Front End Bootcamp
PT. Mahardika Solusi Teknologi

Ariel Arliyanus, S.Kom

Chapter 17 – React JS Context

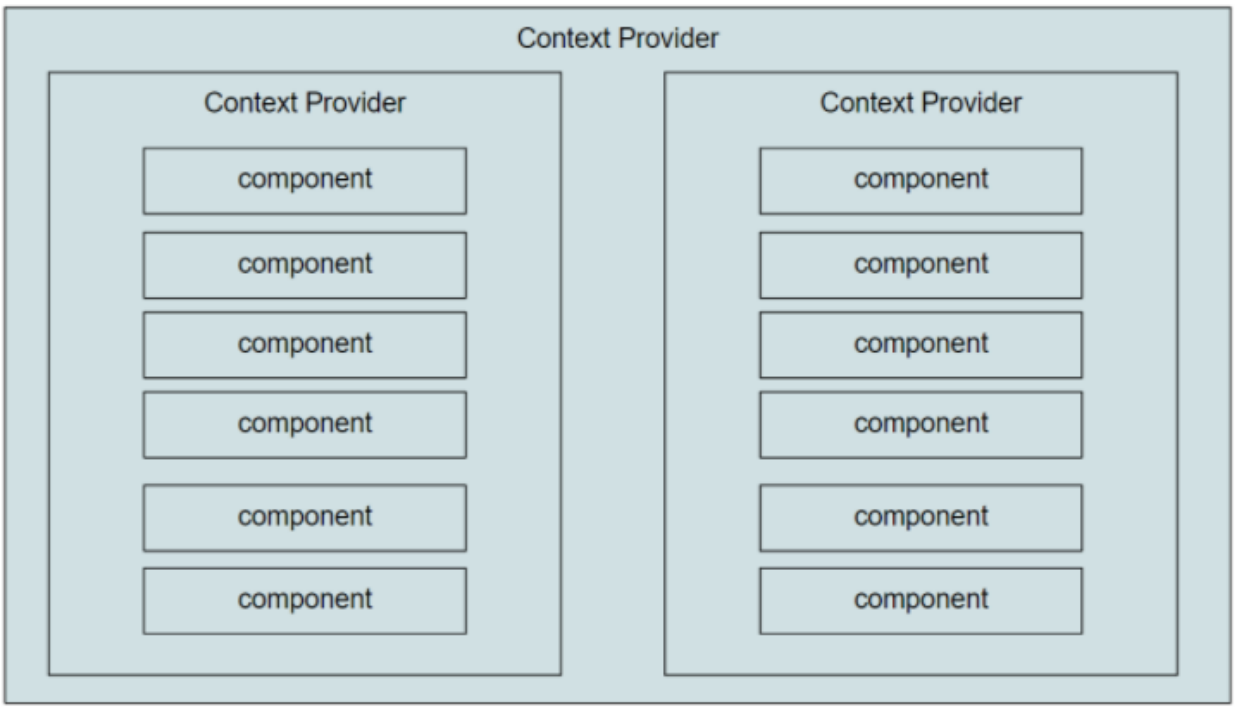
Chapter 17
React JS Context

Selama kita menggunakan state reactjs baik dengan menggunakan class component atau pun dengan hooks pasti pernah merasa kesulitan bagaimana memindahkan data tapi berbeda component atau berbeda file. mungkin dari kita pernah menggunakan solusi passing data atau oper data dengan props, tetapi praktek tersebut sangat tidak disarankan oleh reactjs.

untuk hal tersebut ada solusinya yaitu menggunakan state management, ada beberapa pilihan menggunakan state management, salah satu yang bisa di gunakan adala context yang di kembangkan oleh reactjs

untuk pengimplementasiannya beragam, disini kita akan membahas secara praktisnya saja, disini kita menggunakan beberapa fitur dari context, yaitu useContext, context provider, create context. lalu dalam pengimplementasian componentnya kita menggunakan hooks

berikut ini illustrasi dalam penggunaan context provider



implementasi context

disini kita punya kasus saya punya data seperti di bawah ini

```
let movie = [
    { name: "Harry Potter", lengthOfTime: 120},
    { name: "Sherlock Holmes", lengthOfTime: 125},
    { name: "Avengers", lengthOfTime: 130},
    { name: "Spiderman", lengthOfTime: 124},
  ]
```

lalu biasanya kita akan inisial state di dalam component, tapi disini kita inisialisasi dahulu di dalam context provider, lalu kita buat file js baru yang berisi context dan context provider seperti di bawah ini

```
import React, { useState, createContext } from "react";

export const MovieContext = createContext();

export const MovieProvider = props => {
  const [movie, setMovie] = useState([
    { name: "Harry Potter", lengthOfTime: 120},
    { name: "Sherlock Holmes", lengthOfTime: 125},
    { name: "Avengers", lengthOfTime: 130},
    { name: "Spiderman", lengthOfTime: 124},
  ]);

  return (
    <MovieContext.Provider value={[movie, setMovie]}>
      {props.children}
    </MovieContext.Provider>
  );
};
```

lalu bagaimanakah kita menggunakan context diatas, disini kita akan membuat file baru yang bernama Movie.js yang berisi movie provider dan dan component movie list dan movie form

```
import React from "react"
import {MovieProvider} from "./MovieContext"
import MovieList from "./MovieList"
import MovieForm from "./MovieForm"

const Movie = () =>{
  return(
    <MovieProvider>
      <MovieList/>
      <MovieForm/>
    </MovieProvider>
  )
}

export default Movie
```

kenapa movie list dan movie form component harus berada di dalam movie provider itu karena kita akan menggunakan state yang ada di dalam movie provider

lalu kita isi movie list dengan kode seperti di bawah ini:

```
import React, {useContext} from "react"
import {MovieContext} from "./MovieContext"

const MovieList = () =>{
  const [movie] = useContext(MovieContext)

  return(
    <ul>
      {movie.map(el=>{
        return <li>name: {el.name} {el.lengthOfTime} minutes</li>
      })}
    </ul>
  )

}

export default MovieList
```

disini kita menggunakan state di dalam context provider yang berisi data film lalu kita munculkan isi data, dari sini terlihat tidak berbeda dari kode useState biasa tapi jika di kita lihat lebih teliti lagi, disini kita tidak menginisialisasi lagi karna data sudah kita tampung di provider

dan untuk movieForm kita tulis kode seperti di bawah ini

```
import React, {useContext, useState} from "react"
import {MovieContext} from "./MovieContext"

const MovieForm = () =>{
  const [name, setName] = useState("")
  const [lengthOfTime, setLengthOfTime] = useState(0)
  const [movie, setMovie] = useContext(MovieContext)

  const handleSubmit = (event) =>{
    event.preventDefault()
    setMovie([...movie, {name, lengthOfTime}])
    setName("")
    setLengthOfTime("")
  }
  const handleChangeName = (event) =>{
    setName(event.target.value)
  }

  const handleChangeTime = (event) =>{
    setLengthOfTime(event.target.value)
  }

  return(
    <>
      <form onSubmit={handleSubmit}>
        <input type="text" value={name} onChange={handleChangeName} />
        <input type="number" value={lengthOfTime} onChange={handleChangeTime}
/>
        <button>submit</button>
      </form>
    </>
  )

}

export default MovieForm
```

form disini menggunakan setState yang sudah kita definisikan di dalam provider sehingga permasalahan sebelumnya yang setState selalu harus dalam satu component menjadi lebih mudah karena bisa berbeda component

dari sini kita dapat menyimpulkan bahwa penggunaan context sangat berguna untuk proses development reactjs

Referensi:

- https://id.reactjs.org/docs/context.html

Tugas
Copy semua codingan dari tugas CRUD lalu gunakanlah context di dalam tugas tersebut, create, delete dan edit sudah menggunakan context (tetap menggunakan axios)

agar lebih leluasa menggunakan contextnya buatlah file terpisah untuk tabel dan form. (jika ingin membuat file terpisah lebih dari itu di persilahkan)

kurang lebih outputnya seperti ini (stylingnya sudah boleh bebas
asalkan **tidak** menggunakan **bootstrap** dan sejenisnya: