

# **PRAKTIKUM PENGEMBANGAN APLIKASI BERGERAK**

## **MODUL 1 Instalasi Environment dan Pengenalan React Tingkat Dasar**



**Sistem Informasi**  
Telkom University  
Surabaya

Disusun Oleh:  
Purnama Anaking, S.Kom., M.Kom.

**PROGRAM STUDI S1 SISTEM INFORMASI  
FAKULTAS REKAYASA INDUSTRI  
TELKOM UNIVERSITY SURABAYA  
2025**

# DAFTAR ISI

|  |    |
|--|----|
| DAFTAR ISI   | 2  |
| 1. Instalasi Environment                               | 3  |
| 2. Membuat & Menjalankan Project React Native via Expo | 3  |
| 3. Belajar Javascript Dasar                            | 4  |
| 3.1 Variables  | 5  |
| 3.2 Class  | 6  |
| 3.3 Inheritance  | 6  |
| 3.4 Arrow Function                                     | 7  |
| 3.5 Destructuring                                      | 8  |
| 3.6 Spread Operator                                    | 9  |
| 3.7 Array Method                                       | 10 |
| 4. Belajar React Dasar                                 | 10 |
| 4.1 JSX (Javascript XML)                               | 11 |
| 4.2 Class Component                                    | 12 |
| 4.3 Functional Component                               | 14 |
| 4.4 Event Handler                                      | 15 |
| 4.5 State  | 16 |
| 5. Tugas   | 17 |

# MODUL 1

## INSTALASI ENVIRONMENT & REACT TINGKAT DASAR

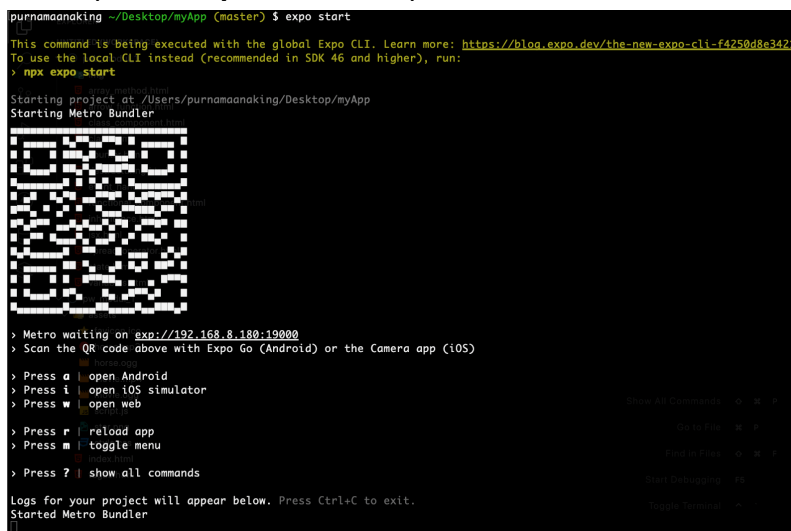
Pada praktikum kali ini kita akan mulai mempelajari tentang bagaimana instalasi environment untuk melakukan pengembangan aplikasi mobile berbasis React Native menggunakan Expo dan juga pengenalan React tingkat dasar. Kegiatan ini dilakukan agar mahasiswa mampu menerapkan konsep dasar React di dalam pemrograman React Native.

### 1. Instalasi Environment

- Download dan Install **NodeJS**: <https://nodejs.org/en/download/>. Jalankan command: **node -v** pada command prompt untuk memastikan bahwa proses instalasi NodeJs berhasil dilakukan.
- Download dan install **Git**: <https://git-scm.com/downloads>. Jalankan command: **git --version** pada command prompt untuk memastikan bahwa proses instalasi Git berhasil dilakukan.
- Download dan Install aplikasi **Expo Go** pada perangkat handphone masing-masing. <https://expo.dev/client>

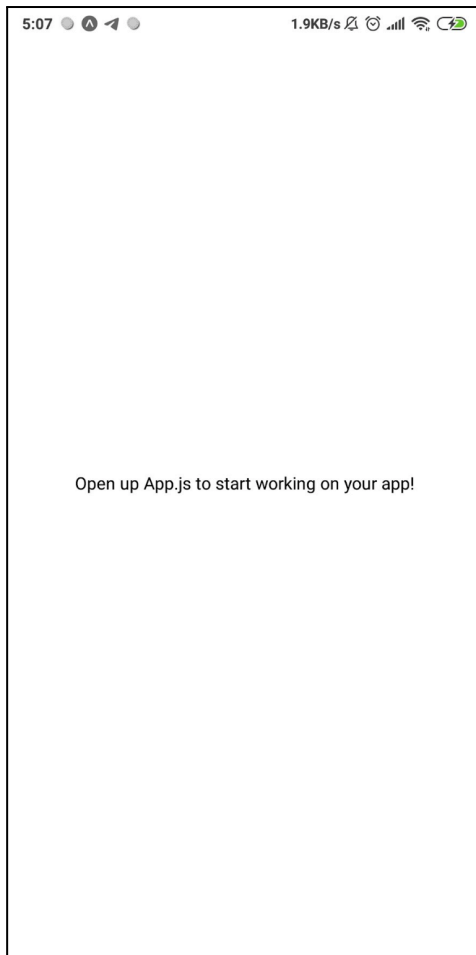
### 2. Membuat & Menjalankan Project React Native via Expo

- Buat sebuah project **React Native** via **Expo** bernama **my-app** dengan menjalankan command:
  - **npx create-expo-app my-app**
- Masuk ke folder project **my-app** yang sudah terbuat dengan menjalankan command:
  - **cd my-app**
- Jalankan project react native anda dengan command:
  - **npx expo start** atau
  - **npx expo start --tunnel**
- Akan muncul sebuah **QRCode** pada command prompt anda
- Buka aplikasi **Expo Go** di handphone, kemudian scan **QRCode** tersebut.



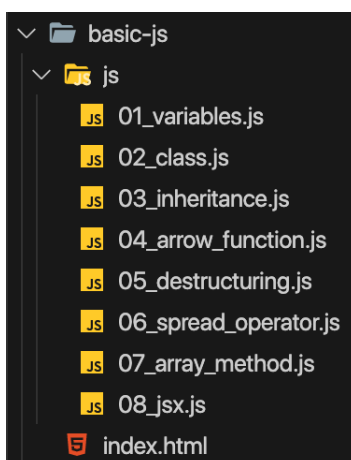
```
purnamaanaking ~/Desktop/myApp (master) $ expo start
This command is being executed with the global Expo CLI. Learn more: https://blog.expo.dev/the-new-expo-cli-f4258d8e3421
To use the local CLI instead (recommended in SDK 46 and higher), run:
> npx expo start
Starting project at /Users/purnamaanaking/Desktop/myApp
Starting Metro Bundler
[QR Code]
> Metro waiting on expo://192.168.8.180:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
> Press a | open Android
> Press i | open iOS simulator
> Press w | open web
> Press r | reload app
> Press m | toggle menu
> Press ? | show all commands
Logs for your project will appear below. Press Ctrl+C to exit.
Started Metro Bundler
```

- Akan muncul Screen Sederhana (pada Handphone anda) dari Project React Native yang telah anda buat.



### 3. Belajar Javascript Dasar

- Buat **folder baru** (di luar dari project react native yang baru saja dibuat pada bab sebelumnya) dengan nama **basic-js**.
- Buat file-file baru sesuai struktur direktori seperti gambar di bawah ini.



- Buka file **index.html** dan tulis kode HTML yang me-link-kan file-file external javascript seperti di bawah ini.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Javascript</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>Javascript</h2>

  <script src="js/01_variables.js"></script>
  <script src="js/02_class.js"></script>
  <script src="js/03_inheritance.js"></script>
  <script src="js/04_arrow_function.js"></script>
  <script src="js/05_destructuring.js"></script>
  <script src="js/06_spread_operator.js"></script>
  <script src="js/07_array_method.js"></script>
</body>
</html>

```

### 3.1 Variables

- Buka file **01\_variables.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Variables -----");

// Var
for (var i = 0; i < 5; i++) {
  console.log(i);
}
console.log(i);

// Var
function looping() {
  for (var j = 0; j < 5; j++) {
    console.log(j);
  }
}
looping();
console.log(j);

// Let
for (let k = 0; k < 5; k++) {
  console.log(k);
}
console.log(k);

// Let
for (let l = 0; l < 5; l++) {

```

```

    console.log(1);
}
if (true) {
    let l = 100;
    console.log("l =",l);
}

// Const
const phi = 3.14;
phi = 3.147;
console.log(phi);

```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.2 Class

- Buka file **02\_class.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Class -----");

class Car {
    constructor(name) {
        this.brand = name;
    }

    present() {
        return "I have a " + this.brand;
    }
}

let myCar = new Car("Ford");

console.log(myCar.present());

```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.3 Inheritance

- Buka file **03\_inheritance.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Inheritance -----");

class Gadget {
    constructor(name) {
        this.name = name;
    }
}

```

```

    present() {
        return "I have an " + this.name;
    }
}

class Handphone extends Gadget {
    constructor(name, mod) {
        super(name);
        this.model = mod;
    }

    show() {
        return this.present() + ", it is " + this.model;
    }
}

let handphone = new Handphone("iphone", "14 Pro Max");

console.log(handphone.show());

```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.4 Arrow Function

- Buka file **04\_arrow\_function.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Arrow Function -----");

// Normal Function
function hello() {
    return "Hello World!";
}

// Normal function
hello = function() {
    return "Hello World!";
}

console.log(hello());

// Arrow Function
hello = () => {
    return "Hello World! This is from Arrow Function";
}

// Arrow Function
hello = () => "Hello World! This is from Arrow Function";

console.log(hello());

// Arrow Function with 1 parameter
hello = (myName) => "Hello " + myName + "! This is from Arrow Function";

```

```
// Arrow Function with 1 parameter
hello = myName => `Hello ${myName}! This is from Arrow Function`;

console.log(hello("Purnama"));
```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.5 Destructuring

- Buka file **05\_destructuring.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```
console.log("----- Destructuring -----");

// Array
let items = ["Table", "Handphone", "Computer"];

// Manual variable declaration
let item1 = items[0];
let item2 = items[1];
let item3 = items[2];

console.log(item1);
console.log(item2);
console.log(item3);

// Variable declaration with destructuring
let [item_1, item_2, item_3] = items

console.log(item_1);
console.log(item_2);
console.log(item_3);

// Object
let student = {
  name: "Ahmad",
  age: 22,
  department: "Information System"
}

// Manual variable declaration
let studentName = student.name;
let studentAge = student.age;
let studentDepartment = student.department;

console.log(studentName);
console.log(studentAge);
console.log(studentDepartment);

// Variable declaration with destructuring
let {name, age, department} = student;
```



```

console.log(name);
console.log(age);
console.log(department);

// Destructuring by providing aliases
let {name: objectName, age: objectAge, department: objectDepartment} = student;

console.log(objectName);
console.log(objectAge);
console.log(objectDepartment);

// Destructuring of a function's return value
const studentData = () => {
    return { name: "Roni", age: 19, department: "Information System" }
}

let {
    name: studentNameData,
    age: studentAgeData,
    department: studentDepartmentData
} = studentData();

console.log(studentNameData);
console.log(studentAgeData);
console.log(studentDepartmentData);

```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.6 Spread Operator

- Buka file **06\_spread\_operator.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Spread Operator -----");

// Array
let arr1 = [10, 20, 30];
let arr2 = [100, 200, 300];

// Without Spread Operator
let arr3 = [arr1, arr2];
console.log(arr3);

// With Spread Operator
let arr4 = [...arr1, ...arr2];
console.log(arr4);

// With Spread Operator
let arr5 = [...arr1, 40, 50];
console.log(arr5);

// With Spread Operator
let arr6 = [30, 40, 50, ...arr1];

```

```

console.log(arr6);

// Object
let employee = {
  name: "Budi",
  age: 22,
  position: "Software Engineer"
};

// Without Spread Operator
let employee1 = {
  employee,
  age: 25,
  city: "Jakarta"
};
console.log(employee1);

// With Spread Operator
let employee2 = {
  ...employee,
  age: 20,
  city: "Surabaya"
};
console.log(employee2);

```

- **Inspect** halaman web anda, lihat hasil kode program pada Tab **Console**.

### 3.7 Array Method

- Buka file **07\_array\_method.js** pada sebuah folder **/js**.
- Kemudian pahami dan terapkan kode program di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

console.log("----- Array Method -----");

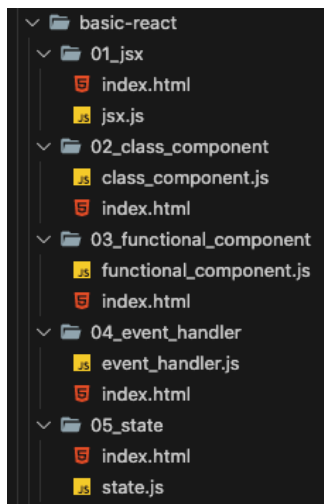
// forEach
let arr = [10, 20, 30, 40];
arr.forEach(function(val, key) {
  console.log(`Array value in index-${key} is ${val}`);
});

// Map
let arrSquare = arr.map((val) => val * val);
console.log(arrSquare);

```

## 4. Belajar React Dasar

- Buat **folder baru** (di luar dari project react native dan **basic-js** yang baru saja dibuat pada bab sebelumnya) dengan nama **basic-react**.
- Buat file-file baru sesuai struktur direktori seperti gambar di bawah ini.



## 4.1 JSX (Javascript XML)

- Buka file ***index.html*** pada folder ***/01\_jsx/***. Tulis kode struktur dasar dokumen HTML seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JSX (Javascript XML)</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>JSX (Javascript XML)</h2>

  <div id="root"></div>

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

  <script type="text/babel" src="jsx.js"></script>
</body>
</html>
```

- Buka file ***jsx.js*** pada folder ***/01\_jsx/***. Kemudian pahami dan terapkan kode program mengenai aturan-aturan dasar JSX di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```
// Hello
let myElement = "Hello World";
```

```
// Basic Rules
myElement = "<h3>Hello React</h3>"; // wrong
myElement = <h3>Hello React</h3>; // correct
```

```
// Basic Rules
myElement = ; // wrong
myElement = ; // correct
```

```
// Basic Rules
// myElement = (
//   <p>Learn Javascript</p>
//   <p>Learn React</p>
// ); // wrong
myElement = (
  <div>
    <p>Learn Javascript</p>
    <p>Learn React</p>
  </div>
); // correct
```

```
// Statement in JSX
let myText = "Learn React";
myElement = (
  <div>
    <h3>{myText}</h3>
  </div>
);
```

```
// Array Processing
let items = ["Table", "Handphone", "Computer"];
myElement = (
  <ul>
    {items.map((item, index) => <li>{item}</li>)}
  </ul>
);

const myApp = myElement;

ReactDOM.createRoot(document.getElementById('root')).render(myApp);
```

- Jalankan **index.html** via extension Live Server pada vsCode.

## 4.2 Class Component

- Buka file **index.html** pada folder **/02\_class\_component/**. Tulis kode struktur dasar dokumen HTML seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Class Component</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>Class Component</h2>

  <div id="root"></div>

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

  <script type="text/babel" src="class_component.js"></script>
</body>
</html>

```

- Buka file **class\_component.js** pada folder **/02\_class\_component/**. Kemudian pahami dan terapkan kode program mengenai aturan-aturan dasar sebuah **Class Component** di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

// Class Component
class Practice1 extends React.Component {
  render() {
    return <h3>Practice: ReactJS</h3>
  }
}
let myElement = <Practice1 />;

```

```

// Props in Class Component
class Practice2 extends React.Component {
  render() {
    return <h3>Practice: {this.props.topic}</h3>
  }
}
myElement = <Practice2 topic="ReactJS"/>;

```

```

// Props Children in Class Component
class Practice3 extends React.Component {
  render() {
    return <h3 id={this.props.module}>Practice: {this.props.children}</h3>
  }
}
myElement = <Practice3 module="01">ReactJS</Practice3>;

const myApp = myElement;

```

```
ReactDOM.createRoot(document.getElementById('root')).render(myApp);
```

- Jalankan **index.html** via extension Live Server pada vsCode.

### 4.3 Functional Component

- Buka file **index.html** pada sebuah folder **/03\_functional\_component/**. Tulis kode struktur dasar dokumen HTML seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Functional Component</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>Functional Component</h2>

  <div id="root"></div>

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

  <script type="text/babel" src="functional_component.js"></script>
</body>
</html>
```

- Buka file **functional\_component.js** pada folder **/03\_functional\_component/**. Kemudian pahami dan terapkan kode program mengenai aturan-aturan dasar sebuah **Functional Component** di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```
// Functional Component
const ReactTitle = () => {
  return <h3>Currently learning ReactJS</h3>
}
const JavascriptTitle = () => {
  return <h3>Currently learning Javascript</h3>
}
const MyComponent = () => {
  return (
    <div>
      <ReactTitle />
      <hr />
      <JavascriptTitle/>
    </div>
  );
};
```

```
}
let myElement = <MyComponent />
```

```
// Props in Functional Component
const Practice = (props) => <h3>Practice: {props.course}</h3>;
myElement = (
  <div>
    <Practice course="Javascript"/>
    <Practice course="ReactJS"/>
    <Practice course="React Native"/>
  </div>
);

const myApp = myElement;

ReactDOM.createRoot(document.getElementById('root')).render(myApp);
```

- Jalankan **index.html** via extension Live Server pada vsCode.

#### 4.4 Event Handler

- Buka file **index.html** pada folder **/04\_event\_handler/**. Tulis kode struktur dasar dokumen HTML seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Event Handler</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>Event Handler</h2>

  <div id="root"></div>

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

  <script type="text/babel" src="event_handler.js"></script>
</body>
</html>
```

- Buka file **event\_handler.js** pada folder **/04\_event\_handler/**. Kemudian pahami dan terapkan kode program mengenai aturan-aturan dasar sebuah **Event Handler** di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```
// Button Component
const MyButton = (props) => {
  const handleClick = () => {
    alert(props.message);
  }

  return (
    <button onClick={handleButtonClick} style={{ margin: "10px" }}>
      {props.children}
    </button>
  );
}
```

```
// App Component
const MyApp = () => {
  return (
    <div>
      <h1>Learn React</h1>
      <MyButton message="Learn React">React</MyButton>
      <MyButton message="Learn JavaScript">JavaScript</MyButton>
    </div>
  );
}

const myElement = <MyApp />;
const myApp = myElement;

ReactDOM.createRoot(document.getElementById('root')).render(myApp);
```

- Jalankan **index.html** via extension Live Server pada vsCode.

## 4.5 State

- Buka file **index.html** pada folder **/05\_state/**. Tulis kode struktur dasar dokumen HTML seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>State</title>
</head>
<body>
  <h1>Basic ReactJS Introduction</h1>
  <h2>State</h2>

  <div id="root"></div>

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

  <script type="text/babel" src="state.js"></script>
</body>
</html>

```

- Buka file **state.js** pada folder **/05\_state/**. Kemudian pahami dan terapkan kode program mengenai aturan-aturan dasar sebuah **State** di bawah ini. Berikan penjelasan pada laporan praktikum dari kode program yang telah anda tulis.

```

// Cat Component
class Cat extends React.Component {
  // State Declaration
  state = {
    isHungry: true
  };

  render() {
    return (
      <div>
        <p>I am {this.props.name}, and I am {this.state.isHungry ? "
hungry" : " full"}!</p>
        <button
          onClick={() => {
            this.setState({ isHungry: false });
          }}
          disabled={!this.state.isHungry}
        >
          {this.state.isHungry ? "Pour me some milk, please!" : "Thank
you!"}
        </button>
      </div>
    );
  }
};

```

```

// Cafe Component
class Cafe extends React.Component {
  render() {
    return (
      <div>
        <Cat name="Munkustrap" />
        <Cat name="Spot" />
      </div>
    );
  }
};

const myElement = <Cafe />;
const myApp = myElement;

ReactDOM.createRoot(document.getElementById('root')).render(myApp);

```

## 5. Tugas

1. Praktekkan seluruh poin praktikum yang ada di atas.
2. Terapkan pendekatan **class component** pada praktikum **Bab 4.4 Event Handler**.
3. Buat react component sederhana yang dapat menampilkan output seperti gambar di bawah ini. Lakukan dengan pendekatan **Functional Component** dan juga **Class Component**. Manfaatkan materi react yang telah dipelajari (props, state, event handler, dll).



4. Dokumentasikan hasil praktikum tersebut (**screenshot kode program, output pada browser, penjelasan kode program yang ditulis**) dalam bentuk Laporan Praktikum.
5. Letakkan source code hasil praktikum dan hasil tugas praktikum (no 2 & 3) ke **Github**. Sertakan info **link repository** source code nya pada Laporan Praktikum.
6. Kumpulkan Laporan Praktikum (**.pdf**) via E-Learning paling lambat sebelum jadwal praktikum minggu depan.