**TUGAS JOBSHEET 4**

**Ditulis untuk memenuhi tugas matakuliah Pemrograman Jaringan**

**[202513430087]**

**Dosen Pengampu:**

**Randi Proska Sandra, S.Pd, M.Sc.**

****

**Disusun Oleh:**

**Florensius Panca Gati**

**(23343006)**

**PROGRAM STUDI INFORMATIKA**

**DEPARTEMEN TEKNIK ELEKTRONIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

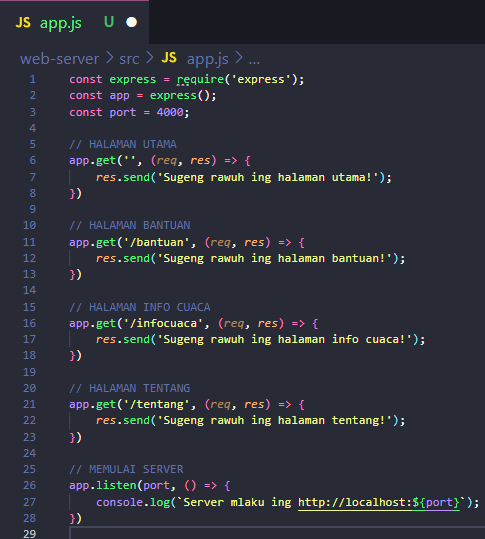
**2025**

**LATIHAN**

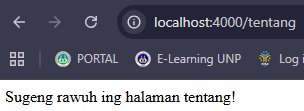
1. **Instalasi Express.js dan membuat halaman menggunakan fungsi app.get**
2. Buatlah folder baru dengan nama web-server dan bukalah folder tersebut pada aplikasi visual studio code anda.
3. Buka direktori ke folder dengan cara klik kanan, lalu pilih Open in Integrated Terminal. Lalu ketikan perintah npm init -y untuk meng-generate file package.json
4. Selanjutnya install express.js versi terbaru yaitu versi 4.18.2 menggunakan perintah npm yang telah anda pelajari sebelumnya. Silakan cek package express.js melalui halaman ini https://www.npmjs.com/package/express. Buka modul sebelumnya jika anda lupa perintah atau cara menginstall package dari npm
5. Anda akan melihat bahwa folder baru node\_modules dan file baru package-lock.json telah berhasil di generate
6. Buatlah dua buah folder baru dalam folder web-server melalui visual studio code anda. Beri nama folder pertama dengan public dan folder kedua dengan src
7. Didalam folder src, buatlah file baru dengan nama app.js
8. Ketikan kode berikut ini

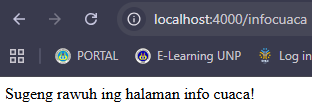
Catatan:

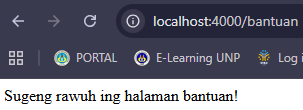
1. app = express() menunjukan pembuatan sebuah variable constant baru dengan nama app dan diambil dari fungsi express yang diimport dari expressjs
2. Method app.get mendefinisikan sebuah rute HTTP GET menggunakan objek app dari Express.js
3. (req, res) => { ... } Bagian dari kode ini mendefinisikan sebuah fungsi callback yang akan dieksekusi ketika rute diakses. Dalam Express.js, fungsi ini memiliki dua argumen: req (objek request) dan res (objek response). Di dalam fungsi ini, Anda mendefinisikan apa yang harus terjadi ketika pengguna mengakses rute ini.
4. res.send(...) digunakan untuk mengirimkan respons kembali ke klien (browser pengguna)
5. Expressjs memiliki banyak jenis method yang dapat diakses lebih jauh melalui dokumentasi berikut <https://expressjs.com/en/4x/api.html>
6. Lanjutkan koding diatas untuk membuat dua halaman lagi yaiut halaman infoCuaca dan halaman tentang. Lakukan sebagaimana mana baris kode halaman bantuan diatas
7. Lalu tambahkan baris kode berikut diakhir semua kode diatas

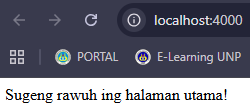


1. Buka direktori ke file app.js tersebut melalui terminal dengan cara klik kanan, lalu pilih Open in Integrated Terminal
2. Pastikan anda berada pada direktori src, lalu ketikan perintah node app.js untuk menjalankan program tersebut
3. Bukalah web browser anda lalu ketikan http://localhost:4000/. Angka 4000 dapat diganti sesuai dengan port yang telah anda set pada bari kode app.listen. Akan tampil seperti gambar dibawah ini
4. Cobalah mengakses halaman lainnya dengan menambahkan url halaman dibelakang url utama. Contoh: http://localhost:4000/infoCuaca. Coba lakukan juga untuk halaman bantuan dan halaman tentang

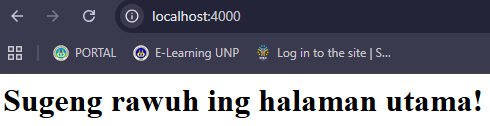
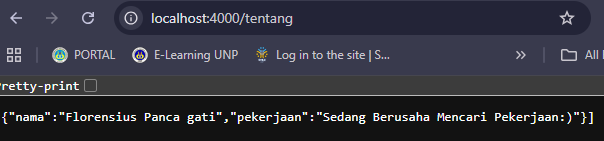
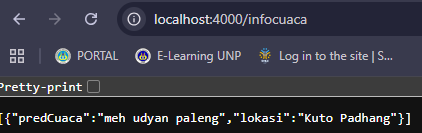








1. **Integrasi HTML dan JSON pada Express.js**
2. Cobalah ganti teks selamat datang di halaman utama dengan kode html sehingga menjadi
3. Jalankan file app.js menggunakan perintah nodemon app.js dan lihat hasilnya pada browser. PENTING! Untuk selanjutnya, anda akan menggunakan perintah nodemon untuk menjalankan program bukan lagi node. Hal ini bertujuan agar perubahan dapat dilihat secara realtime setiap anda mengubah baris kode.
4. Selanjutnya cobalah ganti kode pada halaman tentang dengan kode berikut Catatan: Teks pada nama dan pekerjaan dapat anda sesuaikan dengan keinginan anda
5. Bukalah aplikasi anda di browser dan pastikan anda beradal url halaman tentang. <http://localhost:4000/tentang>
6. Perhatikan bahwa tampilan pada halaman utama berbeda dengan halaman tentang. Hal ini karena halaman utama menggunakan format HTML sedangkan halaman tentang menggunakan format JSON
7. 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. Bukalah setiap halaman di browser anda dan pastikan bahwa semuanya menampilkan tampilan yang diinginkan.
9. PENTING! Untuk selanjutnya, setiap kali anda melakukan perubahan pada program anda, pastikan untuk mengecek halaman yang anda ubah. Hal ini bertujuan untuk memastikan konten yang tampil sudah sesuai dengan yang diinginkan dan tidak ada error yang muncul

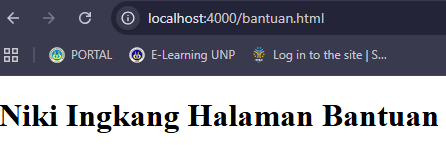
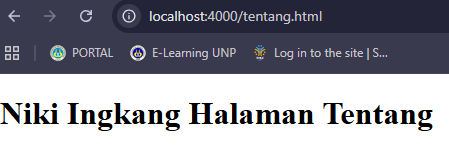
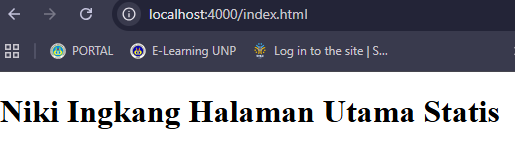
   

1. **Akses Static Assets pada Expressjs menggunakan Path module**
2. Buatlah file baru pada folder public dan beri nama index.html dan tambahkan kode html berikut
3. Selanjutnya, pada file app.js di folder src, tambakan kode berikut ini

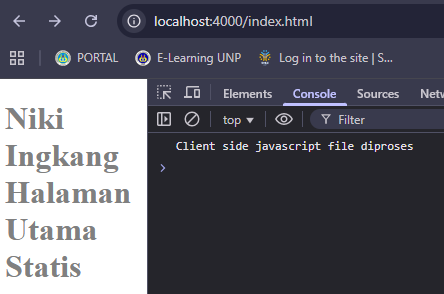
Pastikan! anda hanya menambahkan kode yang belum ada pada baris program anda. Jangan menghapus kode yang sudah ada

Catatan:

1. Module path menyediakan utilitas untuk bekerja dengan path file dan direktori yang dapat dipelajari lebih jauh melalui dokumentasi berikut ini <https://nodejs.org/api/path.html#path>
2. path.join() adalah method untuk menggabungkan semua segmen path yang diberikan menggunakan pemisah yang spesifik, kemudian melakukan normalisasi terhadap jalur yang dihasilkan.
3. \_\_dirname menunjukkan bahwa yang diakses adalah direktori dalam hal ini adalah direktori public. Jika anda mengakses file maka akan menggunakan \_\_filename
4. express.static dapat dipahami melalui dokumentasi berikut <http://expressjs.com/en/api.html#express.static>
5. Lanjutkan program dengan membuat dua file html lagi pada folder public dengan nama tentang.html dan bantuan.html. Silakan isi file html dengan baris kode html seperti pada halaman index.html. Gantilah teks pada baris kode agar sesuai dengan nama halaman.
6. Silakan akses aplikasi pada browser anda untuk melihat tampilannya. Cobalah akses masingmasing halaman dengan menambahkan .html dibelakang. Contoh: <http://localhost:4000/tentang.html>

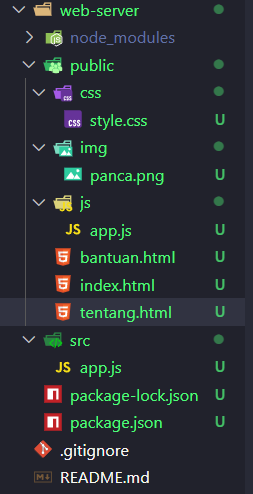
1. Selanjutnya buatlah folder baru pada folder public dengan nama css. Lalu, dalam folder tersebut, buatlah file baru dengan nama styles.css. Lalu tambahkanlah kode berikut ini
2. Selanjutnya tambahkan baris kode berikut pada file index.html pada didalam heading . Lakukan yang sama untuk file bantuan.html dan tentang.html
3. Buatlah satu folder baru lagi di folder public dengan nama js. Lalu buatlah file baru dalam folder tersebut dengan nama app.js dan tambahkan kode berikut ini
4. Lalu tambahkan lagi baris kode berikut pada file index.html, tepat dibawah baris kode pada nomor 6
5. Jalankan program anda di browser. Sembari membuka halaman utama, cobalah inspect elemen pada browser chrome/firefox anda dan perhatikan bahwa fille app.js dalam folder js telah di proses



1. Selanjutnya buatlah folder baru lagi dengan nama img. Lalu masukanlah sebuah gambar berformat .png ke folder tersebut. Direkomendasikan memasukan foto anda
2. Lalu tambahkan baris kode berikut pada file tentang.html tepat dibawah kode pada bagian
3. Silakan buka localhost pada browser anda untuk mengakses halaman tentang, bantuan dan halaman utama. Pastikan akhir halaman anda berekstensi .html. Contoh: <http://localhost:4000/tentang.html>



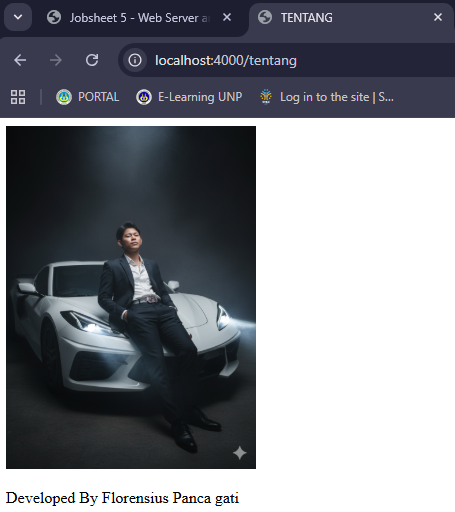
1. Keseluruhan direktori yang ada pada aplikasi anda adalah seperti pada gambar dibawah ini



1. **Templating pada Expressjs menggunakan Handlebars.js**

Pada bagian ini, anda akan menggunakan handlebars. Handlebars.js adalah perluasan dari bahasa templating Mustache yang diciptakan oleh Chris Wanstrath. Handlebars.js dan Mustache keduanya adalah bahasa templating yang tidak memiliki logika yang menjaga pemisahan antara tampilan dan kode. Handlebars dapat dikatakan sebagai sebuah mesin template (template engine) yang dapat digunakan bersama dengan Express.js untuk menghasilkan tampilan dinamis dalam aplikasi web. Handlebars memungkinkan untuk memisahkan logika aplikasi dari tampilan pada aplikasi berbasis web. Handlebars menggunakan sintaksis yang mudah dibaca dan dapat memasukkan variabel, loop, kondisi, dan lainnya ke dalam template HTML.

1. Silakan stop nodemon anda pada terminal di visual studio code dengan menekan ctrl + c
2. Lakukan instalasi hbs menggunakan perintah npm. hbs adalah sebuah mesin tampilan (view engine) express.js untuk handlebars.js. Silakan akses informasi lebih jauh terkait hbs melalui url berikut <https://www.npmjs.com/package/hbs>
3. Setelah instalasi selesai jalankan kembali aplikasi web-server anda menggunakan perintah nodemon app.js
4. Bukalah file app.js yang ada di folder src dan tambahkan baris kode berikut tepat diatas baris kode app.use(express.static(direktoriPublic)) PENTING! lalu, jadikan baris kode app.use diatas menjadi komentar
5. Ubah kembali kode pada baris app.get halaman utama dan app.get halaman tentang pada file app.js di folder src dengan kode berikut
6. Lanjutkan program, lakukan hal yang sama untuk halaman bantuan. Pada baris kode halaman bantuan, tambahkan satu objek lagi yaitu teksBantuan: 'ini adalah teks bantuan'. Khusus halaman infoCuaca biarkan saja seperti kode awal.
7. Lalu, buatlah folder baru pada aplikasi web-server anda dan beri nama views. Didalam folder ini, buatlah file baru dengan nama index.hbs
8. Salinlah semua baris kode yang ada pada file index.html ke file index.hbs. Lalu gantilah kode yang ada dalam dengan baris kode berikut
9. Buat dua buah file lagi dalam folder views dengan nama bantuan.hbs dan tentang.hbs.
10. Lalu salinlah baris kode bantuan.html ke bantuan.hbs dan ubahlah baris kode yang ada didalam dengan kode berikut
11. Lalu salin baris kode tentang.html ke tentang.hbs dan ubahlah baris kode yang ada didalam dengan kode berikut
12. Jika langkah-langkah diatas sudah selesai, buatlah folder baru didalam folder public dengan nama file\_html. Lalu pindahkan semua file .html kedalam folder ini. Hal ini bertujuan agar aplikasi tidak membaca file .html lagi melainkan file .hbs. Dalam praktik sebenarnya, anda dapat menghapus semua file .html karena tidak akan digunakan lagi. Selain itu file .html ini juga bersifat statis. Namun, dalam praktikum ini file .html tetap dibiarkan ada sebagai bukti apa yang sudah anda kerjakan
13. Ubahlah nama folder views dengan nama templates. Lalu pahami error yang tampil pada browser anda. Error ini terjadi karena expressjs menganggap folder views sebagai lokasi default untuk file-file yang terkait dengan tampilan website kita. Hal ini tidak efektif jika memiliki banyak file yang terkait dengan tampilan (views)
14. Untuk mengatasinya maka kita harus meng-kustomisasi path ke direktori folder views. Tambahkanlah kode berikut kedalam file app.js yang ada folder src. Penting! Cukup tambahkan baris kode yang belum ada pada baris kode anda Baris kode baru yang seharusnya anda tambahkan berada pada baris 3, 9 dan 13
15. Cek kembali program anda melalui browser apakah error sudah teratasi.



1. **Mengatur Tampilan Aplikasi menggunakan sistem templating**

Bagian ini hanya melanjutkan bagian d Langkah 15 yang bertujuan untuk mengatur tampilan aplikasi anda sehingga lebih menarik.

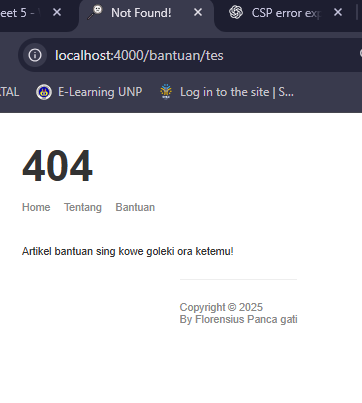
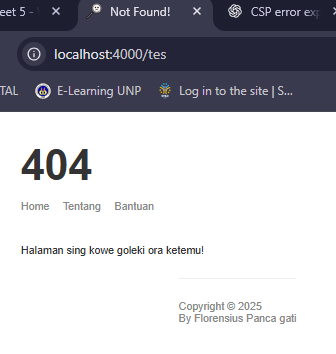
1. Silakan hentikan nodemon pada terminal visual studi code dengan menekan ctrl + c
2. Lalu jalankan kembali dengan perintah berikut nodemon app.js -e js,hbs. PENTING! Untuk selanjutnya, anda akan menggunakan perintah ini untuk menjalankan program anda.
3. Selanjutnya, buatlah dua folder baru didalam folder templates. Beri nama folder tersebut dengan views dan partials. Folder views akan berisi file tampilan utama, sementara folder partials nantinya akan berisi file tampilan partial seperti header dan footer. Hal ini diperlukan untuk membuat beberapa baris kode program menjadi dapat digunakan kembali (reusable).
4. Masukan file index.hbs, bantuan.hbs dan tentang.hbs ke folder views
5. Buatlah dua buah file didalam folder partials. File pertama beri nama header.hbs dan yang kedua footer.hbs
6. Sesuaikanlah baris kode berikut pada file app.js anda. Pastikan anda hanya mengubah yang perlu diubah. Pada tampilan diatas seharusnya anda hanya perlu mengubah baris 9, 10 dan 15
7. Masukan baris kode berikut pada file header.hbs
8. Lalu, masukan baris kode berikut pada file footer.hbs
9. Unduh icon cuaca berikut ini https://cdn-icons-png.flaticon.com/512/1779/1779940.png dan letakan pada folder img di aplikasi web-server anda. Jika anda ingin menggunakan gambar lainnya, silakan cari di google dan pastikan formatnya .png dan tidak memiliki background
10. Tambahkanlah baris kode berikut pada didalam bagian pada file index.hbs Silakan sesuaikan teks yang di blok dengan warna merah dengan nama file icon anda
11. Lakukan kembali langkah 8 untuk file bantuan.hbs dan tentang.hbs. Ubahlah teks yang di blok hijau sesuai dengan judul halaman masing-masing.
12. Ubalah kode pada file style.css dengan kode berikut ini
13. Ubah lagi kode pada file index.hbs, tentang.hbs dan bantuan.hbs yang ada dalam bagian dengan kode berikut
14. Baris kode didalam pada file index.hbs
15. Baris kode didalam pada file tentang.hbs
16. Baris kode didalam pada file bantuan.hbs Catatan: {{>header}} dan {{>footer}} digunakan mengakses file header dan footer yang ada pada folder partials
17. Berikut adalah seharusnya tampilan akhir aplikasi anda

1. **Wildcard route pada Expressjs**

Expressjs memungkinkan pembuatan routing menggunakan wildcard seperti tanda \*. Hal ini biasa digunakan programmer untuk menunjukan bahwa sebuah halaman (page) tidak ada seperti error 404.

1. Silakan tambahkan baris kode berikut pada file app.js yang ada didalam folder src. Letakan tepat diatas app.listen dan dibawah app.get untuk halaman infoCuaca Catatan:
2. app.get(‘/bantuan/\*’……..) digunakan untuk menampilkan pesan kesalahan ketika sebuah halaman artikel pada url …./bantuan tidak ditemukan
3. app.get(‘\*’……..) digunakan untuk menampilkan pesan kesalahan ketika sebuah halaman di url manapun tidak ditemukan
4. Selanjutnya, buatlah file baru pada folder templates/views dengan nama 404.hbs dan tambahkan baris kode berikut ini
5. Silakan ketikan url berikut pada browser anda http://localhost:4000/tes dan http://localhost:4000/bantuan/tes. Lalu pahamilah apa yang ditampilkan.

**TUGAS**

1. Perhatikan bahwa didalam kode anda pada file index.hbs, tentang.hbs, bantuan.hbs dan 404.hbs terjadi penggunaan beberapa baris kode berulang pada bagian . Buatlah kode tersebut menjadi lebih efisien dengan dengan menampungnya pada file .hbs terpisah didalam folder partials. Baris kode pada bagian harus tetap bersifat dinamis sehingga ketika halaman diganti maka title halaman juga berganti.
2. Src/app.js

const path = require('path');

const hbs = require('hbs');

const express = require('express');

const { title } = require('process');

const app = express();

const port = 4000;

// MENDEFINISIKAN PATH U/ KONFIGRUASI EXPRESS

const direktoriPublic = path.join(\_\_dirname, '../public');

const direktoriViews = path.join(\_\_dirname, '../templates/views');

const direktoriPartials = path.join(\_\_dirname, '../templates/partials');

// SETUP HANDLEBARS SEBAGAI VIEW ENGINE

app.set('view engine', 'hbs');

app.set('views', direktoriViews);

hbs.registerPartials(direktoriPartials);

// SETUP DIREKTORI STATIS

app.use(express.static(direktoriPublic));

// HALAMAN UTAMA

app.get('', (*req*, *res*) => {

*res*.render('index', {

        title: 'Aplikasi Cek Cuaca',

        img: '/img/cuaca.png',

        judul: 'Aplikasi cek cuaca nganggo Express.js',

        nama: 'Florensius Panca gati',

    });

})

// HALAMAN BANTUAN

app.get('/bantuan', (*req*, *res*) => {

*res*.render('bantuan', {

        title: 'Bantuan',

        img: '/img/cuaca.png',

        judul: 'Halaman Bantuan Kawula',

        teksBantuan: 'Ini adalah halaman bantuan jika kamu mengalami kesulitan dalam menggunakan aplikasi ini.',

        nama: 'Florensius Panca gati',

    });

})

// HALAMAN INFO CUACA

app.get('/infocuaca', (*req*, *res*) => {

*res*.send([{

        title: 'Info Cuaca',

        img: '/img/cuaca.png',

        predCuaca: 'meh udyan paleng',

        lokasi: 'Kuto Padhang',

    }]);

})

// HALAMAN TENTANG

app.get('/tentang', (*req*, *res*) => {

*res*.render('tentang', {

        title: 'Tentang',

        img: '/img/cuaca.png',

        judul: 'Halaman Tentang Kawula',

        nama: 'Florensius Panca gati',

        pekerjaan: 'Sedang Berusaha Mencari Pekerjaan:)',

    });

})

// HALAMAN 404

app.all('/bantuan/{\*splat}', (*req*, *res*) => {

*res*.render('404', {

        title: 'Not Found!!!',

        img: '/img/nf.png',

        rel: 'icon',

        judul: '404',

        nama: 'Florensius Panca gati',

        pesanError: 'Artikel bantuan sing kowe goleki ora ketemu!',

    })

});

app.all('{\*splat}', (*req*, *res*) => {

*res*.render('404', {

        title: 'Not Found!!!',

        img: '/img/nf.png',

        rel: 'icon',

        judul: '404',

        nama: 'Florensius Panca gati',

        pesanError: 'Halaman sing kowe goleki ora ketemu!',

    })

});

// MEMULAI SERVER

app.listen(port, () => {

    console.log(`Server mlaku ing http://localhost:${port}`);

})

1. Templates/partials/head.hbs

<head>

    <meta *charset=*"UTF-8">

    <meta *name=*"viewport" *content=*"width=device-width, initial-scale=1.0">

    <title>{{*title*}}</title>

    <link *rel=*"icon" *href=*"{{*img*}}">

    <link *rel=*"stylesheet" *href=*"/css/style.css">

    <script *src=*"/js/app.js"></script>

</head>

1. Templates/views/404.hbs

<!DOCTYPE html>

<html *lang=*"en">

    {{>*head*}}

<body>

    <div *class=*"main-content">

        {{>*header*}}

        <p>{{*pesanError*}}</p>

    </div>

    {{>*footer*}}

</body>

</html>

1. Templates/views/index.hbs

<!DOCTYPE html>

<html *lang=*"en">

    {{>*head*}}

<body>

    <div *class=*"main-content">

        {{>*header*}}

        <p>Aplikasi buat cek cuaca!</p>

    </div>

    {{>*footer*}}

</body>

</html>

1. Templates/views/tentang.hbs

<!DOCTYPE html>

<html *lang=*"en">

    {{>*head*}}

<body>

    <div *class=*"main-content">

        {{>*header*}}

        <img *src=*"/img/panca.png" *alt=*"panca" *class=*"portrait">

    </div>

    {{>*footer*}}

</body>

</html>

1. Templates/views/bantuan.hbs

<!DOCTYPE html>

<html *lang=*"en">

    {{>*head*}}

<body>

    <div *class=*"main-content">

        {{>*header*}}

        <p>{{*teksBantuan*}}</p>

    </div>

    {{>*footer*}}

</body>

</html>

1. Hasil

