



Bootcamp Dumbways.id Stage 2

By Ananda Rizky Alvie Nuryahya

Timeline

1. Front-end (Week 1)
2. Back-end (Week 2)
3. Integration (Week 3)
4. Mobile (Week 4)

Week 1

Day 1 - Fundamental React.js

Learning materials

1. JSX
2. Component
3. Embedding Expression
4. Event
5. Props
6. State
7. Conditional Rendering

▼ What is react.js?



React.js is a Javascript library for building user interfaces (UI)

▼ What is react.js about?

1. Facilitate the creation of web applications
2. Very easy to use
3. More interactive interface
4. Component based
5. And many more

▼ Why react.js?

1. Learn once Write anywhere!
2. Developed by Giant (Facebook)
3. Used by Giant (Facebook, Twitter, Instagram, Whatsapp, etc.)

▼ Prerequisite

1. NPX
 - What is and how to
 1. Is A package to execute application
 2. To install react, type `npx create-react-app my-app`
`cd my-app`
`npm start`
in terminal

▼ Learning materials

- Structure

1. React.js default port is 3000
 - How can I change port?
 1. Open package.json
 2. Find `"start"` under `"scripts"` section
 3. Change `"react-scripts start"` to `"PORT=3010(desired port) react-scripts start"`
2. What's inside React.js?
 - a. **.git** contains all information that is necessary for the project and all information relating commits, remote repository address, etc.
It also contains a log that stores the commit history. This log can help you to roll back to the desired version of the code.

- b. **node_module** is used to save all downloaded packages from NPM in your computer for the JavaScript project that you have.
Developers are always recommended to do a fresh install with npm install each time they downloaded a JavaScript project into their computer.
- c. **public** contains static files such as index.html, javascript library files, images, and other assets, etc. which you don't want to be processed by webpack.
Files in this folder are copied and pasted as they are directly into the build folder.
- d. **src** is the heart of React application as it contains JavaScript which needs to be processed by webpack. In this folder, there is a main component App.js, its related styles (App.css), test suite (App.test.js). index.js, and its style (index.css); which provide an entry point into the App. Lastly, it contains registerServiceWorker.js which takes care of caching and updating files for the end user. It helps in offline capability and faster page loading after the first visit.

NOTE:

You will find `function App() {`
`return ()} export default App;` inside app.js.

The function is component based and you have to import it to index.js as component or `<App />` inside `<React.StrictMode>` in order to run display the component

- **JSX**

- ▼ What is JSX?

1. JSX stand for Javascript XML.
2. JSX is a library syntax in Javascript that allows us to create a UI for react.js, JSX might remind you of a template language, the difference is that JSX is equipped with the full power of Javascript.
3. In conclusion, JSX can create elements or tags in the reaction. The general form of JSX is very similiar to HTML

Example:

```
const name = 'Budi';  
  
const element = <h1>Halo, {name}</h1>;
```


```
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

- **Components**

- ▼ **Template**

GitHub - DumbwaysDotId/reactjs-fundamental-template: Meeting 1

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

 <https://github.com/DumbwaysDotId/reactjs-fundamental-template/tree/1.Component>

DumbwaysDotId/**reactjs-fundamental-template**

Meeting 1

3 Contributors 0 Issues 0 Stars 21 Forks

NOTE:

1. You have to use capital in every start of component's name in order for it to work
2. Use `<Componentname />` to call component
3. You can call `<Componentname />` inside another component
4. Components are reusable

EXAMPLE:

in app.js

```
function Component() {
  return (
    <div>
      <h3>Welcome</h3>
    </div>
  )
}

export default Component
```

in index.js

```
import Component from './Component'

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

- **Embedding Expressions**

▼ What is?

1. Embedding expressions is used to call or execute a function or variable using {}

NOTE:

Use parameter or `()` inside embeded expression to call a function

- **Event**

NOTE:

Use camelCase to call events

1. Don't use `onclick=`

use `onClick=`

2. Don't use `onClick=(alert('helo'))`

use `onClick={() => alert('helo')}`

EXAMPLE:

```
onClick={() => alert('helo')}
```

or

```
function Greeting(){
  alert('helo')
```

```
}
```

```
onClick={Greeting}
```

- **Props**

- ▼ What is?

1. terms like attribute in HTML
2. Can be used to send data between components

NOTE:

Use {} for number data type

EXAMPLE:

in list.js

```
function List(props){  
  return(  
    <div>  
      <h1> {props.data}</h1>  
    </div>  
  )  
}  
  
export default List
```

in props.js

```
import List from "../components/list";  
  
function Props() {  
  return(  
    <div>  
      <List data="Honda Civic" />  
      <List data="Mustang" />  
    </div>  
  );  
}  
  
export default Props;
```

- **State**

- ▼ What is?

1. Have the same concept as Javascript variable but kind of different
2. Used as data container and for updating data

NOTE:

State contains an array with 2 indexes when it's value not defined, first index contains data and second index contain function for update data

EXAMPLE:

in state.js

```
import {useState} from "react";
```

```
const [number, setNumber] = useState(35)
```

**The value would be 35 in setValue when logged (console.log)*

```
const Add = () => {  
  setNumber(number - 1)  
}
```

**now the value would be 34 in setValue when logged and will be updated on browser*

```
function state(){  
  
  return(  
    <div>  
  
    </div>  
  );  
}
```

- **Conditional Rendering**

NOTE:

ternary is a simple method for iteration

EXAMPLE:

```
const PrivatePage = () => {
  return(
    <div>
      <h1>Welcome</h1>
      <button onClick={() => setIslogin(false)}>Log out</button>
    </div>
  )
}

function GuestPage(props) {
  const setIsLogin = props.setIsLogin

  function login(){
    setIsLogin(true)
  } *setIsLogin value would be changed to true when a the function
  called in a button

  return(
    <div>
      <h1>Please Login</h1>
      <button onClick={Login}>Login</button> *this button would call
      Login function when clicked
    </div>
  )
}

function ConditionalRendering(){
  const isLogin, setIsLogin = useState(false)
  return {
    <div>
      (isLogin == true ? <PrivatePage setIsLogin={setIs:Login} /> :
      <GuestPage setIsLogin={setIs:Login />)
    </div>
  )
}
```

Day 2 - Fundamental React.js

coming soon...