Nama : Reihan Putra Oktavio
Kelas : APAP - C
NPM : 1606874476

Pertanyaan

1. Pada setiap fungsi di Appointment.js, parameter pertama pasti memiliki prefix "${cors}" yang merefer pada variable cors. Mengapa hal tersebut perlu dilakukan? Apa yang terjadi jika prefix tersebut dihilangkan?

   Jawab : Kegunaannya adalah untuk request restricted sources dari luar domain web. Jika hilang maka web tidak dapat diakses karena request tidak sesuai.

2. Pada fungsi handleFormSubmit(e) di UpdatePasien.jsx terdapat klausa:

```
let last = name.pop()
name.reduce((prev, next) => {
    return prev[next] = prev[next] || {};
}, dataJson)[last] = val
```

   Apa yang dilakukan pada code tersebut? Jika klausa tersebut diganti dengan "dataJson[key] = val", apa yang terjadi dengan isi variable dataJson?

   Jawab : Code tersebut meng-assign variable dataJson dengan index id ke "value id" pasien. Jika parameter diganti "key" akan jadi error 500.

Latihan

1. Pertama, buat fungsi untuk melakukan GET staff farmasi

```
getAllFarmasi() {
    return fetch(`${cors}${baseUrl}/1/getAllStaffFarmasi`, {
        method: "GET"
    })
        .then(response => {
            return response.json();
        })
        .then(jsonResponse => {
            console.log(jsonResponse);
            return jsonResponse;
        });
},
```

Kemudian, buat di DaftarFarmasiRow.jsx pada components

```
import React from "react";

export const DaftarFarmasiRow = props => {
    return (
        <tbody>
        {props.listFarmasi.map(farmasi => {
            return (
                <tr key={farmasi.id}>
                    <td>{farmasi.nama}</td>
                </tr>
            );
        })}
        </tbody>
    );
};
```

Untuk penampilannya, DaftarFarmasi.jsx pada screens

```jsx
import React from "react";
import { Loading } from "../components/Loading";
import { TableContainer } from "../containers/TableContainer";
import { Appointment } from "../utils/Appointment";
import { DaftarFarmasiRow } from "../components/DaftarFarmasiRow";

export class DaftarFarmasi extends React.Component {
    /**
     * TODO: Akses method getAllPasien() pada Appointment dan lakukan update state.
     * TODO: Lakukan pemanggilan pada constructor() atau pada lifecycle componentDidMount()
     */

    constructor(props) {
        super(props);

        this.state = {
            loading: true,
            listPasien: []
        };
    }
    componentDidMount() {
        Appointment.getAllFarmasi().then(response => {
            this.setState({
                loading: false,
                listFarmasi: response.result
            });
        });
    }

    render() {
        if (this.state.loading) {
            return <Loading msg="Fetching Data..." />;
        } else {
            return (
                <TableContainer title="Daftar Staff Farmasi" header={["Nama Staff"]}>
                    <DaftarFarmasiRow listFarmasi={this.state.listFarmasi} />
                </TableContainer>
            );
        }
    }
}
```

Terakhir, tambahkan di App.js untuk run aplikasi

2. Pertama, buat fungsi untuk melakukan POST addLabResult()

```
addLabResult(requestBody) {
    return fetch(`${cors}${baseUrl}/1/addLabResult`, {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(requestBody)
    })
        .then(response => {
            console.log(response);
            return response.json();
        })
        .then(jsonResponse => {
            return jsonResponse;
        });
}
```

Kemudian, buat container FormAddLabResult

```
import React from "react";

export const FormAddLabResult = props => {
    return (
        <form onSubmit={props.onSubmit}>
            <h2>Add Lab Result Pasien {props.pasien.name}</h2>
            <input type="hidden" name="pasien.id" value={props.pasien.id} />
            <div className="form-group">
                <label>Jenis Diagnosa</label>
                <input type="text" className="form-control" name="jenis" />
            </div>
            <div className="form-group">
                <label>Hasil Diagnosa</label>
                <input type="text" className="form-control" name="hasil" />
            </div>
            <div className="form-group">
                <label>Tanggal Pengajuan</label>
                <input type="date" className="form-control" name="tanggalPengajuan" />
            </div>
            <button className="btn btn-success" value="submit">
                Add
            </button>
        </form>
    );
};
```

Untuk penampilannya, AddLabResult.jsx pada screens

```
export class addLabResult extends React.Component {
    /**
     * TODO: Akses method getDetailPasien(idPasien) pada Appointment dan lakukan update state.
     * TODO: Lakukan pemanggilan pada constructor() atau pada lifecycle componentDidMount()
     */

    constructor(props) {
        super(props);
        this.state = {
            loading: true,
            pasien: {}
        };
        Appointment.getDetailPasien(this.props.match.params.id).then(response => {
            if (response.status === 200) {
                this.setState({
                    loading: false,
                    pasien: response.result
                });
            } else {
                alert("Data Tidak Ditemukan");
                this.props.history.push("/all-pasien");
            }
        });
        this.handleFormSubmit = this.handleFormSubmit.bind(this);
    }

    handleFormSubmit(e) {
        e.preventDefault();
        this.setState({
            loading: true
        });
        const data = new FormData(e.target);
        const dataJson = {};

        data.forEach((val, key) => {
            if (val !== "") {
                let name = key.split(".");
                if (name.length > 1) {
                    let last = name.pop();
                    name.reduce((prev, next) => {
                        return (prev[next] = prev[next] || {});
                    }, dataJson)[last] = val;
                } else {
                    dataJson[key] = val;
                }
            }
        });
        Appointment.addLabResult(dataJson).then(response => {
            console.log(dataJson);
            if (response.status === 200) {
                this.setState({
                    loading: false,
```

Terakhir, tambahkan di App.js untuk run aplikasi