

MODUL 3: React Component

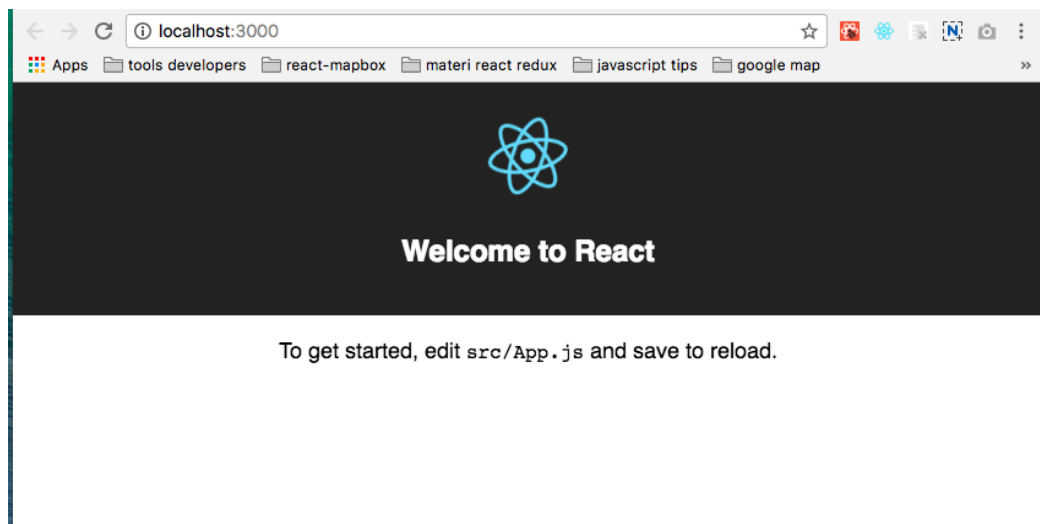
A. TUJUAN

1. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Stateless component & statefull component
2. Mahasiswa dapat mengetahui dan mengimplementasikan kegunaan Dymanic 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 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 * as serviceWorker from './serviceWorker';

function HelloComponent() {
  return HelloComponent
  ReactDOM.render(<HelloComponent />, document.getElementById('root'));

  // If you want your app to work offline and load faster, you can change
  // unregister() to register() below. Note this comes with some pitfalls.
  // Learn more about service workers: https://bit.ly/CRA-PWA
  serviceWorker.unregister();
}
```

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 * as serviceWorker from './serviceWorker';

const HelloComponent()=>{
  return HelloComponent}

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

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

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Membuat Props

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

1. Buatlah Program berikut ini

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

2. Simpan ke Component Footer atau File Footer.js.

Menampilkan props

1. Buatlah Program berikut ini

```
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.

Props pada Class Component

Contoh Kasus:

Kita ingin menampilkan banyak gambar di dalam component list, di mana component list akan menggunakan component image.

Component List

1. Buatlah program berikut ini

```
// 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;
```

2. Pada kode di atas, kita membuat empat buah list di mana masing masing list memanggil Component image. Kemudian Component Image di panggil dengan program berikut ini

Component Image

1. Buatlah Program berikut ini

```
import React, { Component } from 'react';class Image extends Component {
  render() {
    return (
      <img src={ this.props.linkGambar } alt="Food" width='500' />
    );
  }
}export default Image;
```

2. Cara mengaksesnya adalah: `this.props.linkGambar`.
3. Dengan begitu, setiap gambar yang tampil akan memanggil nilai src yang berbeda-beda, berdasarkan yang sudah kita tentukan di Component List di atas.

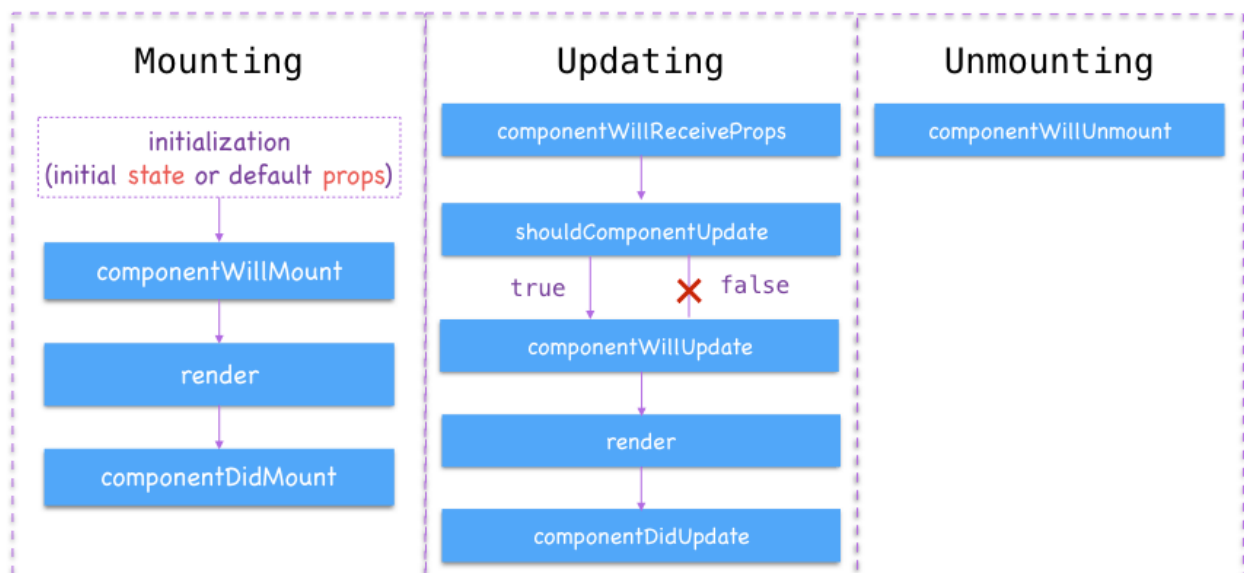
Lifecycle Component

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 di kenal 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>);
  }

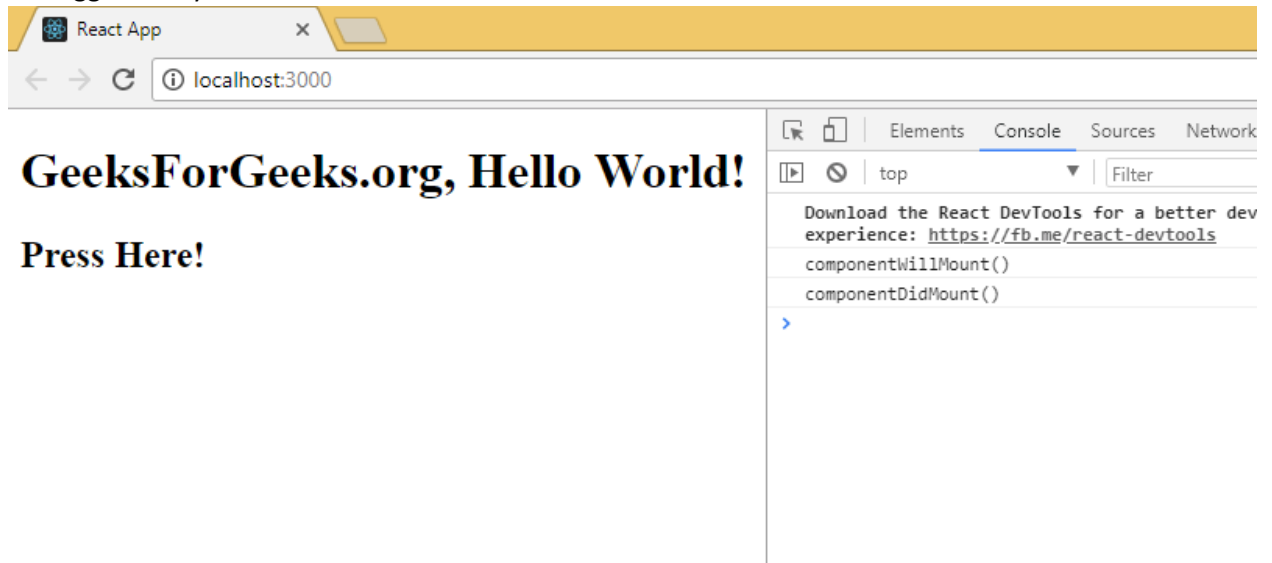
  shouldComponentUpdate(nextProps, nextState)
  {
    console.log("shouldComponentUpdate()");
    return true;
  }

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

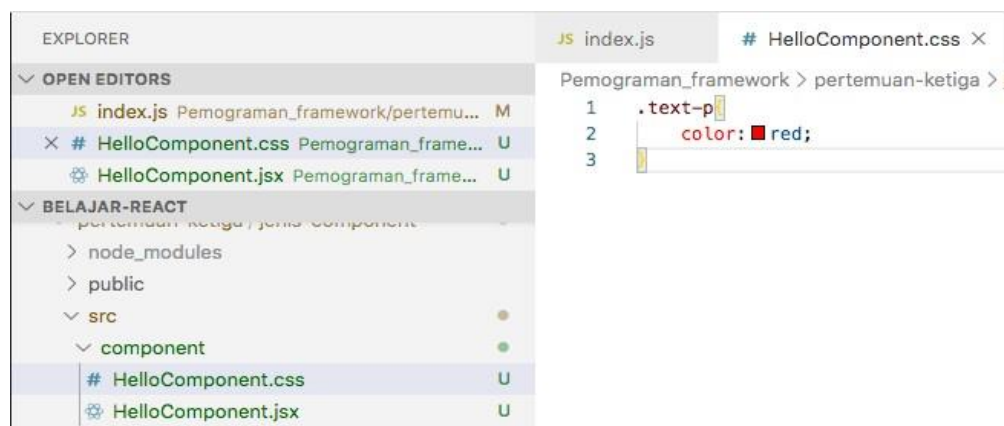
  componentDidUpdate()
  {
    console.log("componentDidUpdate()");
  }
}

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

3. 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)



- Setelah itu import HelloComponent.css pada HelloComponent.js seperti pada berikut

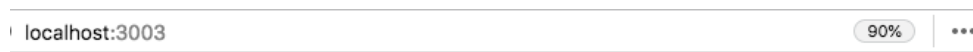
```
JS index.js # HelloComponent.css HelloComponent.jsx X
Pemograman_framework > pertemuan-ketiga > jenis-component > src > component > HelloComponent.jsx >
1 // agar function arrow HelloComponent bisa berjalan di react maka tambahkan react
2 import React from 'react';
3 import './HelloComponent.css';
4
5 const HelloComponent = () =>{
6   return <p className='text-p'> ini adalah arrow function yang ada di folder </p>
7 }
8
9 // agar component ini dapat dipakai dimana aja
10 export default HelloComponent;
```

Hasilnya



ini adalah arrow function yang ada di folder

Buatlah menu login seperti pada gambar menggunakan CSS



Form Login

Tugas Pertemuan ketiga

Username

Password

Login

☒ Remember Me

Cancel