Thinking in React

Ketika hendak membuat website menggunakan react, ada langkah-langkah yang perlu dilakukan. Langkah pertama yaitu membuat design tampilan. Kemudian mengidentifikasi hieraki komponen dan dilanjutkkan dengan membuat versi statis dari tampilan web yang diinginkan. Setelah itu, mulailah mengidentifikasi data mana yang merupakan props dan mana yang merupakan state. Nilai dari state dapat berubah-ubah sedangkan props tidak, tetapi nilai dari state dapat diturunkan ke child componentnya melalui props, sehingga nilai props tersebut “seolah-olah” dinamis. Apabila sudah diketahui data mana yang merupakan props dan state, kemudian dilanjutkan dengan mengidentifikasi letak state dan menambahkannya pada baris kode hingga diperoleh tampilan website yang dinamis.

Versi lengkapnya:

Cara membangun UI web menggukanakan react berdasarkan data sebagai berikut:

[

{type: "Mobil", harga: "Rp 148.000.000", ready: true, nama: "Brio"},

{type: "Mobil", harga: "Rp 205.500.000", ready: true, nama: "Mobilio"},

{type: "Mobil", harga: "Rp 252.500.000", ready: false, nama: "Jazz"},

{type: "Motor", harga: "Rp 16.000.000", ready: true, nama: "Beat"},

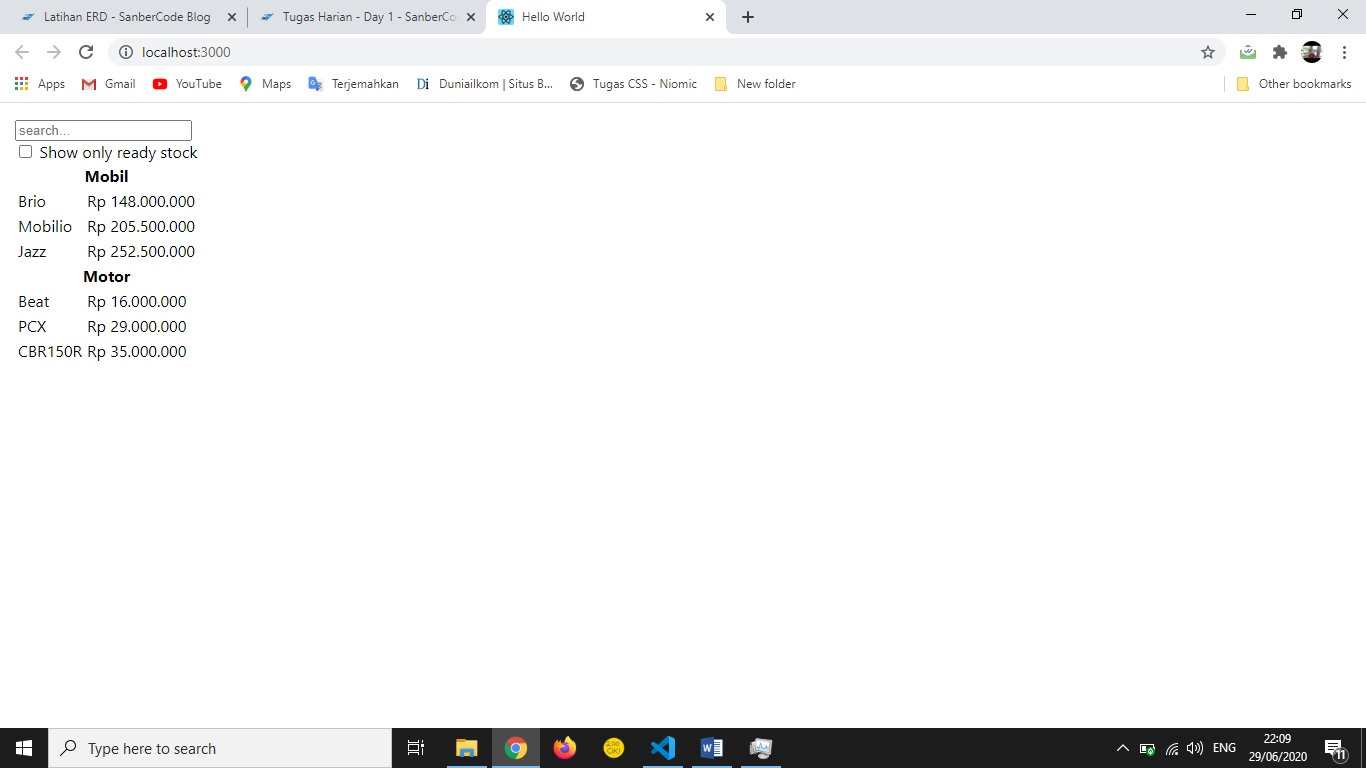
{type: "Motor", harga: "Rp 29.000.000", ready: false, nama: "PCX"},

{type: "Motor", harga: "Rp 35.000.000", ready: true, nama: "CBR150R"}

];

1. Membuat design tampilan

Design tampilan ini bisa berasal dari designer web atau oleh front end developer itu sendiri, contoh design tampilan adalah sebagai berikut:



1. Mengidentifikasi hierarki komponen

Pada React, suatu tampilan/bagian dapat dipecah mejadi komponen-komponen yang lebih kecil. Pada contoh di atas saya membuat hierarki komponen sebagai berikut:

* class FilterableVehicle
* class Search
* class ShowProductTable
* class RowType
* class RowVehicle

1. Membuat versi statis

Pada tahap ini cukup dengan membuat tampilan website awal, sesuai dengan yang diinginkan, seperti contoh nomor 1

1. Mengidentifikasi yang mana props dan state

Data yang dioper dari perent component ke child component sudah pasti props, sedangkan data yang bisa berubah-berubah dapat diidentifikasi sebagai state. Nilai dari state tersebut dapat diturunkan ke child componentnya menggunakan props. Pada contoh di atas data yang merupakan state adalah data yang berasal dari field search dan checkbox, sedangkan data tabel menyesuaikan value dari search dan checkbox

1. Menentukan letak state

Pada contoh di atas, state terletak pada class FilterableVehicle yang nilai statenya diatur dari component dibawahnya, yakni class Search. Perubahan state seperti ini mengacu kepada konsep lifting state up. Kemudian, nilai dari state tersebut akan diturunkan ke component ShowProductTable menggunakan props. Component ShowProductTable akan memfilter data berdasarkan nilai props tersebut

1. Mengimplementasikan penggunaan state

Ini merupakan tahap terakhir untuk membuat UI menjadi dinamis. Props hanya dapat dikirimkan dari atas ke bawah (parent ke child), sedangkan state dapat diubah dari child componentnya menggunakan konsep lifting state up, sehingga dengan menambahkan state, dapat dihasilkan Singe Web Page yang dinamis.

import React from 'react';

// import logo from './logo.svg';

import './App.css';

class RowVehicle extends React.Component{

  render(){

    return(

      <tr>

        <td>{this.props.vehicle.nama}</td>

        <td>{this.props.vehicle.harga}</td>

      </tr>

    )

  }

}

class RowType extends React.Component{

  render(){

    return(

      <tr>

        <th colSpan="2">{this.props.vehicle.type}</th>

      </tr>

    )

  }

}

class ShowProductTable extends React.Component{

  render(){

    const {search,checkbox}=this.props.filter;

    const listProduct=[];

    let lastType= null;

    this.props.vehicle.forEach((value)=>{

      if (value.type.indexOf(search)==-1 && value.nama.indexOf(search)==-1) {

        return

      }

      if (checkbox===true && value.ready===false) {

        return

      }

      if (lastType!==value.type) {

        listProduct.push(<RowType vehicle={value}/>)

      }

      listProduct.push(<RowVehicle vehicle={value}/>)

      lastType=value.type;

    })

    return(

      <>

        {listProduct}

      </>

    )

  }

}

class Search extends React.Component{

  constructor(props){

    super(props)

  }

  handleSearch = (event)=>{

    this.props.onSearch(event.target.value)

  }

  handleCheckbox = (event)=>{

    this.props.onCheckbox(event.target.checked)

  }

  render(){

    const {search,checkbox} = this.props.filter

    return(

      <>

        <input type="search" value={search} placeholder="search..." onChange={this.handleSearch}/><br/>

        <input type="checkbox" checked={checkbox} id="cekStok" onChange={this.handleCheckbox}/>

        <label htmlFor="cekStok">Show only ready stock</label>

      </>

    )

  }

}

class FilterableVehicle extends React.Component{

  constructor(props){

    super(props)

    this.state = {

      search :"",

      checkbox: false,

    }

  }

  onSearch = (search)=>{

    this.setState({search})

  }

  onCheckbox = (checked)=>{

    this.setState({checkbox:checked})

  }

  render(){

    console.log(this.state.checkbox)

    return (

      <>

        <div>

          <Search filter={this.state} onSearch={this.onSearch} onCheckbox={this.onCheckbox}/>

        </div>

        <div>

          <table>

            <tbody>

              <ShowProductTable vehicle={this.props.vehicle} filter={this.state}/>

            </tbody>

          </table>

        </div>

      </>

    )

  }

}

class App extends React.Component {

  render(){

    const vehicle=[

                    {type: "Mobil", harga: "Rp 148.000.000", ready: true, nama: "Brio"},

                    {type: "Mobil", harga: "Rp 205.500.000", ready: true, nama: "Mobilio"},

                    {type: "Mobil", harga: "Rp 252.500.000", ready: false, nama: "Jazz"},

                    {type: "Motor", harga: "Rp 16.000.000", ready: true, nama: "Beat"},

                    {type: "Motor", harga: "Rp 29.000.000", ready: false, nama: "PCX"},

                    {type: "Motor", harga: "Rp 35.000.000", ready: true, nama: "CBR150R"}

                  ];

    return (

      <div>

        <FilterableVehicle vehicle={vehicle}/>

      </div>

    )

  }

}

export default App;