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.

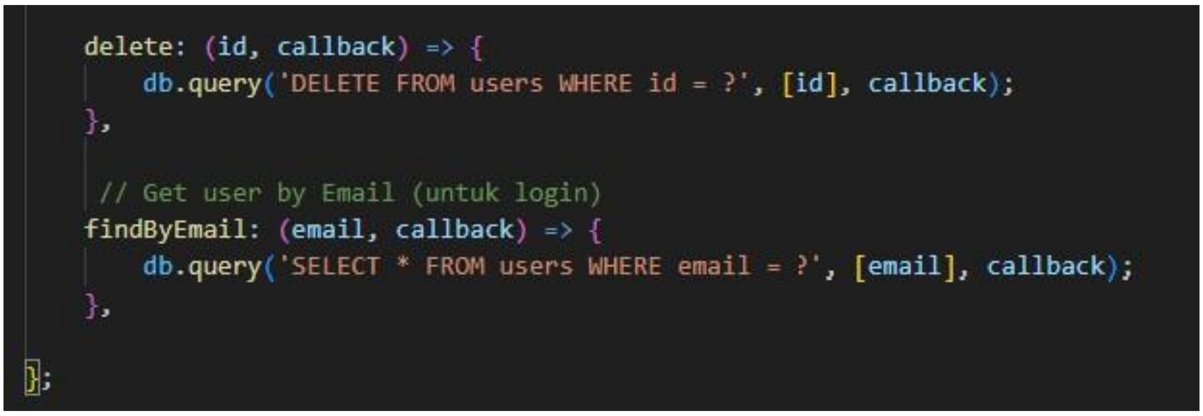
A. Membuat JSON Web Token (Dynamic Bearer Token)

1. Lanjutkan Project Praktikum 8-9, dengan menggunakan file yang sama (copy)
2. Install library JWT
   npm install jsonwebtoken bcryptjs

3. Tambahkan file auth.controller.js, auth.middleware.js, dan auth.routes.js
4. Buat file .env disamping server.js (root folder)
   Isi file .env dengan variable sebagai berikut:
   JWT_SECRET=”KUNCI-RAHASIA”
   JWT_EXPIRE=1d

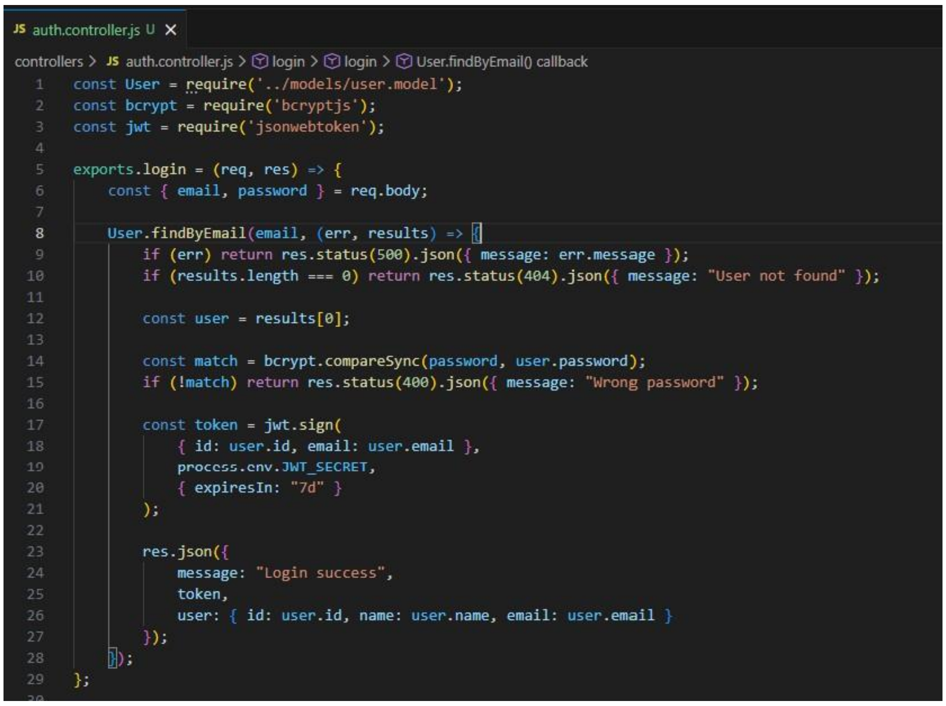
5. Tambahkan script berikut di server.js
   ```
   require('dotenv').config();
   ```

6. Revisi model sebelumnya pada user.model.js dengan menambahkan fungsi baru seperti berikut, tambahkan findByEmail



```
    delete: (id, callback) => {
        db.query('DELETE FROM users WHERE id = ?', [id], callback);
    },

    // Get user by Email (untuk login)
    findByEmail: (email, callback) => {
        db.query('SELECT * FROM users WHERE email = ?', [email], callback);
    },

};
```

7. Masukan script berikut pada auth.controller.js yang telah dibuat

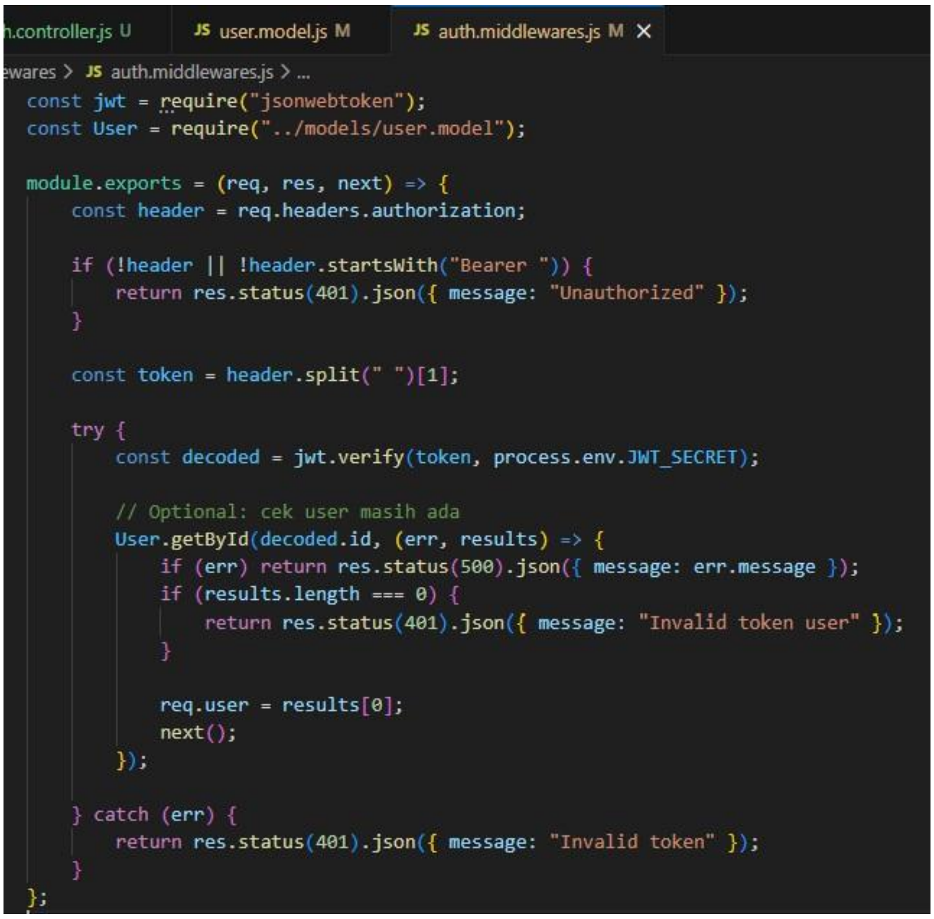


JS auth.controller.js U ×

controllers > JS auth.controller.js > login > login > User.findByEmail() callback

```javascript
const User = require('../models/user.model');
const bcrypt = require('bcryptjs');
const jwt = require('jsonwebtoken');

exports.login = (req, res) => {
    const { email, password } = req.body;

    User.findByEmail(email, (err, results) => {
        if (err) return res.status(500).json({ message: err.message });
        if (results.length === 0) return res.status(404).json({ message: "User not found" });

        const user = results[0];

        const match = bcrypt.compareSync(password, user.password);
        if (!match) return res.status(400).json({ message: "Wrong password" });

        const token = jwt.sign(
            { id: user.id, email: user.email },
            process.env.JWT_SECRET,
            { expiresIn: "7d" }
        );

        res.json({
            message: "Login success",
            token,
            user: { id: user.id, name: user.name, email: user.email }
        });
    });
};
```

8. Ubah auth.middleware.js yang sebelumnya menggunakan token biasa, menjadi json web token seperti gambar dibawah ini

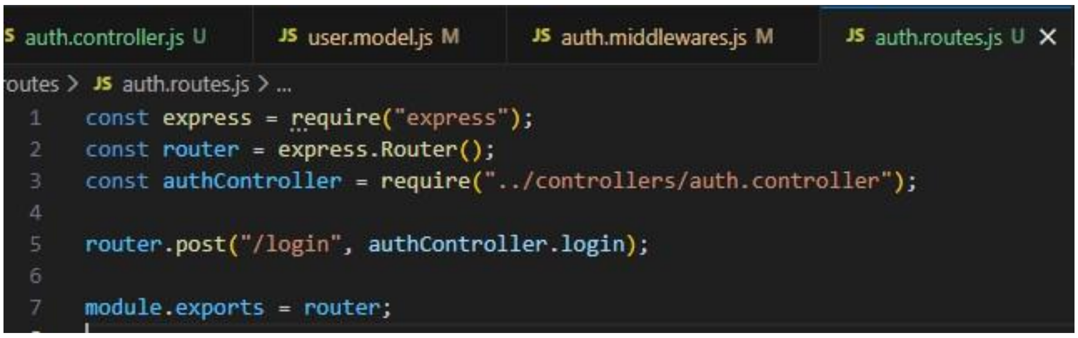


h.controller.js U   JS user.model.js M   JS auth.middlewares.js M ×

ewares > JS auth.middlewares.js > ...

```javascript
const jwt = require("jsonwebtoken");
const User = require("../models/user.model");

module.exports = (req, res, next) => {
    const header = req.headers.authorization;

    if (!header || !header.startsWith("Bearer ")) {
        return res.status(401).json({ message: "Unauthorized" });
    }

    const token = header.split(" ")[1];

    try {
        const decoded = jwt.verify(token, process.env.JWT_SECRET);

        // Optional: cek user masih ada
        User.getById(decoded.id, (err, results) => {
            if (err) return res.status(500).json({ message: err.message });
            if (results.length === 0) {
                return res.status(401).json({ message: "Invalid token user" });
            }

            req.user = results[0];
            next();
        });

    } catch (err) {
        return res.status(401).json({ message: "Invalid token" });
    }
};
```

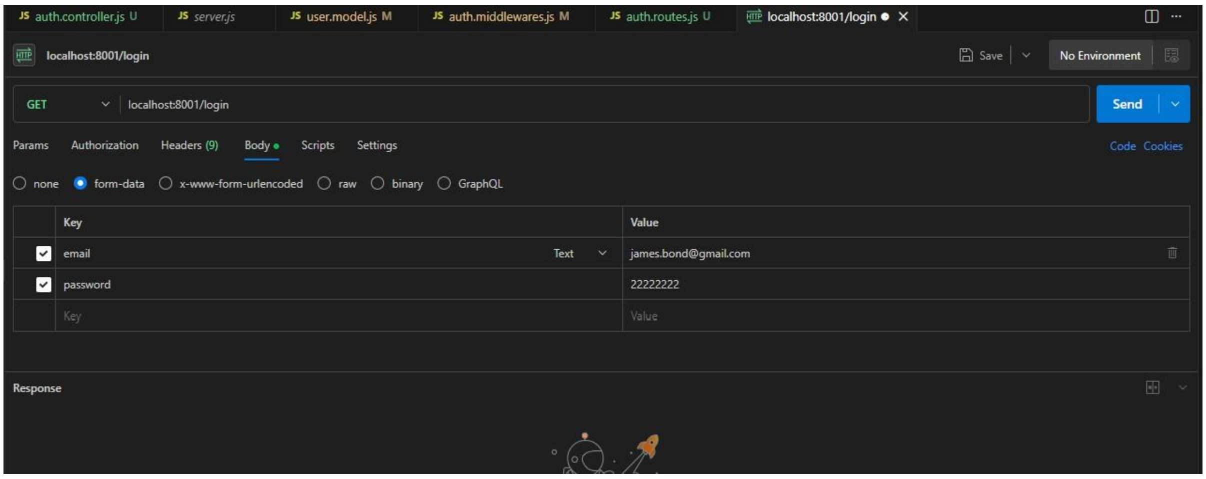
9. Tambahkan Routes untuk mengakses login pada auth.routes.js



```javascript
const express = require("express");
const router = express.Router();
const authController = require("../controllers/auth.controller");

router.post("/login", authController.login);

module.exports = router;
```

B. Gunakan POSTMAN dapatkan Token BEARER

1. Install postman di visual code, dan lakukan login berdasarkan email dan password yang terdaftar di database
2. Dapatkan bearer dengan memanggil API endpoints /login



GET localhost:8001/login

Body: form-data

Key	Value
email	james.bond@gmail.com
password	22222222

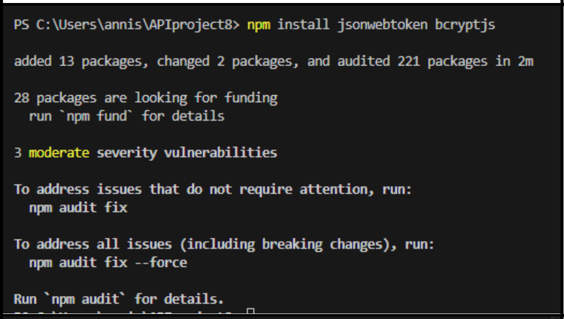
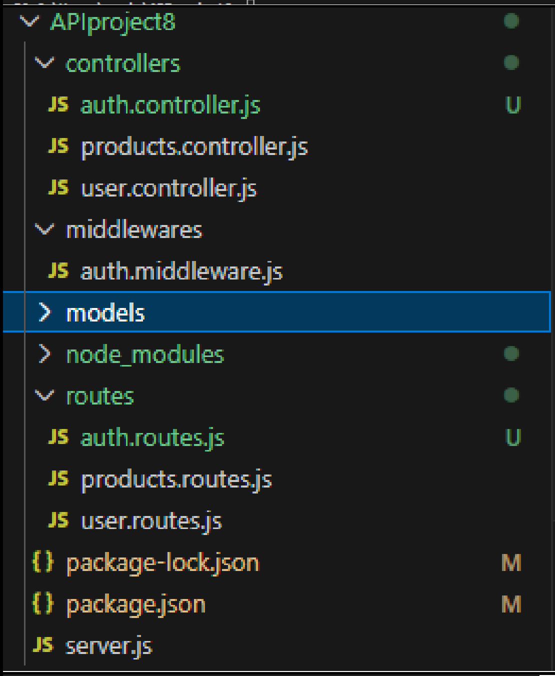
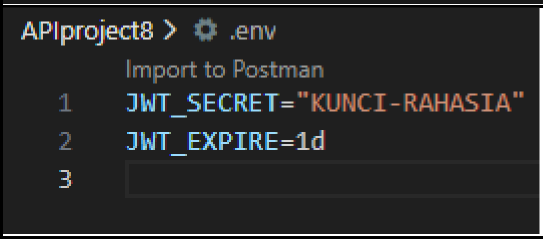
3. Catat bearer yang di dapatkan, lalu gunakan bearer tersebut untuk memanggil endpoints lainya yang pada praktikum 9 telah di proteksi.

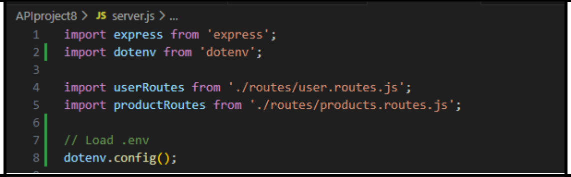
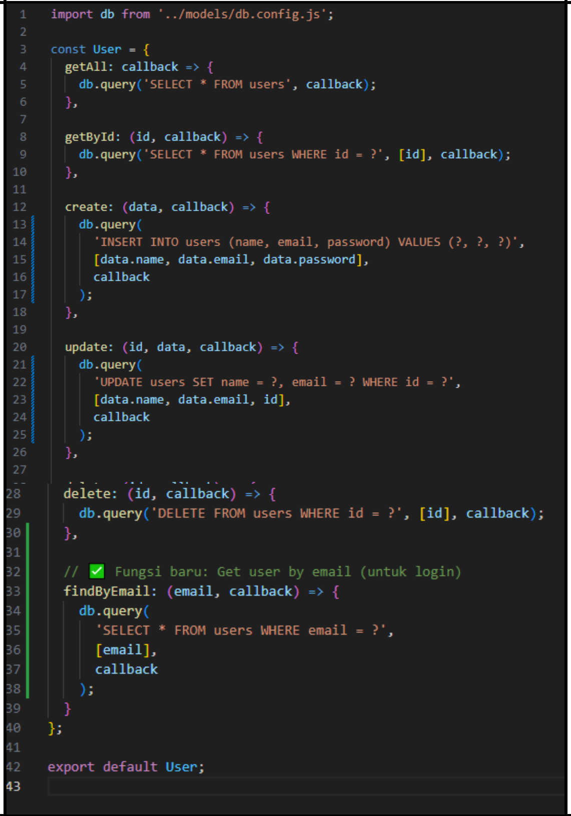
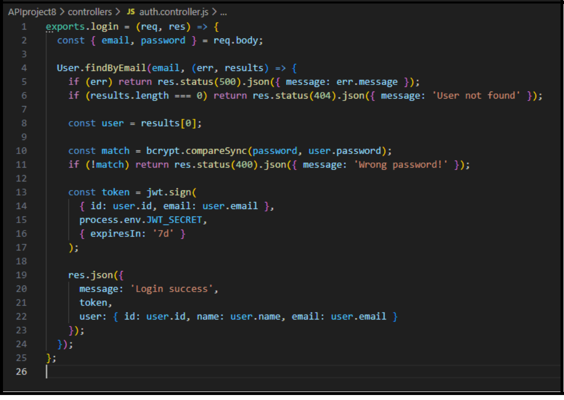
F. Github + Visual Code

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.	**Membuat JSON Web Token (Dynamic Bearer Token)**		
1.	1. Lanjutkan Project Praktikum 8-9, dengan menggunakan file yang sama (copy) 2. Install library JWT npm install jsonwebtoken bcryptjs	 PS C:\Users\annis\APIproject8> npm install jsonwebtoken bcryptjs added 13 packages, changed 2 packages, and audited 221 packages in 2m 28 packages are looking for funding run `npm fund` for details 3 moderate severity vulnerabilities To address issues that do not require attention, run: npm audit fix To address all issues (including breaking changes), run: npm audit fix --force Run `npm audit` for details.	-
2.	3. Tambahkan file auth.controller.js, auth.middleware.js, dan auth.routes.js	 APIproject8 controllers auth.controller.js U products.controller.js user.controller.js middlewares auth.middleware.js models node_modules routes auth.routes.js U products.routes.js user.routes.js package-lock.json M package.json M server.js	-
3.	4. Buat file .env disamping server.js (root folder) Isi file .env dengan variable sebagai berikut: JWT_SECRET="KUNCI-RAHASIA"	 APIproject8 > .env Import to Postman JWT_SECRET="KUNCI-RAHASIA" JWT_EXPIRE=1d	-

	JWT_EXPIRE=1d		
4.	5. Tambahkan script berikut di server.js require('dotenv').config();		-
5.	6. Revisi model sebelumnya pada user.model.js dengan menambahkan fungsi baru seperti berikut, tambahkan findByEmail		-
6.	7. Masukan script berikut pada auth.controller.js yang telah dibuat		-

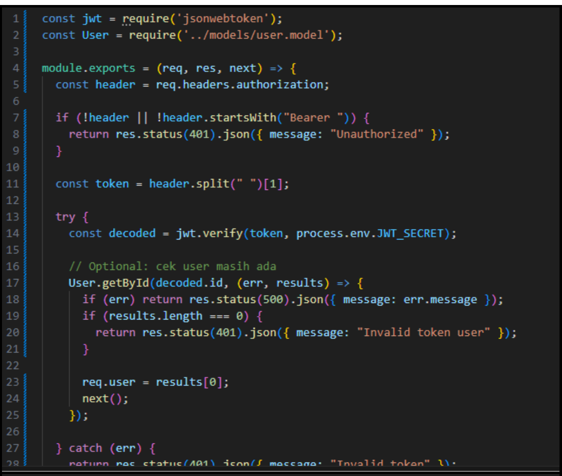
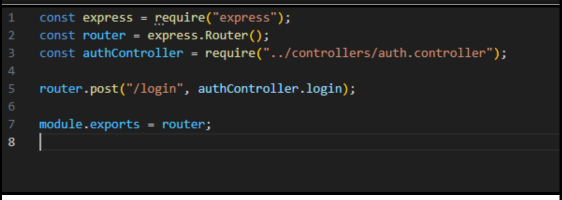
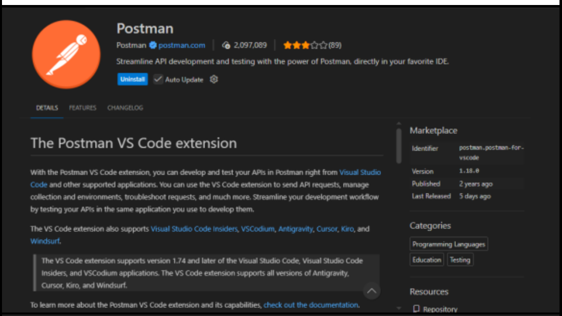
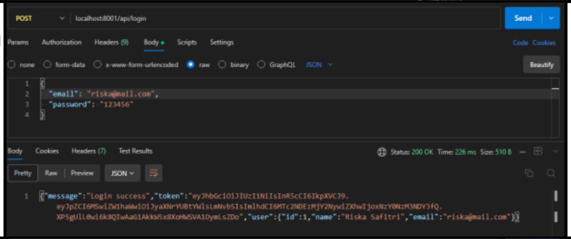
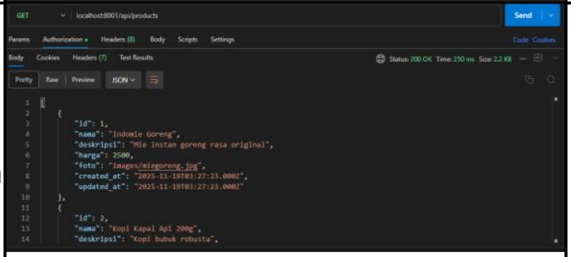
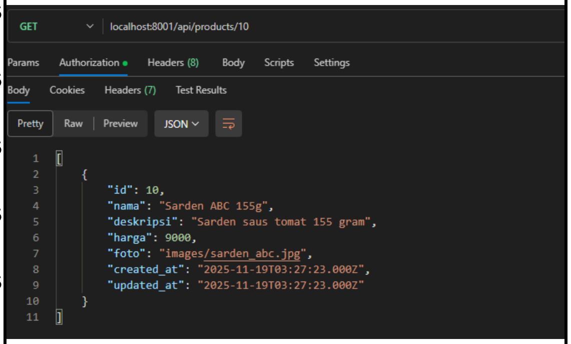
APIproject8 > JS server.js > ...

```js
import express from 'express';
import dotenv from 'dotenv';

import userRoutes from './routes/user.routes.js';
import productRoutes from './routes/products.routes.js';

// Load .env
dotenv.config();
```

```js
import db from '../models/db.config.js';

const User = {
  getAll: callback => {
    db.query('SELECT * FROM users', callback);
  },

  getById: (id, callback) => {
    db.query('SELECT * FROM users WHERE id = ?', [id], callback);
  },

  create: (data, callback) => {
    db.query(
      'INSERT INTO users (name, email, password) VALUES (?, ?, ?)',
      [data.name, data.email, data.password],
      callback
    );
  },

  update: (id, data, callback) => {
    db.query(
      'UPDATE users SET name = ?, email = ? WHERE id = ?',
      [data.name, data.email, id],
      callback
    );
  },

  delete: (id, callback) => {
    db.query('DELETE FROM users WHERE id = ?', [id], callback);
  },

  // ✅ Fungsi baru: Get user by email (untuk login)
  findByEmail: (email, callback) => {
    db.query(
      'SELECT * FROM users WHERE email = ?',
      [email],
      callback
    );
  }
};

export default User;
```

APIproject8 > controllers > JS auth.controller.js > ...

```js
exports.login = (req, res) => {
  const { email, password } = req.body;

  User.findByEmail(email, (err, results) => {
    if (err) return res.status(500).json({ message: err.message });
    if (results.length === 0) return res.status(404).json({ message: 'User not found' });

    const user = results[0];

    const match = bcrypt.compareSync(password, user.password);
    if (!match) return res.status(400).json({ message: 'Wrong password!' });

    const token = jwt.sign(
      { id: user.id, email: user.email },
      process.env.JWT_SECRET,
      { expiresIn: '7d' }
    );

    res.json({
      message: 'Login success',
      token,
      user: { id: user.id, name: user.name, email: user.email }
    });
  });
};
```

7.	8. Ubah auth.middleware.js yang sebelumnya menggunakan token biasa, menjadi json web token seperti gambar dibawah ini		-
8.	9. Tambahkan Routes untuk mengakses login pada auth.routes.js		-
B.	**Gunakan POSTMAN dapatkan Token BEARER**		
1.	1. Install postman di visual code, dan lakukan login berdasarkan email dan password yang terdaftar di database		-
2.	2. Dapatkan bearer dengan memanggil API endpoints /login		
3.	3. Catat bearer yang di dapatkan, lalu gunakan bearer tersebut untuk memanggil endpoints lainya yang pada praktikum 9 telah di proteksi. 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	 	

```js
const jwt = require('jsonwebtoken');
const User = require('../models/user.model');

module.exports = (req, res, next) => {
  const header = req.headers.authorization;

  if (!header || !header.startsWith("Bearer ")) {
    return res.status(401).json({ message: "Unauthorized" });
  }

  const token = header.split(" ")[1];

  try {
    const decoded = jwt.verify(token, process.env.JWT_SECRET);

    // Optional: cek user masih ada
    User.getById(decoded.id, (err, results) => {
      if (err) return res.status(500).json({ message: err.message });
      if (results.length === 0) {
        return res.status(401).json({ message: "Invalid token user" });
      }

      req.user = results[0];
      next();
    });

  } catch (err) {
    return res.status(401).json({ message: "Invalid token" });
```

```js
const express = require("express");
const router = express.Router();
const authController = require("../controllers/auth.controller");

router.post("/login", authController.login);

module.exports = router;
```

Postman
Postman postman.com | 2,097,089 | ★★★☆☆ (89)
Streamline API development and testing with the power of Postman, directly in your favorite IDE.
Uninstall ✓ Auto Update

DETAILS FEATURES CHANGELOG

The Postman VS Code extension

With the Postman VS Code extension, you can develop and test your APIs in Postman right from Visual Studio Code and other supported applications. You can use the VS Code extension to send API requests, manage collection and environments, troubleshoot requests, and much more. Streamline your development workflow by testing your APIs in the same application you use to develop them.

The VS Code extension also supports Visual Studio Code Insiders, VSCodium, Antigravity, Cursor, Kiro, and Windsurf.

> The VS Code extension supports version 1.74 and later of the Visual Studio Code, Visual Studio Code Insiders, and VSCodium applications. The VS Code extension supports all versions of Antigravity, Cursor, Kiro, and Windsurf.

To learn more about the Postman VS Code extension and its capabilities, check out the documentation.

Marketplace

Identifier	postman.postman-for-vscode
Version	1.18.0
Published	2 years ago
Last Released	5 days ago

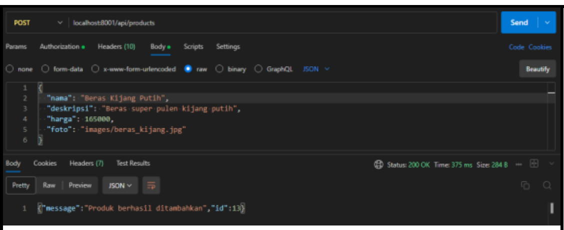
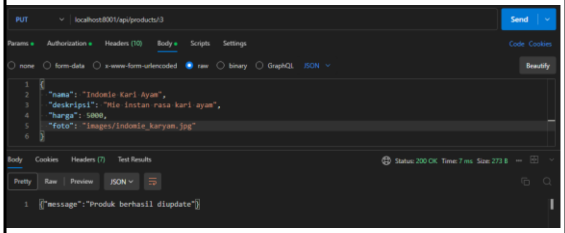
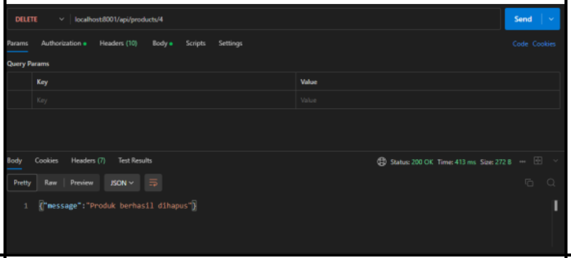
Categories: Programming Languages, Education, Testing

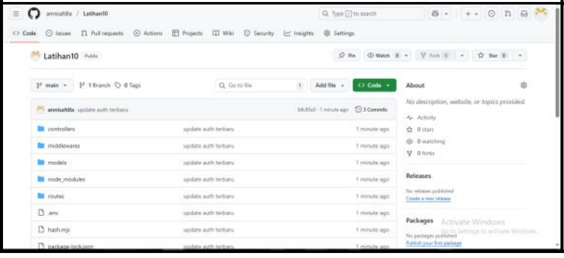
Resources: Repository

```
POST  localhost:8001/api/login
Body: raw JSON
{
  "email": "riska@mail.com",
  "password": "123456"
}

Status: 200 OK  Time: 226 ms  Size: 510 B
{"message":"Login success","token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJpZCI6MSwiZW1haWwiOiJyaXNrYUBtYWlsLmNvbSIsImlhdCI6MTc2NDEzMjY2NywiZXhwIjoxNzY0NzM3NDY3fQ.
XP5gUlL0wi6k8QIwAaG1AkkW5x8XoHW5VA1OymLsZDo","user":{"id":1,"name":"Riska Safitri","email":"riska@mail.com"}}
```

```
GET  localhost:8001/api/products
Authorization
Status: 200 OK  Time: 250 ms  Size: 2.2 KB
[
    {
        "id": 1,
        "nama": "Indomie Goreng",
        "deskripsi": "Mie instan goreng rasa original",
        "harga": 2500,
        "foto": "images/miegoreng.jpg",
        "created_at": "2025-11-19T03:27:23.000Z",
        "updated_at": "2025-11-19T03:27:23.000Z"
    },
    {
        "id": 2,
        "nama": "Kopi Kapal Api 200g",
        "deskripsi": "Kopi bubuk robusta",
```

```
GET  localhost:8001/api/products/10
Authorization
[
    {
        "id": 10,
        "nama": "Sarden ABC 155g",
        "deskripsi": "Sarden saus tomat 155 gram",
        "harga": 9000,
        "foto": "images/sarden_abc.jpg",
        "created_at": "2025-11-19T03:27:23.000Z",
        "updated_at": "2025-11-19T03:27:23.000Z"
    }
]
```

		POST localhost:8001/api/products
		Body: raw JSON
		`{`
		`"nama": "Beras Kijang Putih",`
		`"deskripsi": "Beras super pulen kijang putih",`
		`"harga": 165000,`
		`"foto": "images/beras_kijang.jpg"`
		`}`
		Status: 200 OK Time: 375 ms Size: 284 B
		`{"message":"Produk berhasil ditambahkan","id":13}`
		PUT localhost:8001/api/products/:3
		Body: raw JSON
		`{`
		`"nama": "Indomie Kari Ayam",`
		`"deskripsi": "Mie instan rasa kari ayam",`
		`"harga": 5000,`
		`"foto": "images/indomie_karyam.jpg"`
		`}`
		Status: 200 OK Time: 7 ms Size: 273 B
		`{"message":"Produk berhasil diupdate"}`
		DELETE localhost:8001/api/products/4
		Query Params: Key / Value
		Status: 200 OK Time: 413 ms Size: 272 B
		`{"message":"Produk berhasil dihapus"}`	

C.	**Github + Visual Code**		
1.	1. Buat proyek di Github dengan nama Latihan10	annisafdla / Latihan10 Latihan10 Public main · 1 Branch · 0 Tags · 3 Commits annisafdla update auth terbaru · 1 minute ago controllers — update auth terbaru — 1 minute ago middlewares — update auth terbaru — 1 minute ago models — update auth terbaru — 1 minute ago node_modules — update auth terbaru — 1 minute ago routes — update auth terbaru — 1 minute ago .env — update auth terbaru — 1 minute ago hash.mjs — update auth terbaru — 1 minute ago package-lock.json — update auth terbaru — 1 minute ago About: No description, website, or topics provided. 0 stars · 0 watching · 0 forks Releases: No releases published Packages: No packages published	-