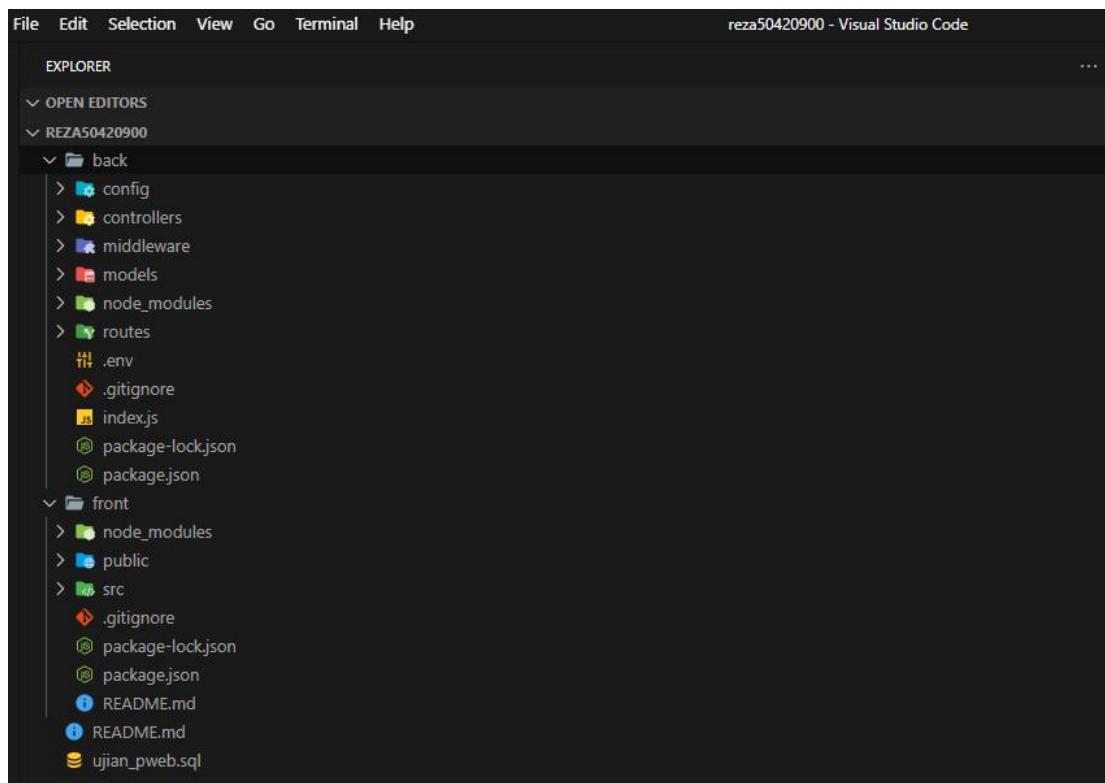


## **UJIAN PEMEROGRAMAN WEB**

Mata Praktikum : Pemrograman Web  
Kelas : 3IA20  
Praktikum ke- : 7  
Tanggal : 7 January 2023  
Materi : React Js ( Ujian)  
NPM : 50420900  
Nama : Muhammad reza hidayat  
Ketua Asisten : Andre  
Nama Asisten :  
Paraf Asisten :  
Jumlah Lembar : 19



**LABORATORIUM TEKNIK INFORMATIKA**  
**UNIVERSITAS GUNADARMA**  
**2023**



```
Database.js
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;
9
```



UserController.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 export const getUserById = async(req, res) =>{
13     try {
14         const response = await User.findOne({
15             where:{
16                 id: req.params.id
17             }
18         });
19         res.status(200).json(response);
20     } catch (error) {
21         console.log(error.message);
22     }
23 }
```

```
UserController.js

25 export const createUser = async(req, res) =>{
26     try {
27         await User.create(req.body);
28         res.status(201).json({msg: "User Created"});
29     } catch (error) {
30         console.log(error.message);
31     }
32 }
33
34 export const updateUser = async(req, res) =>{
35     try {
36         await User.update(req.body,{
37             where:{
38                 id: req.params.id
39             }
40         });
41         res.status(200).json({msg: "User Updated"});
42     } catch (error) {
43         console.log(error.message);
44     }
45 }
46
47 export const deleteUser = async(req, res) =>{
48     try {
49         await User.destroy({
50             where:{
51                 id: req.params.id
52             }
53         });
54         res.status(200).json({msg: "User Deleted"});
55     } catch (error) {
56         console.log(error.message);
57     }
58 }
```

```
UserModel.js

1 import {Sequelize} from "sequelize";
2 import db from "../config/Database.js";
3
4 const {DataTypes} = Sequelize;
5
6 const User = db.define('users',{
7     name: DataTypes.STRING,
8     email: DataTypes.STRING,
9     gender: DataTypes.STRING
10 },{
11     freezeTableName:true
12 });
13
14 export default User;
15
16 (async()=>{
17     await db.sync();
18 })();
```

```
UserRoute.js

1 import express from "express";
2 import {
3     getUsers,
4     getUserById,
5     createUser,
6     updateUser,
7     deleteUser
8 } from "../controllers/UserController.js";
9
10 const router = express.Router();
11
12 router.get('/users', getUsers);
13 router.get('/users/:id', getUserById);
14 router.post('/users', createUser);
15 router.patch('/users/:id', updateUser);
16 router.delete('/users/:id', deleteUser);
17
18 export default router;
```



index.js

```
1 import express from "express";
2 import cors from "cors";
3 import UserRoute from "../routes/UserRoute.js";
4
5 const app = express();
6 app.use(cors());
7 app.use(express.json());
8 app.use(UserRoute);
9
10 app.listen(5000, () => console.log("Server up and running..."));
11
```



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;
19
```



index.js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App';
4 import "bulma/css/bulma.css";
5
6 ReactDOM.render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10  document.getElementById('root')
11 );
12
13
```



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;
79
```



```
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;
91

```



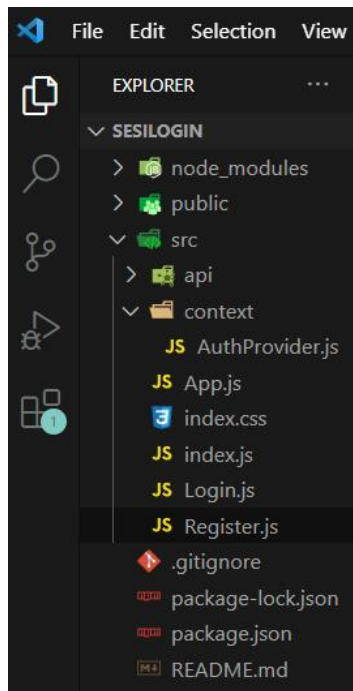
UserList.js

```
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 }
```

```

UserList.js
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

```



Pada laporan ini membuat program login dan register dengan menggunakan react js. Pertama buat folder sesi login seperti gambar diatas.

```
AuthProvider.js
1 import { createContext, useState } from "react";
2
3 const AuthContext = createContext({});
4
5 export const AuthProvider = ({ children }) => {
6   const [auth, setAuth] = useState({});
7
8   return <AuthContext.Provider value={{ auth, setAuth }}>{children}</AuthContext.Provider>;
9 };
10
11 export default AuthContext;
```

Task diatas merupakan penggunaan context pada react js. Dimana contex digunakan untuk menyediakan data yang dapat diakses oleh komponen di dalam aplikasi secara cepat tanpa menggunakan props. Perintah createContext digunakan untuk membuat contex dengan nama AuthContext, dimana variabel tersebut berfungsi untuk mengakses data yang telah tersimpan di dalamnya. Pada line 5 AuthProvider akan menggunakan variabel AuthContext dimana komponen yang digunakan yaitu useState yang berfungsi menyimpan data ke dalam state yang bernama auth.

```

App.js
1 // import Register from "../Register";
2 import Login from "../Login";
3
4 function App() {
5   return (
6     <main className='App'>
7       <Login />
8     </main>
9   );
10 }
11
12 export default App;

```

Task diatas menjelaskan bahwa register tidak akan digunakan karena script tersebut dijadikan komentar. Hanya komponen login yang akan ditampilkan jika program telah dieksekusi. Pada blok tersebut terdapat func app yang berfungsi untuk mengembalikan sebuah elemen yang telah berisi sebuah variabel login, dimana nanti nya akan ditampilkan pada saat func app berhasil di eksekusi. Kemudian variabel app akan di ekspor agar dapat digunakan di program lain.

```

index.css
1 @import url("https://fonts.googleapis.com/css2?family=Nunito&display=swap");
2
3 * {
4   margin: 0;
5   padding: 0;
6   box-sizing: border-box;
7 }
8
9 html {
10   font-family: "Nunito", sans-serif;
11   font-size: 22px;
12   color: #fff;
13 }
14
15 body {
16   min-height: 100vh;
17   background-color: dodgerblue;
18 }
19
20 .App {
21   display: flex;
22   flex-direction: column;
23   justify-content: center;
24   align-items: center;
25   min-height: 100vh;
26   padding: 1rem 0.5rem;
27 }
28
29 section {
30   width: 100%;
31   max-width: 420px;
32   min-height: 400px;
33   display: flex;
34   flex-direction: column;
35   justify-content: flex-start;
36   padding: 1rem;
37   background-color: rgba(0, 0, 0, 0.4);
38 }

```

Task diatas merupakan salah satu potongan script css yang berfungsi untuk membuat tampilan website menjadi lebih menarik. Dimana jenis font yang digunakan sans serif dengan ukuran 22px. Untuk bagian body memiliki tinggi 100

dengan background yang digunakan dodgerblue. Untuk class app menggunakan justify content center dan tinggi 100.

```
index.js
1 import React from "react";
2 import ReactDOM from "react-dom/client";
3 import "./index.css";
4 import App from "./App";
5 import { AuthProvider } from "../context/AuthProvider";
6
7 const root = ReactDOM.createRoot(document.getElementById("root"));
8 root.render(
9   <React.StrictMode>
10     <AuthProvider>
11       <App />
12     </AuthProvider>
13   </React.StrictMode>
14 );
15
16 // If you want to start measuring performance in your app, pass a function
17 // to log results (for example: reportWebVitals(console.log))
18 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
19
```

Task diatas pada line 1 variabel react akan mengimport react kemudian pada blok program diatas menggunakan ReactDOM yang berfungsi untuk merender komponen react ke dalam DOM. Sebelum di render, user menggunakan AuthProvider yang digunakan untuk memproses komponen data dari context. Kemudian perintah pada line 7 berfungsi untuk mengembalikan sebuah object yang bisa digunakan untuk merender komponen react ke dalam DOM. Dimana createRoot akan mencari elemen tersebut.



```

Login.js
1 import { useRef, useState, useEffect, useContext } from "react";
2 import AuthContext from "../context/AuthProvider";
3
4 import axios from "../api/axios";
5 const LOGIN_URL = "/auth";
6
7 const Login = () => {
8   const { setAuth } = useContext(AuthContext);
9   const userRef = useRef();
10  const errRef = useRef();
11
12  const [user, setUser] = useState("");
13  const [pwd, setPwd] = useState("");
14  const [errMsg, setErrMsg] = useState("");
15  const [success, setSuccess] = useState(false);
16
17  useEffect(() => {
18    userRef.current.focus();
19  }, []);
20
21  useEffect(() => {
22    setErrMsg("");
23  }, [user, pwd]);
24
25  const handleSubmit = async (e) => {
26    e.preventDefault();
27
28    try {
29      const response = await axios.post(LOGIN_URL, JSON.stringify({ user, pwd }), {
30        headers: { "Content-Type": "application/json" },
31        withCredentials: true,
32      });
33      console.log(JSON.stringify(response?.data));
34      //console.log(JSON.stringify(response));
35      const accessToken = response?.data?.accessToken;
36      const roles = response?.data?.roles;
37      setAuth({ user, pwd, roles, accessToken });
38      setUser("");
39      setPwd("");
40      setSuccess(true);
41    } catch (err) {
42      if (!err?.response) {
43        setErrMsg("No Server Response");
44      } else if (err.response.status === 400) {
45        setErrMsg("Missing Username or Password");
46      } else if (err.response.status === 401) {
47        setErrMsg("Unauthorized");
48      } else {
49        setErrMsg("Login Failed");
50      }
51      errRef.current.focus();
52    }
53  };

```

Pada task diatas menggunakan beberapa hook react seperti useRef, useStates, useEffect dan useContext. Dimana useStates berfungsi untuk menambahkan state bernama user, pwd, errMsg dan success. Sedangkan jenis hook useRef berfungsi untuk menyimpan referensi ke dalam elemen DOM. Pada blok program ini mengimport object axios yang digunakan untuk melakukan pamanggilan API yang nanti nya tersimpan dalam variabel login\_URL Pada blok program register memiliki logika yang hampir sama.

