

PRAKTIKUM PEMROGRAMAN WEB



MANUAL BOOK

“CRUD MENGGUNAKAN REACT JS DAN NODEJS ”

NAMA ANGGOTA:

- | | |
|--------------------------|------------|
| 1. ALI AKBAR SAID | (50421119) |
| 2. ALWAN DZAKI SYAEFUDIN | (50421135) |
| 3. ANGGA CAHYO KRISNANTO | (50421165) |
| 4. TRESNA SATYA SANRAE | (51421483) |

KELAS : 3 IA 03

FAKULTAS : TEKNOLOGI INDUSTRI

JURUSAN : INFORMATIKA

PJ : PUTRA MUHAMMAD

Ditulis Guna Melengkapi Sebagai Syarat Praktikum Pemrograman Web

Jenjang S1

Universitas Gunadarma

2023

DAFTAR ISI

DAFTAR ISI.....	i
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan	1
BAB II.....	2
PEMBAHASAN	2
2.1 Pengertian.....	2
2.2 ReactJs.....	2
2.2.1 Manfaat Reactjs	3
2.2.2 Kerugian Reactjs.....	3
2.3 NodeJs	3
2.3.1 Manfaat NodeJs	4
2.3.2 Kerugian NodeJs.....	4
BAB III.....	5
ANALISA DAN PERANCANGAN	5
BAB IV.....	22
PENUTUP.....	22
KESIMPULAN	22
SARAN	22

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era sekarang, website sudah menjadi kebutuhan utama yang tidak bisa diabaikan. Seluruh sektor bisnis atau edukasi dapat memanfaatkan website sebagai alat untuk promosi, tukar informasi, dan lainnya. Web Programming atau Pemrograman Web merupakan istilah yang erat kaitannya dengan internet dan website. Jika diumpamakan, web programming itu seperti arsitek yang merancang sebuah rumah. Tukang bangunan dengan segala alat dan bahannya adalah komputer. Lalu rumah adalah program atau website yang berhasil dibangun. Awalnya, web terdiri dari halaman-halaman statis dengan tautan antar halaman yang sederhana. Namun, dengan kemajuan teknologi dan munculnya bahasa pemrograman membuat web berkembang menjadi platform yang dinamis. Perkembangan ini memungkinkan pembangunan situs dan aplikasi yang tidak hanya menampilkan informasi tetapi juga berinteraksi dengan pengguna secara real-time.

1.2 Tujuan

Dalam penulisan manual book kali ini bertujuan untuk membantu dalam memahami dan mengetahui langkah - langkah pembuatan website CRUD menggunakan reactjs dan nodejs.

BAB II

PEMBAHASAN

1.1 Pengertian

Pemrograman web atau web programming adalah proses pengkomunikasian perintah pada komputer untuk membangun suatu program berbasis website agar dapat diakses menggunakan internet. Pemrograman web diambil dari 2 suku kata, yaitu pemrograman dan web. Pemrograman yang dalam bahasa inggris ialah programming dan diartikan proses, cara, pembuatan program. Definisi web ialah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks, grafik, suara, dan sumber daya animasi melalui protokol transfer hypertext.

1.2 ReactJs

ReactJS atau React adalah library JavaScript populer buatan Facebook yang digunakan dalam proses pengembangan aplikasi mobile dan web. React dirancang untuk membangun antarmuka pengguna (UI) yang efisien dan memungkinkan pengembangan aplikasi web yang scalable. React digunakan untuk membangun antarmuka pengguna yang dinamis dan responsif dengan memecah UI menjadi komponen-komponen yang terpisah. ReactJS bukanlah framework JavaScript, karena hanya bertugas untuk merender komponen area tampilan aplikasi.

1.2.1 Manfaat Reactjs

Banyak pengguna yang menggunakan framework ini karena memiliki banyak kelebihan yang menarik, yaitu penulisan coding-nya yang deklaratif. Oleh karena itu, pengguna bisa memahami kode javascript yang ada dengan mudah. Selain itu, framework ini dapat dipakai untuk project dengan skala kecil dan besar. Framework ini dapat menampilkan perubahan data yang terjadi secara kompleks dengan lengkap dan akurat.

1.2.2 Kerugian Reactjs

Meskipun memiliki banyak manfaat, framework ini juga memiliki kerugian, salah satunya ialah dokumentasi yang berantakan. Secara umum, dokumentasi framework lainnya lebih rapi dan lengkap dari reactjs. Selain itu, kompatibilitas framework ini memerlukan versi browser terbaru. Pengguna harus meng-update versi browser terlebih dahulu jika ingin menggunakan framework ini.

1.3 NodeJs

Node.js adalah runtime environment untuk JavaScript yang bersifat open-source dan cross-platform. Dengan Node.js kita dapat menjalankan kode JavaScript di mana pun, tidak hanya terbatas pada lingkungan browser. Node.js menjalankan V8 JavaScript engine (yang juga merupakan inti dari Google Chrome) di luar browser. Ini memungkinkan Node.js memiliki performa yang tinggi. Node.js juga menyediakan banyak library/module JavaScript yang membantu menyederhanakan pengembangan aplikasi web. Node.js sangat populer untuk pengembangan aplikasi real-time seperti aplikasi obrolan atau chat, permainan daring, dan aplikasi kolaborasi langsung karena kemampuannya menangani banyak koneksi secara bersamaan.

1.3.1 Manfaat NodeJs

Manfaat menggunakan nodejs ialah open source, yaitu gratis untuk digunakan bagi pengguna dan dapat dijalankan di seluruh os, yaitu windows, mac os, dan linux. Framework ini menggunakan mesin v8 yang dikembangkan oleh google, sehingga eksekusi kode menjadi cepat dan efisien.

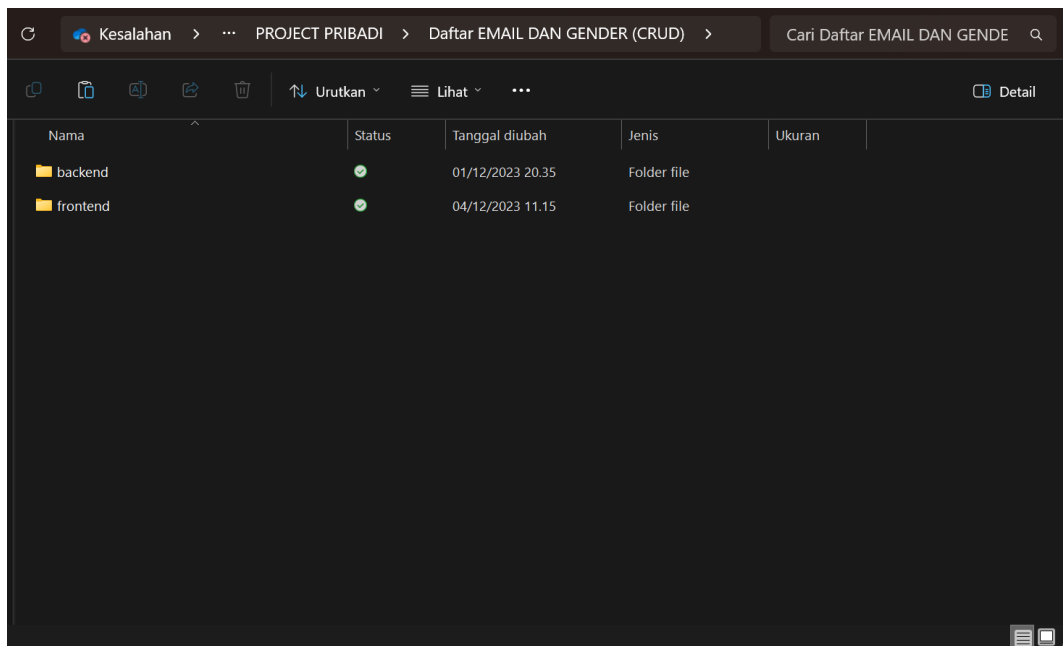
1.3.2 Kerugian NodeJs

Terdapat beberapa kerugian memakai framework ini, salah satunya ialah nodejs tidak efektif dengan aplikasi skala besar. Nodejs tidak mendukung pemrograman multi-utas. Nodejs bersifat single-threaded yang dapat menjadi kerugian jika aplikasi membutuhkan pemrosesan berat secara bersamaan.

BAB III

ANALISA DAN PERANCANGAN

Project yang kita buat yaitu bertema Crud dengan menggunakan React JS dan library-nya menggunakan nodemon. Tampilan dibawah ini merupakan folder dari library nodemon.



Nama	Status	Tanggal diubah	Jenis	Ukuran
backend	✓	01/12/2023 20:35	Folder file	
frontend	✓	04/12/2023 11:15	Folder file	

Masuk ke folder frontend -> src -> index.js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import App from './App';
4 import "bulma/css/bulma.css";
5
6 const root = ReactDOM.createRoot(document.getElementById('root'));
7 root.render(
8   <React.StrictMode>
9     <App />
10  </React.StrictMode>
11 );
```

Penjelasan:

- **import React from 'react'**: Mengimpor library React untuk membuat dan mengelola komponen React.
- **import ReactDOM from 'react-dom/client'**: Mengimpor library ReactDOM dari package `react-dom/client` untuk rendering komponen React ke DOM.
- **import App from './App'**: Mengimpor komponen `App` dari file `./App`. Komponen ini kemungkinan besar merupakan komponen utama dari aplikasi.
- **"bulma/css/bulma.css"**: Mengimpor file CSS dari Bulma, sebuah framework CSS, untuk mempercantik tata letak dan tampilan aplikasi.
- **Const root**: Membuat root (akar) dari aplikasi React menggunakan `createRoot` dari ReactDOM, dan menetapkan elemen HTML dengan id 'root' sebagai tempat di mana aplikasi akan dirender.
- **root.render(...)** : Menggunakan metode `render` pada root untuk merender komponen React. Dalam kasus ini, menggunakan `React.StrictMode` untuk memberikan peringatan tambahan selama pengembangan. Komponen `App` di-render di dalam `StrictMode`.

Masuk ke folder frontend -> src -> App.js

```
1 import { BrowserRouter, Routes, Route } from 'react-router-dom';
2 import UserList from './components/UserList';
3 import AddUser from './components/AddUser';
4 import EditUser from './components/EditUser';
5
6 function App() {
7   return (
8     <BrowserRouter>
9       <Routes>
10        <Route path="/" element={<UserList />} />
11        <Route path="add" element={<AddUser />} />
12        <Route path="edit/:id" element={<EditUser />} />
13      </Routes>
14    </BrowserRouter>
15  );
16 }
17
18 export default App;
```

Penjelasan:

- **import { BrowserRouter, Routes, Route } from 'react-router-dom':** Baris ini mengimpor komponen yang diperlukan dari pustaka react-router-dom, termasuk BrowserRouter untuk menangani rute menggunakan browser history, Routes untuk mendefinisikan rute-rute, dan Route untuk menetapkan komponen yang harus ditampilkan sesuai dengan rute tertentu.
- **function App() { ... }:** Ini adalah fungsi komponen utama yang akan merender komponen-komponen terkait dengan rute-rute yang telah didefinisikan.
- **<BrowserRouter>:** Ini adalah komponen pembungkus yang menyediakan konteks untuk pengelolaan rute. Dengan menggunakan BrowserRouter, kita dapat mengaktifkan routing berbasis peramban (browser) dalam aplikasi React.

- **<Routes>**: Ini adalah komponen yang digunakan untuk mendefinisikan rute-rute dalam aplikasi. Di dalamnya, kita menetapkan rute-rute menggunakan elemen **<Route>**.
- **<Route path="/" element={<UserList />} />**: Ini mendefinisikan rute utama atau beranda (/) dan menetapkan komponen **<UserList />** untuk ditampilkan ketika rute ini dicocokkan.
- **<Route path="/add" element={<AddUser />} />**: Ini mendefinisikan rute dengan path "/add" dan menetapkan komponen **<AddUser />** untuk ditampilkan ketika rute ini dicocokkan.
- **<Route path="/edit/:id" element={<EditUser />} />**: Ini mendefinisikan rute dinamis dengan path "/edit/:id", di mana ":id" adalah parameter dinamis yang dapat berubah. Komponen **<EditUser />** akan ditampilkan dan memiliki akses ke nilai parameter ":id".
- **export default App**: Mengekspor fungsi komponen App agar dapat digunakan di file lain.

Masuk ke folder frontend -> src -> components -> AddUser.js

```
1 import React, {useState } from "react";
2 import axios from "axios";
3 import {useNavigate} from "react-router-dom";
4
5 const AddUser = () => {
6   const [name, setName] = useState("");
7   const [email, setEmail] = useState("");
8   const [gender, setGender] = useState("Male");
9   const navigate = useNavigate();
10
11   const saveUser = async (e) => {
12     e.preventDefault();
13     try {
14       await axios.post("http://localhost:5000/users", {
15         name,
16         email,
17         gender,
18       });
19       navigate("/");
20     } catch (error) {
21       console.log(error);
22     }
23   };
24
25   return (
26     <div className="columns mt-5 is-centered">
27       <div className="column is-half">
28         <form onSubmit={saveUser}>
29           <div className="field">
30             <label className="label">Name</label>
31             <div className="control">
32               <input
33                 type="text"
34                 className="input"
35                 value={name}
36                 onChange={(e) => setName(e.target.value)}
37                 placeholder="Name"
38               />
39             </div>
40           </div>
41           <div className="field">
42             <label className="label">Email</label>
43             <div className="control">
44               <input
45                 type="text"
46                 className="input"
47                 value={email}
48                 onChange={(e) => setEmail(e.target.value)}
49                 placeholder="Email"
50               />
51             </div>
52           </div>
53           <div className="field">
54             <label className="label">Gender</label>
55             <div className="control">
56               <div className="select is-fullwidth">
57                 <select
58                   value={gender}
59                   onChange={(e) => setGender(e.target.value)}
60                 >
61                   <option value="Male">Male</option>
62                   <option value="Female">Female</option>
63                 </select>
64               </div>
65             </div>
66           </div>
67           <div className="field">
68             <button type="submit" className="button is-success">
69               Save
70             </button>
71           </div>
72         </form>
73       </div>
74     </div>
75   );
76 };
77
78 export default AddUser;
```

Penjelasan:

- **State Hooks:** Membuat variabel name, email, dan gender untuk menyimpan nilai dari formulir pengguna menggunakan useState.
- **Navigasi:** Menggunakan hook useNavigate dari React Router untuk menangani navigasi antar halaman.
- **Fungsi saveUser:** Mencegah pengiriman formulir, mengirimkan data pengguna baru ke server menggunakan Axios, dan mengarahkan pengguna kembali ke halaman utama setelah berhasil menambahkan pengguna.
- **Render Formulir:** Membuat tampilan formulir dengan input untuk nama, email, dan pilihan gender.
- **Event Handlers:** Mengatur fungsi-fungsi yang akan dijalankan saat nilai input berubah, untuk memperbarui state yang sesuai.
- **Tombol Simpan:** Menambahkan tombol "Save" yang akan memicu fungsi saveUser saat ditekan.

Masuk ke folder frontend -> src -> components -> EditUser.js

```
1 import React, {useState, useEffect} from "react";
2 import axios from "axios";
3 import {useNavigate, useParams} from "react-router-dom";
4
5 const EditUser = () => {
6   const [name, setName] = useState("");
7   const [email, setEmail] = useState("");
8   const [gender, setGender] = useState("Male");
9   const navigate = useNavigate();
10  const { id } = useParams();
11
12  useEffect(() => {
13    getUserById();
14  }, []);
15
16  const updateUser = async (e) => {
17    e.preventDefault();
18    try {
19      await axios.patch( `http://localhost:5000/users/${id}`, {
20        name,
21        email,
22        gender,
23      });
24      navigate("/");
25    } catch (error) {
26      console.log(error);
27    }
28  };
29
30  const getUserById = async () => {
31    const response = await axios.get( `http://localhost:5000/users/${id}` );
32    setName(response.data.name);
33    setEmail(response.data.email);
34    setGender(response.data.gender);
35  };
36
37  return (
38    <div className="columns mt-5 is-centered">
39      <div className="column is-half">
40        <form onSubmit={updateUser}>
41          <div className="field">
42            <label className="label">Name</label>
43            <div className="control">
44              <input
45                type="text"
46                className="input"
47                value={name}
48                onChange={(e) => setName(e.target.value)}
49                placeholder="Name"
50              />
51            </div>
52          </div>
53          <div className="field">
54            <label className="label">Email</label>
55            <div className="control">
56              <input
57                type="text"
58                className="input"
59                value={email}
60                onChange={(e) => setEmail(e.target.value)}
61                placeholder="Email"
62              />
63            </div>
64          </div>
65          <div className="field">
66            <label className="label">Gender</label>
67            <div className="control">
68              <div className="select is-fullwidth">
69                <select
70                  value={gender}
71                  onChange={(e) => setGender(e.target.value)}
72                >
73                  <option value="Male">Male</option>
74                  <option value="Female">Female</option>
75                </select>
76              </div>
77            </div>
78          </div>
79          <div className="field">
80            <button type="submit" className="button is-success">
81              Update
82            </button>
83          </div>
84        </form>
85      </div>
86    </div>
87  );
88 };
89
90 export default EditUser;
```

Penjelasan:

- **State Hooks:** Membuat variabel name, email, dan gender menggunakan useState untuk menyimpan nilai dari formulir pengguna.
- **Navigasi dan Parameter URL:** Menggunakan useNavigate dan useParams dari React Router untuk menangani navigasi dan mendapatkan parameter ID dari URL.
- **Efek Samping (Effect Hook):** Menggunakan useEffect untuk menjalankan fungsi getUserById saat komponen pertama kali di-mount (saat halaman dimuat).
- **Fungsi updateUser:** Mencegah pengiriman formulir dengan memanggil e.preventDefault(). Menggunakan Axios untuk melakukan permintaan PATCH ke endpoint "http://localhost:5000/users/\${id}" dengan data yang diperbarui. Mengarahkan pengguna kembali ke halaman utama setelah berhasil memperbarui.
- **Fungsi getUserById:** Menggunakan Axios untuk mendapatkan data pengguna berdasarkan ID dari server. Mengatur nilai name, email, dan gender berdasarkan data pengguna yang diterima.
- **Render Formulir:** Membuat tampilan formulir dengan input untuk nama, email, dan pilihan gender.
- **Event Handlers:** Mengatur fungsi-fungsi yang akan dijalankan saat nilai input berubah, untuk memperbarui state yang sesuai.
- **Tombol Update:** Tombol "Update" akan memicu fungsi updateUser saat ditekan.

Masuk ke folder frontend -> src -> components -> UserList.js

```
1 import React, { useState, useEffect } from 'react';
2 import axios from "axios";
3 import {Link} from "react-router-dom";
4
5 const UserList = () => {
6   const [users, setUser] = useState([]);
7
8   useEffect(() => {
9     getUsers();
10  }, []);
11
12  const getUsers = async () => {
13    const response = await axios.get("http://localhost:5000/users");
14    setUser(response.data);
15  };
16
17  const deleteUser = async (id) => {
18    try {
19      await axios.delete(`http://localhost:5000/users/${id}`);
20      getUsers();
21    } catch (error) {
22      console.log(error);
23    }
24  };
25
26  return (
27    <div className="columns mt-5 is-centered">
28      <div className="column is-half">
29        <Link to={'add'} className="button is-success">
30          Add New
31        </Link>
32        <table className="table is-striped is-fullwidth">
33          <thead>
34            <tr>
35              <th>No</th>
36              <th>Name</th>
37              <th>Email</th>
38              <th>Gender</th>
39              <th>Actions</th>
40            </tr>
41          </thead>
42          <tbody>
43            {users.map((user, index) => (
44              <tr key={user.id}>
45                <td>{index + 1}</td>
46                <td>{user.name}</td>
47                <td>{user.email}</td>
48                <td>{user.gender}</td>
49                <td>
50                  <Link
51                    to={'edit/${user.id}'}
52                    className="button is-small is-info mr-2"
53                  >
54                    Edit
55                  </Link>
56                  <button
57                    onClick={() => deleteUser(user.id)}
58                    className="button is-small is-danger"
59                  >
60                    Delete
61                  </button>
62                </td>
63              </tr>
64            ))}
65          </tbody>
66        </table>
67      </div>
68    </div>
69  );
70 };
71
72 export default UserList;
73
```

Penjelasan:

- State Hooks: Membuat variabel users menggunakan useState untuk menyimpan daftar pengguna yang akan ditampilkan.
- Efek Samping (Effect Hook): Menggunakan useEffect untuk menjalankan fungsi getUsers saat komponen pertama kali di-mount (saat halaman dimuat).
- Fungsi getUsers: Menggunakan Axios untuk mendapatkan daftar pengguna dari server. Memperbarui state users dengan data pengguna yang diterima.
- Fungsi deleteUser: Menggunakan Axios untuk mengirim permintaan DELETE ke server untuk menghapus pengguna berdasarkan ID. Setelah penghapusan berhasil, memanggil kembali fungsi getUsers untuk memperbarui daftar pengguna.
- Render Tampilan: Menampilkan tombol "Add New" menggunakan Link dari React Router untuk mengarahkan ke halaman penambahan pengguna. Menampilkan tabel dengan daftar pengguna yang diterima dari server. Setiap baris tabel menunjukkan informasi pengguna dan menyediakan tautan "Edit" untuk mengedit, serta tombol "Delete" untuk menghapus.
- Looping Melalui Daftar Pengguna: Menggunakan metode map untuk melakukan iterasi melalui setiap pengguna dalam daftar users. Membuat baris tabel untuk setiap pengguna dengan menampilkan nomor urut, nama, email, gender, dan tombol untuk mengedit dan menghapus.
- Tautan dan Tombol Aksi: Tombol "Add New" mengarahkan ke halaman penambahan pengguna. Tombol "Edit" di setiap baris tabel mengarahkan ke halaman pengeditan pengguna dengan menggunakan Link. Tombol "Delete" di setiap baris tabel memicu fungsi deleteUser untuk menghapus pengguna.

UserControl.js

```
1 import User from "../models/UserModel.js";
2
3 export const getUsers = async(req, res) =>{
4   try {
5     const response = await User.findAll();
6     res.status(200).json(response);
7   } catch (error) {
8     console.log(error.message);
9   }
10 }
11
12 //Mengambil Single Data
13 export const getUserById = async(req, res) =>{
14   try {
15     const response = await User.findOne({
16       where:{
17         id: req.params.id
18       }
19     });
20     res.status(200).json(response);
21   } catch (error) {
22     console.log(error.message);
23   }
24 }
25
26 //CREATE
27 export const createUser = async(req, res) =>{
28   try {
29     await User.create(req.body);
30     res.status(201).json({msg: "User Created"});
31   } catch (error) {
32     console.log(error.message);
33   }
34 }
35
36 //UPDATE
37 export const updateUser = async(req, res) =>{
38   try {
39     await User.update(req.body,{
40       where:{
41         id: req.params.id
42       }
43     });
44     res.status(200).json({msg: "User Updated"});
45   } catch (error) {
46     console.log(error.message);
47   }
48 }
49
50 //DELETE
51 export const deleteUser = async(req, res) =>{
52   try {
53     await User.destroy({
54       where:{
55         id: req.params.id
56       }
57     });
58     res.status(200).json({msg: "User Deleted"});
59   } catch (error) {
60     console.log(error.message);
61   }
62 }
```

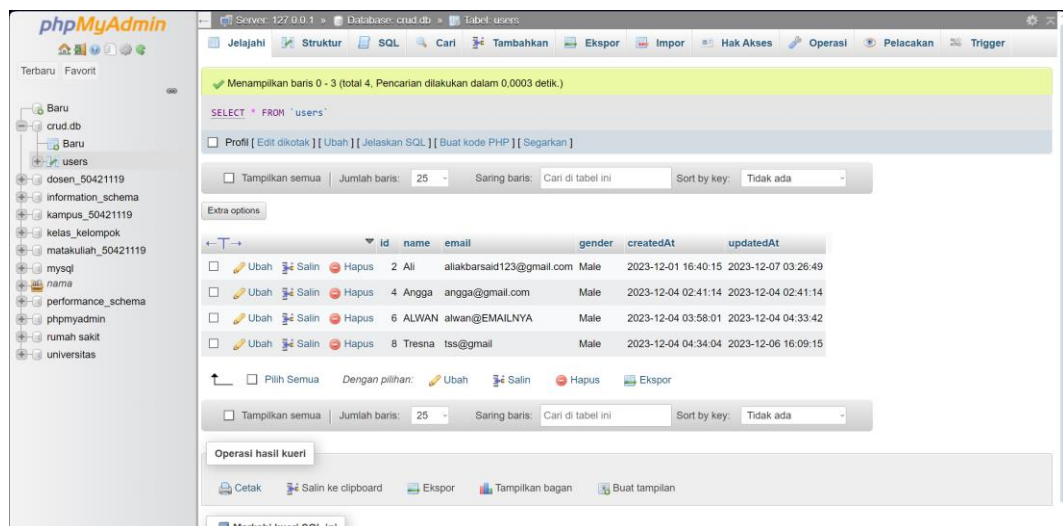


```
1  GET http://localhost:5000/users
2
3  ###
4  GET http://localhost:5000/users/1
5
6  ###
7  POST http://localhost:5000/users
8  Content-Type: application/json
9
10 {
11     "name": "Ali Akbar Said",
12     "email": "aliakbarsaid@gmail.com",
13     "gender": "Male"
14 }
15
16 ###
17 PATCH http://localhost:5000/users/1
18 Content-Type: application/json
19
20 {
21     "name": "a.a.said_18",
22     "email": "aliakbarsaid123@gmail.com",
23     "gender": "Male"
24 }
25
26 ###
27 DELETE http://localhost:5000/users/1
```

Menyambungkan ke Database

```
1 import {Sequelize} from "sequelize";
2
3 const db = new Sequelize('crud.db', 'root', '',{
4     host: 'localhost',
5     dialect: 'mysql'
6 });
7
8 export default db;
```

Database yang digunakan:



Melakukan running cd backend -> nodemon index

```
History restored
PS C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)> cd backend
PS C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)\backend> nodemon index
[nodemon] 3.0.1
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node index.js'
Server up and running...
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = 'users' AND TABLE_SCHEMA = 'crud.db'
Executing (default): SHOW INDEX FROM 'users'
Executing (default): SELECT 'id', 'name', 'email', 'gender', 'createdAt', 'updatedAt' FROM 'users' AS 'users';
Executing (default): SELECT 'id', 'name', 'email', 'gender', 'createdAt', 'updatedAt' FROM 'users' AS 'users';
[]
```

cd frontend -> npm start

```
Windows PowerShell
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)>code .
C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)>cd cd frontend
The system cannot find the path specified.
C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)>cd frontend
C:\Users\Ali Akbar Said\OneDrive\Dokumen\PROJECT PRIBADI\Daftar EMAIL DAN GENDER (CRUD)\frontend>npm start

> frontend@0.1.0 start
> react-scripts start

(node:5716) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is de
precated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:5716) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is
deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

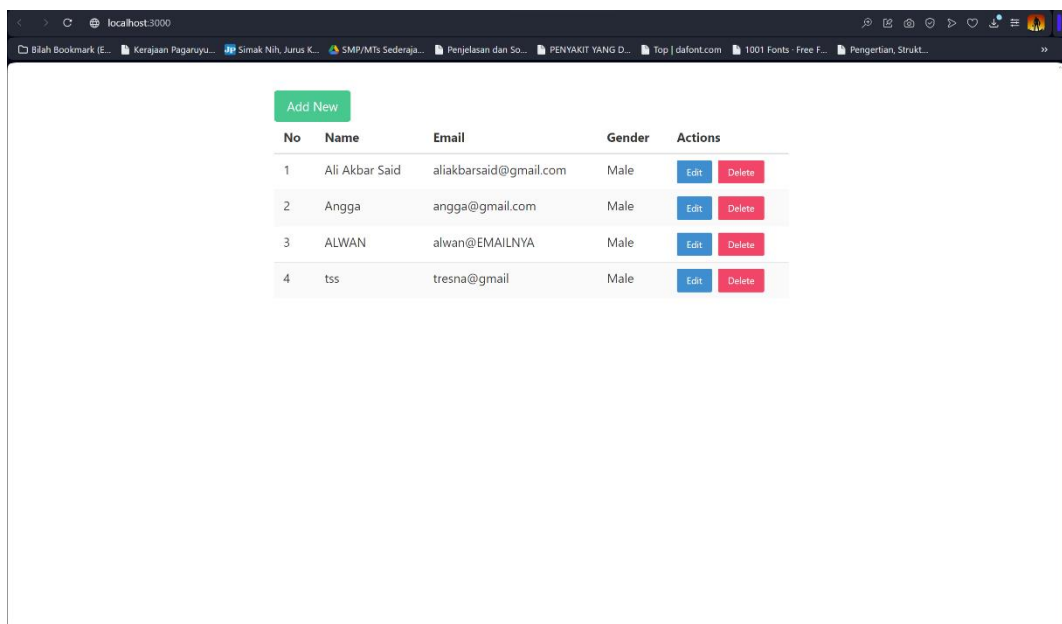
You can now view frontend in the browser.

Local:      http://localhost:3000
On Your Network:  http://192.168.100.162:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Maka tampilan yang akan terlihat seperti ini



Jika mengedit maka seperti ini

localhost:3000/edit/8

Bilah Bookmark (E...Kerajaan Paganuyu...Simak Nih, Jurus K...SMP/MTs Sederaja...Penjelasan dan So...PENYAKIT YANG D...Top | dafont.com1001 Fonts - Free F...Pengertian, Strukt...

Name

Tresna

Email

tss@gmail

Gender

Male

Update

localhost:3000

Bilah Bookmark (E...Kerajaan Paganuyu...Simak Nih, Jurus K...SMP/MTs Sederaja...Penjelasan dan So...PENYAKIT YANG D...Top | dafont.com1001 Fonts - Free F...Pengertian, Strukt...

Add New

No	Name	Email	Gender	Actions
1	Ali Akbar Said	aliakbarsaid@gmail.com	Male	<div>EdikDelete</div>
2	Angga	angga@gmail.com	Male	<div>EdikDelete</div>
3	ALWAN	alwan@EMAILNYA	Male	<div>EdikDelete</div>
4	Tresna	tss@gmail	Male	<div>EdikDelete</div>

Jika di add maka tampilannya seperti ini

The image shows two screenshots of a web application running on localhost:3000. The top screenshot displays a form for adding a new user with fields for Name, Email, and Gender, and a Save button. The bottom screenshot displays a table of existing users with columns for No, Name, Email, Gender, and Actions (Edit, Delete).

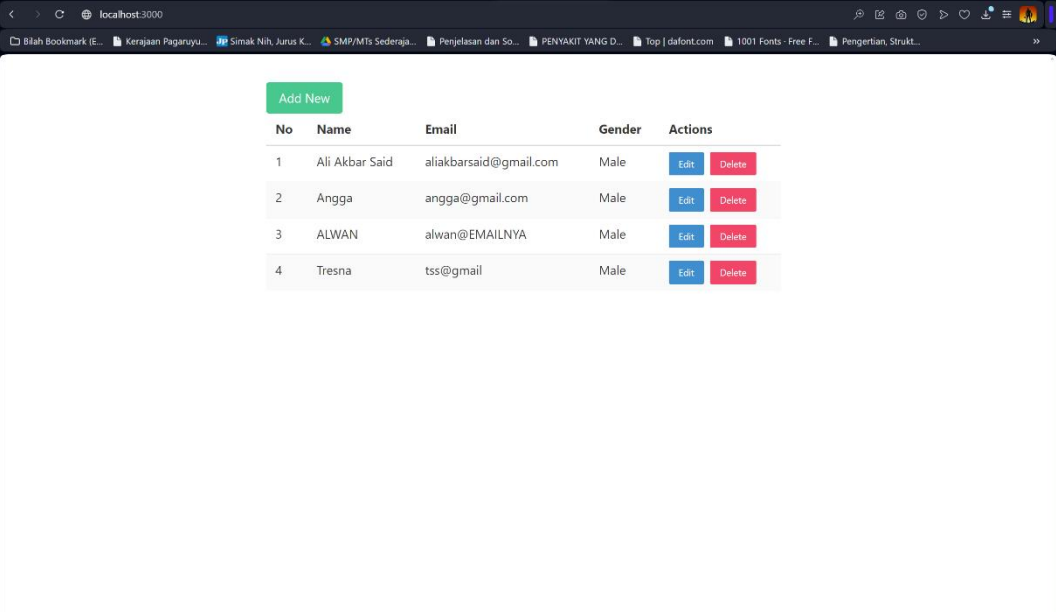
Form Fields:

- Name: orang baru
- Email: lg@email.nya
- Gender: Female
- Save button

Table of Existing Users:

No	Name	Email	Gender	Actions
1	Ali Akbar Said	aliakbarsaid@gmail.com	Male	Edit Delete
2	Angga	angga@gmail.com	Male	Edit Delete
3	ALWAN	alwan@EMAILNYA	Male	Edit Delete
4	Tresna	tss@gmail	Male	Edit Delete
5	orang baru	lg@email.nya	Female	Edit Delete

Jika di delete maka tampilannya seperti ini



localhost:3000

Blah Bookmark (E... Kerajaan Paganuyu... Simak Nih, Jurus K... SMP/MTs Sederaja... Penjelasan dan So... PENYAKIT YANG D... Top | dafont.com 1001 Fonts - Free F... Pengertian, Strukt...

Add New

No	Name	Email	Gender	Actions
1	Ali Akbar Said	aliakbarsaid@gmail.com	Male	Edit Delete
2	Angga	angga@gmail.com	Male	Edit Delete
3	ALWAN	alwan@EMAILNYA	Male	Edit Delete
4	Tresna	tss@gmail	Male	Edit Delete

BAB IV

PENUTUP

KESIMPULAN

Dengan adanya praktikum ini, kami dapat mempelajari tentang struktur perancangan pembuatan web dengan mengintegrasikannya bersama database, serta dapat mengimplementasikannya ke dalam project yang telah kami buat ini.

SARAN

Pada project ini, diharapkan mempersiapkan beberapa tools untuk menjalankan program CRUD. Pastikan mensetting tools dengan baik untuk menjalankan program.