Get Started
React.js Library

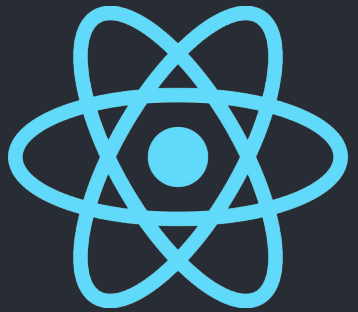


Table of Contents

Fun Fact

React	React Native
React !== React Native	
Library	Framework
Frontend Library	Mobile Framework
Building UI	Building Native Mobile
JavaScript	
Facebook	

Prerequisites

Prerequisites

Basic	Advanced
Variable: let and const	Destructing: Array and Object
Data Types	Rest parameters and spread
Template Literals	Array methods: forEach, map, find, filter
Conditional	Modules
Looping	Promise + Async Await
Function: Declaration, Expression, Arrow	
Array and Object	

Referensi: JavaScript Info - Modern JavaScript.

Glossary

Glossary

Term	Description
SPA (Single Page Application)	Aplikasi yang hanya terdiri 1 halaman
JSX	Syntax extension untuk React
Elements	Building blocks dari aplikasi React
Components	Reusable UI
Props	Input pada component
State	Data pada component

Referensi: React - Glossary of React Terms.

Install

Install

React diinstall hanya dengan menambahkan script react.

- Buat dom container (div).
- Tambahkan script react dan jsx (optional).
- Buat dan render component.

Referensi: React - Add React to Website.

index.html

```
<body>
  <!--
    Membuat div dengan id root.
    Digunakan untuk menampung hasil render react
  -->
  <div id="root"></div>

  <!--
    Tambahkan script react
   -->
  <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

  <!--
    Tambahkan script JSX.
   -->
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

  <!--
    Kodingan React
   -->
  <script src="./js/script.js" type="text/babel"></script>
</body>
```

script.js

```
/**
 * Membuat Component Hello
 * Component Hello mengemballikan/mencetak UI (elements)
 */
function Hello() {
  return (
    <div>
      <h2>Hello React</h2>
      <p>Saya Frontend Engineer</p>
    </div>
  );
}

// Akses element yang memiliki id root
const rootElement = document.getElementById("root");

// Buat React root untuk menampilkan Component di browser
const root = ReactDOM.createRoot(rootElement);

// Render component Hello ke root
root.render(<Hello />);
```

Congrats
You’re React Developer

Introduction JSX

JSX

Memudahkan penulisan HTML di React.

- Syntax extension untuk React.
- Mempermudah pembuatan UI.
- Dapat menyematkan expression: {expression}.
- Menggunakan camelCase untuk nama property: className, tabIndex

Referensi: React - Writing Markup with JSX.

script.js

```
function Hello() {
  const name = "Aufa Billah";

  return (
    <div>
      <h2>Hello React</h2>
      <p>Saya {name} - Frontend Engineer</p>
    </div>
  );
}
```

Rendering Elements

Elements

- Building blocks/satuan terkecil dari aplikasi React
- React Element is regularly HTML Element.: <p>, <h1>, <a>, <button>, <input>
- Mendeskripsikan apa yang ingin dilihat di layar.

Referensi: React - Rendering Elements.

script.js

```
/**
 * Membuat element
 */
const name = "Aufa Billah";
const element = <h1>Halo {name}</h1>;

/**
 * Render Component Hello ke div#root
 */
const root = document.getElementById("root");
ReactDOM.render(element, root);
```

Components and Props

Components

- Membagi UI menjadi bagian independen yang dapat digunakan kembali.
- Disusun oleh banyak elements.
- Sama seperti function.
- Menerima input/parameter yang disebut props.
- Mengembalikan elements yang merepresentasikan UI yang ingin ditampilkan.

Jenis:

- Functional Components: Dibuat menggunakan function.
- Class Components: Dibuat menggunakan class.

Referensi: React - Components and Props.

script.js

```
/**
 * Membuat component Header.
 * Component Header menampilkan navigasi.
 */
function Header() {
  return (
    <nav>
      <ul>
        <li>Home</li>
        <li>About</li>
        <li>Contact</li>
      </ul>
    </nav>
  );
}
```

script.js

```
/**
 * Membuat component Main.
 * Component Main menampung konten utama.
 */
function Main() {
  return (
    <main>
      <Hello />
      <Hello />
      <Hello />
      <Hello />
      <Hello />
    </main>
  );
}
```

script.js

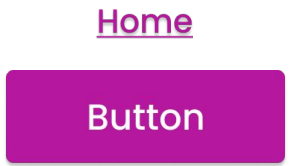
```
/**
 * Membuat component Footer.
 * Component Footer menampilkan footer.
 */
function Footer() {
  return (
    <footer>
      <h2>Copyright @aufaroot18</h2>
      <p>Created by React.js</p>
    </footer>
  );
}
```

script.js

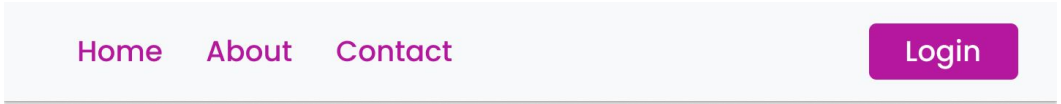
```
/**
 * Membuat component App.
 * Component utama yang menampung components lain.
 */
function App() {
  return (
    <div>
      <Header />
      <Main />
      <Footer />
    </div>
  );
}
```

Element vs Component

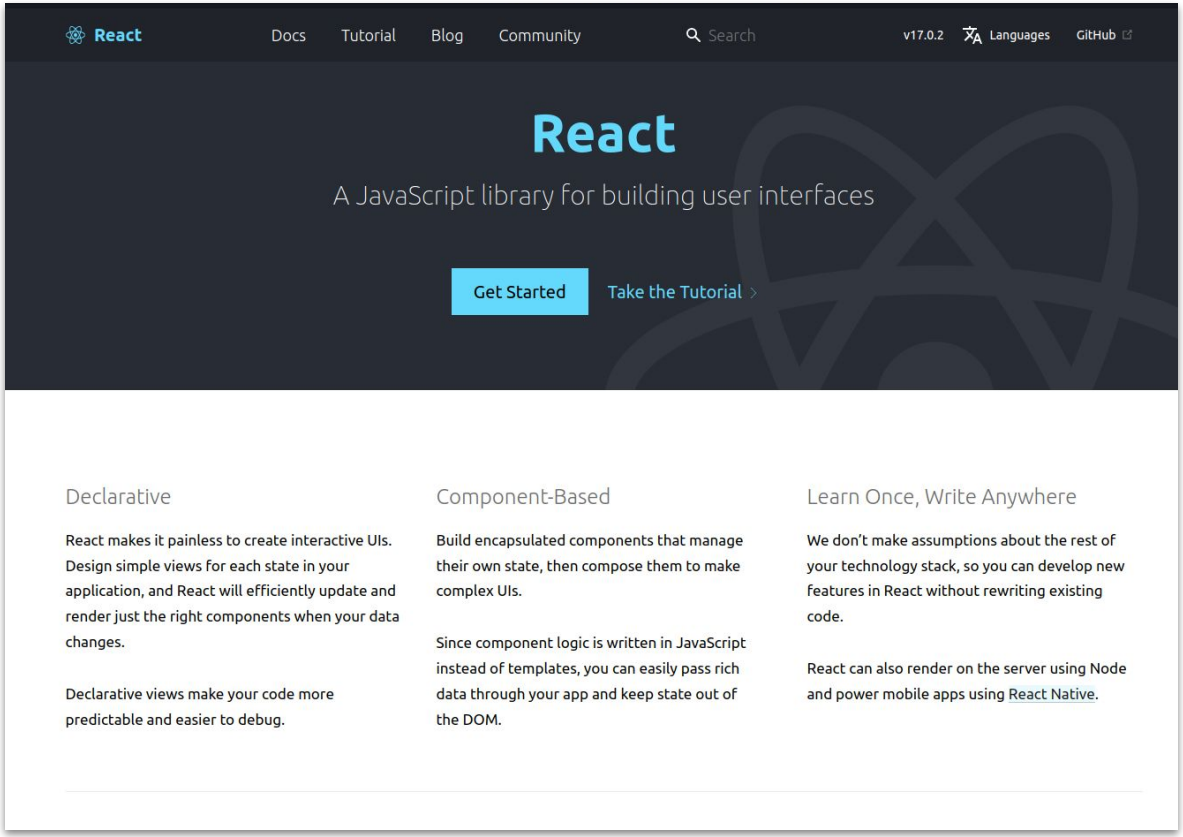
Element vs Component



Element



Component



[React - Homepage]

Page: Home

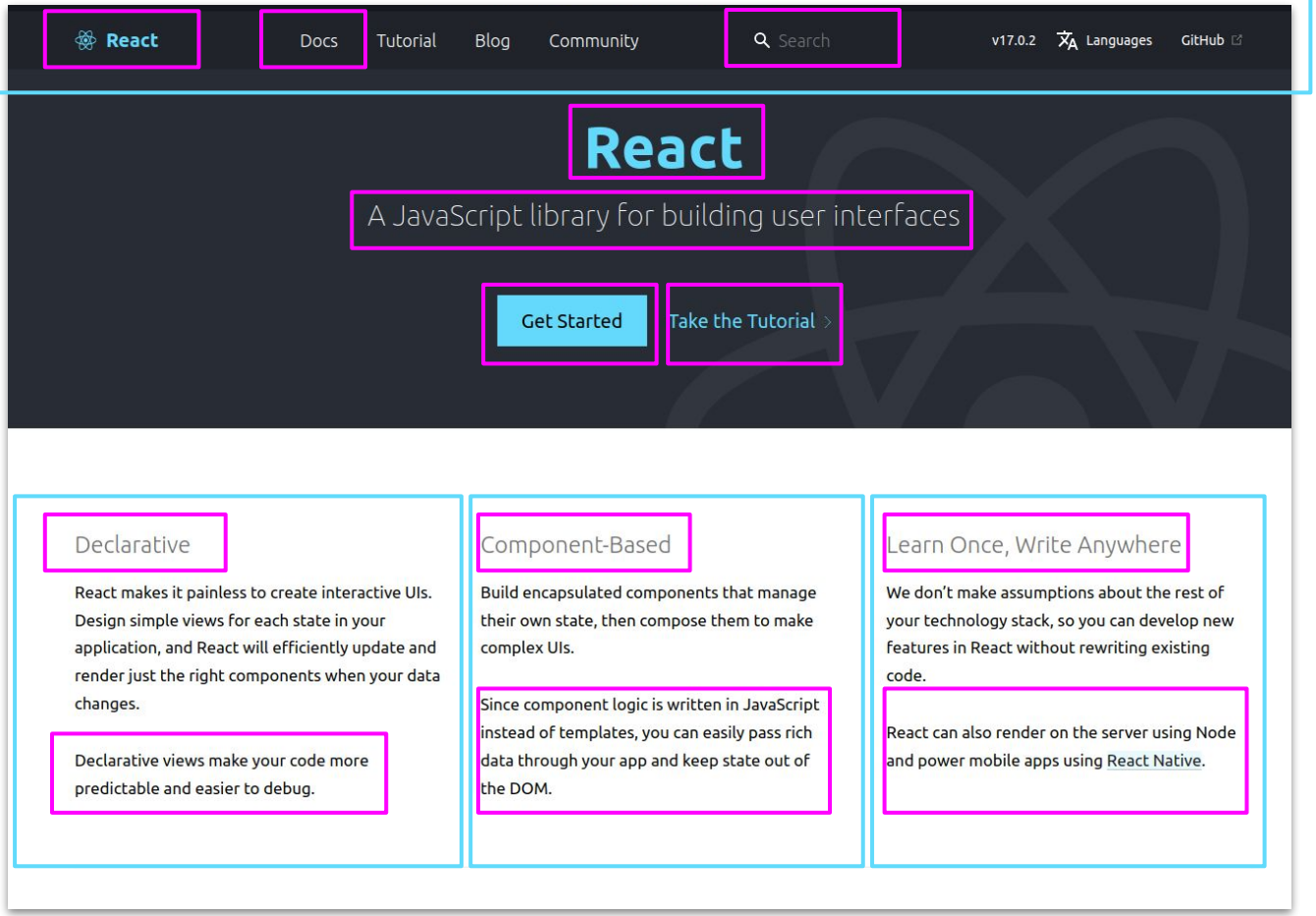
Components: Navbar, Card, Header

Elements: Link, Heading, Button, Text, Image

Navbar Component:

- Image: <img />
- Link: <a>
- Input Search: <input />

Card Component?



- Component
- Element

Props

- Component dapat menerima input/parameter yang disebut props.
- Singkatan dari properties.
- Membuat component menjadi lebih dinamis dan reusable.
- Props bersifat read-only (tidak boleh diubah).

Referensi: React - Components and Props.

script.js

```
function Main() {
  return (
    <main>
      {/**
       * Mengirim props ke component Hello.
       * nama props: name
       */}
      <Hello name="Aufa" />
      <Hello name="Mikel" />
      <Hello name="Hannah" />
      <Hello name="Jonas" />
      <Hello name="Martha" />
    </main>
  );
}
```

script.js

```
/**
 * Membuat component Hello.
 * Component Hello menerima inputan: props.
 */
function Hello(props) {
  // Melakukan destructing props (object)
  const { name } = props;

  return (
    <div>
      <h2>Hello React</h2>
      <p>Saya {name} - Frontend Engineer</p>
    </div>
  );
}
```

Recap

Recap

- JSX: Menuliskan HTML di JavaScript React.
- Elements: Building blocks terkecil dari React.
- Components: UI yang dapat digunakan kembali.
- Props: Input/parameter pada components.

Referensi: React - Components and Props.

CRA

Toolchain

- Scaling apps.
- Integrating third-party library.
- Detecting problem.
- Live-editing CSS and JS.
- Optimizing production.
- Zero config.

Referensi: React - Start a New React Project.

Toolchain

Alat yang mempercepat pengembangan aplikasi React.

- Create React App: Learning React or Creating single-page app.
- Next.js: Server-rendered website.
- Gatsby: Static content.

Create from scratch:

- Package manager: NPM or Yarn.
- Bundler: Webpack or Parcel.
- Compiler: Babel

Referensi: React - Start a New React Project.

Create React App

- npx create-react-app movie-database
- cd movie-database
- npm start

Referensi: Create React App - Getting Started.

Folder Structure

- public/index.html: Page template.
- src/index.js: JavaScript entry point.
- src/App.js: Main/first component.
- src: Coding.

Referensi: Create React App - Folder Structure.

App.js

```
import "./App.css";

function App() {
  return (
    <div>
      <h2>This is Create React App</h2>
    </div>
  );
}

export default App;
```

Folder components

- Sebelumnya menuliskan semua components dalam 1 file.
- Sebaiknya setiap component disimpan di file terpisah (module).
- Membuat component menjadi lebih dinamis dan reusable.
- Ketentuan: component disimpan di folder src/components.

Referensi: Create React App - Folder Structure.

components/Hello.js

```
function Hello(props) {
  const { name } = props;

  return (
    <div>
      <h2>Hello React</h2>
      <p>Saya {name} - Frontend Engineer</p>
    </div>
  );
}

export default Hello;
```

components/Hello.js

```
import "./App.css";
// Import Component Hello
import Hello from "./components/Hello";

function App() {
  return (
    <div>
      <h2>This is Create React App</h2>
      {/*
        Memanggil Component Hello.
        Mengirim props name
       */}
      <Hello name="Aufa" />
    </div>
  );
}

export default App;
```

Congrats Again
You’re React Developer

QnA

Take Away

Task

Description:

- Task boilerplate and description: Link.
- Practice Create React App.
- Practice component and module.

Assignment:

- Push code to Repository Github (use branch for task management).
- Task tidak perlu dizip dan folder node_nodules tidak perlu diupload.
- Submit link repository to elena: Link.

TODO

- Pindahkan component sebelumnya ke Create React App.
- Pisahkan setiap component menjadi component terpisah (module):
    - Hello.js: Component Hello
    - Header.js: Component Header
    - Main.js: Component Main
    - Footer: Component Footer
- Simpan component di folder src/components.
- Susun kembali semua component di App Component.
- Pastikan hasil Create React App sama seperti hasil sebelumnya.

Optional:

- Ubah semua component menjadi Arrow Function.

Attendance

Thanks