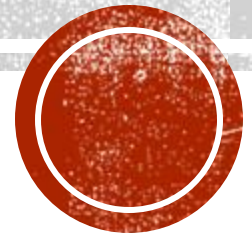


STATELESS COMPONENT DAN STATEFULL COMPONENT



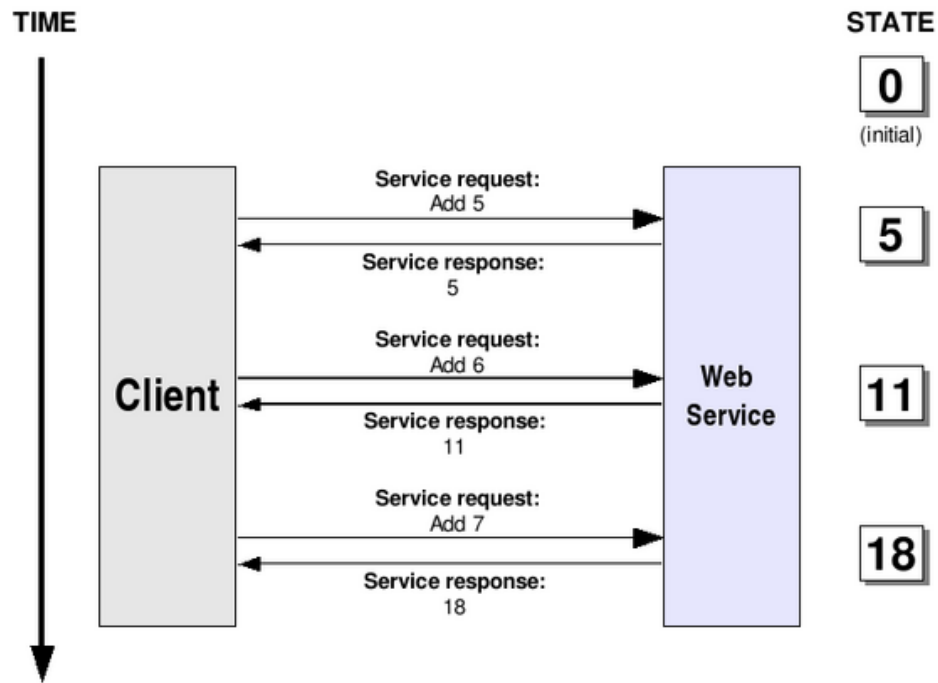
STATELESS COMPONENT DAN STATEFULL COMPONENT

- Untuk membuat component terdapat 2 cara yaitu
 - Statefull Component atau disebut class component
 - Component ini membawa semua fitur dan lifecycle dari ReactJs.
 - Di Component ini, kita bisa melakukan operasi logic, perubahan state, maupun fetch data dari server.
 - Stateless Component atau disebut juga function component
 - Stateless Component atau Component tanpa state.
 - Component ini hanya menerima props dan context pada react.
 - Component ini juga tidak membawa lifecycle react di dalamnya.
 - Jadi model component seperti sangat cocok untuk menyelesaikan permasalahan seperti pembuatan Badge, Ticker, Label dsb.

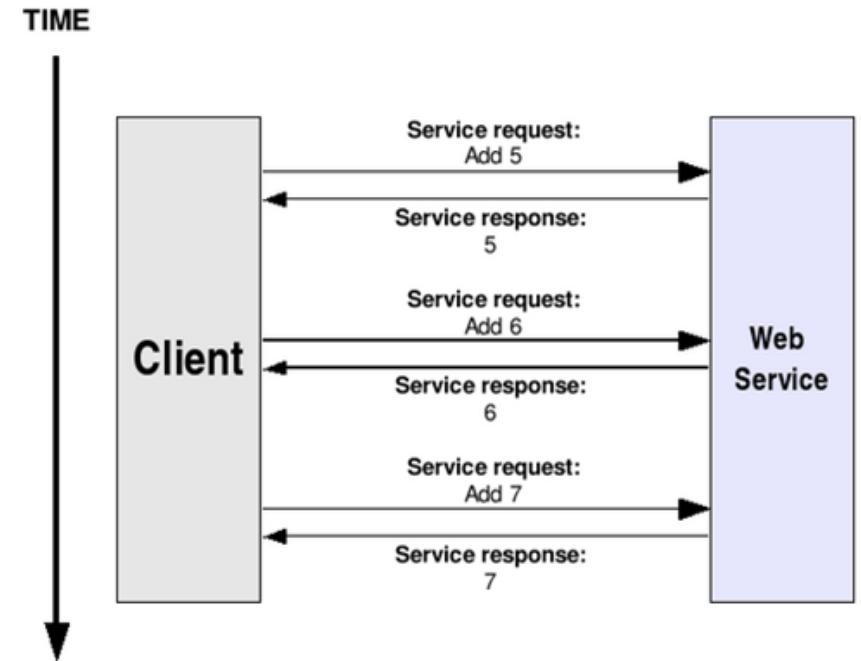


STATELESS COMPONENT DAN STATEFULL COMPONENT

STATEFUL



STATELESS



No state information is kept!



STATELESS COMPONENT DAN STATEFULL COMPONENT

- Jika dilihat pada gambar diatas saat statefull kondisi state tersimpan kondisi state yang terakhir 0, 5, 11 dan 18, jadi saat client melakukan service request 5 maka webservice akan mengembalikan service responnya 5 saat service request kedua yaitu 6 maka web service akan mengembalikan service responnya 11 (5 yg berasal state awal ditambah dengan 6 request dari client)
- Sedangkan pada stateless kondisi state tidak tersimpan jadi saat client melakukan service request 5 maka webservice akan mengembalikan service responnya 5

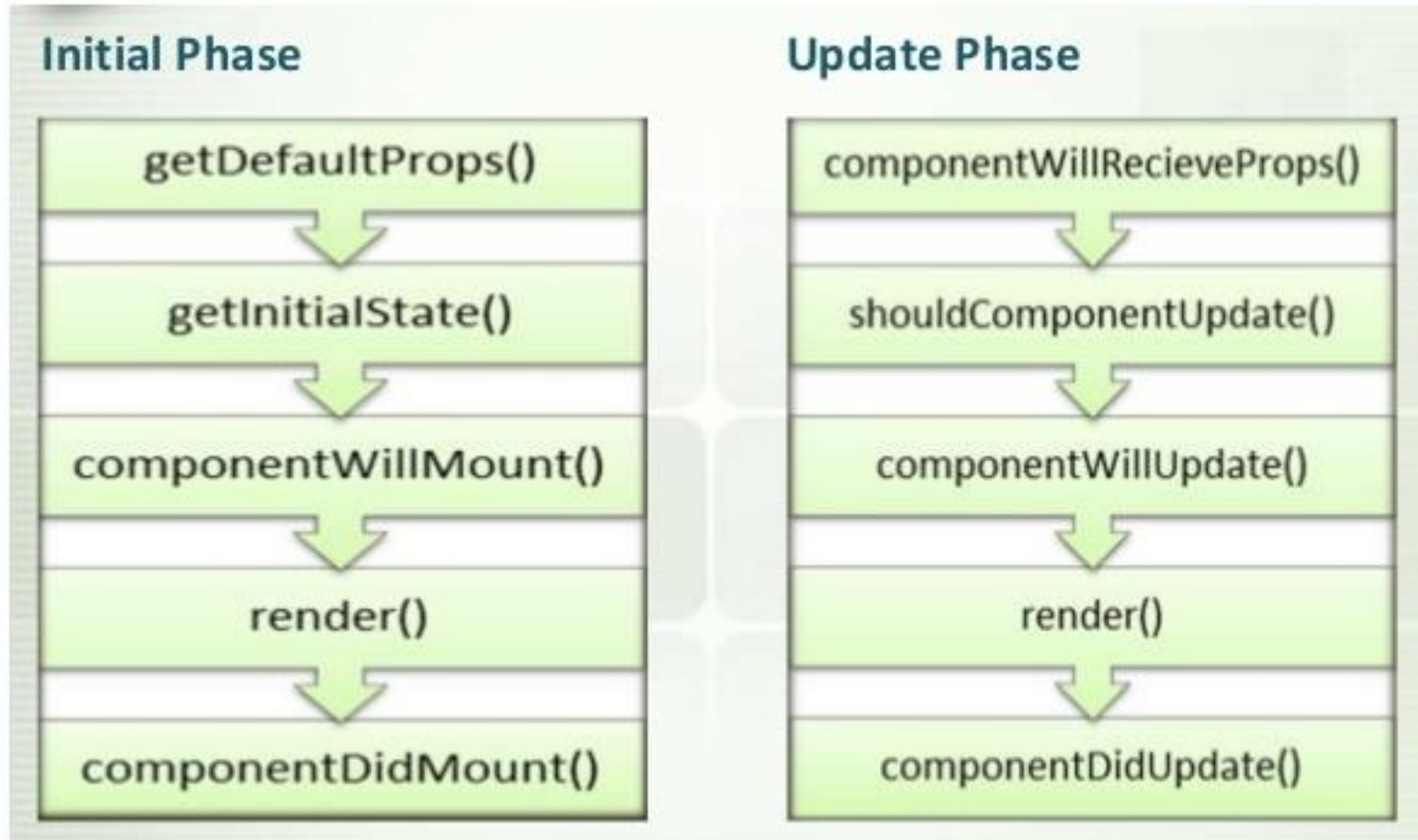


COMPONENT LIFE CYCLE

- ***Component life cycle*** adalah tahap berurutan dari kondisi *component* ketika digunakan.
- Di tiap kondisi, kita bisa melakukan macam-macam hal seperti memanggil data dari server (ajax), memanipulasi data, validasi, etc.
- Tahapnya sendiri dikelompokkan menjadi dua yaitu ***initial phase*** dan ***update phase*** Perhatikan gambar berikut



COMPONENT LIFE CYCLE



component life cycle (sumber : <https://goo.gl/APm1bR>)

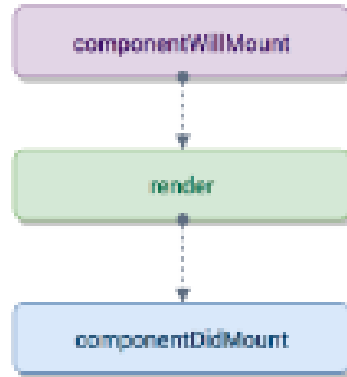


COMPONENT LIFE CYCLE

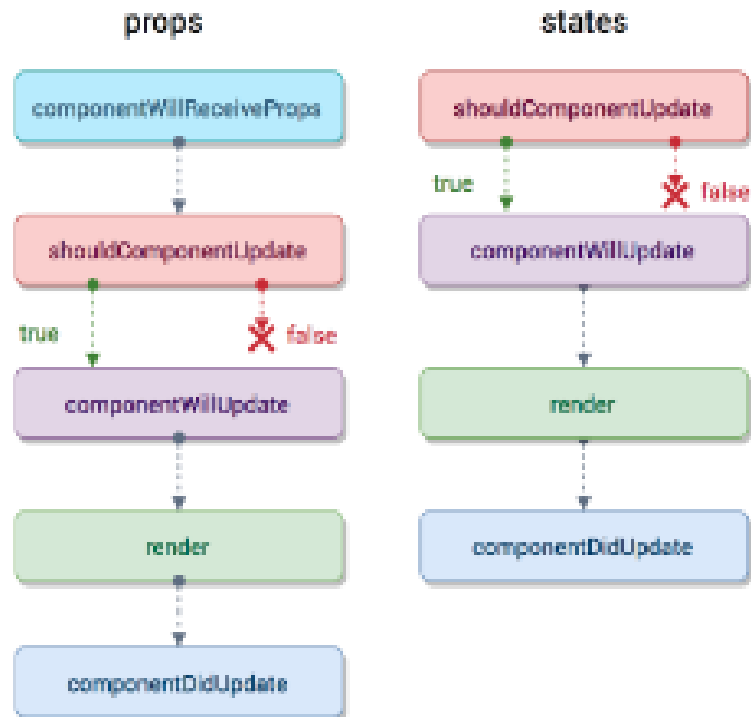
Initialization

setup props and state

Mounting



Updation



Unmounting

`componentWillUnmount`

COMPONENT LIFE CYCLE

- **initial phase** yaitu tahap ketika component baru pertama kali di panggil. Tahap-tahapnya :
 - `getDefaultProps()` dan `getInitialState()` : tahap ketika data props dan data state di deklarasikan. khusus untuk ES6, tahap ini dijalankan di *constructor function*.
 - `componentWillMount()` : tahap sebelum *component* di render.
 - `componentDidMount()` : tahap setelah *component* di render. pada tahap ini proses pemanggilan *ajax* dan perubahan isi *state* setelah proses *ajax* di panggil.



COMPONENT LIFE CYCLE

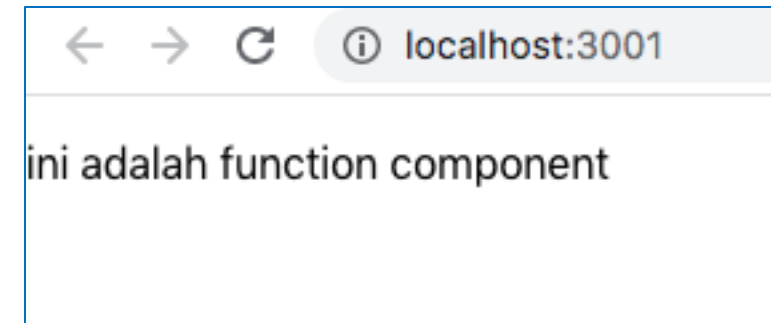
- **update phase** yaitu tahap ketika component mengalami perubahan “*model*” data. Tahap-tahapnya :
 - `componentWillReceiveProps()` : tahap ketika terjadi perubahan pada props sebelum component di render.
 - `shouldComponentUpdate()` : tahap untuk menentukan apakah perubahan akan di tampilkan atau tidak. isinya di set *true* atau *false*.
 - `componentWillUpdate()` : tahap sebelum *component* di *render*.
 - `componentDidUpdate()` : tahap setelah *component* di *render*.



PERCOBAAN STATELESS COMPONENT ATAU FUNCTION COMPONENT

- Buka index.js dan tambahkan code berikut (line 7 sampai 11)

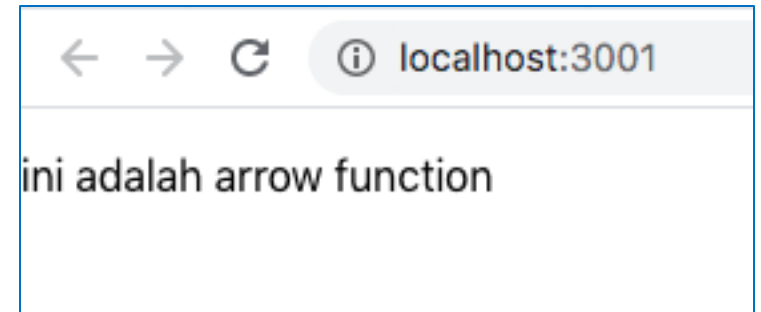
```
JS App.js JS index.js ×
Pemograman_framework > pertemuan-ketiga > jenis-component > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  function HelloWorld ()
8  {
9    return <p> ini adalah function component </p>
10 }
11 ReactDOM.render(<HelloWord />, document.getElementById('root'));
12
13 // If you want your app to work offline and load faster, you can change
14 // unregister() to register() below. Note this comes with some pitfalls.
15 // Learn more about service workers: https://bit.ly/CRA-PWA
16 serviceWorker.unregister();
17
```



PERCOBAAN STATELESS COMPONENT ATAU FUNCTION COMPONENT

- Karena sudah menggunakan ES6 maka function component dapat diganti dengan arrow function
- Buka index.js dan tambahkan code berikut (line 12 sampai 14)

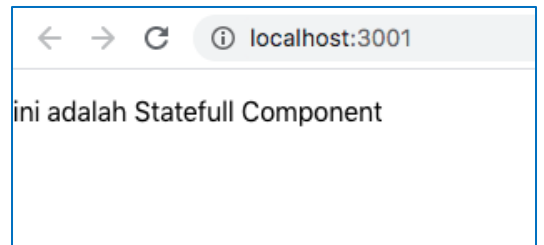
```
JS App.js JS index.js x
Pemograman_framework > pertemuan-ketiga > jenis-component > src > JS index.js > [HelloWord]
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  // function HelloWorld ()
8  // {
9  //   return <p> ini adalah function component </p>
10 // }
11
12 const HelloWorld = () =>{
13   return <p> ini adalah arrow function </p>
14 }
15 ReactDOM.render(<HelloWord />, document.getElementById('root'));
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
```



PERCOBAAN STATEFULL COMPONENT ATAU CLASS COMPONENT

- Buka index.js dan tambahkan code berikut

```
JS App.js JS index.js X
Pemograman_framework > pertemuan-ketiga > jenis-component > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  // function HelloWorld ()
8  // {
9  //   return <p> ini adalah function component </p>
10 // }
11
12 // const HelloWorld = () =>{
13 //   return <p> ini adalah arrow function </p>
14 // }
15
16 class Statefullcomponent extends React.Component
17 {
18   render(){
19     return <p> ini adalah Statefull Component</p>
20   }
21 }
22 ReactDOM.render(<Statefullcomponent />, document.getElementById('root'));
23
24 // If you want your app to work offline and load faster, you can change
25 // unregister() to register() below. Note this comes with some pitfalls.
26 // Learn more about service workers: https://bit.ly/CRA-PWA
27 serviceWorker.unregister();
```



PERCOBAAN STATEFULL COMPONENT ATAU CLASS COMPONENT

- Jika kita lihat pada line 16
 - class Statefullcomponent extends React.Component
 - Terdapat sebuah class yang merupakan turunan dari react.component
 - <https://reactjs.org/docs/react-api.html#reactcomponent>

React.Component

`React.Component` is the base class for React components when they are defined using [ES6 classes](#):

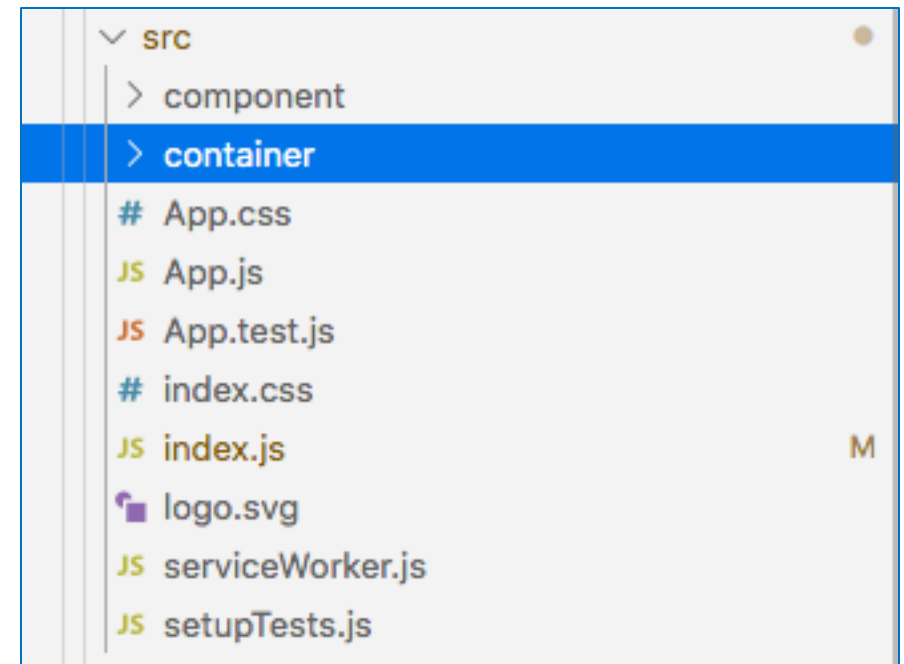
```
class Greeting extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

See the [React.Component API Reference](#) for a list of methods and properties related to the base `React.Component` class.



REACT.JS

- Untuk mempermudah penggunaan component kita dapat menempatkan component-component tersebut pada sebuah folder
- Langkah-langkahnya
 - Buat dua folder container pada src yaitu
 - Folder component → berisi stateless component
 - Folder container → berisi statefull component



REACT.JS FOLDER COMPONENT => STATELESS COMPONENT

- Setelah itu tambahkan file jsx pada folder component dengan nama HelloComponent
- untuk penamaan function dan nama component diusahakan sama

```
JS index.js  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
4  const HelloComponent = () =>{
5    |   return <p> ini adalah arrow function yang ada di folder </p>
6  }
7
8  // agar component ini dapat dipakai dimana aja
9  export default HelloComponent; |
```



REACT.JS

- Keterangan per line

- Line 2

- `import React from 'react';` → Digunakan agar component dapat berjalan pada react

- Line 4-6

- Function arrow

- Line 9

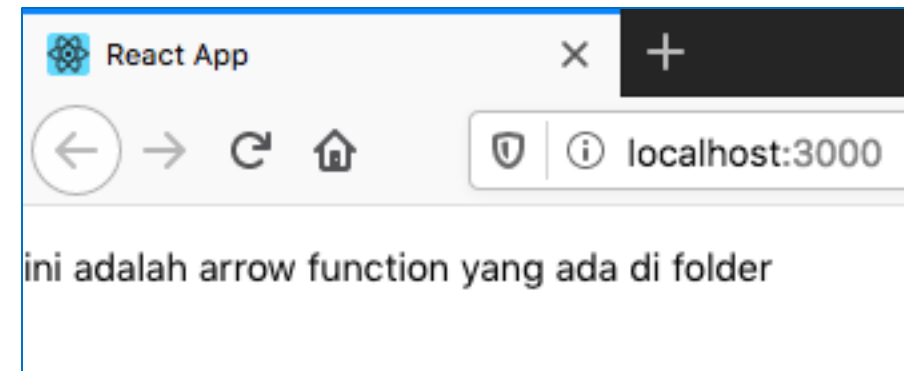
- `export default HelloComponent;` → Digunakan agar component dapat digunakan dipanggil dimana saja



REACT.JS

- Untuk menjalankan function arrow maka buka halaman index.js dan tambahkan code pada line 6 dan line 23

```
JS index.js x HelloComponent.jsx
Pemograman_framework > pertemuan-ketiga > jenis-component > src > JS index.js > ...
6  import HelloComponent from './component/HelloComponent';
7
8  // function HelloWord ()
9  // {
10 //   return <p> ini adalah function component </p>
11 // }
12
13 // const HelloWord = () =>{
14 //   return <p> ini adalah arrow function </p>
15 // }
16
17 class Statefullcomponent extends React.Component
18 {
19   render(){
20     return <p> ini adalah Statefull Component</p>
21   }
22 }
23 ReactDOM.render(<HelloComponent />, document.getElementById('root'));
```



REACT.JS

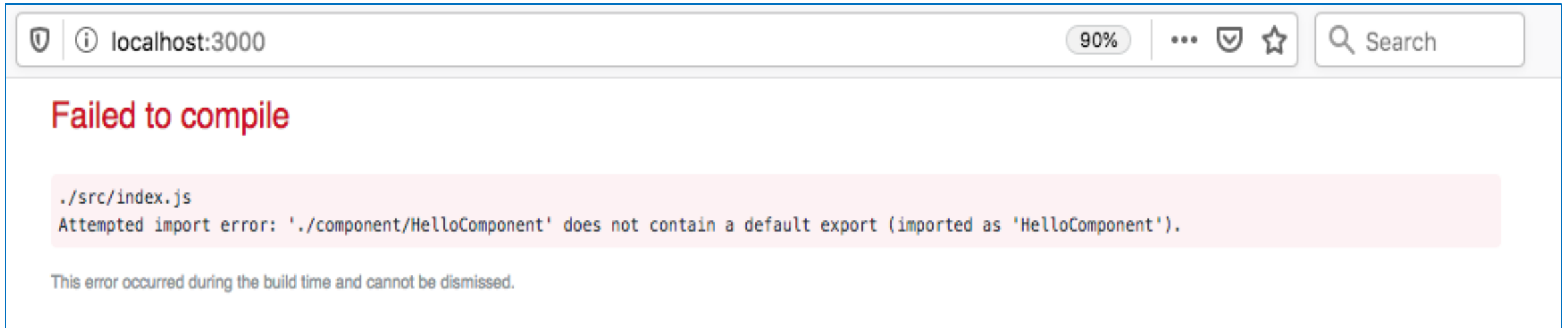
- Jika kita comment export pada HelloComponent seperti pada gambar dibawah ini

```
JS index.js  HelloComponent.jsx ×
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
4  const HelloComponent = () =>{
5    |   return <p> ini adalah arrow function yang ada di folder </p>
6  }
7
8  // agar component ini dapat dipakai dimana aja
9  // export default HelloComponent; |
```



REACT.JS

- Maka saat dijalankan akan terjadi pesan error seperti berikut



REACT.JS FOLDER CONTAINER => STATEFULL COMPONENT

- Setelah kita membuat stateless component pada folder component maka selanjutnya kita akan membuat statefull component pada folder container
- Langkah-langkah sebagai berikut :
 - Buat file dengan nama StateFullComponent.jsx pada folder container
 - Tambahkan code berikut

```
JS index.js  StateFullComponent.jsx X  HelloComponent.jsx
Pemograman_framework > pertemuan-ketiga > jenis-component > src > container > StateFullComponent.jsx > ...
1  import React from 'react';
2
3  class StateFullComponent extends React.Component
4  {
5      render(){
6          return <p> ini adalah Statefull Component</p>
7      }
8  }
9
10 export default StateFullComponent;
```



REACT.JS

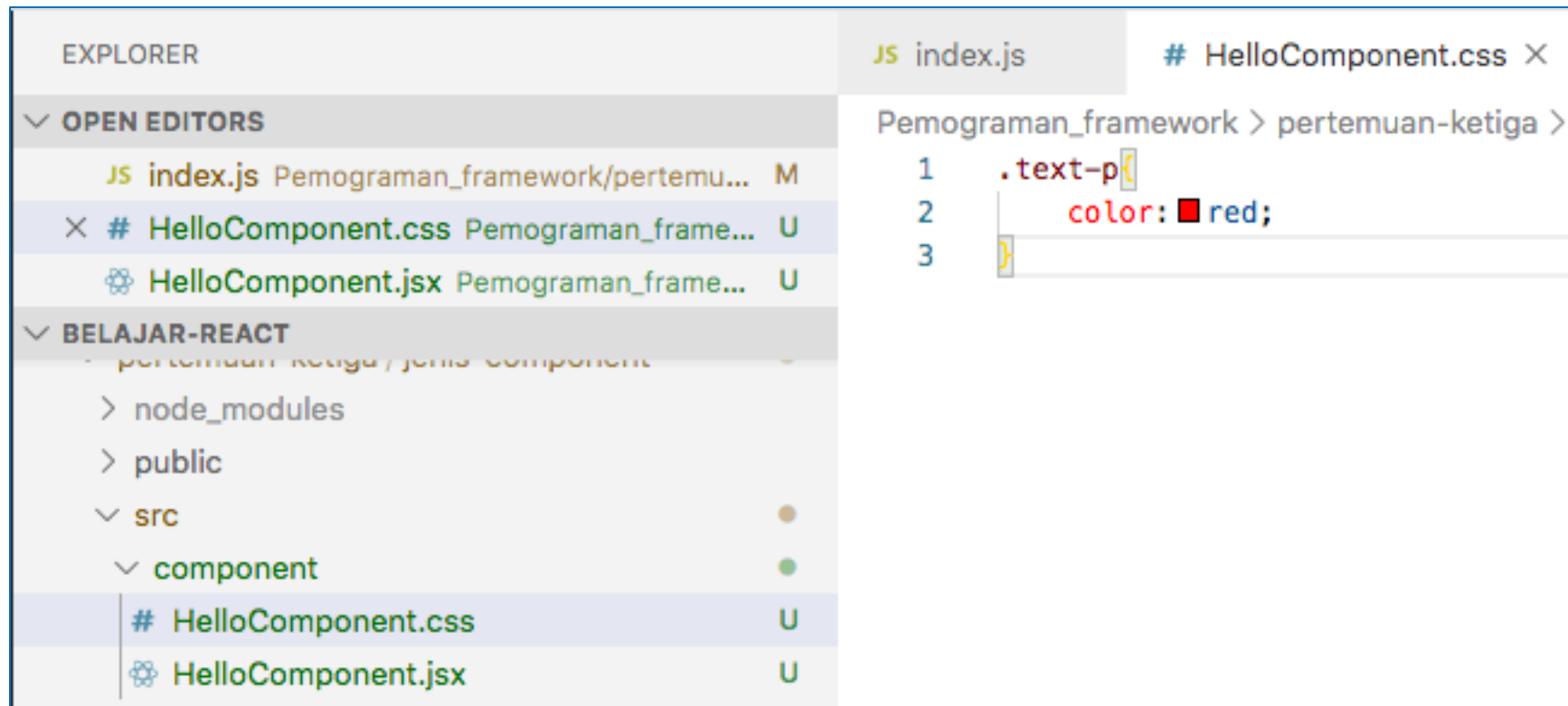
- Untuk menjalankan StateFullComponentnya dengan cara tambahkan beberapa code pada index.js seperti gambar dibawah ini (line 7 dan 10)

```
JS index.js  X  StateFullComponent.jsx  HelloComponent.jsx
Pemograman_framework > pertemuan-ketiga > jenis-component > src > JS index.js
3  import './index.css';
4  // import App from './App';
5  import * as serviceWorker from './serviceWorker';
6  import HelloComponent from './component/HelloComponent';
7  import StateFullComponent from './container/StateFullComponent';
8
9
10 ReactDOM.render(<StateFullComponent />, document.getElementById('root'));
11
12 // If you want your app to work offline and load faster, you can change
13 // unregister() to register() below. Note this comes with some pitfalls.
14 // Learn more about service workers: https://bit.ly/CRA-PWA
15 serviceWorker.unregister();
16
```



MENAMBAHKAN FILE CSS PADA FOLDER COMPONENT

- 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)



MENAMBAHKAN FILE CSS PADA FOLDER COMPONENT

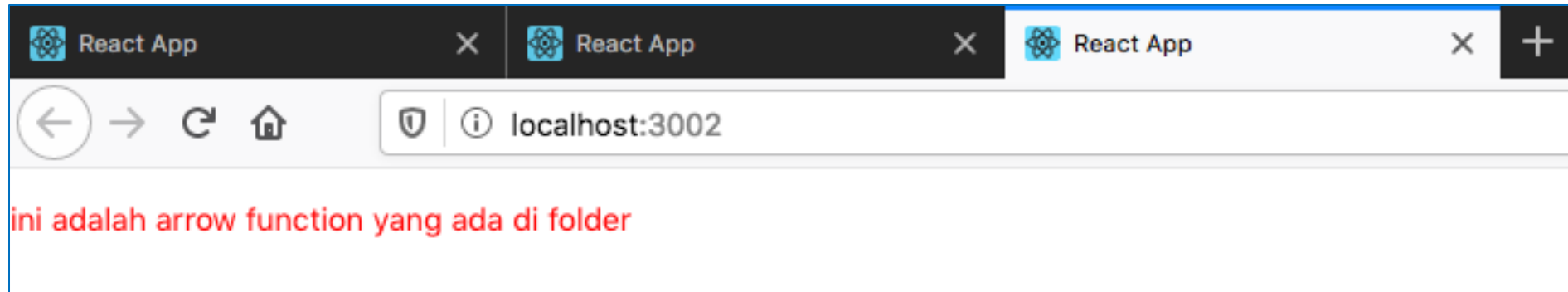
- Setelah itu import HelloComponent.css pada HelloComponent.js seperti pada berikut

```
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;
```



MENAMBAHKAN FILE CSS PADA FOLDER COMPONENT

- Hasilnya



TUGAS

- Buatlah menu login seperti pada gambar menggunakan CSS

localhost:3003

90%

...

Form Login

Tugas Pertemuan ketiga

Username

masukan username

Password

Masukkan Password Anda

Login

☒ Remember Me

Cancel

