{

"compilerOptions": { "target": "ES5", "module": "commonjs", "outDir": "dist",

"rootDir": "src", "strict": true,

"esModuleInterop": true, "skipLibCheck": true,

"forceConsistentCasingInFileNames": true

}

}

Listing 3.1 Isi *tsconfig.json*

import \* as yargs from 'yargs';

// <1>

const argv = yargs.options({ url: {

alias: 'u', demandOption: true,

description: 'URL yang akan diambil'

},

output: {

alias: 'o',

default: "output.unk",

description: 'Hasil disimpan di file ini'

}

}).argv;

// <2>

// <3>

let url: string = argv.url as string;

let fhasil: string = argv.output as string;

// <4>

// <5>

console.log('Anda memasukkan URL ', argv.url);

console.log('Anda memasukkan nama file output:', argv.output);

Listing 3.2

Mengolah Argumen dari *Command Line / Shell*

$ ts-node src/index.ts Pilihan:

--help Lihat bantuan

--version Lihat nomor versi

-u, --url URL yang akan diambil

-o, --output Hasil disimpan di file ini

[boolean]

[boolean] [diperlukan]

[bawaan: "output.unk"]

***Kurang argumen wajib: url***

$ ts-node src/index.ts --url [http://www.nodejs.org](http://www.nodejs.org/) --output node.html Anda memasukkan URL [http://www.nodejs.org](http://www.nodejs.org/)

Anda memasukkan nama file output: node.html

$

Listing 3.3

Menjalankan Perintah ts-node pada *File* index.ts

import axios from 'axios';

const AxiosInstance = axios.create(); let requestedData: string = '';

// <1>

// <2>

// <3>

[AxiosInstance.get('http://www](http://www.nodejs.org/%27)).nodejs.org/')

.then( response => {

requestedData = response.data; console.log(requestedData);

}).catch(error=> { console.log(error.message);

}

);

// <4>

// <5>

// <6>

Listing 3.4

Mengambil *File* dari URL di Internet dengan Axios

*$ ts-node src/index.ts*

Hasil : Isi dari Web di URL [http://www.nodejs.org](http://www.nodejs.org/) akan ditampilkan ke layar.

let HtmlTableToJson = require('html-table-to-json'); let jsonTables = HtmlTableToJson.parse(`

<p>Tabel 1</p>

<table>

<tr>

<th>NIM</th>

<th>Nama</th>

<th>Alamat</th>

</tr>

<tr>

<td>123456</td>

<td>Mahasiswa</td>

<td>Satu</td>

</tr>

<tr>

<td>7891011</td>

<td>Mahasiswa</td>

<td>Dua</td>

</tr>

</table>

<p>Tabel 2</p>

<table>

<tr>

<th>NIDN</th>

<th>Nama Dosen</th>

</tr>

<tr>

// <1>

// <2>

<td>9876</td>

<td>Dosen Satu</td>

</tr>

<tr>

<td>98789</td>

<td>Dosen dua</td>

</tr>

</table>

`);

console.log('Jumlah tabel: ', jsonTables.count); console.log(jsonTables.results);

// <3>

// <4>

Listing 3.5

Mengubah Tabel HTML Menjadi Data JSON

$ ts-node src/html-json.ts Jumlah tabel: 2

[

[

{ NIM: '123456', Nama: 'Mahasiswa', Alamat: 'Satu' },

{ NIM: '7891011', Nama: 'Mahasiswa', Alamat: 'Dua' }

], [

{ NIDN: '9876', 'Nama Dosen': 'Dosen Satu' },

{ NIDN: '98789', 'Nama Dosen': 'Dosen dua' }

]

]

Listing 3.6

Menjalanlan *Script* HTML-tabel to JSON

import \* as fs from "fs/promises";

// <1>

let f: string = 'test.txt';

// <2>

let requestedData = 'ini contoh teks yang ditulis ke file';

// <3>

(async function(f) { try {

await fs.writeFile(f, requestedData); console.log(`Disimpan ke file ${f}`);

} catch (error) { console.log('Error:', error);

}

})(f);

// <4>

Listing 3.7 Penulisan ke *File*

**MSIM4401**/MODUL 3 **3.11**



|  |
| --- |
| let HtmlTableToJson = require('html-table-to-json');  // menyiapkan berbagai opsi untuk yargs const argv = yargs.options({  url: {  alias: 'u', demandOption: true,  description: 'URL yang akan diambil'  },  output: {  alias: 'o',  default: "output.unk",  description: 'Hasil disimpan di file ini'  }  }).argv;  let url: string = argv.url as string;  let fhasil: string = argv.output as string;  // persiapan untuk mengambil file menggunakan axios const AxiosInstance = axios.create();  let requestedData: string = ''; let ftype: string = '';  let dataToBeWritten: string = ''; let jsonTables: any;  AxiosInstance.get(url)  .then(  // file berhasil diambil response => {  requestedData = response.data;  ftype = response.headers['content-type'].substring(0,9);  // Jika HTML, maka cari table. Jika ada, isikan data tabel  // ke JSON.  if (ftype === 'text/html') {  jsonTables = HtmlTableToJson.parse(requestedData); console.log('Jumlah tabel: ', jsonTables.count);  if (jsonTables.count > 0) {  dataToBeWritten = JSON.stringify(jsonTables.results);  } else {  console.log('Tidak ada tabel di HTML, menulis HTML ke file'); dataToBeWritten = requestedData;  }  } else {  // jika bukan HTML, langsung ditulis apa adanya. dataToBeWritten = requestedData;  } |
| // Simpan ke nama file (async function(f) { |

try {

await fs.writeFile(f, dataToBeWritten); console.log(`Disimpan ke file ${f}`);

} catch (error) {

console.log('Error:', error.message);

}

})(fhasil);

}).catch(error=> { console.log(error.message);

}

);

Listing 3.9

Kode Sumber *Scraping*

$ npm run start url "https://datatables.net/examples/data\_sources/dom"

--output salary.json

* [scraper@1.0.0](mailto:scraper@1.0.0) start
* node dist/index.js "--url" "https://datatables.net/examples/data\_sources/ dom" "--output" "salary.json"

Jumlah tabel: 1

Disimpan ke file salary.json

$ more < salary.json

[[{“Name”:”Tiger Nixon”,”Position”:”System

Architect”,”Office”:”Edinburgh”,”Age”:”61”,”Start

date”:”2011/04/25”,”Salary”:”$320,800”},

…

…

…

{“Name”:”Donna Snider”,”Position”:”Customer Support”,”Office”:”New York”,”Age”:”27”,”Start date”:”2011/01/25”,”Salary”:”$112,000”}]]

Listing 3.10

Mengambil *File* HTML yang Mengandung Tabel di Dalam *File* HTML

// paket-paket yang digunakan import axios from 'axios'; import \* as yargs from 'yargs';

import \* as fs from "fs/promises";

let HtmlTableToJson = require('html-table-to-json'); import \* as validateURI from "valid-url"

import \* as fsNode from "fs"; import \* as path from "path";

|  |
| --- |
| // menyiapkan berbagai opsi untuk yargs const argv = yargs.options({  url: {  alias: 'u', demandOption: true,  description: 'URL yang akan diambil'  },  output: {  alias: 'o',  default: "output.unk",  description: 'Hasil disimpan di file ini'  }  }).argv;  let url: string = argv.url as string;  let fhasil: string = argv.output as string;  // Latihan 1: memeriksa validitas URL  // menggunakan paket valid-url  // jangan lupa npm install valid-url --save-dev if (!validateURI.isUri(url)) {  console.log('URL yang anda masukkan tidak valid'); process.exit(1);  }  // Latihan 2: memeriksa apakah file sudah ada atau belum  // Jika sudah, keluar  if (fsNode.existsSync(fhasil)) { console.log('file sudah ada'); process.exit(1);  } else {  console.log('file ', fhasil, ' belum ada');  }  // Latihan 3: memeriksa akses direktori tempat file output try {  fsNode.accessSync(path.dirname(fhasil), fsNode.constants.R\_OK | fsNode.constants.W\_OK | fsNode.constants.F\_OK);  console.log('Hak akses direktori OK');  } catch (err) {  console.log('Hak akses direktori tidak cukup'); process.exit(1);  }  // persiapan untuk mengambil file menggunakan axios const AxiosInstance = axios.create();  let requestedData: string = ''; let ftype: string = '';  let dataToBeWritten: string = ''; let jsonTables: any; |
| AxiosInstance.get(url)  .then(  // file berhasil diambil |

response => {

requestedData = response.data;

ftype = response.headers['content-type'].substring(0,9);

// Jika HTML, maka cari table. Jika ada, isikan data tabel

// ke JSON.

if (ftype === 'text/html') {

jsonTables = HtmlTableToJson.parse(requestedData); console.log('Jumlah tabel: ', jsonTables.count);

if (jsonTables.count > 0) {

dataToBeWritten = JSON.stringify(jsonTables.results);

} else {

console.log('Tidak ada tabel di HTML, menulis HTML ke file'); dataToBeWritten = requestedData;

}

} else {

// jika bukan HTML, langsung ditulis apa adanya. dataToBeWritten = requestedData;

}

// Simpan ke nama file (async function(f) {

try {

await fs.writeFile(f, dataToBeWritten); console.log(`Disimpan ke file ${f}`);

} catch (error) {

console.log('Error:', error.message);

}

})(fhasil);

}).catch(error=> { console.log(error.message);

}

);

Listing 3.11

Kode Sumber Latihan 1

Konfigurasi terbagi menjadi 2 bagian, yaitu package.json dan tsconfig.json:

* + 1. *Package.json*

{

"name": "mhs-endpoint",

"version": "1.0.0",

"description": "",

"main": "index.js", "scripts": {

"start:dev": "nodemon dist/index.js",

"build:dev": "tsc --watch --preserveWatchOutput", "dev": "concurrently \"npm:build:dev\" \"npm:start:dev\""

},

"keywords": [],

"author": "",

"license": "ISC", "devDependencies": { "@types/cors": "^2.8.9", "@types/express": "^4.17.10", "cors": "^2.8.5",

"express": "^4.17.1"

}

}

Listing 3.12 Konfigurasi *Package.json*

Paket yang terdapat pada devDependencies harus di-*install* terlebih dahulu menggunakan **npm install <nama-paket> --save-dev**.

* + 1. *tsconfig.json*

*File* ini digunakan untuk mengkonfigurasi kompilator TypeScript.

{

"compilerOptions": { "target": "ES5", "module": "commonjs", "outDir": "./dist",

"rootDir": "./src", "strict": true,

"esModuleInterop": true, "skipLibCheck": true,

"forceConsistentCasingInFileNames": true, "resolveJsonModule": true

}

}

Listing 3.13 Konfigurasi Kompilator TypeScript

import { Request, Response } from 'express'; import { CrudController } from '../CrudController';

import dataMhsJson from '../../resources/dataMhs.json';

export class MhsController extends CrudController {

public read(req: Request<import("express-serve-static-core"). ParamsDictionary>,

res: Response): void {

res.json(dataMhsJson);

}

}

Listing 3.14 src/controllers/Mhs/Mhs.ts

[

{

"nim": 1,

"nama": "Mahasiswa 1",

"tglLahir": "25-08-2003",

"e-mail": ["mhs1@@kampus.ac.id",](mailto:mhs1@@kampus.ac.id) "prodi": "Informatika"

},

{

"nim": 2,

"nama": "Mahasiswa 2",

"tglLahir": "01-03-2002",

"e-mail": ["mhs2@kampus.ac.id",](mailto:mhs2@kampus.ac.id) "prodi": "Informatika"

},

{

"nim": 3,

"nama": "Mahasiswa 3",

"tglLahir": "14-11-2004",

"e-mail": ["mhs3@kampus.ac.id",](mailto:mhs3@kampus.ac.id) "prodi": "Sistem Informasi"

},

{

"nim": 4,

"nama": "Mahasiswa 4",

"tglLahir": "21-02-2002",

"e-mail": ["mhs4@kampus.ac.id",](mailto:mhs4@kampus.ac.id) "prodi": "Sistem Informasi"

},

{

"nim": 5,

"nama": "Mahasiswa 5",

"tglLahir": "10-01-2002",

"e-mail": ["mhs5@kampus.ac.id",](mailto:mhs5@kampus.ac.id) "prodi": "Sains Data"

}

]

Listing 3.15 Resources/dataMhs.json

import express from 'express'; import cors from 'cors';

import { PORT } from './config/constants'; import { mhsRouter } from './routes';

const app = express(); app.use(express.json());

app.use(cors()); app.get('/', (req, res) => {

res.send('Selamat datang di RESTful API gateway - SI Akademik');

});

app.use('/mhs', mhsRouter);

app.listen(PORT, () => {

console.log(`Endpoint Akademik sudah siap dan bisa diakses di port

${PORT}`);

});

Listing 3.16 src/index.ts

**3.28** Praktikum-1: *Typescript* dan *Vue*



Beberapa *file* akan diuraikan di bagian berikutnya.

* 1. *package.json*

{

"name": "vue-client",

"version": "0.1.0", "private": true, "scripts": {

"serve": "vue-cli-service serve", "build": "vue-cli-service build", "lint": "vue-cli-service lint"

},

"dependencies": { "core-js": "^3.6.5",

"vue": "^3.0.0"

},

"devDependencies": { "@types/axios": "^0.14.0",

"@vue/cli-plugin-babel": "~4.5.0", "@vue/cli-plugin-eslint": "~4.5.0", "@vue/cli-service": "~4.5.0",

"@vue/compiler-sfc": "^3.0.0",

"axios": "^0.21.1",

"babel-eslint": "^10.1.0",

"eslint": "^6.7.2",

"eslint-plugin-vue": "^7.0.0-0"

},

"eslintConfig": { "root": true,

"env": { "node": true

},

"extends": [

"plugin:vue/vue3-essential", "eslint:recommended"

],

"parserOptions": { "parser": "babel-eslint"

},

"rules": {}

},

"browserslist": [ "> 1%",

"last 2 versions", "not dead"

]

}

Listing 3.17

TypeScript Sebagai *Client* RESTful API

Untuk paket yang digunakan; bisa digunakan pada devDependencies. Jika *file* ini sudah ada, tinggal dilakukan “**npm install**”.

* 1. *src/App.vue*

<template>

<img alt="Vue logo" src="./assets/logo.png">

<HalamanAwal msg="Daftar Mahasiswa"/>

</template>

<script>

import HalamanAwal from './components/HalamanAwal.vue'

export default { name: 'App', components: { HalamanAwal

}

}

</script>

<style> #app {

font-family: Avenir, Helvetica, Arial, sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale; text-align: center;

color: #2c3e50; margin-top: 60px;

}

</style>

Listing 3.18 src/App.vue

<template>

<div class="hello">

<h2>{{ msg }}</h2>

<p>

<button @click='tampilkanMhs'> Tampilkan Data Mahasiswa

</button>

// <1>

**3.30** Praktikum-1: *Typescript* dan *Vue*



|  |
| --- |
| </p>  <p v-if=tampil>  <table class="center">  <caption>Daftar Mahasiswa Universitas Terbuka</caption>  <thead>  <tr>  <th>NIM</th>  <th>Nama</th>  <th>Tanggal Lahir</th>  <th>E-mail</th>  <th>Program Studi</th>  </tr>  </thead>  <tbody>  <tr v-for="dataMhs in dataMhsList"  :key="dataMhs.id"> // <2>  <td>{{ dataMhs.nim }}</td>  <td>{{ dataMhs.nama }}</td>  <td>{{ dataMhs.tglLahir }}</td>  <td>{{ dataMhs.email }}</td>  <td>{{ dataMhs.prodi }}</td>  </tr>  </tbody>  </table>  </p>  </div>  </template>  <script>  import axios from 'axios'; // <3>  export default {  name: 'HalamanAwal', data() {  return { tampil: false,  dataMhsList: [] // <4>  };  },  props: { msg: String  },  methods: {  tampilkanMhs() { // <5>  let AxiosInstance = axios.create(); let requestedData;  AxiosInstance.get('http://localhost:4000/mhs') // <6>  .then(  response => { // <7>  requestedData = response.data; |
| this.dataMhsList = requestedData; this.tampil = true; |

}).catch(error=> { console.log(error.message);

});

}

}

}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->

<style scoped> h3 {

margin: 40px 0 0;

}

ul {

list-style-type: none; padding: 0;

}

li {

display: inline-block; margin: 0 10px;

}

a {

color: #42b983;

}

.center {

margin-left: auto; margin-right: auto;

}

</style>

Listing 3.19 src/components/HalamanAwal.vue

{

"compilerOptions": { "target": "ES5", "module": "commonjs",

"strict": true, "esModuleInterop": true,

"experimentalDecorators": true, "emitDecoratorMetadata": true,

"skipLibCheck": true, "forceConsistentCasingInFileNames": true

}

}

Listing 3.20

*tsconfig.json* untuk Decorator

function buatMax(jumlah: number) {

// <1>

return (target: Object, propertyKey: string, descriptor: TypedPropertyDescriptor<any>) => {

// <2>

let oldMethod = descriptor.value

// <3>

descriptor.value = function(...args: any[]) {

// <4>

let returnedResult: string;

let result = oldMethod.apply(this, arguments)

// <5>

if (result.length > jumlah) {

returnedResult = result.substring(0, jumlah);

} else {

returnedResult = result;

}

// <6>

return returnedResult;

// <7>

}

}

}

class ClassA { // <8>

str1: string; constructor(teks: string) {

this.str1 = teks;

}

method1(newStr1: string): string { this.str1= newStr1;

return this.str1;

}

@buatMax(5)

method2(newStr2: string): string { this.str1= newStr2;

return this.str1;

}

// <9>

}

let kelasA = new ClassA("isi awal"); console.log(kelasA.str1);

console.log(kelasA.method1('lebih dari 5 karakter')); console.log(kelasA.method2('lebih dari 5 karakter'));

// <10>

// <11>

Listing 3.21

*Penggunaan Decorator*

$ tsc

$ node decorator.js isi awal

lebih dari 5 karakter lebih

$

**vue create ts-vue**

$ vue create ts-vue Vue CLI v4.5.10

? Please pick a preset:

Default ([Vue 2] babel, eslint)

Default (Vue 3 Preview) ([Vue 3] babel, eslint)

❯ Manually select features

? Check the features needed for your project:

* Choose Vue version
* Babel

❯◉ TypeScript

◯ Progressive Web App (PWA) Support

◯ Router

◯ Vuex

◯ CSS Pre-processors

* Linter / Formatter

◯ Unit Testing

◯ E2E Testing

? Choose a version of Vue.js that you want to start the project with 2.x

❯ 3.x (Preview)

? Use class-style component syntax? (y/N) y

? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes

? Pick a linter / formatter config:

❯ ESLint with error prevention only

? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection)

❯◉ Lint on save

◯ Lint and fix on commit

? Where do you prefer placing config for Babel, ESLint, etc.? (Use arrow keys)

❯ In dedicated config files

? Save this as a preset for future projects? (y/N) N

…

…

…

⚓ Running completion hooks...

📄 Generating README.md...

🎉 Successfully created project ts-vue.

👉 Get started with the following commands:

$ cd ts-vue

$ npm run serve

$

…

…

<script lang="ts">

import { Options, Vue } from 'vue-class-component';

// <1>

// <2>

@Options({ props: { msg: String

}

})

export default class HelloWorld extends Vue { msg!: string

}

</script>

…

…

// <3>

// <4>

Listing 3.22 src/components/HelloWorld.vue

$ vue add typescript

WARN There are uncommitted changes in the current repository, it's recommended to commit or stash them first.

? Still proceed? Yes

📦 Installing @vue/cli-plugin-typescript...

+ @vue/cli-plugin-typescript@4.5.11

added 49 packages from 42 contributors and audited 1371 packages in 10.165s

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

found 0 vulnerabilities

✔ Successfully installed plugin: @vue/cli-plugin-typescript

? Use class-style component syntax? Yes

? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes

? Convert all .js files to .ts? Yes

? Allow .js files to be compiled? No

? Skip type checking of all declaration files (recommended for apps)? Yes

🚀 Invoking generator for @vue/cli-plugin-typescript...

📦 Installing additional dependencies...

added 16 packages from 7 contributors and audited 1387 packages in 17.797s 76 packages are looking for funding

run `npm fund` for details

found 0 vulnerabilities

⚓ Running completion hooks...

✔ Successfully invoked generator for plugin: @vue/cli-plugin-typescript

Solusi Latihan

## KEGIATAN BELAJAR 3.1

Hasil akhir untuk kode sumber yang dibuat adalah sebagai berikut:

// paket-paket yang digunakan import axios from 'axios'; import \* as yargs from 'yargs';

import \* as fs from "fs/promises";

let HtmlTableToJson = require('html-table-to-json'); import \* as validateURI from "valid-url"

import \* as fsNode from "fs"; import \* as path from "path";

// menyiapkan berbagai opsi untuk yargs const argv = yargs.options({

url: {

alias: 'u', demandOption: true,

description: 'URL yang akan diambil'

},

output: {

alias: 'o',

default: "output.unk",

description: 'Hasil disimpan di file ini'

}

}).argv;

let url: string = argv.url as string;

let fhasil: string = argv.output as string;

// Latihan 1: memeriksa validitas URL

// menggunakan paket valid-url

// jangan lupa npm install valid-url --save-dev if (!validateURI.isUri(url)) {

console.log('URL yang anda masukkan tidak valid'); process.exit(1);

}

// Latihan 2: memeriksa apakah file sudah ada atau belum

// Jika sudah, keluar

if (fsNode.existsSync(fhasil)) { console.log('file sudah ada'); process.exit(1);

} else {

console.log('file ', fhasil, ' belum ada');

}

// Latihan 3: memeriksa akses direktori tempat file output

**MSIM4401**/MODUL 3 **3.49**



|  |
| --- |
| try { fsNode.accessSync(path.dirname(fhasil),  fsNode.constants.R\_OK | fsNode.constants.W\_OK | fsNode.constants.F\_OK); console.log('Hak akses direktori OK');  } catch (err) {  console.log('Hak akses direktori tidak cukup'); process.exit(1);  }  // persiapan untuk mengambil file menggunakan axios const AxiosInstance = axios.create();  let requestedData: string = ''; let ftype: string = '';  let dataToBeWritten: string = ''; let jsonTables: any;  AxiosInstance.get(url)  .then(  // file berhasil diambil response => {  requestedData = response.data;  ftype = response.headers['content-type'].substring(0,9);  // Jika HTML, maka cari table. Jika ada, isikan data tabel  // ke JSON.  if (ftype === 'text/html') {  jsonTables = HtmlTableToJson.parse(requestedData); console.log('Jumlah tabel: ', jsonTables.count);  if (jsonTables.count > 0) {  dataToBeWritten = JSON.stringify(jsonTables.results);  } else {  console.log('Tidak ada tabel di HTML, menulis HTML ke file'); dataToBeWritten = requestedData;  }  } else {  // jika bukan HTML, langsung ditulis apa adanya. dataToBeWritten = requestedData;  }  // Simpan ke nama file (async function(f) {  try {  await fs.writeFile(f, dataToBeWritten); console.log(`Disimpan ke file ${f}`);  } catch (error) { |
| console.log('Error:', error.message);  } |

})(fhasil);

}).catch(error=> { console.log(error.message);

}

);

## KEGIATAN BELAJAR 3.2

**Endpoint**

*src/index.ts*

import express from 'express'; import cors from 'cors';

import { PORT } from './config/constants'; import { mhsRouter } from './routes'; import { dosenRouter } from './routes';

const app = express(); app.use(express.json());

app.use(cors()); app.get('/', (req, res) => {

res.send('Selamat datang di RESTful API gateway - SI Akademik');

});

app.use('/mhs', mhsRouter); app.use('/dosen', dosenRouter);

app.listen(PORT, () => {

console.log(`Endpoint Akademik sudah siap dan bisa diakses di port

${PORT}`);

});

*src/controllers/index.ts*

import { MhsController } from './Mhs/Mhs'; import { DosenController } from './Dosen/Dosen';

const mhsController = new MhsController(); const dosenController = new DosenController();

export { mhsController, dosenController

};

**MSIM4401**/MODUL 3 **3.51**



*src/controllers/Dosen/Dosen.ts*

import { Request, Response } from 'express'; import { CrudController } from '../CrudController';

import dataDosenJson from '../../resources/dataDosen.json'; export class DosenController extends CrudController {

public read(req: Request<import("express-serve-static-core").

ParamsDictionary>, res: Response): void { res.json(dataDosenJson);

}

}

*src/resources/dataDosen.json*

[

{

"nidn": 123456789,

"nama": "Dosen 1",

"tglLahir": "10-05-1980",

"alamat": "Alamat Dosen 1", "homebase": "Sistem Informasi"

},

{

"nidn": 139893842,

"nama": "Dosen 2",

"tglLahir": "29-06-1979",

"alamat": "Alamat Dosen 2", "homebase": "Informatika"

},

{

"nidn": 142323223,

"nama": "Dosen 3",

"tglLahir": "11-12-1985",

"alamat": "Alamat Dosen 3", "homebase": "Akuntansi"

},

{

"nidn": 153232243,

"nama": "Dosen 4",

"tglLahir": "12-08-1990",

"alamat": "Alamat Dosen 4", "homebase": "Informatika"

},

{

"nidn": 167384738,

"nama": "Dosen 5",

"tglLahir": "19-01-1985",

"alamat": "Alamat Dosen 5", "homebase": "Sains Data"

}

]

*src/routes/index.ts*

import { router as mhsRouter } from './Mhs/Mhs'; import { router as dosenRouter } from './Dosen/Dosen';

export { mhsRouter, dosenRouter

};

*src/routes/Dosen/Dosen.ts*

import express, { Request, Response } from 'express'; import { dosenController } from '../../controllers';

export const router = express.Router({ strict: true

});

router.get('/', (req: Request, res: Response) => { dosenController.read(req, res);

});

## Client

*src/App.vue (parsial)*

<template>

<img alt="Vue logo" src="./assets/logo.png">

<HalamanAwal msg="Daftar Mahasiswa dan Dosen"/>

…

*src/components/HalamanAwal.vue (parsial - tanpa style)*

<template>

<div class="hello">

<h2>{{ msg }}</h2>

<p>

<button @click='tampilkanMhsDosen'>Tampilkan Data Mahasiswa dan Dosen</button>

</p>

<p v-if=tampil>

<table class="center">

**MSIM4401**/MODUL 3 **3.53**



|  |
| --- |
| <caption>Daftar Dosen Universitas Terbuka</caption>  <thead>  <tr>  <th>NIDN</th>  <th>Nama</th>  <th>Tanggal Lahir</th>  <th>Alamat</th>  <th>Homebase</th>  </tr>  </thead>  <tbody>  <tr v-for="dataDosen in dataDosenList" :key="dataDosen.id">  <td>{{ dataDosen.nidn }}</td>  <td>{{ dataDosen.nama }}</td>  <td>{{ dataDosen.tglLahir }}</td>  <td>{{ dataDosen.alamat }}</td>  <td>{{ dataDosen.homebase }}</td>  </tr>  </tbody>  </table>  <table class="center">  <caption>Daftar Mahasiswa Universitas Terbuka</caption>  <thead>  <tr>  <th>NIM</th>  <th>Nama</th>  <th>Tanggal Lahir</th>  <th>E-mail</th>  <th>Program Studi</th>  </tr>  </thead>  <tbody>  <tr v-for="dataMhs in dataMhsList" :key="dataMhs.id">  <td>{{ dataMhs.nim }}</td>  <td>{{ dataMhs.nama }}</td>  <td>{{ dataMhs.tglLahir }}</td>  <td>{{ dataMhs.email }}</td>  <td>{{ dataMhs.prodi }}</td>  </tr>  </tbody>  </table>  </p>  </div>  </template>  <script>  import axios from 'axios'; |
| export default {  name: 'HalamanAwal', data() { |

return { tampil: false,

dataMhsList: [], dataDosenList: []

};

},

props: { msg: String

},

methods: { tampilkanMhsDosen() {

let AxiosInstance = axios.create(); let requestedMhs;

let requestedDosen;

AxiosInstance.get('http://192.168.1.4:4000/mhs')

.then( response => {

requestedMhs = response.data;

this.dataMhsList = requestedMhs; this.tampil = true;

}).catch(error=> { console.log(error.message);

});

AxiosInstance.get('http://192.168.1.4:4000/dosen')

.then( response => {

requestedDosen = response.data;

this.dataDosenList = requestedDosen; this.tampil = true;

}).catch(error=> { console.log(error.message);

});

}

}

}

</script>

…

…

…

## KEGIATAN BELAJAR 3.3

Buat proyek Vue baru menggunakan vue-cli. Pilihlah fitur vue-class-component.

Setelah itu ubah *src/components/HelloWorld.vue*.

<template>

<div class="hello">

<h2>{{ msg }}</h2>

<p>Jumlah detik: {{ counter }}</p>

<p>

<button @click="counterOnOff">{{ teksButton }}</button>

</p>

</div>

</template>

<script lang="ts">

import { Options, Vue } from 'vue-class-component';

@Options({ props: { msg: String

},

data() { return { counter: 0,

teksButton: 'Hitung!', interval: 0

}

}

})

export default class HelloWorld extends Vue {

// menggunakan ! supaya tidak perlu masuk

// ke konstruktur. Bagian ini sama dengan

// data()

counter!: number; teksButton!: string; interval!: number;

// akan dikerjakan setiap setInterval tambahSatu() {

this.counter += 1;

}

counterOnOff() {

if (this.teksButton === 'Hitung!') { this.teksButton = 'Berhenti Menghitung!';

// setiap 1 detik, akan mengerjakan tambahSatu this.interval = setInterval(this.tambahSatu, 1000);

} else {

this.teksButton = 'Hitung!';

// Jika sudah selesai / Off, maka bersihkan interval

// clearInterval(this.interval);

}

}

}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->

<style scoped> h3 {

margin: 40px 0 0;

}

ul {

list-style-type: none; padding: 0;

}

li {

display: inline-block; margin: 0 10px;

}

a {

color: #42b983;

}

</style>