

MODUL 3: React Component

A. TUJUAN

1. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Stateless component & statefull component
2. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Dynamic component menggunakan props
3. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Update parent oleh child component
4. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Lifecycle component 1&2

B. TINJAUAN PUSTAKA

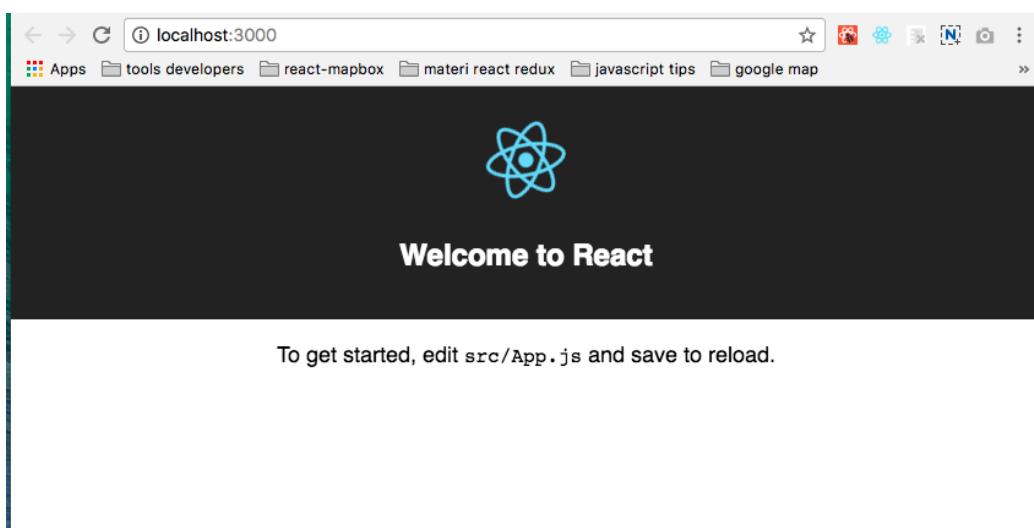
C. PRAKTIKUM

- a. Membuat Komponen Pada React
- b. Membuat Stateless component & statefull component

Project sebelumnya create hello world

Bagaimana cara membuat komponen dengan stateless component :

1. Membuka halaman react



2. Buka index.js
3. Membuat function baru dengan hello component (stateless component)

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';


function HelloComponent(){
  return <h1> HelloComponent</h1>;
}

ReactDOM.render(
  <React.StrictMode>
    <HelloComponent />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

4. Run di browser dan catat hasilnya
5. Buat class component / statefull component dengan memanggil React.Component
6. Memanggil StateFullComponent didalam react

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';


function HelloComponent(){
  return <h1> HelloComponent</h1>;
}

class StateFullComponent extends React.Component{
  render(){
    return <p>StateFullComponent</p>;
  }
}

ReactDOM.render(
  <React.StrictMode>
    <StateFullComponent />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```

7. Run di browser dan catat hasilnya!
8. Membuat konstruktor komponen, adalah sebagai berikut!

```
class PercobaanKontruktor extends React.Component{  
  constructor(){  
    super();  
    this.state={name: "Hallo Mahasiswa Polinema"}  
  }  
  
  render(){  
    return<p>Hello</p>;  
  }  
}  
  
ReactDOM.render(  
  <React.StrictMode>  
    <PercobaanKontruktor| />  
  </React.StrictMode>,  
  document.getElementById('root')  
>;
```

9. Run di browser dan catat hasilnya!

b. Membuat State dan Props Pada Komponen React

State dan props adalah objek khusus yang menyimpan data untuk komponen. **State** adalah objek yang digunakan untuk menyimpan data di **dalam komponen**, sedangkan **props** adalah objek yang digunakan untuk menyimpan data yang diterima dari **luar komponen**.

Membuat State

1. Tuliskan kode program berikut ini di dalam file index.js

```
class PercobaanState extends React.Component{
  constructor(){
    super();
    this.state={
      judul1: "Ayo Belajar React",
      judul2: "Bersama mahasiswa Politeknik"
    };
  }

  render(){
    return(
      <div>
        <h1>{this.state.judul1}</h1>
        <h2>{this.state.judul2}</h2>
      </div>
    )
  }
}

ReactDOM.render(
  <React.StrictMode>
    <PercobaanState />
  </React.StrictMode>,
  document.getElementById('root')
);
```

Note: state dibuat didalam konstruktor dan nilainya bisa langsung di isi didalamnya!

2. Run di browser dan catat hasilnya!
3. Ubahlah nilai state pada judul2 menjadi **Saya suka Coding** dengan menggunakan **setState**

```
//ubah judul
ChangeJudul2 = () => {
  this.setState({
    judul2: "Saya Suka Coding"
  });
}

render(){
  return(
    <div>
      <h1>{this.state.judul1}</h1>
      <h2>{this.state.judul2}</h2>
      <button onClick={this.ChangeJudul2}>Update Judul 2</button>
    </div>
  )
}
}

ReactDOM.render(
  <React.StrictMode>
    <PercobaanState />
  </React.StrictMode>,
  document.getElementById('root')
);
```

4. Run di browser dan catat hasilnya!

Membuat Props

Props atau properties digunakan untuk menampung nilai yang nantinya dikirimkan ke suatu Component.

1. Buatlah fungsi Welcome dan fungsi Nama seperti kode program d bawah ini!

```

//membuat props
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}

function Nama() {
  return (
    <div>
      <Welcome name="Vivin" />
      <Welcome name="Ayu" />
      <Welcome name="Lestari" />
    </div>
  );
}

ReactDOM.render(
  <React.StrictMode>
    <Nama />
  </React.StrictMode>,
  document.getElementById('root')
);

```

2. Run dibrowser dan catat hasilnya!
3. Props juga dapat dibuat dengan memisahkan file menjadi beberapa file. Misalnya kita akan membuat 3 buah file .js yaitu Footer.js, Image.js dan List.js .

Footer.js

```

import React from "react";

// Component menggunakan Function
const Footer = (props) => {
  return (
    <div>
      <h3>Halaman Footer</h3>
      <h3>Component ini dibuat menggunakan Function bukan Class</h3>
      <p>Nilai ini ditampilkan dari props: {props.judul} </p>
      <p>Nama Saya: {props.nama} </p>
    </div>
  );
};

export default Footer;

```

Pada kode di atas, kirim mengirimkan sebuah parameter props ke fungsi (anonymous function), lalu parameter tersebut akan kita gunakan untuk mengakses variabel yang dikirimkan dari file App.js.

Image.js

```
import React, { Component } from "react";

// Class Component Image
class Image extends Component {
  render() {
    return <img src={this.props.linkGambar} alt="Food" width="500" />;
  }
}

export default Image;
```

Kode program di atas merupakan Component image yang digunakan untuk mengakses gambar yang akan kita buat di file list.js. Cara mengaksesnya adalah: this.props.linkGambar. Dengan begitu, setiap gambar yang tampil akan memanggil nilai src yang berbeda-beda, berdasarkan yang sudah kita tentukan di Component List di atas.

List.js

```
// import React, { Component } from 'react';
import React, { Component } from 'react';
import Image from './Image';

class List extends Component {
  render() {
    return (
      <div>
        <ol>
          <li>
            Satu
            <Image linkGambar='https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/food1.jpg' />
          </li>
          <li>
            Dua
            <Image linkGambar='https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/nasipadang.jpg' />
          </li>
          <li>
            Tiga
            <Image linkGambar='https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/sate.png' />
          </li>
          <li>
            Empat
            <Image linkGambar='https://s3-ap-southeast-1.amazonaws.com/niomic/img/sample/sotolamongan.png' />
          </li>
        </ol>
      </div>
    );
  }
}

export default List;
```

Pada kode di atas, kita membuat empat buah list di mana masing masing list memanggil Component image.

4. Import file Footer.js, list.js dan Image.js di file App.js

```
import React, { Component } from "react";
import Footer from "./Footer";
import Header from "./Header";
import List from "./List";

class App extends Component {
  render() {
    return (
      <div>
        <Header />
        <h1>Component dari Class App</h1>
        <List />
        <Footer judul="Halaman Footer" nama="Aufa" />
      </div>
    );
  }
}

export default App;
```

5. Run dibrowser dan catat hasilnya!
- 6.

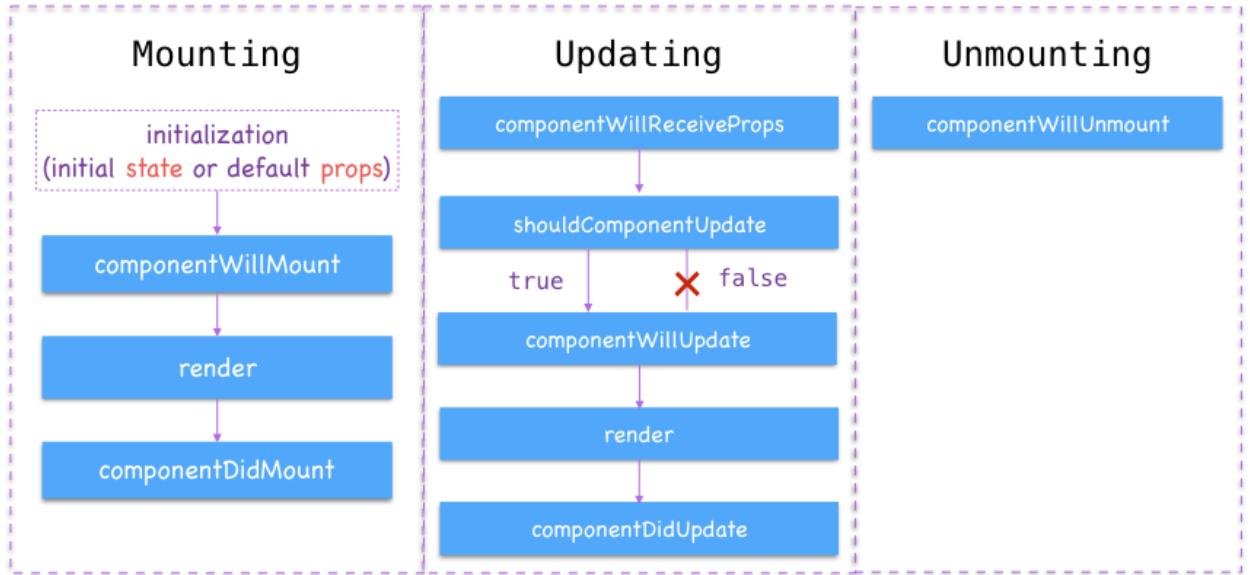
c. Lifecycle Component

Component — component di react setidaknya akan melewati tiga fase, yaitu :

1. **Mounting**
2. **Updating**
3. **Unmounting**

Pada setiap fase, components akan mengeksekusi fungsi atau method yang berbeda-beda yang dikenal sebagai lifecycle methods.

Lebih jelasnya perhatikan gambar di bawah ini !



<https://rangle.github.io/react-training/react-lifecycles/>

Buatlah Program dibawah ini dan deteksilah bagian dari lifecycle Component

```

import React from 'react';
import ReactDOM from 'react-dom';

class Test extends React.Component {
  constructor(props)
  {
    super(props);
    this.state = { hello : "World!" };
  }

  componentWillMount()
  {
    console.log("componentWillMount() ");
  }

  componentDidMount()
  {
    console.log("componentDidMount() ");
  }

  changeState()
  {
    this.setState({ hello : "Geek!" });
  }

  render()
  {
    return (
      <div>
        <h1>GeeksForGeeks.org, Hello{ this.state.hello }</h1>
        <h2>
          <a onClick={this.changeState.bind(this)}>Press Here!</a>
        </h2>
      </div> );
  }
}

```

```

shouldComponentUpdate(nextProps, nextState)
{
    console.log("shouldComponentUpdate()");
    return true;
}

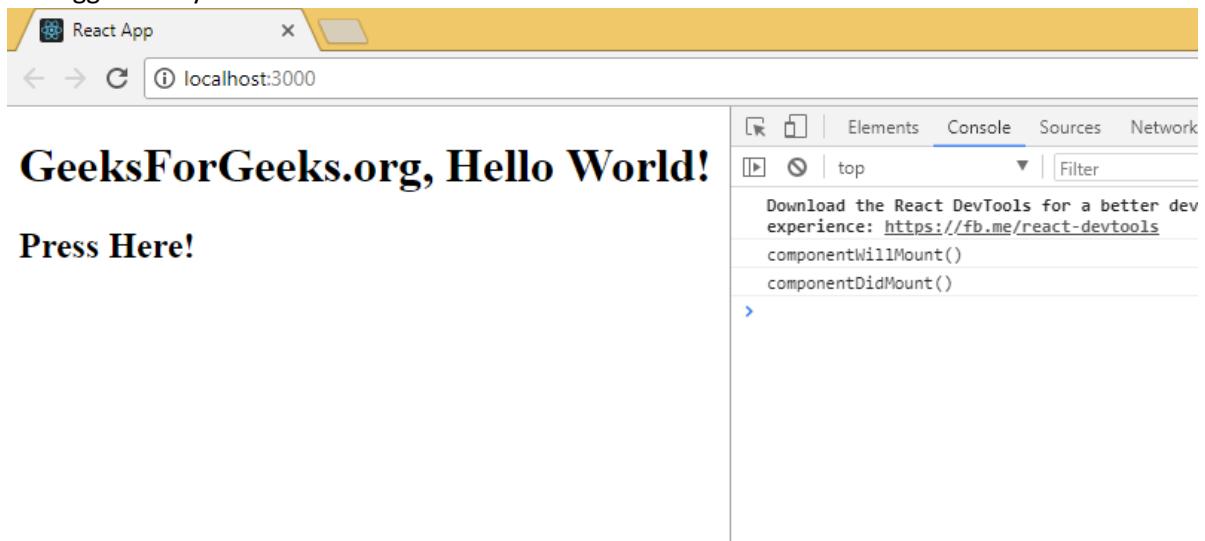
componentWillUpdate()
{
    console.log("componentWillUpdate()");
}

componentDidUpdate()
{
    console.log("componentDidUpdate()");
}
}

ReactDOM.render(
<Test />,
document.getElementById('root'));

```

1. Sehingga hasilnya berikut ini



- Selain kita dapat membuat file stateless component pada folder component, kita juga dapat menambahkan file css seperti pada gambar berikut (sebaiknya nama file css disamakan dengan nama file js nya dan tambahkan code seperti berikut)

```

.text-p {
    color: red;
}

```

- Buatlah folder component di src kemudian buatlah file HelloComponent.js , setelah itu import HelloComponent.css pada HelloComponent.js seperti pada berikut

```
// agar function arrow HelloComponent bisa berjalan di react maka tambahkan react
import React from "react";
import "./HelloComponent.css";

const HelloComponent = () => {
  return <p className="text-p">ini adalah arrow function yang ada di folder</p>;
};

// agar component ini dapat dipakai dimana saja
export default HelloComponent;
```

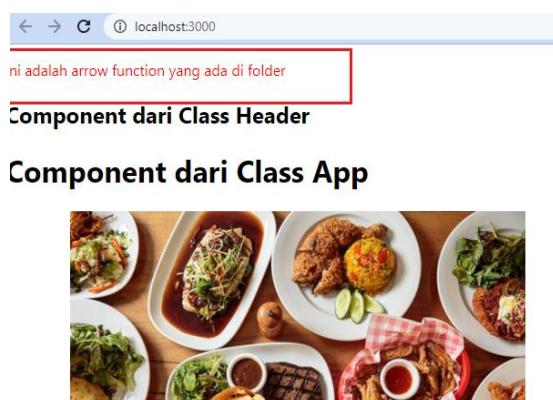
- Kemudian import HelloComponent.js di file App.js

```
import React, { Component } from "react";
import Footer from "./Footer";
import Header from "./Header";
import List from "./List";
import HelloComponent from "./component/HelloComponent";

class App extends Component {
  render() {
    return (
      <div>
        <HelloComponent/>
        <Header />
        <h1>Component dari Class App</h1>
        <List />
        <Footer judul="Halaman Footer" nama="Aufa" />
      </div>
    );
  }
}

export default App;
```

- Run di browser dan hasilnya adalah sebagai berikut:



Tugas Kelompok!

Buatlah menu login seperti pada gambar menggunakan CSS. Desain CSS bebas disesuaikan dengan judul proyek masing-masing!

The screenshot shows a browser window with the URL 'localhost:3003' in the address bar. The page title is 'Form Login'. The main content is a login form with the following structure:

- Tugas Pertemuan ketiga**: The title of the application.
- Username**: A text input field labeled 'masukan username'.
- Password**: A text input field labeled 'Masukkan Password Anda'.
- Login**: A green button labeled 'Login'.
- Remember Me**: A checked checkbox labeled 'Remember Me'.
- Cancel**: A red button labeled 'Cancel'.