

Front End Bootcamp PT. Mahardika Solusi Teknologi

Ariel Arliyanus, S.Kom

Chapter 16 – React JS Hooks & menggunakan REST API dengan axios

2021 © All Right Reserved

Dilarang memperbanyak dan/atau meng-copy sebagian atau seluruh material dalam dokumen ini tanpa persetujuan tertulis dari PT. Mahardika Solusi Teknologi

Hooks & menggunakan REST API dengan axios

Hooks merupakan fitur baru di React 16.8 atau versi lebih baru dari itu. Dengan Hooks, kita dapat menggunakan state dan fitur React yang lain tanpa perlu menulis sebuah kelas baru. Hooks merupakan fungsi yang memungkinkan Anda untuk “mengaitkan” state dan fitur-fitur lifecycle React dari function component. Hooks tidak dapat berfungsi didalam kelas dan Hooks memungkinkan menggunakan React tanpa kelas

Function Component dengan hooks

seperti yang sudah kita ketahui bahwa component dapat dibuat dengan dua cara yaitu menggunakan react subclass component atau dengan function component. sebelum React versi 16.8 beberapa frontend developer cenderung menggunakan class component karena di dalamnya kita bisa menggunakan state, namun permasalahan tersebut sudah ada solusinya yaitu hooks

berikut ini contoh penggunaan hooks:

```
import React, { useState } from 'react';

const Example = () => {
  // Deklarasi variabel state baru yang kita sebut "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>Anda menekan sebanyak {count} kali</p>
      <button onClick={() => setCount(count + 1)}>
        Klik saya
      </button>
    </div>
  );
}

export default Example
```

seperti yang kita lihat diatas pada function component diatas itu menggunakan useState, useState di panggil dalam function component untuk menambahkan suatu state lokal. React akan menyimpan state antar render. useState memberikan dua hal: nilai state saat ini dan fungsi untuk memperbarui nilai tersebut. Anda dapat memanggil fungsi ini dari sebuah event handler atau dimanapun. Hal ini serupa dengan this.setState pada kelas, tetapi tidak menggabungkan state lama dan baru menjadi satu

mari kita bandingkan dengan kode tersebut dengan contoh kode class component di bawah ini:

```
import React from 'react';

class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  render() {
    return (
      <div>
        <p>Anda menekan sebanyak {this.state.count} kali</p>
        <button onClick={() => this.setState({ count: this.state.count + 1
      ))}>
          Klik saya
        </button>
      </div>
    );
  }
}

export default Example
```

dari kode diatas terlihat perbedaannya, mulai dari tidak ada constructor, tidak ada this, method render pun sudah tidak ada di function component tapi langsung return nilai yang diinginkan

jika sudah terlihat perbedaannya mari kita ubah kode lists dan forms pada materi sebelumnya dengan function component dan hooks

mengganti constructor dengan inisialisasi hooks

```
const [pesertaLomba, setPesertaLomba] = useState([ 'Budi', 'Susi', 'Lala', 'Agung' ])
const [inputName, setInputName] = useState('')
const [indexOfForm, setIndexofForm] = useState(-1)
```

mengganti setiap handle menjadi arrow function dan merubah setState nya sesuai dengan inisialisasi di atas

```
const handleDelete = (event) => {
  let index = event.target.value
  let newPesertaLomba = pesertaLomba
  let editedPeserta = newPesertaLomba[indexOfForm]
  newPesertaLomba.splice(index, 1)

  if (editedPeserta !== undefined){
    // array findIndex baru ada di ES6
    var newIndex = newPesertaLomba.findIndex((el) => el === editedPeserta)
    setPesertaLomba([...newPesertaLomba])
    setIndexOfForm(newIndex)
  }else{
    setPesertaLomba([...newPesertaLomba])
  }
}

const handleEdit = (event) =>{
  let index = event.target.value
  let peserta = pesertaLomba[index]
  setInputName(peserta)
  setIndexOfForm(index)
}

const handleChange = (event) =>{
  setInputName(event.target.value);
}

const handleSubmit = (event) =>{
  // menahan submit
  event.preventDefault()

  let name = inputName

  if (name.replace(/\s/g, '') !== ''){
    let newPesertaLomba = pesertaLomba
    let index = indexOfForm

    if (index === -1){
      newPesertaLomba = [...newPesertaLomba, name]
    }else{
      newPesertaLomba[index] = name
    }

    setPesertaLomba(newPesertaLomba)
    setInputName('')
  }
}
```

terakhir kita perlu merubah render() menjadi hanya return saja dan beberapa handle yang menggunakan this di ubah

```
return(  
  <>  
    <h1>Daftar Peserta Lomba</h1>  
    <table>  
      <thead>  
        <tr>  
          <th>No</th>  
          <th>Nama</th>  
          <th>Aksi</th>  
        </tr>  
      </thead>  
      <tbody>  
        {  
          pesertaLomba.map((val, index)=>{  
            return(  
              <tr key={index}>  
                <td>{index+1}</td>  
                <td>{val}</td>  
                <td>  
                  <button onClick={handleEdit}  
value={index}>Edit</button>  
                    &nbsp; <button onClick={handleDelete}  
value={index}>Delete</button>  
                </td>  
              </tr>  
            )  
          })  
        }  
      </tbody>  
    </table>  
    { /* Form */}  
    <h1>Form Peserta</h1>  
    <form onSubmit={handleSubmit}>  
      <label>  
        Masukkan nama peserta:  
      </label>  
      <input type="text" value={inputName} onChange={handleChange}/>  
      <button>submit</button>  
    </form>  
  </>  
)
```

Implementasi useEffect hooks

seperti yang sudah dijelaskan di atas bahwa hooks memungkinkan untuk kita menerapkan state lifecycle tanpa harus menggunakan class component, lalu selain handle state apa penerapan lifecycle yang lain.

useEffect adalah salah satu bagian dari hook yang berfungsi menggantikan lifecycle method yang ada dalam class component, useEffect memungkinkan Anda melakukan efek samping (side effects) didalam function component

Jika Anda familiar dengan React class lifecycle methods, Anda dapat menganggap Hook useEffect sebagaicomponentDidMount, componentDidUpdate, dan componentWillUnmount yang disatukan

berikut ini contoh penerapan useEffect:

```
import React, { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  // Mirip dengan componentDidMount dan componentDidUpdate:
  useEffect(() => {
    // Memperbarui judul dokumen menggunakan API browser
    document.title = "You clicked " + count + " times";
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

pada kode diatas di dalam useEffect terdapat kode yang selalu mentrigger perubahan setiap kali ada perubahan dalam count maka title di browser akan selalu berubah

penerapan REST API dengan axios

seperti yang kita ketahui salah satu tugas di dalam frontend development ialah mengoperasikan data dari server, dan hal tersebut biasanya menggunakan apa yang di sebut dengan REST API,

REST (Representational State Transfer) API adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi

lalu bagaimanakah cara mengimplementasikan nya di dalam reactjs, tentunya kita butuh tools yang bisa mendukung perpindahan data dengan mudah, disini kita akan menggunakan axios, axios merupakan packages yang memungkinkan kita dapat mengkonsumsi api

untuk menggunakan axios silahkan install axios di project reactjs anda:

`npm install axios`

tunggu hingga proses selesai, setelah selesai mari kita terapkan axios ini dengan menggunakan Lists yang sudah kita buat di atas

ubahlah inialisasi peserta lomba menjadi array kosong:

```
const [pesertaLomba, setPesertaLomba] = useState([])
```

lalu tambahkan `useEffect` sebelum `return` yang isinya adalah mengambil data dari server lalu kita tampilkan tabelnya isi tabelnya saja

```
import React, {useState, useEffect} from "react"
import axios from "axios"
```

```
const Peserta= () =>{
  const [pesertaLomba, setPesertaLomba] = useState([])
  const [inputName, setInputName] = useState("")
  const [currentId, setCurrentId] = useState(null)

  useEffect( () => {
    const fetchData = async () => {
      const result = await
    axios.get(`http://backendexample.sanbercloud.com/api/contestants`)

      setPesertaLomba(result.data.map(x=>{ return {id: x.id, name: x.name} }) )
    }

    fetchData()
  }, [])
}
```

```

return(
  <>
    { pesertaLomba !== null &&
      (<div style={{width: "70%", margin: "0 auto", textAlign: "center"}}>
        <h1>Daftar Peserta Lomba</h1>
        <table className="peserta-lomba">
          <thead>
            <tr>
              <th>No</th>
              <th>Nama</th>
            </tr>
          </thead>
          <tbody>
            {
              pesertaLomba.map((item, index)=>{
                return(
                  <tr key={index}>
                    <td>{index+1}</td>
                    <td>{item.name}</td>
                  </tr>
                )
              })
            }
          </tbody>
        </table>
      </div>)
    }
  </>
)
}

```

export default Peserta

lalu tambahkan button edit dan delete yang memiliki trigger onclick untuk edit dan delete

```

<td>
  <button onClick={handleEdit} value={item.id}>Edit</button>
  &nbsp;
  <button onClick={handleDelete} value={item.id}>Delete</button>
</td>

```

lalu di dalam handleEdit dan handleDelete tambahkan axios get dan delete

```

const handleEdit = (event) =>{
  let idPeserta = event.target.value
  axios.get(`http://backendexample.sanbercloud.com/api/contestants/${idPeserta}`)
    .then(res => {
      let data = res.data
      setInputName(data.name)
      setCurrentId(data.id)
    })
}

```



```
const handleDelete = (event) =>{
  let idPeserta = parseInt(event.target.value)
  axios.delete(`http://backendexample.sanbercloud.com/api/contestants/${idPeserta}`)
  .then(() => {
    let newPesertaLomba = pesertaLomba.filter(el=> {return el.id !== idPeserta})
    setPesertaLomba(newPesertaLomba)
  })
}
```

lalu tambahkan form di bawah tabelnya

```
{/* Form */}
<h1>Form Peserta</h1>
<form style={{paddingBottom: "20px"}} onSubmit={handleSubmit}>
  <label>
    Masukkan nama peserta:
  </label>
  <input type="text" value={inputName} onChange={handleChange}/>
  <button>submit</button>
</form>
```

lalu tambahkan handle change

```
const handleChange = (event) =>{
  let inputValue = event.target.value
  setInputName(inputValue)
}
```

lalu tambahkan function handle submit yang di dalamnya menggunakan axios post

```
const handleSubmit = (event) =>{
  event.preventDefault()

  if (currentId === null){
    // untuk create data baru
    axios.post(`http://backendexample.sanbercloud.com/api/contestants`, {name:
inputName})
    .then(res => {
      let data = res.data
      setPesertaLomba([...pesertaLomba, {id: data.id, name: data.name}])
    })
  }else{
    axios.put(`http://backendexample.sanbercloud.com/api/contestants/${currentId}`,
{name: inputName})
    .then(() => {
      let singlePeserta = pesertaLomba.find(el=> el.id === currentId)
      singlePeserta.name= inputName
      setPesertaLomba([...pesertaLomba])
    })
  }
  setInputName("")
  setCurrentId(null)
}
```

referensi

- <https://id.reactjs.org/docs/hooks-intro.html>
- <https://id.reactjs.org/docs/hooks-overview.html>
- <https://id.reactjs.org/docs/hooks-state.html>
- <https://id.reactjs.org/docs/hooks-effect.html>
- <https://medium.com/@kiddy.xyz/restful-api-apaan-tuh-dbcfa434761e>
- <https://www.digitalocean.com/community/tutorials/react-axios-react>

Tugas

Buatlah Lists & forms (CRUD) menggunakan function components dan hooks, lalu implementasi kan REST API di dalamnya dengan menggunakan axios, berikut ini api yang di gunakan

GET <http://backendexample.sanbercloud.com/api/student-scores>

POST <http://backendexample.sanbercloud.com/api/student-scores>

PUT http://backendexample.sanbercloud.com/api/student-scores/{ID_STUDENT}

DELETE http://backendexample.sanbercloud.com/api/student-scores/{ID_STUDENT}

lalu pada app.js bagian component yang menampilkan tugas sebelumnya di comment.

Daftar Nilai Mahasiswa

No	Nama	Mata Kuliah	Nilai	Indeks Nilai	Aksi
1	John	Algoritma	80	A	<button>Edit</button> <button>Delete</button>
2	Doe	Matematika Diskrit	70	B	<button>Edit</button> <button>Delete</button>
3	Frank	Kalkulus	60	C	<button>Edit</button> <button>Delete</button>
4	Jason	Basis Data	50	D	<button>Edit</button> <button>Delete</button>

Form Nilai Mahasiswa

Nama:

Mata Kuliah:

Nilai:

submit

ketentuan:

- pada form di atas nilai hanya bisa di input dari 0 hingga 100
- indeks nilai memiliki kondisi: nilai ≥ 80 indeksnya A, nilai ≥ 70 dan nilai < 80 indeksnya B, nilai ≥ 60 dan nilai < 70 , indeksnya c nilai ≥ 50 dan nilai < 60 indeksnya D, nilai < 50 indeksnya E

pada tugas ini **tidak di perbolehkan** menggunakan bootstrap ataupun sejenisnya