/anisafrityamelia/praktikum8.git">https://github.com/anisafrityamelia/praktikum8.git</a>	