

Laporan Praktikum
Mata Kuliah PEMWEB dan PBO



JWT Web Token

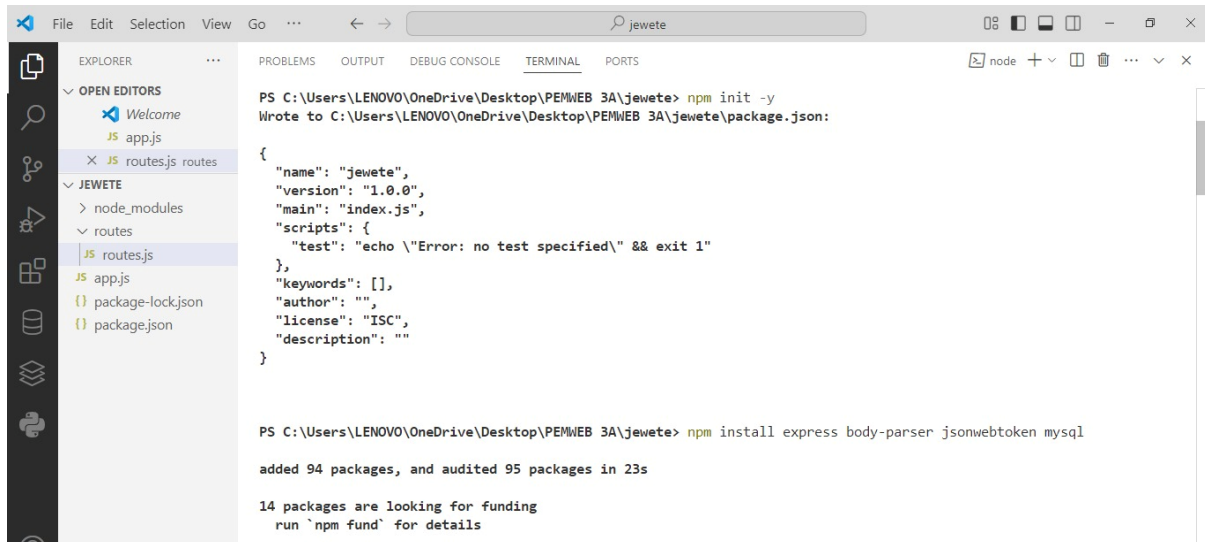
Dosen Pengampu : Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh:
Fauziah Salsabila (2307311)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

Langkah-langkah JWT Web Token

1. Lakukan inisialisasi node.js pada terminal, Ini akan membuat file package.json dengan konfigurasi default. Selanjutnya lakukan paket instalasi sesuai dengan kebutuhan yang di perlukan menggunakan npm.



The screenshot shows the VS Code interface with the Explorer, Search, and Run and Debug views on the left. The Explorer view shows the file structure of the 'JEWETE' project, including 'node_modules', 'routes', 'app.js', 'package-lock.json', and 'package.json'. The Search view shows the search results for 'routes.js' and 'routes'. The Run and Debug view shows the terminal output of the 'npm init -y' command, which created the 'package.json' file. The terminal output shows the command 'npm init -y' and the resulting 'package.json' file content. The 'package.json' file content is as follows:

```
{
  "name": "jewete",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

The terminal also shows the command 'npm install express body-parser jsonwebtoken mysql' and the output 'added 94 packages, and audited 95 packages in 23s'.

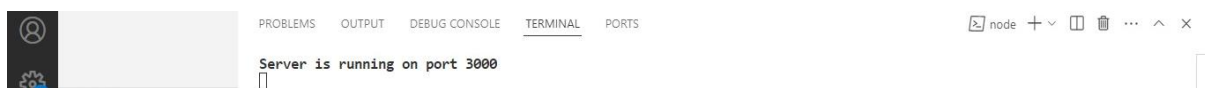
2. Langkah berikutnya membuat file utama yaitu app.js yang berfungsi untuk mengatur konfigurasi awal server, middleware, koneksi database, dan routing global.



The screenshot shows the VS Code interface with the Explorer, Search, and Run and Debug views on the left. The Explorer view shows the file structure of the 'JEWETE' project, including 'node_modules', 'routes', 'app.js', 'package-lock.json', and 'package.json'. The Search view shows the search results for 'app.js' and 'app.js'. The Run and Debug view shows the code in 'app.js'.

```
1 const express = require("express");
2 const bodyParser = require("body-parser");
3 const routes = require("./routes/routes");
4
5 const app = express();
6
7 app.use(bodyParser.json());
8 app.use("/", routes);
9
10 app.listen(3000, () => {
11   console.log("Server is running on port 3000");
12 });
```

3. Berikut merupakan jika server berhasil dijalankan, maka pada terminal akan bertuliskan "Server is running on port 3000"



The screenshot shows the terminal output of the 'npm start' command, which is 'Server is running on port 3000'.

4. Langkah selanjutnya membuat file routes.js yang berfungsi untuk mendefinisikan endpoint API secara modular, memisahkan logika routing dari app.js sehingga rute dalam sebuah aplikasi/web agar kode lebih terorganisir atau rapih. Pada tampilan gambar dibawah ini kita sedang membuat web token untuk bagian login.

