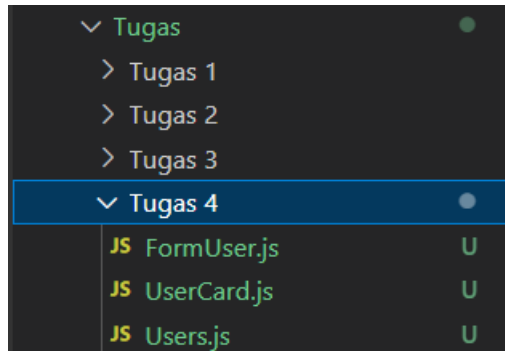


Nama : Yudhistira Kusuma
NPM : 202310067
Kelas : 6 TI-20-PA1
Matkul : Pemrograman Web Lanjut

1. Pada project sebelumnya buatlah sebuah package baru didalam folder components dengan nama latihan-4.



2. Silakan anda lanjutkan untuk membuat form data master User, dimana end-point Rest Api yang dapat diikuti berdasarkan Table 4.2. Buatlah cara merender seluruh data User, melihat detail dari data user yang dipilih dan dapat menambahkan data User baru

Source Code
<pre>import React, { useState } from "react"; import DatePicker from "react-datepicker"; import "react-datepicker/dist/react-datepicker.css"; import { openModal } from "../../../../../layouts/components/modals/ModalPopUp"; import axios from "axios"; export default function FormProfile(props) { var today = new Date(); const currDate = formatDate(today); const objProfile = { npm: "", email: "", firstname: "", middlename: "", lastname: "", birthdate: currDate, }; const [profile, setProfile] = useState(objProfile); const submitForm = (e) => { e.preventDefault(); const config = { method: "POST", url: "http://localhost:8080/api/students", headers: { "Content-Type": "application/json" },</pre>

```

    data: profile,
  };
  axios
    .request(config)
    .then((response) => {
      let results = response.data.data;
      if (results) {
        openModal({ header: "Info", message: "Successfully submitted" });
        props.getUser();
      } else {
        openModal({ header: "Error", message: "Failed submit" });
      }
    })
    .catch((response) => {
      openModal({ header: "Error", message: response.error });
    });
};

return (
  <div className="card ">
    <div className="card-header">
      <h3 className="card-title align-items-start flex-column">
        <span className="card-label fw-bolder text-dark">
          Form add data user
        </span>
        <span className="text-gray-400 mt-1 fw-bold fs-6">
          Please fill up the form with correctly
        </span>
      </h3>
    </div>
    <div className="card-body">
      <form
        className="form-profile"
        method="post"
        onSubmit={(e) => submitForm(e)}
        id="form-profile"
      >
        <div className="fv-row mb-10 fv-plugins-icon-container">
          <label className="required form-label fs-6 fw-bolder text-dark">
            NPM
          </label>
          <input
            type="text"
            name="npm"
            required
            className="form-control form-control-lg form-control-solid"
            pattern="^[0-9]*$"
            defaultValue={profile.npm}
            onChange={(e) => setProfile({ ...profile, npm: e.target.value })}
          />
        </div>
      </form>
    </div>
  </div>

```

```

<div className="fv-row mb-10 fv-plugins-icon-container">
  <label className="required form-label fs-6 fw-bolder text-dark">
    Email
  </label>
  <input
    type="email"
    name="email"
    required
    className="form-control form-control-lg form-control-solid"
    defaultValue={profile.firstname}
    onChange={(e) =>
      setProfile({ ...profile, email: e.target.value })
    }
  />
</div>
<div className="fv-row mb-10 fv-plugins-icon-container">
  <label className="required form-label fs-6 fw-bolder text-dark">
    First Name
  </label>
  <input
    type="text"
    name="firstname"
    required
    className="form-control form-control-lg form-control-solid"
    defaultValue={profile.firstname}
    onChange={(e) =>
      setProfile({ ...profile, firstname: e.target.value })
    }
  />
</div>
<div className="fv-row mb-10 fv-plugins-icon-container">
  <label className="form-label fs-6 fw-bolder text-dark">
    Middle Name
  </label>
  <input
    type="text"
    name="middlename"
    className="form-control form-control-lg form-control-solid"
    defaultValue={profile.middlename}
    onChange={(e) =>
      setProfile({ ...profile, middlename: e.target.value })
    }
  />
</div>
<div className="fv-row mb-10 fv-plugins-icon-container">
  <label className="required form-label fs-6 fw-bolder text-dark">
    Last Name
  </label>
  <input
    type="text"
    name="lastname"

```

```

        required
        className="form-control form-control-lg form-control-solid"
        defaultValue={profile.lastname}
        onChange={(e) =>
            setProfile({ ...profile, lastname: e.target.value })
        }
    />
</div>
<div className="fv-row mb-10 fv-plugins-icon-container">
    <label className="required form-label fs-6 fw-bolder text-dark">
        Birthdate
    </label>
    <DatePicker
        selected={new Date(profile.birthdate)}
        onChange={(value) =>
            setProfile({ ...profile, birthdate: formatDate(value) })
        }
        className="form-control form-control-lg form-control-solid"
        dateFormat="yyyy-MM-dd"
        placeholderText="Choose Date"
    />
</div>

<div className="text-center">
    <button type="button" className="btn btn-lg btn-light">
        Clear
    </button>
    <button
        type="submit"
        className="btn btn-lg btn-primary mx-2 btn-submit"
    >
        Submit
    </button>
</div>
</form>
</div>
</div>
);
}

const formatDate = (datestring) => {
    const today = new Date(datestring);
    const yyyy = today.getFullYear();
    let mm = today.getMonth() + 1; // Months start at 0!
    let dd = today.getDate();

    if (dd < 10) dd = "0" + dd;
    if (mm < 10) mm = "0" + mm;

    return yyyy + "-" + mm + "-" + dd;
};

```

```

import React from "react";

export default function UserCard({ data }) {
  const UserList = ({ user }) => {
    return (
      <div class="card mb-3" style={{ maxWidth: "540px", background: "white" }}>
        <div class="row g-0">
          <div class="col-md-4">
            <img
              src={require("../..../asset/user.png")}
              className="img-fluid rounded-start p-2 mt-2"
              alt="..."
            />
          </div>
          <div class="col-md-8">
            <div class="card-body">
              <h5 class="card-title">NPM : {user.npm}</h5>
              <p class="card-text">
                Name : {user.firstname} {user.middlename} {user.lastname}{""}
              </p>
              <p class="card-text">Birthdate : {user.birthdate}</p>
            </div>
          </div>
        </div>
      </div>
    );
  };

  return (
    <div className="user-list">
      <div className="row d-flex mb-2">
        {Object.values(data).length > 0 ? (
          data.map((v, index) => (
            <div className="col-5" key={index}>
              <UserList user={v} />
            </div>
          ))
        ) : (
          <p className="text-danger">No record found</p>
        )}
      </div>
    </div>
  );
}

```

```

import React, { useEffect, useState } from "react";
import FormUser from "../FormUser";
import UserCard from "../UserCard";
import axios from "axios";
import LoadingSpin from "../../layouts/components/loadings/LoadingSpin";
import AlertInfo from "../../layouts/components/alerts/AlertInfo";
// import PackageJSON from "../../package.json";

export default function Users() {
  //const [product, setProduct] = useState([]);
  const [user, setUser] = useState({
    loading: false,
    data: [],
    message: ""
  });

  const getUser = () => {
    setUser({ ...user, loading: true });
    let config = {
      method: 'get',
      maxBodyLength: Infinity,
      url: "http://localhost:8080/api/students",
      headers: {}
    };

    axios.request(config)
      .then((response) => {
        // console.log(JSON.stringify(response.data));
        //setProduct(response.data.data)
        setUser({ ...user, loading: false, data: response.data.data, message:"" });
        //console.log(product)
      })
      .catch((error) => {
        //console.log(error);
        setUser({ ...user, loading: false, message: error, data:[] });
      });
  }

  useEffect(() => {
    getUser()
  }, [])

  return (
    <div id="product-master">
      <div className="row">
        <div className="col-sm-12 col-lg-8">
          <div className="d-flex flex-wrap flex-stack pb-7">
            <div className="d-flex flex-wrap align-items-center my-1">
              <h3 className="fw-bolder me-5 my-1">{Object.values(user.data).length}
items of user</h3>
            </div>

```

```

<div className="d-flex flex-wrap my-1">
  <div className="d-flex my-0">
    <div className="input-group me-3">
      <input
        type="text"
        name="search"
        className="form-control"
        placeholder="Search here"
        autoComplete="off"
      />
      <span className="input-group-text">
        <i className="bi bi-search"></i>
      </span>
    </div>
    <select name="sort" className="form-select">
      <option value="">Sort by</option>
      <option value="">Name</option>
      <option value="">NPM</option>
    </select>
  </div>
</div>

{{(user.loading) ? <LoadingSpin /> : ''}
{{(user.message) ? <AlertInfo message={user.message.name+"
"+user.message.message} variant="danger" /> : ''}}
  <div className="product-items">
    <UserCard data={user.data} />
  </div>

</div>
<div className="col-sm-12 col-lg-4">
  <FormUser getUser={getUser} />
  {/* <FormUser /> */}
</div>
</div>
</div>
);
}

```

Hasil Program

The screenshot shows a web application interface with a top navigation bar containing links for Home, Explore, Messages, Master Data, and Log Out. Below the navigation bar, there are two tabs: 'Products' and 'Users'. The 'Users' tab is active, displaying a list of 7 items of user data. Each item is represented by a card with a profile icon, NPM number, Name, and Birthdate. To the right of the list is a 'Form add data user' section with input fields for NPM, Email, First Name, Middle Name, Last Name, and Birthdate. A search bar and a 'Sort by' dropdown are located above the list.

Products
[Master data control Products](#)

Users
Master data control Users

7 items of user

Search here Sort by

Form add data user
Please fill up the form with correctly.

NPM *

Email *

First Name *

Middle Name

Last Name *

Birthdate *

Clear Submit

The screenshot shows the same web application interface, but now displaying 8 items of user data. A modal window titled 'Info' is open, showing a 'Successfully submitted' message. The 'Form add data user' section is also visible, with the NPM field filled with '202310000' and the Email field filled with '202310000@gmail.com'. The search bar and 'Sort by' dropdown are still present.

Products
[Master data control Products](#)

Users
Master data control Users

8 items of user

Search here Sort by

Form add data user
Please fill up the form with correctly.

NPM *

Email *

First Name *

Middle Name

Last Name *

Birthdate *

Clear Submit

Info
Successfully submitted

© 2023 Felxy O.F. Inc. [Contact](#)

The screenshot shows the web application interface with 0 items of user data. The 'Users' tab is active, and the list is empty. The 'Form add data user' section is visible on the right, with the NPM field filled with '202310000' and the Email field filled with '202310000@gmail.com'. The search bar and 'Sort by' dropdown are still present.

Products
[Master data control Products](#)

Users
Master data control Users

0 items of user

Search here Sort by

Form add data user
Please fill up the form with correctly.

NPM *

Email *

First Name *

Middle Name

Last Name *

Birthdate *

Clear Submit

