# **MODUL 3: React Component**

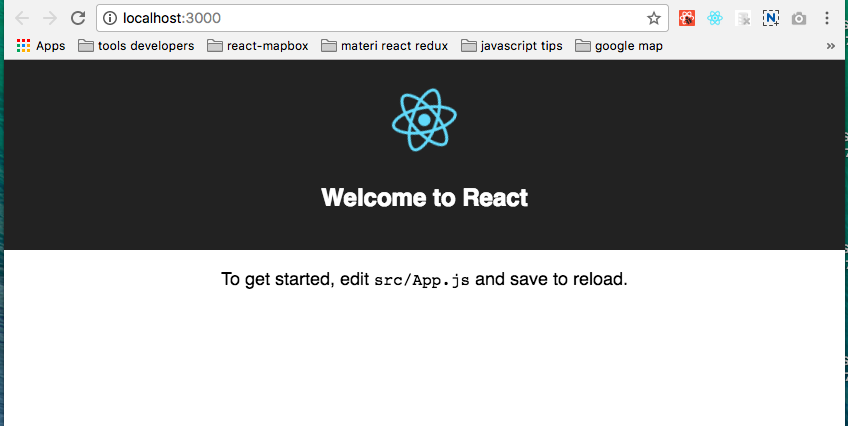
## **TUJUAN**

1. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan Stateless component & statefull component
2. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan Dymanic component menggunakan props
3. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan Update parent oleh child component
4. Mahasiswa dapat mengetahui dan mengimplemantasikan kegunaan Lifecycle component 1&2
5. **TINJAUAN PUSTAKA**
6. **PRAKTIKUM**
7. Membuat Stateless component & statefull component

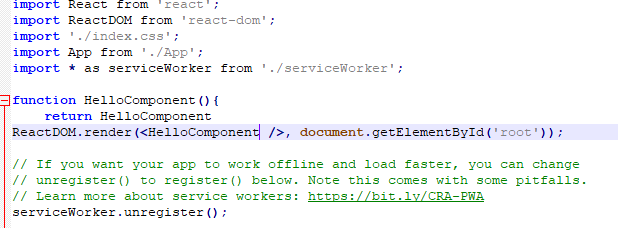
Project sebelummnya create hello world

Bagaimana cara membuat komponen dengan stateless component :

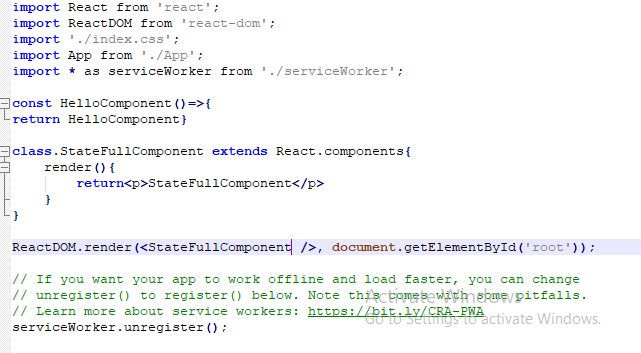
1. Membuka halaman react



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



1. Run di browser dan catat hasilnya
2. Buat class component / statefull component dengan memanggil react .component
3. Memanggil StateFullComponent didalam react



**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;

1. 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;

1. 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;

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

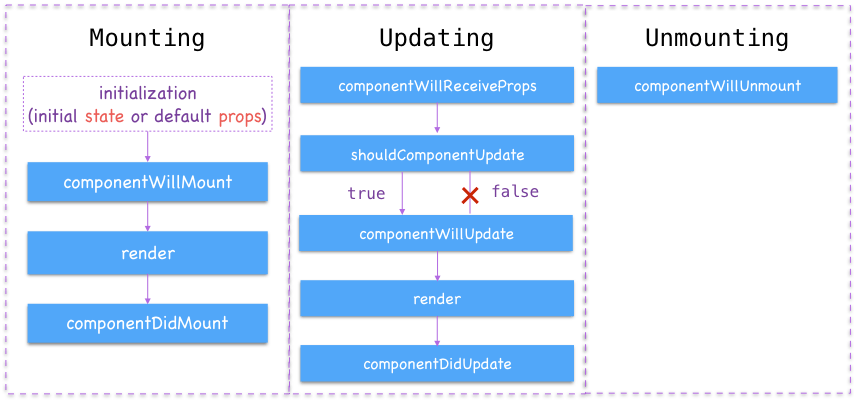
**Lifecycle Component**

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()");

    }

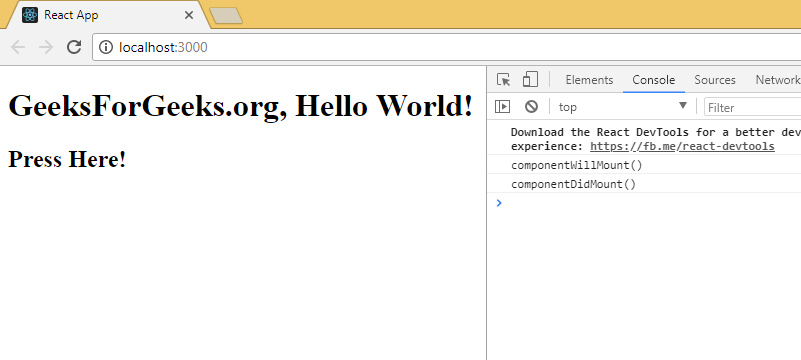
}

ReactDOM.render(

    <Test />,

    document.getElementById('root'));

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)

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

Hasilnya

Buatlah menu login seperti pada gambar menggunakan CSS