

# React JS

Ning Ratwastuti

Sisia Dika Ariyanto

Roni Prasetyo

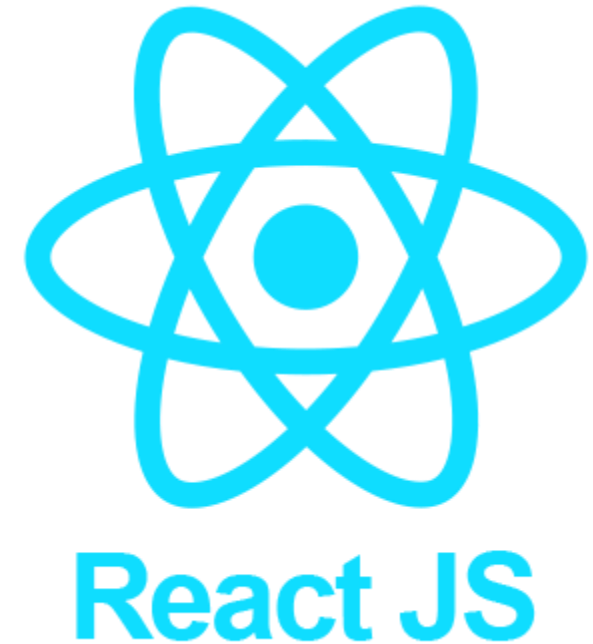
Elora Manuella Amei

# Overview

- Pengenalan React JS
- Virtual DOM
- Komponen React JS
- Library React JS
- React Router
- Axios
- Install React JS

# Pengenalan React JS

- React JS merupakan **library JavaScript** yang digunakan untuk membuat **user interface (UI) website**. React JS berisi **library kode-kode JavaScript** yang juga berfungsi untuk **mempermudah dan mempercepat web development**.
- Konsep utama React JS adalah **Document Object Model (DOM) virtual**. Artinya, **React JS adalah tree** yang dibuat berdasarkan **komponen JavaScript yang meniru tree dari DOM**.



# Virtual DOM

- Library JavaScript ini menggunakan **salinan dari DOM yang disebut Virtual DOM**. DOM standar adalah cara menyajikan file XML dan HTML yang menentukan struktur logis dokumen dan metode untuk mengakses serta menanganinya. Virtual DOM memiliki properti yang sama dengan DOM asli, tetapi **tidak secara langsung mengaplikasikan perubahan**.
- React menggunakan komponen yang memiliki **state**. Saat state berubah, React memperbarui Virtual DOM lalu membandingkannya dengan versi sebelum state berubah dan menggunakan algoritma yang dioptimalkan dengan baik untuk menjaga semuanya berjalan dengan cepat. Singkatnya, menggunakan **Virtual DOM memungkinkan interface untuk memperbarui data yang dibutuhkan setiap kali terjadi perubahan, tanpa harus mengubah seluruh halaman, perubahan menyeluruh biasanya lebih memakan waktu**.

# Virtual DOM

## Real & Virtual DOMs



# Komponen React

1. **Functional Component**
2. **Class Component**

# Functional Component

Merupakan komponen yang **tidak memiliki statusnya sendiri (object state)** dan **tidak memiliki method render**, biasanya hanya mengembalikan suatu **HTML** untuk ditampilkan. Functional component biasa disebut sebagai komponen stateless. Namun, sebagai gantinya functional component memiliki sesuatu yang dinamakan hook state.

```
function Header() {  
  return (  
    <div>  
      <h1>Memahami Konsep Dasar ReactJs</h1>  
    </div>  
  );  
}
```

# Class Component

Merupakan komponen yang **dapat menampung dan mengelola statusnya dan memiliki metode render() untuk mengembalikan JSX** di layar. Komponen ini juga disebut komponen stateful, karena mereka dapat memiliki status. State pada component ini berbentuk object yang nilainya dapat dimanipulasi.

```
class Clock extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { date: new Date() };  
  }  
  
  render() {  
    return (  
      <div>  
        <h1>  
{this.state.date.toLocaleTimeString()}</h1>  
      </div>  
    );  
  }  
}
```



# State

State adalah **suatu object** penyimpan data yang **nilainya dapat diperbarui** dan dapat berfungsi **untuk memicu pembaruan tampilan dari website**. State bersifat mutable yang **nilainya dapat diperbarui menggunakan bantuan method `setState()`**. Tiap saat nilai dari suatu state diubah, maka tampilan dari website yang mengandung state tersebut akan ikut diperbarui.

```
class Header extends React.Component {
  state = {
    title: "Memahami Konsep Dasar ReactJs"
  };

  changeTitle = () => {
    this.setState({
      title: "Memahami Konsep Dasar ReactJs Secara
Lengkap"
    });
  }

  render() {
    return (
      <div>
        <h1>{this.state.title}</h1>
        <button onClick={this.changeTitle}>Ubah
Judul</button>
      </div>
    );
  }
}
```

# Props

Pada dasarnya, props memiliki fungsi serupa seperti state. 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.** Props bertindak sebagai atribut yang dapat di kirim ke component yang lain (child component). Data yang disimpan di dalam props tidak bisa diubah karena sifatnya **read-only**.

```
1  import React from 'react';
2
3  const WelcomeMessage = (props) => {
4    |   return <h1>Welcome, {props.name}</h1>;
5    |
6    |
7  }
8
9  const App = () => {
10   |   return (
11   |     <div>
12   |       <h1>My App</h1>
13   |       <WelcomeMessage name="John" />
14   |     </div>
15   |   );
16   |
17 }
18
19 export default App;
```

# Install React JS

Pada dasarnya untuk menjalankan react js kita harus menambahkan **NodeJS Package Manager (NPM)** atau sejenisnya.


nodeJS dapat didownload pada link berikut.

<https://nodejs.org/en/download/>

```
C:\Users\lab_mi>node -v  
v20.11.1
```

## Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) 

Downloads Node.js **v20.11.1**<sup>1</sup> with long-term support.  
Node.js can also be installed via [package managers](#).

Want new features sooner? Get **Node.js v21.7.1**<sup>1</sup> instead.

# Struktur Project React

```

✓ my-app
  > node_modules
  > public
  > src
  ◆ .gitignore
  {} package-lock.json
  {} package.json
  ⓘ README.md
  
```

**node\_modules** → module/library yang akan digunakan.

**public** → berisi halaman primer untuk index.html


**src** → berisi komponen dan halaman-halaman.

**.gitignore** → pengabaian file/direktori yang tidak perlu di push.

**package.json** → file render reactJS

# Cara Menambahkan Bootstrap di React JS

1. Menggunakan Bootstrap CDN.
2. Menginstall Bootstrap melalui NPM.




### Install via package manager

Install Bootstrap's source Sass and JavaScript files via npm, RubyGems, Composer, or Meteor. Package managed installs don't include documentation or our full build scripts. You can also [use any demo from our Examples repo](#) to quickly jumpstart Bootstrap projects.

```
$ npm install bootstrap@5.3.3
```

```
$ gem install bootstrap -v 5.3.3
```

[Read our installation docs](#) for more info and additional package managers.



### Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/c:"
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/:"
```

Buka website bootstrap: <https://getbootstrap.com/>

# Library React JS

**React-Router** → digunakan untuk melakukan proses routing pada web yang menggunakan library ReactJS.

**Redux** → digunakan untuk mengatur state pada aplikasi ReactJS.

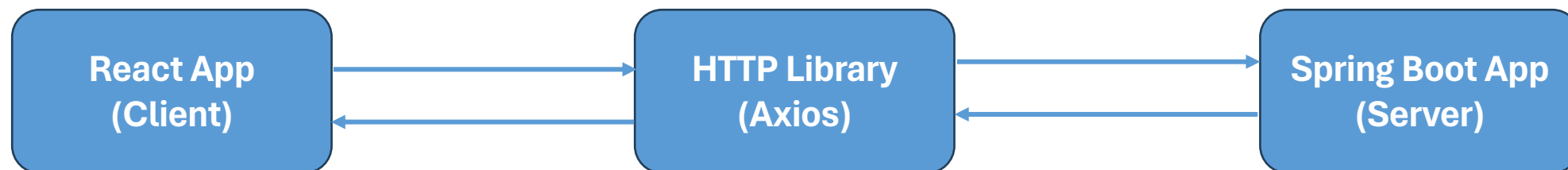
**Axios** → digunakan untuk melakukan proses HTTP request pada website dengan React JS.

**Material UI** → digunakan untuk mengimplementasikan desain dari Google Material Design.

**Styled Components** → digunakan untuk melakukan styling pada tampilan sebuah website, bisa dibilang Styled Components ini bisa seperti CSS di dalam Javascript

# Axios

- Axios merupakan **Library Opensource** yang digunakan untuk **request data melalui http**.
- Axios memungkinkan pengembang untuk melakukan permintaan (requests) ke server untuk mengambil data dari API atau mengirim data ke server dengan mudah.



# Axios

```
import React from "react";
import axios from "axios";
export default class MyList extends React.Component {
  state = {
    blogs: [],
  };
  componentDidMount() {
    axios
      .get(
        `https://jsonplaceholder.typicode.com/posts`
      )
      .then((response) => {
        const posts = response.data;
        this.setState({ posts });
      });
  }
  render() {
    return (
      <ul>
        {this.state.posts.map((post) => {
          post.title;
        })}
      </ul>
    );
  }
}
```

## GET Request

```
handleSubmit = event => {
  event.preventDefault();
  axios.delete(
    `https://jsonplaceholder.typicode.com/posts/${this.state.postName}`
  )
    .then(res => {
      console.log(res);
      console.log(res.data);
    })
}
```

## DELETE Request



# Axios

```
import React from "react";
import axios from "axios";
export default class AddPost extends React.Component {
  state = {
    postTitle: "",
  };
  handleChange = (event) => {
    this.setState({ postTitle: event.target.value });
  };
  handleSubmit = (event) => {
    event.preventDefault();
    const post = {
      postName: this.state.postName,
    };
    axios
      .post(
        `https://jsonplaceholder.typicode.com/posts`,
        { post }
      )
      .then((res) => {
        console.log(res);
        console.log(res.data);
      });
  };
};
```

```
render() {
  return (
    <div>
      <form onSubmit="{this.handleSubmit}">
        <label>
          Post Name:
          <input
            type="text"
            name="name"
            onChange="{this.handleChange}"
          />
        </label>
        <button type="submit">Add</button>
      </form>
    </div>
  );
}
```

## POST Request

# React Router

- **React Router** adalah sebuah library yang digunakan dalam pengembangan aplikasi web dengan menggunakan React. Library ini memungkinkan untuk **membuat navigasi pada aplikasi web** menjadi cepat dan SPA atau Single Page Application. Routing memungkinkan navigasi antar halaman tanpa perlu melakukan refresh pada halaman.

Langkah-langkah untuk menambahkan React Router pada aplikasi React:

## 1. Install React Router DOM pada terminal/CMD

```
npm install react-router-dom
```

# React Router

Langkah-langkah untuk menambahkan React Router pada aplikasi React:

## 2. Import React Router DOM dan Membungkus Komponen dengan Router (index.js)

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { BrowserRouter } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);

// 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();
```

# React Router

Langkah-langkah untuk menambahkan React Router pada aplikasi React:

## 3. Membuat Navbar (Navbar.js)

```
import { Link } from "react-router-dom";

const Navbar = () => {
  return (
    <nav>
      <ul>
        <li>
          <Link to="/">Home</Link>
        </li>
        <li>
          <Link to="/about">About</Link>
        </li>
        <li>
          <Link to="/contact">Contact</Link>
        </li>
      </ul>
    </nav>
  );
}

export default Navbar
```

**<Link>** digunakan untuk mengatur URL dan mengaktifkan navigasi ke bagian-bagian berbeda dari aplikasi web.

# React Router

Langkah-langkah untuk menambahkan React Router pada aplikasi React:

## 4. Membuat File Komponen React pada setiap halaman

```
//Home.js
import React from 'react'
import Navbar from './Navbar'

const Home = () => {
  return (
    <div>
      <Navbar />
      <h1>Home</h1>
    </div>
  )
}

export default Home
```

```
//About.js
import React from 'react'
import Navbar from './Navbar'

const About = () => {
  return (
    <div>
      <Navbar />
      <h1>About</h1>
    </div>
  )
}

export default About
```

```
//Contact.js
import React from 'react'
import Navbar from './Navbar'

const Contact = () => {
  return (
    <div>
      <Navbar />
      <h1>Contact</h1>
    </div>
  )
}

export default Contact
```

# React Router

Langkah-langkah untuk menambahkan React Router pada aplikasi React:

## 5. Menambahkan Route di App.js

```
//App.js
import { Route, Routes } from 'react-router-dom';
import './App.css';
import Home from './Home';
import About from './About';
import Contact from './Contact';

function App() {
  return (
    <div className="App">

      </div>
    );
}

export default App;
```

```
<div className="App">
  <Routes>
    <Route path="/" element={<Home/>} />
    <Route path="/about" element={<About/>} />
    <Route path="/contact" element={<Contact/>} />
  </Routes>
</div>
```

- Elemen-elemen **Route** dibungkus di dalam **Routes**.
- **Path** mewakili path URL ke komponen yang direpresentasikannya.
- **element** menggambarkan komponen yang ditunjuk oleh path.

# Terimakasih