

[NAMA MODUL]

Pathway	:	
Sub Pathway	:	
Kompetensi	:	
Key Element	:	
Level	:	
Developer	:	
Approval	:	1. Mgr Learning Development 2. SM School of Dig Platform & Service

} Sesuai
Dirkom-10

Nama/ NIK/
Bidang
Atau Nama
Vendor

Modul Pembelajaran	:	React JSX
Deskripsi Modul	:	<i>React JSX adalah syntax HTML yang diinput dalam Javascript agar bisa membikin komponen user-interface secara modular, dinamis, dan gampang diolah.</i>
Enabling Learning Objective	:	Memahami tentang React JSX dan bagaimana bisa menggunakan nya untuk komponen UI
Daftar Topik	:	<div> 1. Why use JSX ? 2. Nested Elements in JSX 3. JSX attributes 4. JSX comments 5. JSX styling </div> <div>6. Installation</div>
Searching Keyword	:	1. ReactJS 2. create-react-app 3. JSX 4. XML 5. Virtual DOM 6. styled-components

REACTJS

React JSX

Calvin Susanto Putra

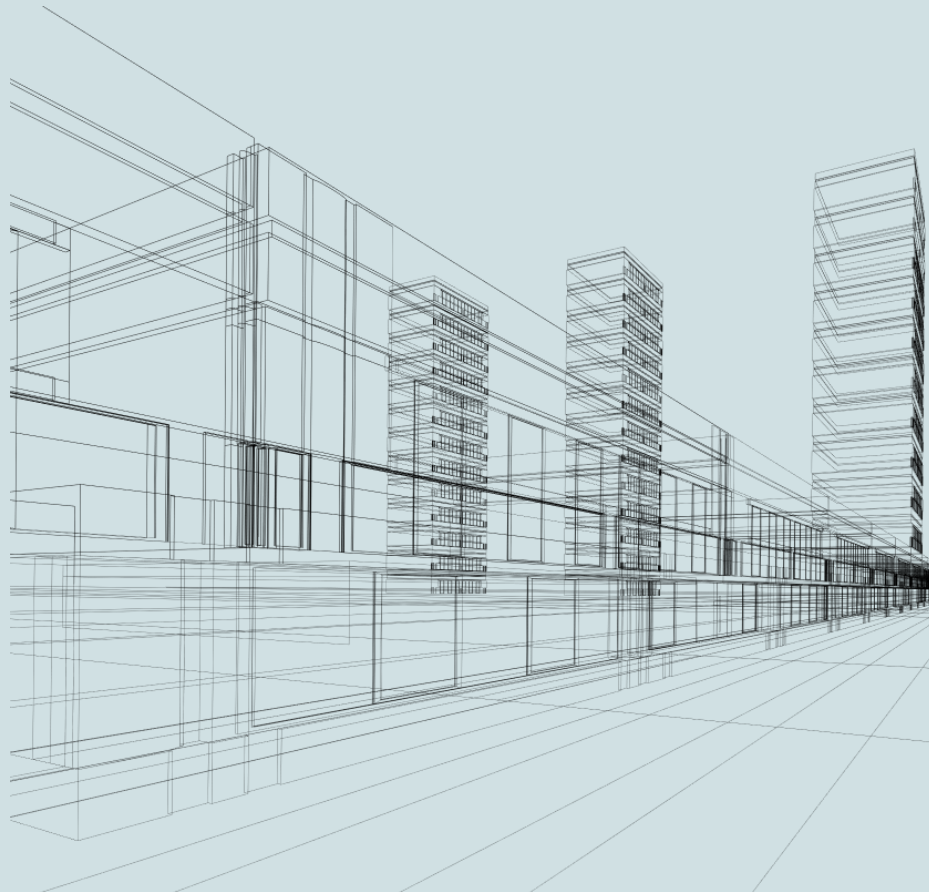
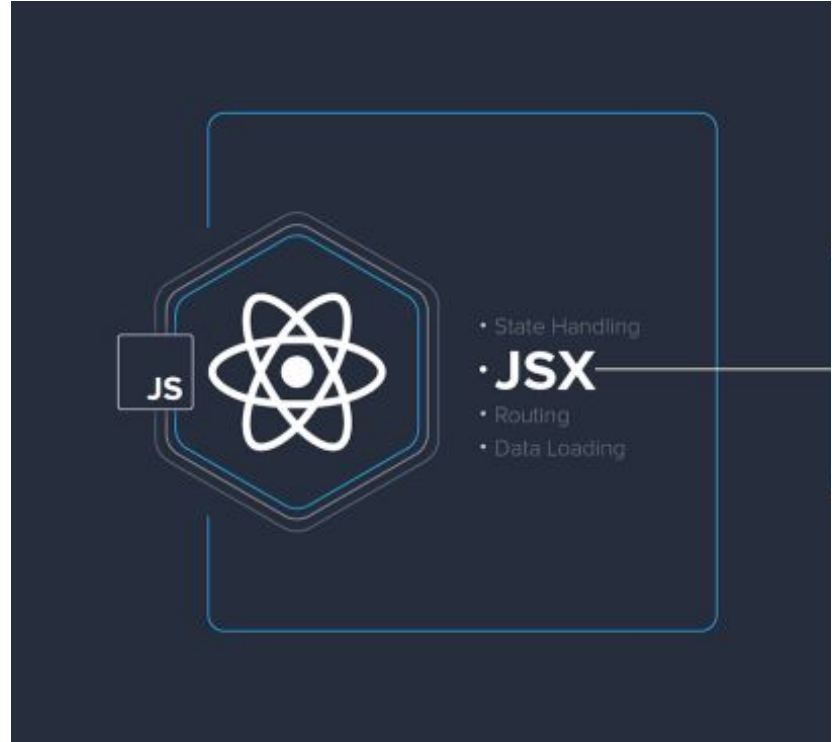


Table of Contents

- a. Why use JSX ?
- b. Nested elements in JSX
- c. JSX attributes
- d. JSX comments
- e. JSX styling

Why use JSX ?

- stands for Javascript XML, it is a form of HTML written in Javascript and used to build React UI components
- can access a virtual DOM, which replicates the real DOM of a standard HTML document and perform real-time updates to the structure
- allows for the inclusion of programming logic to directly handle states and events from the user
- has an active developer community and official documentation, surpassing jQuery as the most popular web developer framework by [40.14%](#)



Why use JSX ?

- An example of JSX as a variable declaration:

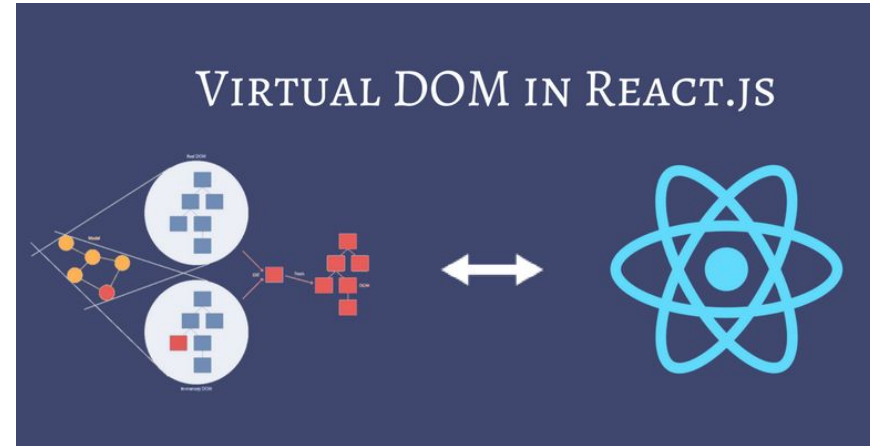
```
const helloWorld = <h1> Hello World </h1>;
```

- Another example as a React component:

```
const helloWorld = () => {  
  return (  
    <div>  
      <h1> Hello World </h1>  
    </div>  
  );  
}  
  
export default helloWorld;
```

Nested Elements in JSX

- JSX allows for the nesting of elements as components within other Javascript files
- compatible files are in '.js, .jsx, .ts, .tsx' format, where 'TS' stands for Typescript, a more advanced class-based language derived from Javascript
- this allows for components or elements to be utilized and reused in multiple ways and sections



Nested Elements in JSX

- Nesting the helloWorld component in another React element; i.e. HomePage.jsx

```
import helloWorld from
'../components/helloWorld';

const HomePage = () => {

  return (
    <div>
      <helloWorld />
    </div>
  );
}

export default HomePage;
```


Nested Elements in JSX

- Nesting the HomePage component into the App.js file; the application is wrapped with a BrowserRouter to render pages

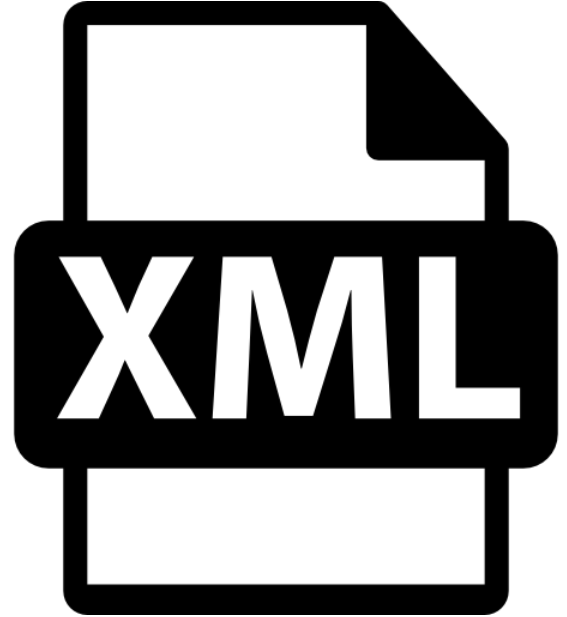
```
import HomePage from '../routes/HomePage';

function App() {
  return (
    <>
      <Routes>
        <Route path="/" element={<HomePage />}/>
      </Routes>
    </>
  );
}

export default App;
```

JSX attributes

- JSX attributes are written in a syntax not too dissimilar from those of a normal HTML document
- differences in JSX to HTML attribute types include the conversion of 'class' to 'className', due to the former being a reserved keyword in Javascript
- Otherwise, custom attributes for elements could also be created with a 'data-' syntax
- For example, 'data-testCase'



JSX attributes

- An example of JSX attributes in use:

```
return (  
  <div>  
    <div className="container" id="wrapper">  
      <a href="https://www.google.com/">  
        Google  
      </a>  
    </div>  
  </div>  
>);  
>
```

JSX attributes

- For the implementation of custom attributes:

```
return (  
  <div>  
    <a data-testCase="test-case" href="https://www.google.com/">  
      Google  
    </a>  
  </div>  
>);  
>
```

JSX attributes

- More advanced examples of JSX attributes allows users to perform callbacks on functions and variables

```
const dummy = [  
  name: "React JSX",  
  module: "04",  
  author: "Telkom Corporate University",  
  date: "01/07/22"  
];  
  
function reactJSX(dummy) => {  
  return `  
    ${dummy.name}  
    ${dummy.module}  
    ${dummy.author}  
    ${dummy.date}  
  `;  
}
```

```
import { dummy, reactJSX } from  
  '../components/reactJSX'  
  
function App(dummy) {  
  return (  
    <div>  
      <h1>  
        { reactJSX(dummy) }  
      </h1>  
    </div>  
  );  
}  
  
export default App;
```

JSX Comments

- the difference in React JSX, is that comments are written as both a block comment with curly braces

```
function App() {  
  return (  
    <div>  
      { /* Hello World, this is a JSX comment */ }  
    </div>  
  );  
}
```

- .comments are used to notify or deliver information about the code, making it vital for documentation and refactoring purposes

JSX Comments

- comments in JSX can be written anywhere in the code regardless of structure, assuming that the aforementioned syntax is used
- comments are used to notify or deliver information about the code, making it vital for documentation and refactoring purposes



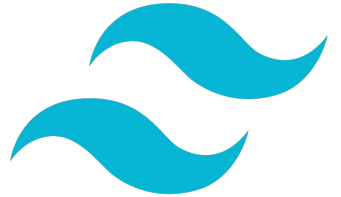
JSX Comments

- An example in appropriate comment usage describing App.js:

```
{/* function declaration for App.js */}  
  
function App() {  
  
  {/* Stateful component */}  
  
  const [data, dummyData] = useState();  
  
  return (  
    <div>  
  
      { /* Hello World, this is a JSX comment */ }  
  
    </div>  
  );  
}  
  
{/* Export App.js for rendering */}  
  
export default App;
```


JSX Styling

- styling in JSX can be done with various methods, all of which are supported internally and externally
- the primary method of styling requires a `index.css` file stored in the “src” folder; all styling done in this file will affect any JSX elements in `App.js`
- Otherwise, inline-styling and external frameworks or libraries are available with NPM



JSX Styling

- styling JSX conventionally requires the `style.css` file to be imported, with standard attribution:

```
import './styled.css';

function App() {

  return (
    <div className="App">
      <h1 className="heading"> Hello World </h1>
    </div>
  );
}

export default App;
```

JSX Styling

- An example of inline-styling in JSX:

```
import './styled.css';

function App() {

  return (
    <div className="App">
      <h1 style={{ color: "#000", margin-top: "1rem", text-align: "center" }}>
        Hello World
      </h1>
    </div>
  );
}

export default App;
```

- .inline-styling passes an object to the style property of 'color', 'margin-top', and 'text-align'

JSX Styling

- Otherwise, external libraries or frameworks could be installed using the 'npm install' command in a Node.js terminal
- An example of the external framework, Tailwind CSS, being used to style a component

```
function App() {  
  return (  
    <div className="grid items-center justify-center p-5">  
      <h1 className="text-4xl font-bold text-blue-500 m-auto">  
        Hello World  
      </h1>  
    </div>  
  );  
}  
  
export default App;
```

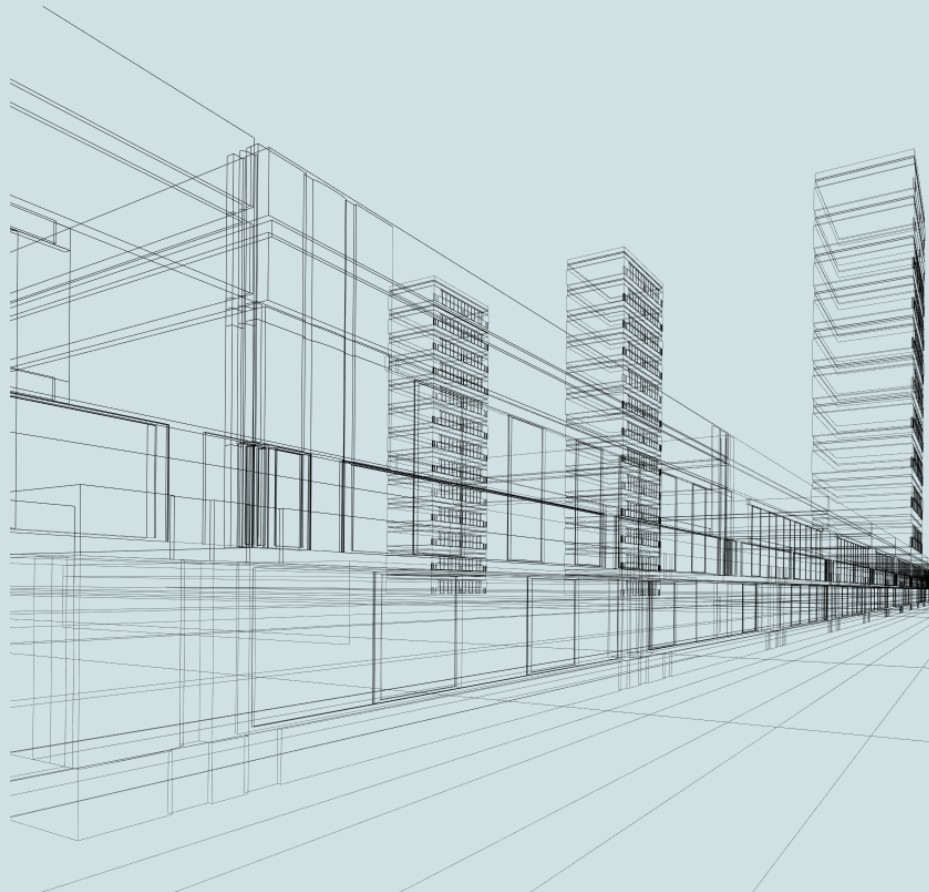
References

- <https://reactjs.org/docs/introducing-jsx.html>
- <https://create-react-app.dev/>
- <https://riptutorial.com/reactjs/example/3846/nesting-components>
- <https://medium.com/@ralph1786/how-to-style-react-components-5b3111d77ab2>

REACTJS

React JSX

Calvin Susanto Putra



Kelengkapan Modul

- Pre & Post Test (minimal 10 soal, pilihan ganda a-d)
- Pre Reading
- Teaching Notes
- Referensi

Assessment – [Pre & Post Test]

Pre-Reading



ReactJS is a Javascript library for building user interfaces. This PDF ebook is a portable offline reference guide for the community, can be printed or synced with your kindle for a better reading experience. It's based on ReactJS version v.16.13.1

This is an excellent reference format for ReactJS beginners, is not a tutorial just the official documentation for your reference

TEACHING NOTES*



Lesson/Topics



Case Study



Clips



Games



Others :

OBJECTIVES

Memahami tentang JSX dan penggunaan nya untuk bikin komponent

PERSIAPAN

*Offline: Laptop
Online: Link Zoom*

OVERVIEW

React JSX adalah syntax HTML yang diinput dalam Javascript agar bisa membikin komponen user-interface yang modular dan gampang diolah.

DIRECTION

- Menjelaskan kepentingan JSX untuk bikin komponen dalam proyek React
- Menjelaskan perbedaan atribut dan komen diantara syntax web development biasa
- Demontrasikan metode untuk style elemen atau komponen JSX
- Mengatur dan mengelola arah dan jalan-nya diskusi;
- Memberikan contoh-contoh, kasus dan ilustrasi;
- Memberikan apresiasi dalam tanya jawab;
- Memberikan debrief atau insight/pesan dari kegiatan;
- Memberikan dan menyampaikan kesimpulan dan di tiap sesi atau di akhir sesi

PERALATAN YANG DIBUTUHAN

*Offline : Laptop
Online: Link Zoom*

**Catatan : jumlah halaman teaching Note dapat ditambah seauai kebutuhan*

DIRECTION		
Section (slide/clip/case/etc)	Activities (Hal apa yang akan dilakukan untuk setiap section-nya)	Description (hal-hal spesifik yang harus dilakukan, misal : penjelasan dan atau penekanan materi, pengelolaan arah dan jalannya diskusi, wrap up/insight atau debrief, etc)
5-6	Presentasi, Diskusi	Menjelaskan kepentingan JSX untuk membikin komponen dalam proyek React
7-9	Presentasi, Diskusi, Case Study	Menjelaskan tentang penggunaan nested elements dalam ReactJS
10-13	Presentasi, Diskusi, Case Study	Menjelaskan tentang atribut dan tipe data untuk JSX
14-17	Presentasi, Diskusi, Case Study	Menjelaskan tentang syntax, penggunaan, dan kepentingan komen di JSX
18-21	Presentasi, Diskusi, Case Study	Menjelaskan metode untuk styling elemen ReactJS