

LAPORAN

PRATIKUM PEMROGRAMAN JARINGAN



Dosen Pengampu:
Ade Kurniawan, S.Pd., M.Pd.T

Disusun Oleh:
Najwa Alawiyah Siregar
(22346040)

PRODI INFORMATIKA
FAKULTAS ELEKRONIKA
UNIVERSITAS NEGERI
PADANGTAHUN AJARAN

A. Web Server

Server web (web server) adalah entitas berupa komputer fisik atau perangkat lunak yang berfungsi sebagai pusat penyimpanan dan pengiriman data dalam bentuk halaman web kepada pengguna yang meminta akses. Ketika seorang pengguna mengakses suatu situs web, permintaannya diteruskan ke server web yang bersangkutan. Selanjutnya, server web ini mengolah permintaan tersebut dan mengirimkan halaman web yang sesuai ke perangkat pengguna melalui jaringan internet. Istilah web server dapat merujuk pada perangkat keras atau perangkat lunak, atau keduanya yang saling bekerja bersama.

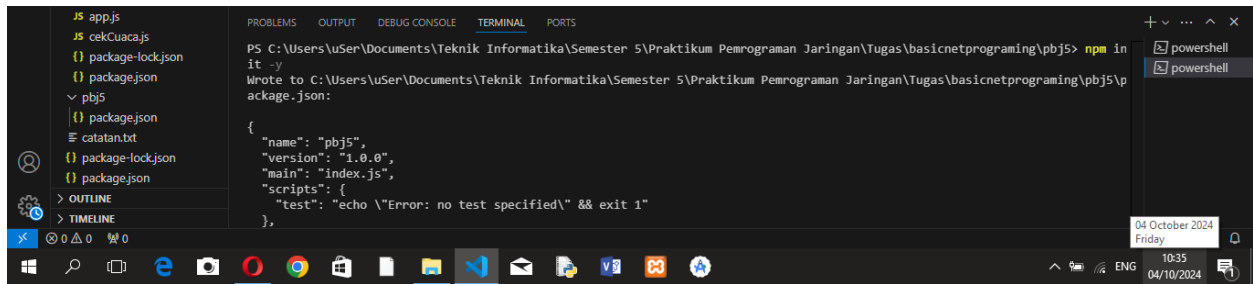
B. ExpressJS

Express.js, atau sering disebut hanya "Express," adalah framework web yang berjalan di atas bahasa pemrograman JavaScript. Ini merupakan salah satu framework yang paling populer dan umum digunakan untuk pengembangan aplikasi web di sisi server (server-side) dengan menggunakan bahasa JavaScript.

Express.js dan Node.js memiliki hubungan erat, di mana Express.js adalah sebuah framework web yang dibangun di atas platform Node.js. Ini berarti Express.js adalah salah satu dari banyak framework yang dapat digunakan untuk mengembangkan aplikasi web dengan menggunakan Node.js sebagai lingkungan runtime-nya. Node.js adalah platform runtime JavaScript yang memungkinkan Anda menjalankan kode JavaScript di sisi server (server-side). Ini sangat efisien untuk menangani banyak permintaan I/O yang bersifat non-blocking, yang merupakan fitur kunci dalam pengembangan aplikasi web yang skalabel dan responsif. Express.js, di sisi lain, adalah framework yang dirancang khusus untuk memudahkan pengembangan aplikasi web dengan menggunakan Node.js. Ini menyediakan alat dan fitur tingkat tinggi untuk mengatur permintaan HTTP, menangani rute, mengelola sesi, dan banyak lagi. Express.js memungkinkan pengembang untuk membangun aplikasi web dengan cepat dan efisien dengan menggunakan Node.js sebagai basisnya. Pada tabel berikut dapat dilihat perbedaan express.js dan node.js

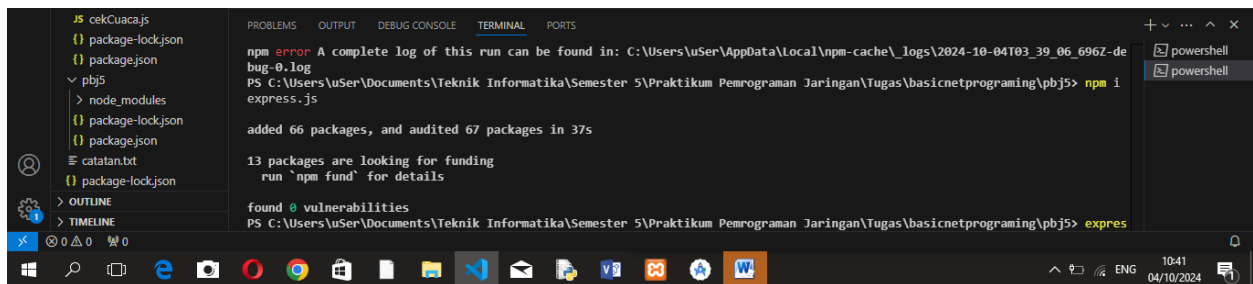
C. Latihan

- a. Instalasi Express.js dan membuat halaman menggunakan fungsi `app.get`
 1. Membuat folder baru pada visual studio code dengan nama 'PBJ5'
 2. Buka terminal pada direktori tersebut dan ketik kode `'npm init -y'` untuk *generate* package.json



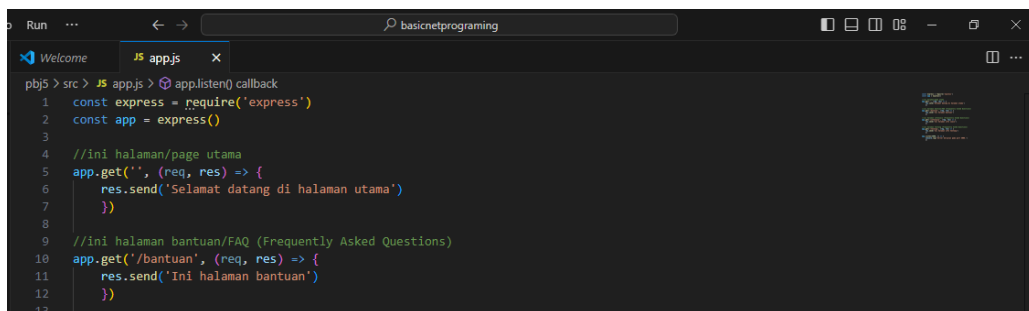
(Gambar a1. Melakukan generate package.json)

3. Lakukan instalasi express.js menggunakan perintah npm



(Gambar a2. Instalasi express.js)

4. Dapat dilihat pada gambar a2, node_module dan file baru package.json telah muncul.
5. Selanjutnya, membuat 2 folder baru pada folder baru dengan nama 'public' dan 'src'.
6. Di dalam folder src, buat file baru dengan nama 'app.js'
7. Ketikkan kode berikut ini



(Gambar a3. Kode membuat halaman utama dan bantuan)

8. Di bawah ini merupakan kodingan lanjutan dari kodingan sebelumnya untuk membuat halaman 'tentang' dan 'infoCuaca'.

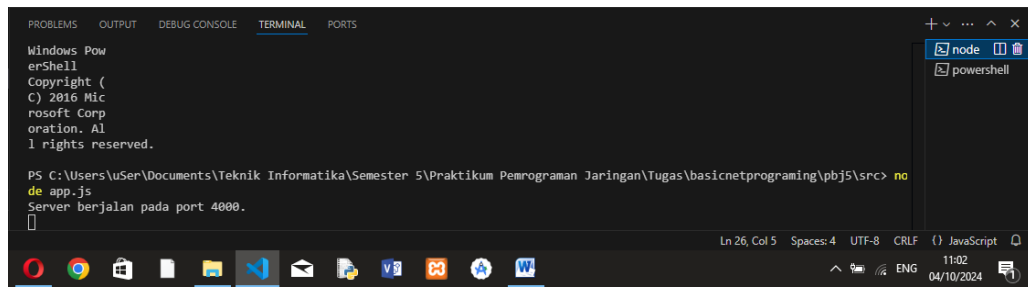
```

13
14 //ini halaman infoCuaca (Frequently Asked Questions)
15 app.get('/infoCuaca', (req, res) => {
16     res.send('Ini halaman info cuaca')
17 })
18
19 //ini halaman tentang (Frequently Asked Questions)
20 app.get('/tentang', (req, res) => {
21     res.send('Ini halaman info tentang')
22 })
23
24 app.listen(4000, () => {
25     console.log('Server berjalan pada port 4000.')
26 })
27
28

```

(Gambar a4. Kode membuat halaman tentang dan infoCuaca)

9. Buka terminal pada deriktori src dan ketik perintah 'node.js' untuk menjalankan program.



```

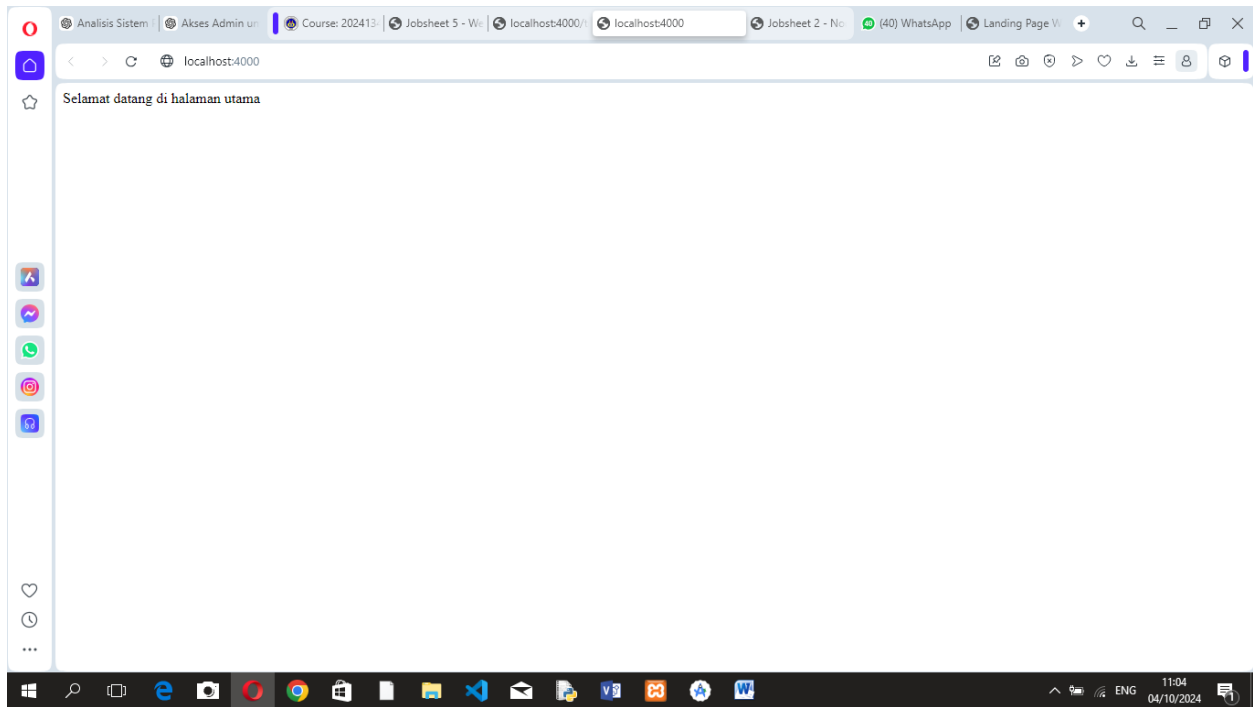
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\user\Documents\Teknik Informatika\Semester 5\Praktikum Pemrograman Jaringan\Tugas\basicnetprogramming\pbjs\src> node app.js
Server berjalan pada port 4000.

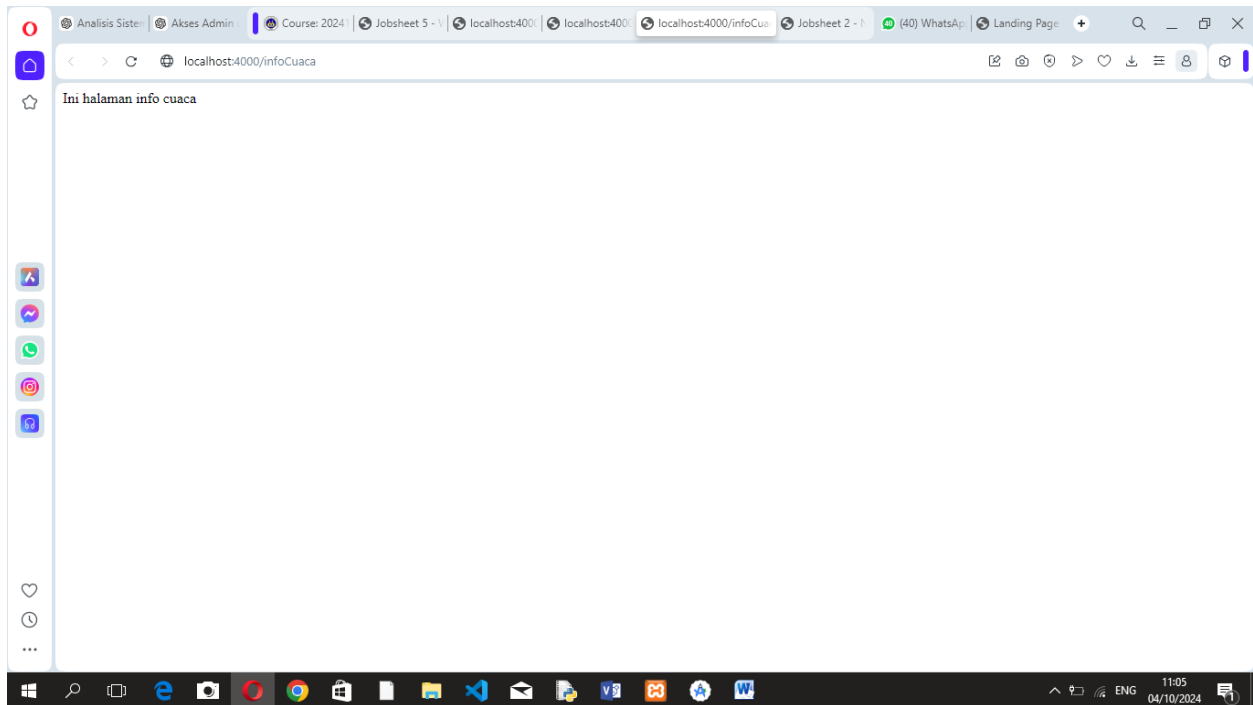
```

(Gambar a5. Output program pada terminal)

10. Buka link ini <http://localhost:4000/>



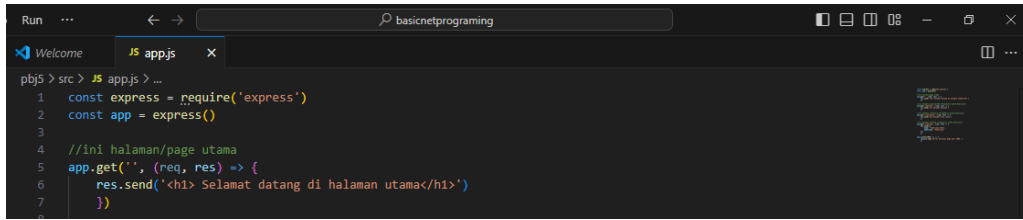
(Gambar a6. Output program pada web untuk halaman utama)



(Gambar a7. Output program pada web untuk halaman infoCuaca)

b. Integrasi HTML dan JSON pada Express.js

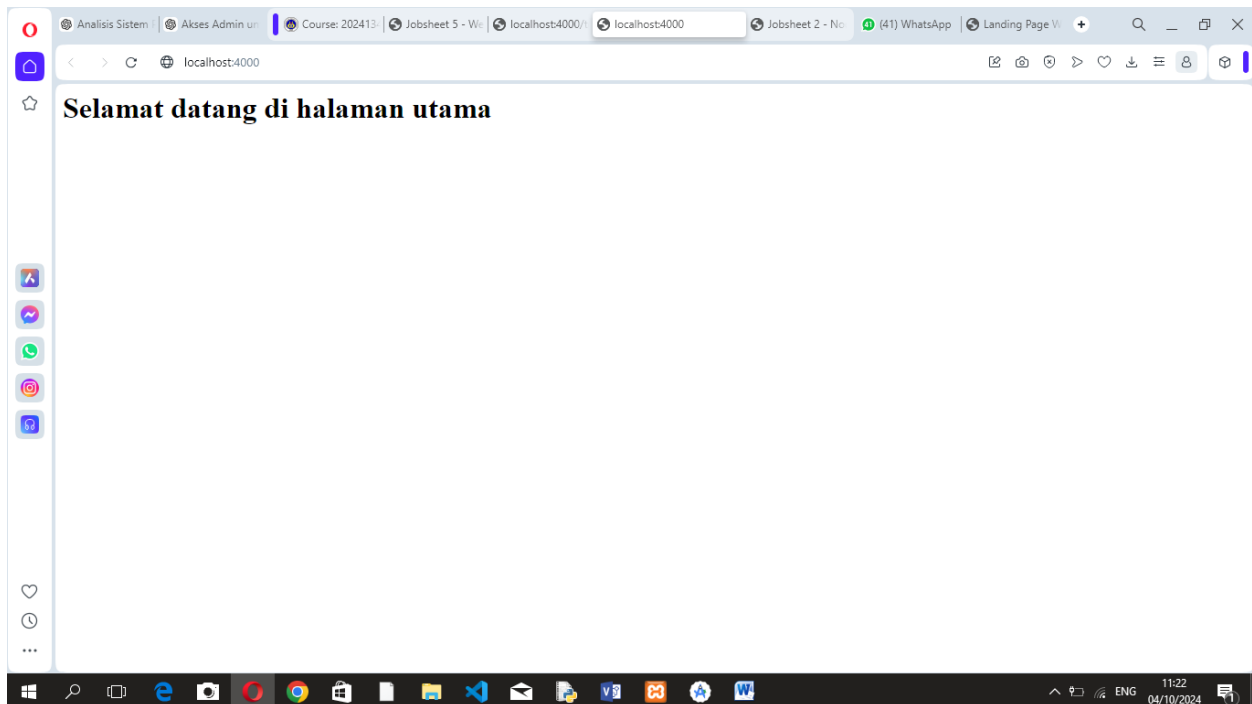
1. Mengganti teks 'selamat datang di halaman utama' dengan kode <h1>



```
Run ... basicnetprogramming
Welcome JS app.js x
pbjs > src > JS app.js > ...
1 const express = require('express')
2 const app = express()
3
4 //ini halaman/page utama
5 app.get('/', (req, res) => {
6   res.send('<h1> Selamat datang di halaman utama</h1>')
7 })
```

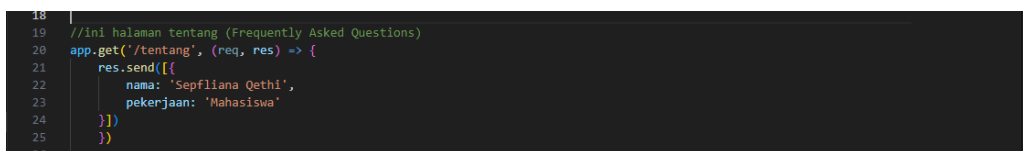
(Gambar b1. Mengganti teks pada kode halaman utama)

2. Jalankan file app.js menggunakan perintah 'nodemon app.js' dan cek kembali hasilnya pada browser.



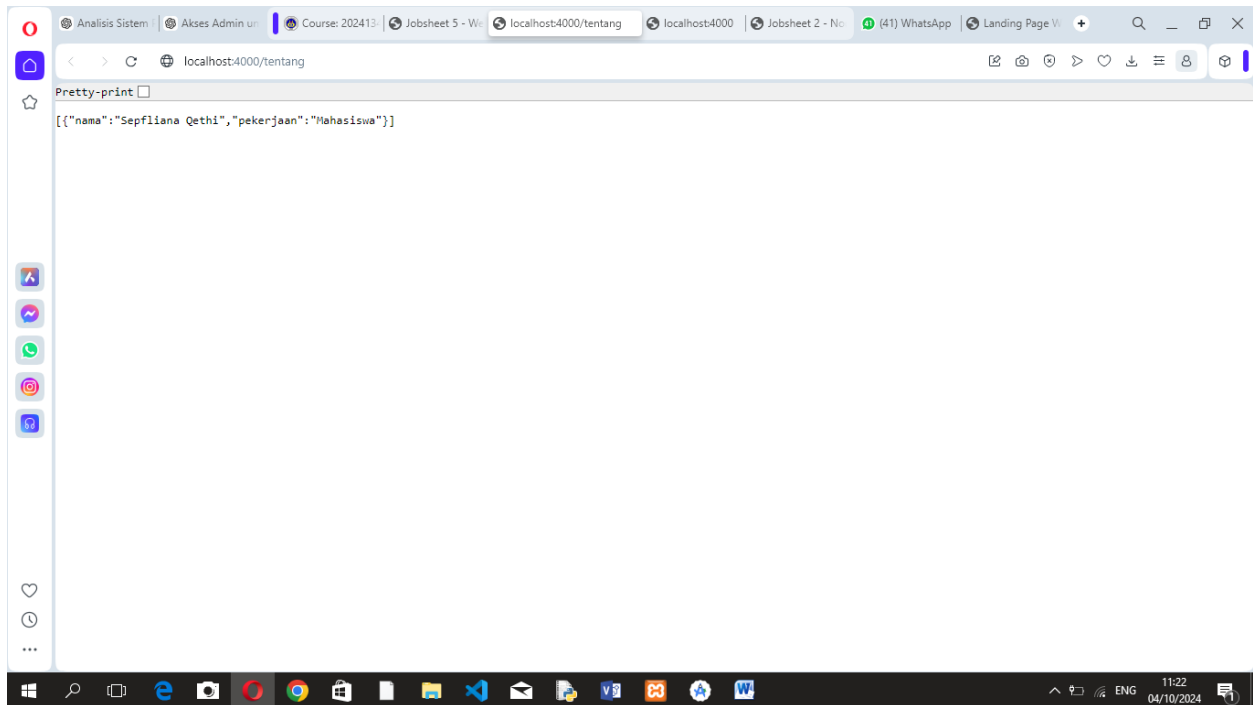
(Gambar b2. Perubahan teks pada halaman utama)

3. Mengganti kode pada halaman tentang seperti berikut



```
18
19 //ini halaman tentang (Frequently Asked Questions)
20 app.get('/tentang', (req, res) => {
21   res.send({
22     nama: 'Sepfiliana Qethi',
23     pekerjaan: 'Mahasiswa'
24   })
25 })
26
```

(Gambar b3. Mengganti teks pada kode halaman tentang)

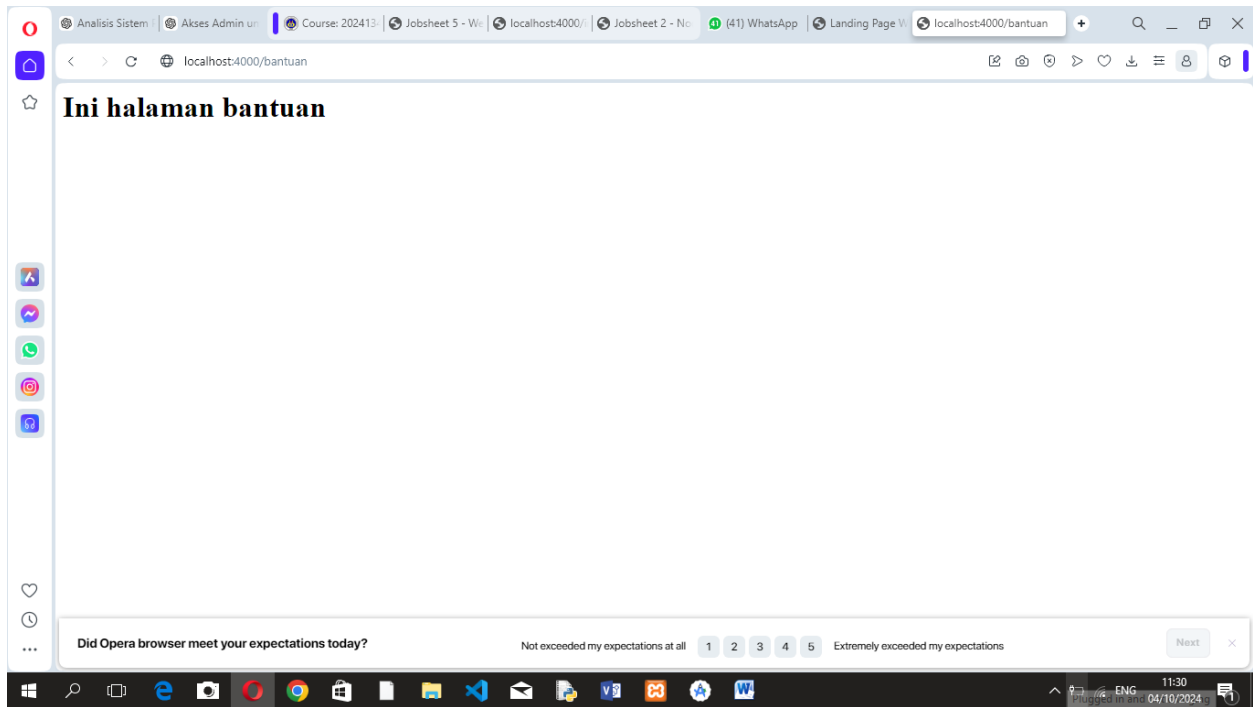


(Gambar b4. Output program pada web untuk halaman tentang)

- Ubah juga teks untuk halaman bantuan dan halaman infoCuaca. Tambahkan teks pada halaman bantuan dengan kode HTML dan teks infoCuaca dengan kode JSON yang berisi dua objek yaitu prediksiCuaca: 'cuaca berpotensi hujan' dan lokasi: 'Padang', sama seperti yang anda lakukan pada halaman tentang.

```
8
9 //ini halaman bantuan/FAQ (Frequently Asked Questions)
10 app.get('/bantuan', (req, res) => {
11   res.send('<h1>Ini halaman bantuan</h1>')
12 })
13
```

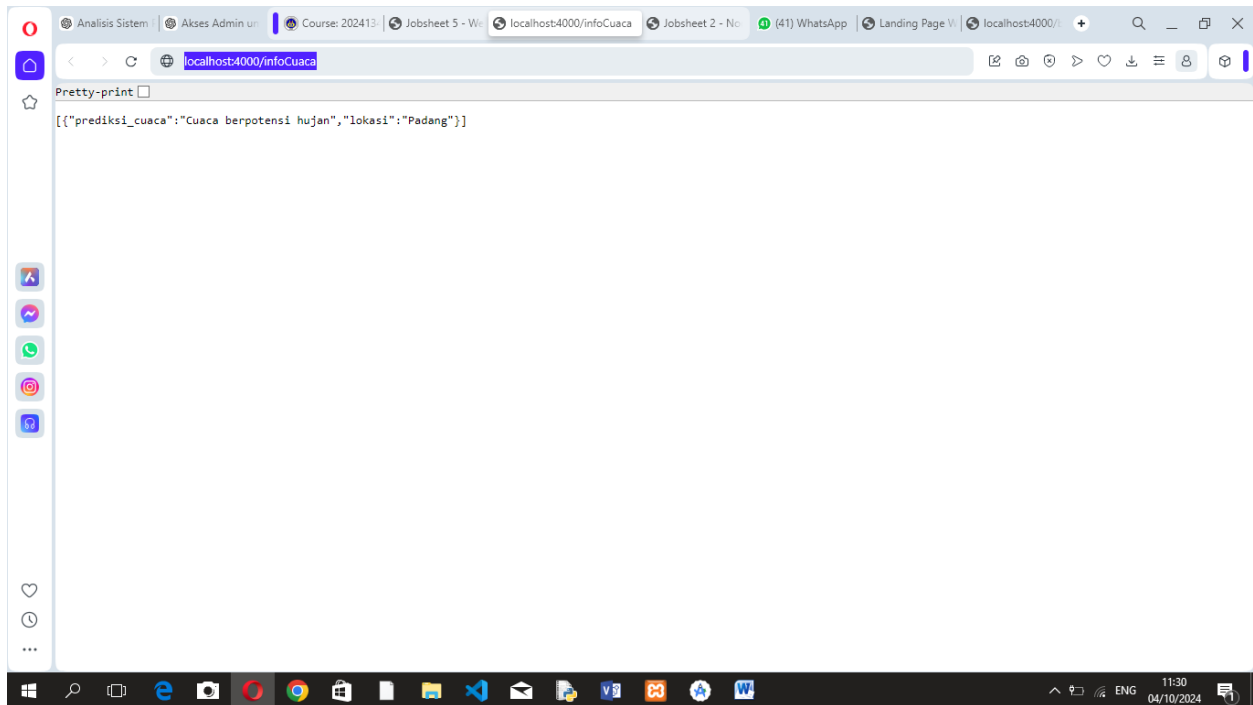
(Gambar b5. Mengganti kode pada halaman bantuan)



(Gambar b6. Perubahan teks pada web untuk halaman bantuan)

```
13 |  
14 | //Ini halaman infoCuaca (Frequently Asked Questions)  
15 | app.get('/infoCuaca', (req, res) => {  
16 |   res.send({  
17 |     prediksi_cuaca: 'Cuaca berpotensi hujan',  
18 |     lokasi: 'Padang'  
19 |   })  
20 | })  
21 |
```

(Gambar b7. Mengganti kode pada halaman infoCuaca)



(Gambar b8. Output program pada web untuk halaman infoCuaca)

c. Akses Static Assets pada Expressjs Menggunakan Path module

1. Membuat file baru pada folder public dengan nama 'index.html' dan ketik kode berikut

```

1 <!DOCTYPE html>
2 <html>
3 <head></head>
4 <body>
5 <h1>Ini adalah halaman utama statis</h1>
6 </body>
7 </html>
8

```

(Gambar c1. Kode html untuk halaman utama)

2. Selanjutnya, tambah kode seperti di bawah ini pada file app.js

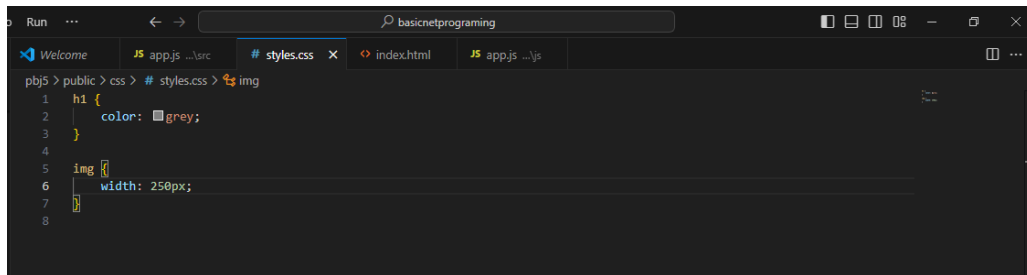
```

33
34 const path = require('path')
35 const direktoriPublic = path.join(__dirname, '../public')
36 app.use(express.static(direktoriPublic))

```

(Gambar c2. Penambahan kode module path pada file app.js)

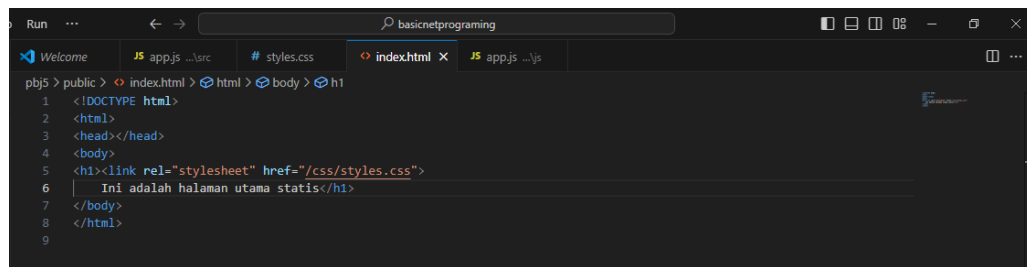
3. Lanjutkan membuat kode untuk halaman bantuan dan tentang dalam file html yang berbeda.
4. Selanjutnya membuat folder baru pada folder public dengan nama 'css'. Pada folder tersebut, buat file baru dengan nama 'styles.css' dan ketik kode berikut.



```
1 h1 {  
2   color: grey;  
3 }  
4  
5 img {  
6   width: 250px;  
7 }  
8
```

(Gambar c3. Kode pada file styles.css)

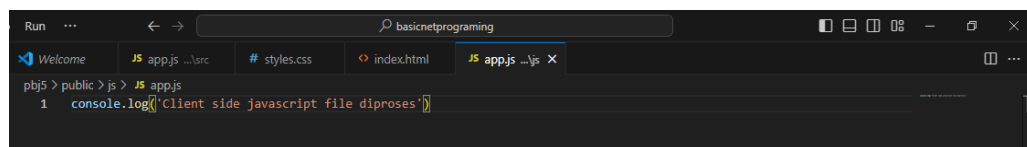
5. Tambahkan kode pada file index.html untuk memanggil fungsi css pada file html.



```
1 <!DOCTYPE html>  
2 <html>  
3 <head></head>  
4 <body>  
5 <h1><link rel="stylesheet" href="/css/styles.css">  
6   Ini adalah halaman utama statis</h1>  
7 </body>  
8 </html>  
9
```

(Gambar c4. Penambahan sumber css pada file html)

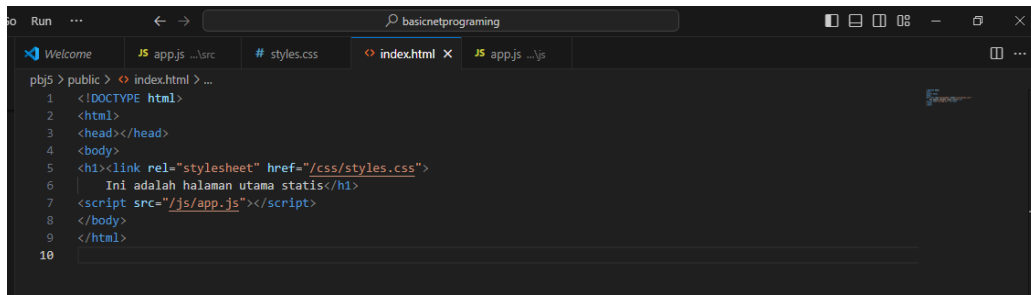
6. Membuat folder baru pada folder public dengan nama 'js'. Pada folder tersebut, buat file baru dengan nama app.js dan ketik kode berikut.



```
1 console.log('client side javascript file diproses');
```

(Gambar c5. Kode pada file app.js di folder public)

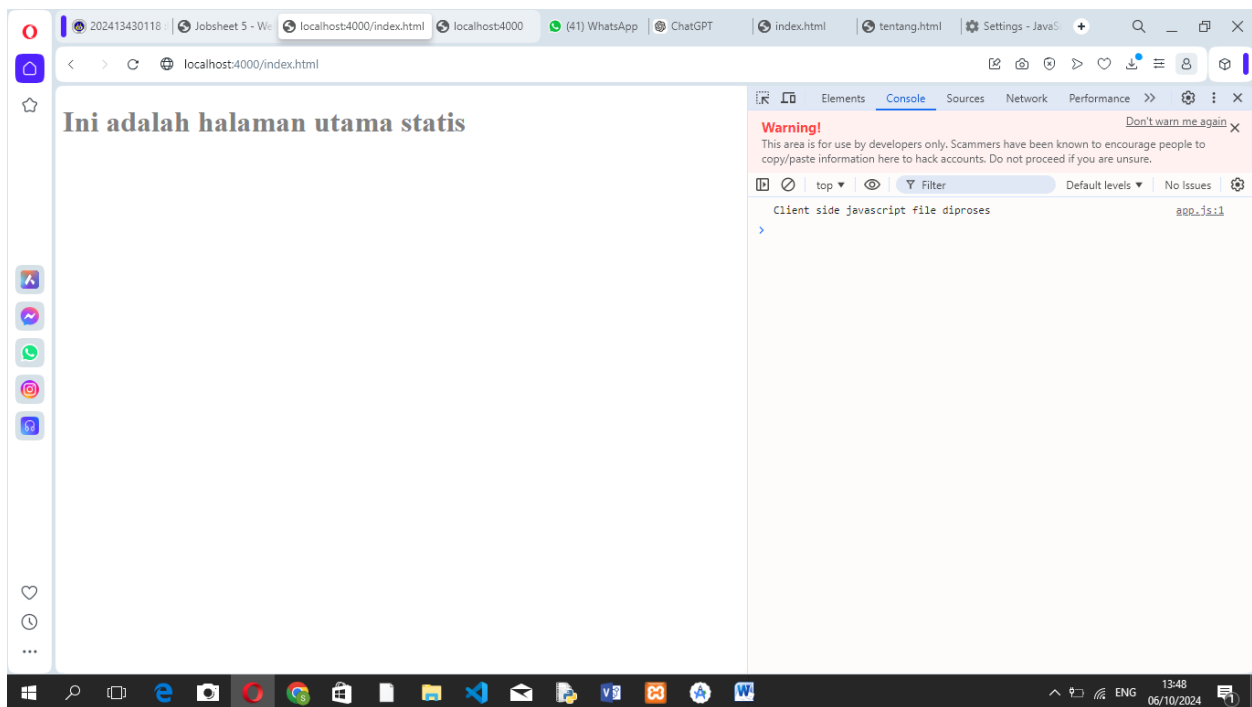
7. Tambahkan lagi baris kode seperti di bawah ini pada file index.html



```
1 <!DOCTYPE html>
2 <html>
3 <head></head>
4 <body>
5 <h1><link rel="stylesheet" href="/css/styles.css">
6   Ini adalah halaman utama statis</h1>
7 <script src="/js/app.js"></script>
8 </body>
9 </html>
10
```

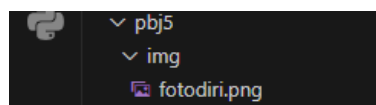
(gambar c6. Penambahan kode pada html untuk memanggil program app.js)

8. Jalankan program anda di browser. Sembari membuka halaman utama, lakukan inspect elemen pada browser chrome/firefox anda dan perhatikan bahwa file app.js dalam folder js telah di proses



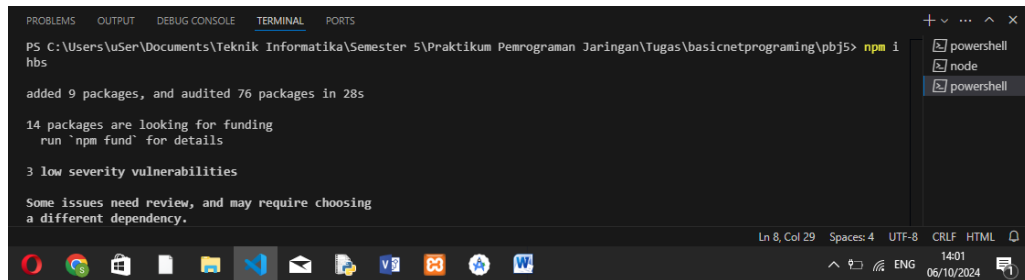
(Gambar c7. Output program html, css, dan app.js untuk halaman utama)

9. Membuat folder baru dengan nama 'img;'. Masukkan sebuah foto pada folder tersebut dengan format png.



(Gambar c8. Folder img dan file png)

- d. Templating pada Expressjs menggunakan Handlebars.js
1. Hentikan nodemon pada terminal dengan menekan ctrl + c
 2. Install hbs menggunakan perintah npm



```
PS C:\Users\user\Documents\Teknik Informatika\Semester 5\Praktikum Pemrograman Jaringan\Tugas\basicnetprograming\pbj5> npm i hbs

added 9 packages, and audited 76 packages in 28s

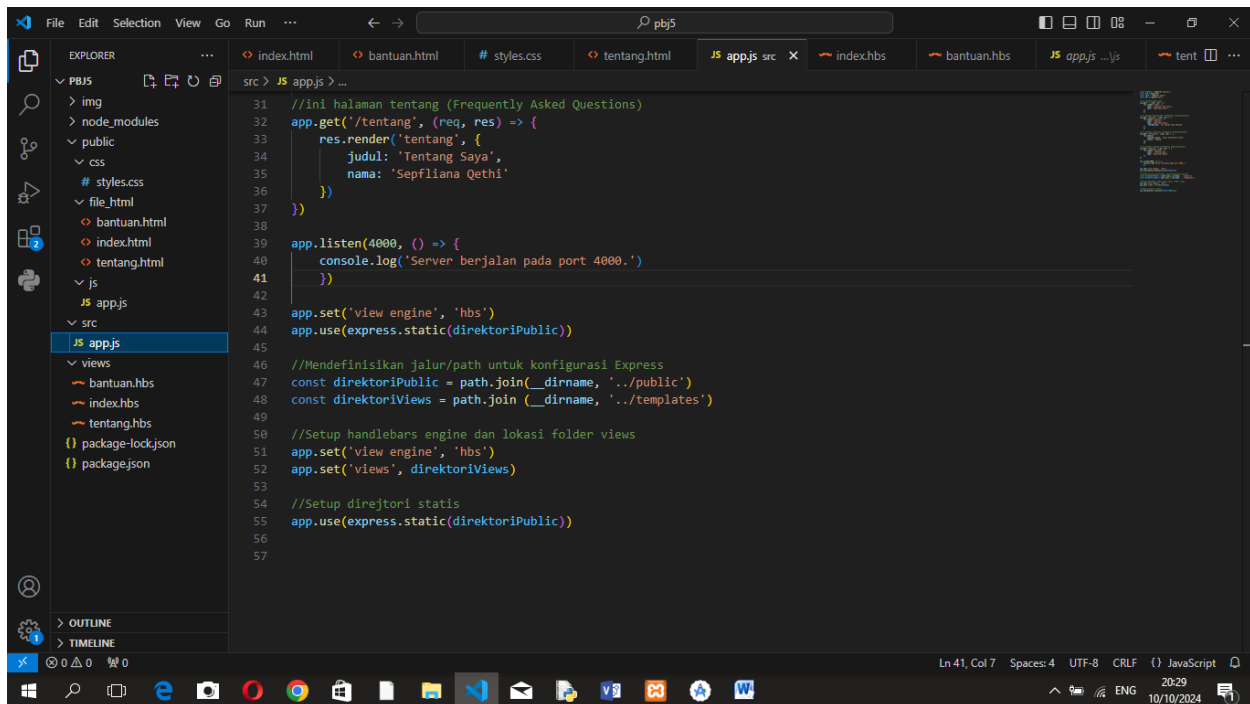
14 packages are looking for funding
  run `npm fund` for details

3 low severity vulnerabilities

Some issues need review, and may require choosing
a different dependency.
```

(Gambar d1. Instalasi hbs)

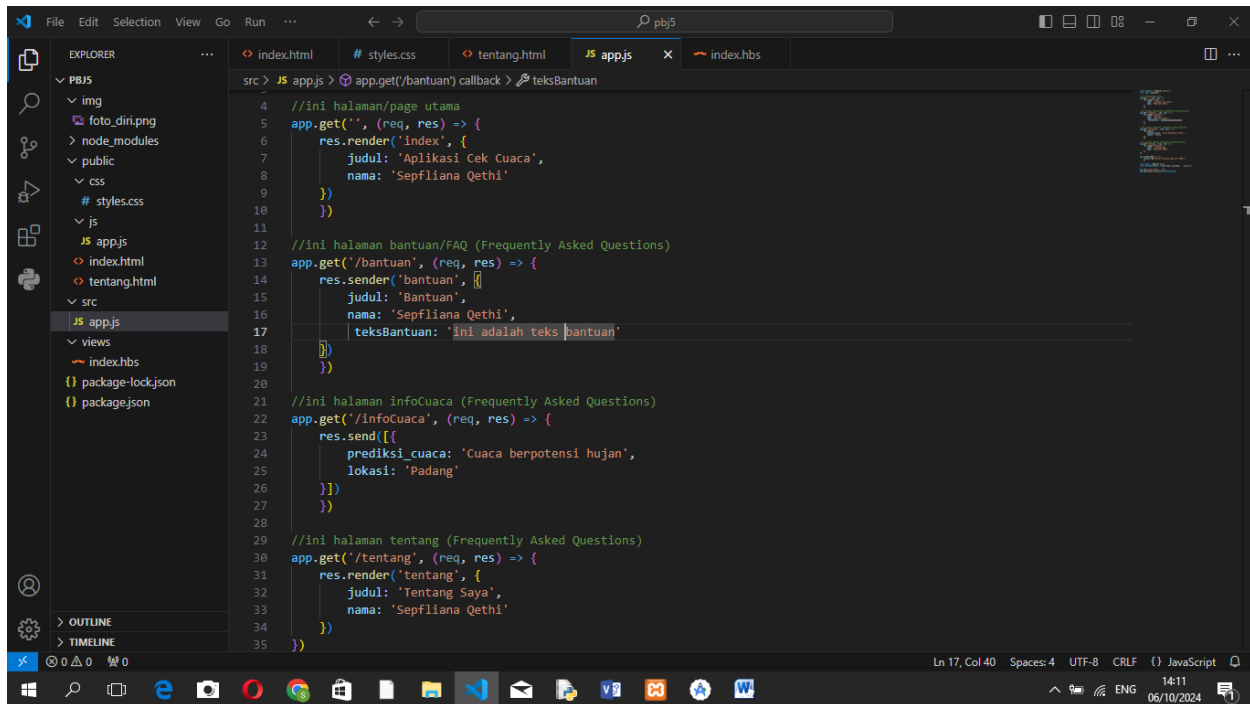
3. Jalankan kembali menggunakan perintah nodemon app.js
4. Tambahkan kode baru pada file app.js yang ada pada folder src



```
31 //ini halaman tentang (Frequently Asked Questions)
32 app.get('/tentang', (req, res) => {
33   res.render('tentang', {
34     judul: 'Tentang Saya',
35     nama: 'Sefliana Qethi'
36   })
37 })
38
39 app.listen(4000, () => {
40   console.log('Server berjalan pada port 4000.')
41 })
42
43 app.set('view engine', 'hbs')
44 app.use(express.static(direktoriPublic))
45
46 //Mendefinisikan jalur/path untuk konfigurasi Express
47 const direktoriPublic = path.join(__dirname, '../public')
48 const direktoriViews = path.join(__dirname, '../templates')
49
50 //Setup handlebars engine dan lokasi folder views
51 app.set('view engine', 'hbs')
52 app.set('views', direktoriViews)
53
54 //Setup direktori statis
55 app.use(express.static(direktoriPublic))
56
57
```

(Gambar d2. Penambahan kode pada file app.js di folder src)

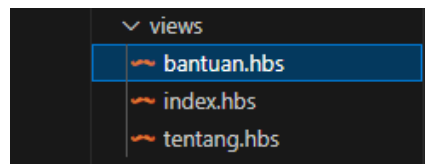
- Ubah kembali kode pada baris `app.get` halaman utama dan `app.get` halaman tentang pada file `app.js` di folder `src` dengan kode berikut



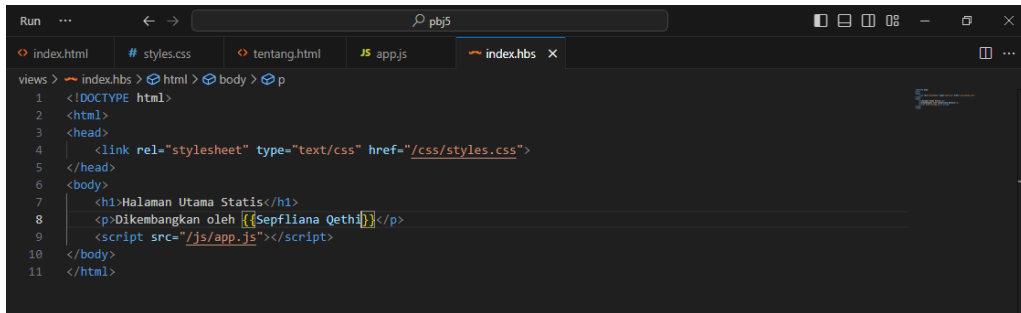
```
4 //ini halaman/page utama
5 app.get('/', (req, res) => {
6   res.render('index', {
7     judul: 'Aplikasi Cek Cuaca',
8     nama: 'Sepfilana Qethi'
9   })
10 })
11
12 //ini halaman bantuan/FAQ (Frequently Asked Questions)
13 app.get('/bantuan', (req, res) => {
14   res.render('bantuan', {
15     judul: 'Bantuan',
16     nama: 'Sepfilana Qethi',
17     teksBantuan: 'ini adalah teks bantuan'
18   })
19 })
20
21 //ini halaman infoCuaca (Frequently Asked Questions)
22 app.get('/infoCuaca', (req, res) => {
23   res.send({
24     prediksi_cuaca: 'Cuaca berpotensi hujan',
25     lokasi: 'Padang'
26   })
27 })
28
29 //ini halaman tentang (Frequently Asked Questions)
30 app.get('/tentang', (req, res) => {
31   res.render('tentang', {
32     judul: 'Tentang Saya',
33     nama: 'Sepfilana Qethi'
34   })
35 })
```

(Gambar d3. Mengubah kodingan pada halaman utama, halaman tentang dan halaman bantuan)

- Lalu, buatlah folder baru pada aplikasi web-server anda dan beri nama `views`. Didalam folder ini, buatlah file baru dengan nama `index.hbs`
- Salinlah semua baris kode yang ada pada file `index.html` ke file `index.hbs`. Lalu gantilah kode yang ada dalam `<body>` dengan baris kode seperti pada gambar d5.

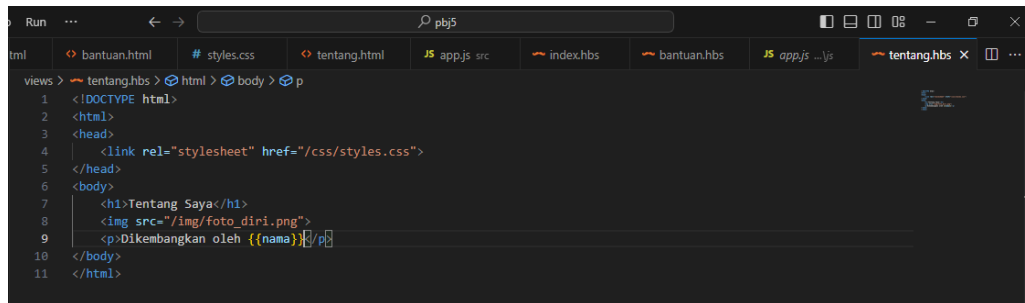


(Gambar d5. File hbs di dalam folder views)



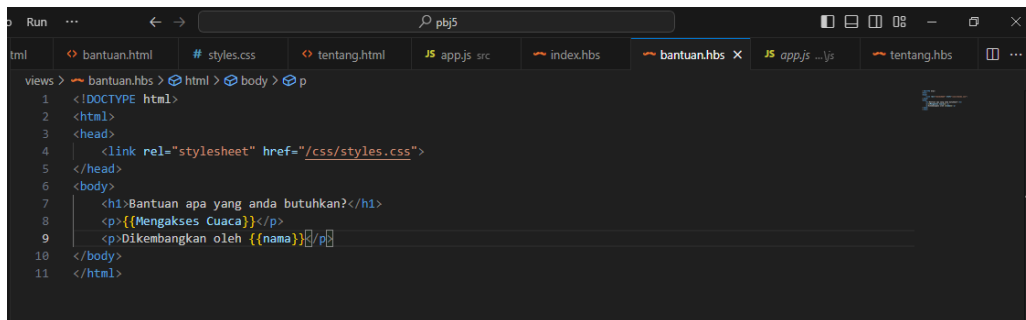
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" type="text/css" href="/css/styles.css">
5 </head>
6 <body>
7   <h1>Halaman Utama Statis</h1>
8   <p>Dikembangkan oleh {{Sepfliana Qethi}}</p>
9   <script src="/js/app.js"></script>
10 </body>
11 </html>
```

(Gambar d6. Kode pada file index.hbs)



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" href="/css/styles.css">
5 </head>
6 <body>
7   <h1>Tentang Saya</h1>
8   
9   <p>Dikembangkan oleh {{nama}}</p>
10 </body>
11 </html>
```

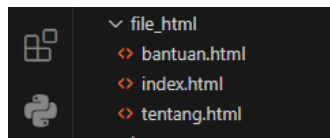
(Gambar d7. Kode pada file tentang.hbs)



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link rel="stylesheet" href="/css/styles.css">
5 </head>
6 <body>
7   <h1>Bantuan apa yang anda butuhkan?</h1>
8   <p>{{Mengakses Cuaca}}</p>
9   <p>Dikembangkan oleh {{nama}}</p>
10 </body>
11 </html>
```

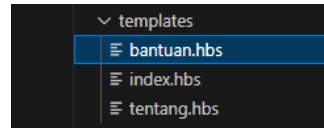
(Gambar d8. Kode pada file bantuan.js)

8. Membuat folder baru dengan nama ‘file_html’ di dalam folder public, lalu pindahkan semua file html ke dalamnya.



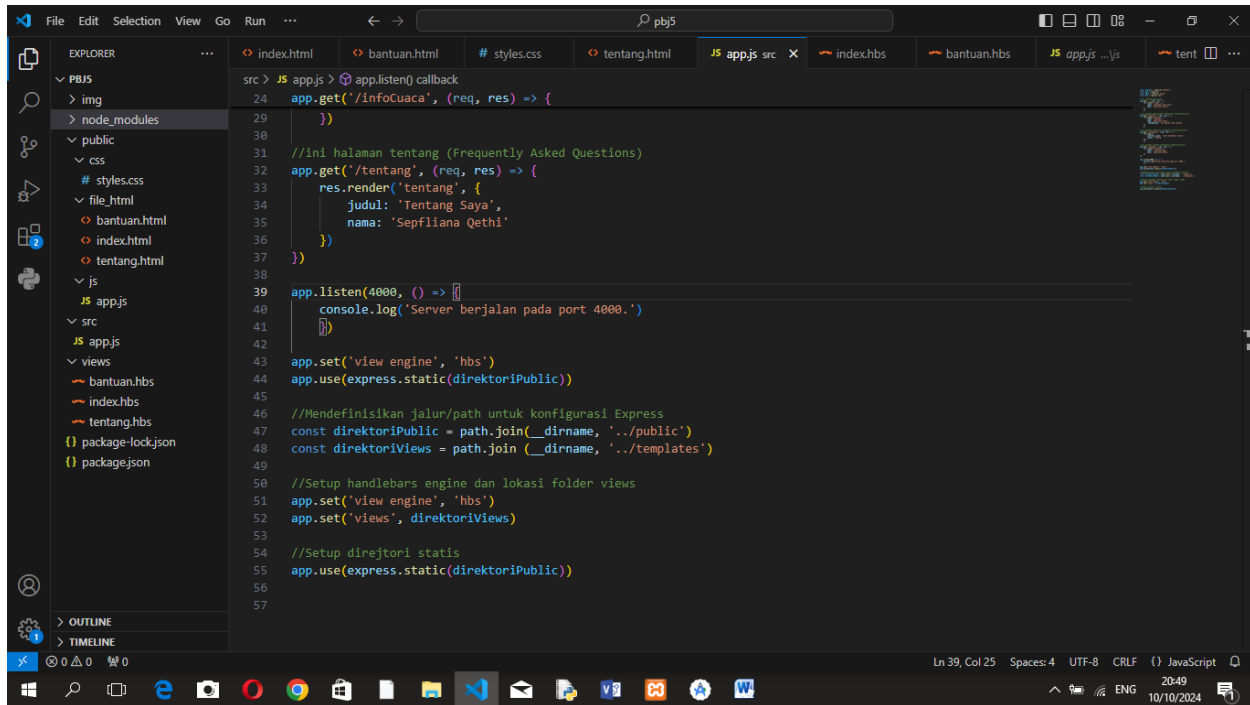
(Gambar d9. Isi folder html)

9. Ubah nama folder views dengan nama templates.



(Gambar d10. Mengubah nama folder menjadi templates)

10. Tambahkan kode berikut pada app.js di folder src



(Gambar d11. Menambahkan kode pada app.js di folder src)