

# **TUGAS PERTEMUAN 7**

Disusun Guna Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Framework

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



Disusun Oleh :

Akhmad Ramadani (2041720002)

TI-3B

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**2023**

## 1. Latihan

### a. Event Handler di Class Component

#### i. Kode Program

slide3.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

      class MyApp extends React.Component {
        render() {
          return (
            <div>
              <h1>Belajar React</h1>
              <button onClick={() =>
alert("Hello React")}>Click me</button>
            </div>
          )
        }
      }

ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

  </script>
</body>

</html>
```

#### ii. Hasil

# Belajar React

Click me

127.0.0.1:5500 says

Hello React

OK

## iii. Penjelasan

1. Sama di JavaScript biasa, event handler di React juga bisa ditulis dalam bentuk inline (langsung ke dalam atribut HTML).
2. Terdapat sebuah kelas yang bernama MyApp yang merupakan komponen React. Pada method render(), komponen MyApp akan mengembalikan sebuah div yang berisi elemen h1 dan button. Pada button tersebut, terdapat event onClick yang apabila button tersebut diklik maka akan menampilkan pesan alert dengan tulisan "Hello React".
3. ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />); yang digunakan untuk me-render komponen MyApp dan menempatkannya pada div dengan id "root" pada halaman website.

## b. Event Handler di Class Component

### i. Kode Program

slide5.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class MyApp extends React.Component {

      handleButtonClick() {
        alert("Hello React")
      }

      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <button
onClick={this.handleButtonClick}>Click me</button>
          </div>
        )
      }
    }

    ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

  </script>
</body>

</html>
```

### ii. Hasil

# Belajar React

Click me

127.0.0.1:5500 says

Hello React

OK

## iii. Penjelasan

1. Selain menulis langsung secara inline, kode untuk event handler juga bisa dipisah menjadi sebuah method di dalam class component
2. Pada kode ini, terdapat sebuah method baru yaitu `handleButtonClick()` yang akan dijalankan saat button diklik. Method tersebut akan menampilkan pesan alert yang sama dengan sebelumnya, yaitu "Hello React".
3. Pada elemen button, event `onClick` dipanggil untuk menjalankan method `handleButtonClick()` yang telah dibuat sebelumnya. Ini berarti saat button tersebut diklik, method `handleButtonClick()` akan dijalankan dan menampilkan pesan alert.

## c. Event Handler di Class Component

### i. Kode Program

slide8.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      handleButtonClick() {
        alert("Hello React")
      }

      render() {
        return <button
onClick={this.handleButtonClick}>Click me</button>
      }
    }

    class MyApp extends React.Component {

      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol />
          </div>
        )
      }
    }

    ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

  </script>
</body>

</html>
```

ii. Hasil

# Belajar React

Click me

127.0.0.1:5500 says

Hello React

OK

iii. Penjelasan

1. Kelas Tombol digunakan untuk membuat komponen button dengan method `handleButtonClick()` yang sama seperti sebelumnya. Pada method `render()`, kelas Tombol hanya mengembalikan elemen button dengan event `onClick` yang dipanggil untuk menjalankan method `handleButtonClick()`.
2. Kelas MyApp digunakan untuk membuat komponen yang berisi elemen `h1` dan Tombol. Pada method `render()`, kelas MyApp akan mengembalikan sebuah `div` yang berisi elemen `h1` dan Tombol yang diambil dari kelas Tombol.

## d. Event Binding

### i. Kode Program

slide9.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      handleButtonClick() {
        alert(this.props.pesan); // ini akan
error
      }

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

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>
            <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
          </div>
        )
      }
    }

  }
```



```
ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

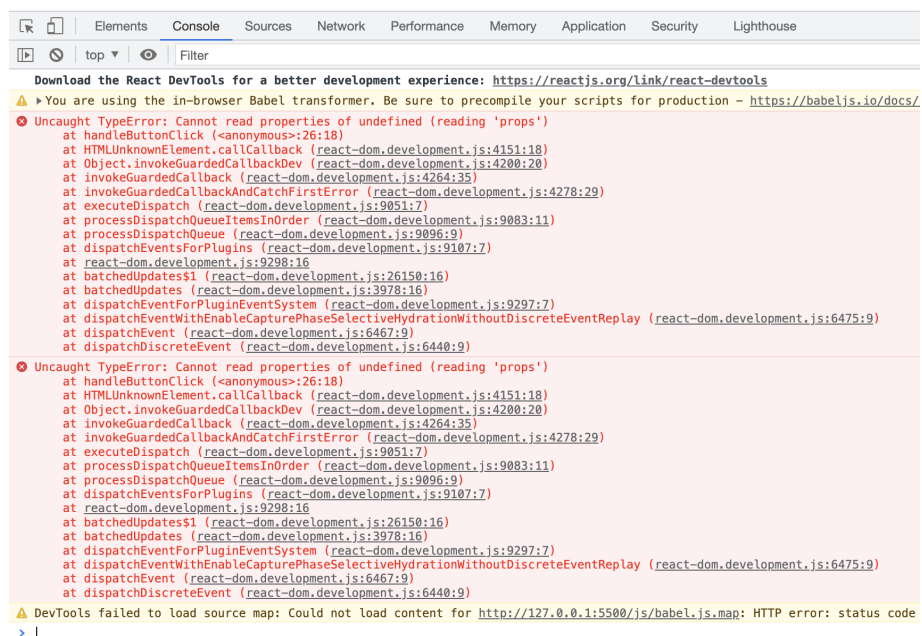
</script>
</body>

</html>
```

## ii. Hasil

### Belajar React

React JavaScript



## iii. Penjelasan

1. Kelas Tombol digunakan untuk membuat komponen button dengan method handleClick() yang akan menampilkan pesan alert dengan nilai dari prop "pesan" saat button diklik. Pada method render(), kelas Tombol mengembalikan elemen button dengan event onClick yang dipanggil untuk menjalankan method handleClick(). Selain itu, Tombol juga menerima children sebagai prop yang berisi teks yang akan ditampilkan pada button.
2. Kelas MyApp digunakan untuk membuat komponen yang berisi elemen h1 dan 2 Tombol dengan prop "pesan" yang berbeda-beda. Tombol pertama akan menampilkan pesan "Belajar React" saat diklik, sedangkan Tombol kedua akan menampilkan pesan "Belajar JavaScript" saat diklik.
3. Namun, terdapat kesalahan pada kode yaitu saat method handleClick() dipanggil, nilai dari this.props.pesan tidak

bisa diakses. Hal ini disebabkan karena `this` pada method `handleButtonClick()` mengacu pada konteks Tombol bukan konteks `MyApp`. Oleh karena itu, kita perlu mem-bind method `handleButtonClick()` dengan konteks Tombol agar dapat mengakses nilai dari `this.props.pesan`.

## e. Event Binding

### i. Kode Program

slide16.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      handleButtonClick() {
        alert(this.props.pesan);
      }

      render() {
        // kurang disarankan karena fungsi ini
akan terus dibuat setiap kali proses render
        return (
          <button
onClick={this.handleButtonClick.bind(this)} style={{
margin: "10px" }}>
            {this.props.children}
          </button>
        )
      }
    }

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>
            <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
          </div>
        )
      }
    }
```

```
    }

    ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

  </script>
</body>

</html>
```

## ii. Hasil

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar JavaScript

OK

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar React

OK

## iii. Penjelasan

1. Kode di atas menambahkan method `handleButtonClick` pada komponen Tombol yang akan menampilkan pesan yang diteruskan melalui prop pesan ketika tombol diklik. Kemudian, method `handleButtonClick` ditambahkan sebagai event handler untuk `onClick` event di dalam render method.
2. Namun, perlu diperhatikan bahwa ketika kita menambahkan event handler seperti itu, konteks `this` dalam method `handleButtonClick` akan menjadi `undefined`. Oleh karena itu, kita perlu mengikat `this` ke instance Tombol di dalam method render atau dengan menggunakan arrow function seperti di atas
3. Namun, cara ini tidak disarankan karena setiap kali proses render, fungsi baru akan dibuat, yang dapat mempengaruhi performa aplikasi ketika komponen menjadi lebih besar. Sebaliknya, kita dapat menggunakan `Function.prototype.bind()` untuk mengikat `this` ke instance Tombol dalam method `handleButtonClick`. Hal ini juga membuat kode menjadi lebih mudah dibaca.

## f. Event Binding

### i. Kode Program

slide17.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      handleButtonClick() {
        alert(this.props.pesan);
      }

      render() {
        // kurang disarankan karena fungsi ini
akan terus dibuat setiap kali proses render
        return (
          <button onClick={() =>
this.handleButtonClick()} style={{ margin: "10px" }}>
            {this.props.children}
          </button>
        )
      }
    }

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>
            <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
          </div>
        )
      }
    }

  }
```

```
ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);
```

```
</script>  
</body>
```

```
</html>
```

## ii. Hasil

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar JavaScript

OK

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar React

OK

## iii. Penjelasan

1. Mengubah pada slide ke-16 cara bindingnya menggunakan function arrow.
2. Namun, cara ini tidak disarankan karena setiap kali proses render, fungsi baru akan dibuat, yang dapat mempengaruhi performa aplikasi ketika komponen menjadi lebih besar. Sebaliknya, kita dapat menggunakan `Function.prototype.bind()` untuk mengikat `this` ke instance Tombol dalam method `handleButtonClick`. Hal ini juga membuat kode menjadi lebih mudah dibaca.

## g. Event Binding

### i. Kode Program

slide19.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../../js/react.development.js"></script>
  <script
src="../../js/react-dom.development.js"></script>
  <script src="../../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      constructor() {
        super(); // tidak perlu props karena
tidak pakai props disini. Yapi jika ingin ditulis
silahkan
        this.handleClick =
this.handleClick.bind(this);
      }

      handleClick() {
        alert(this.props.pesan);
      }

      render() {
        return (
          <button
onClick={this.handleClick} style={{ margin: "10px"
}}>
            {this.props.children}
          </button>
        )
      }
    }

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>

```

```

        <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
      </div>
    )
  }
}

ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

</script>
</body>

</html>

```

## ii. Hasil

### Belajar React

React JavaScript



### Belajar React

React JavaScript



## iii. Penjelasan

1. Kode di atas memperbaiki kesalahan pada kode sebelumnya dengan mengikat this ke metode handleClick() menggunakan konstruktor di dalam komponen Tombol. Dalam konstruktor, this.handleClick diikat ke this sehingga metode tersebut dapat digunakan dengan benar sebagai fungsi pemanggilan kembali dalam penanganan klik tombol. Kode ini lebih disukai daripada menggunakan fungsi arrow atau bind() dalam render karena konstruktor hanya dipanggil sekali saat komponen pertama kali dibuat, sementara fungsi arrow atau bind() akan dipanggil setiap kali komponen di-render ulang, yang dapat mempengaruhi kinerja aplikasi secara keseluruhan.



## h. Event binding

### i. Kode Program

slide21.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      // Class Properties (Stage 3 Proposal), buat
      menggunakan arrow function
      handleClick = () => {
        alert(this.props.pesan);
      }

      render() {
        return (
          <button
onClick={this.handleClick} style={{ margin: "10px"
}}>
            {this.props.children}
          </button>
        )
      }
    }

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>
            <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
          </div>
        )
      }
    }
```

```

    }

ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

</script>
</body>

</html>

```

## ii. Hasil

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar JavaScript

OK

### Belajar React

React JavaScript

127.0.0.1:5500 says  
Belajar React

OK

## iii. Penjelasan

1. Kode tersebut menunjukkan penggunaan class properties (proposal stage 3) untuk membuat method `handleButtonClick` sebagai arrow function di dalam class `Tombol`. Dengan menggunakan arrow function, maka `this` dalam method tersebut akan merujuk ke instance dari class `Tombol` yang dibuat saat render, sehingga tidak perlu lagi mengikat `this` seperti pada contoh sebelumnya.
2. Dalam kode tersebut, saat button di-click, maka method `handleButtonClick` akan dipanggil dan alert box akan muncul menampilkan pesan yang diambil dari prop pesan yang dilewatkan dari parent component.

## i. Event Object

### i. Kode Program

slide24.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

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

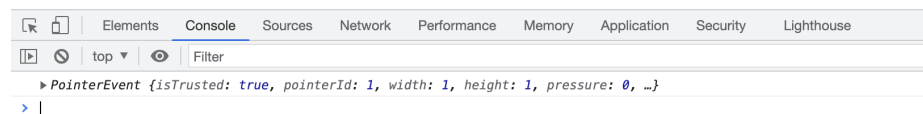
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <h1 id="judul">Belajar JavaScript</h1>
  <script>
    var h1Node = document.getElementById("judul");
    h1Node.addEventListener("click", function
(event) {
      console.log(event);
    });
  </script>
</body>

</html>
```

### ii. Hasil

## Belajar JavaScript



### iii. Penjelasan

1. Script di atas menambahkan event listener "click" ke elemen h1 dengan id "judul". Ketika elemen tersebut diklik, akan di-log informasi tentang event tersebut ke konsol. Informasi tersebut mencakup properti event seperti type, target, timeStamp, dan lain-lain.

## j. Event Object

### i. Kode Program

slide29.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      handleButtonClick = (event) => {
        console.log(event); // event object
        console.log(event.target); // node
object tempat event terjadi
        console.log(event.target.innerHTML); //
ambil isi HTML dari node object. Untuk form, biasanya
ambil nialai value
      }

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

    class MyApp extends React.Component {
      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol pesan="Belajar
React">React</Tombol>
            <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
          </div>
        )
      }
    }
  </script>
</body>
</html>
```

```

    )
  }

}

ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

</script>
</body>

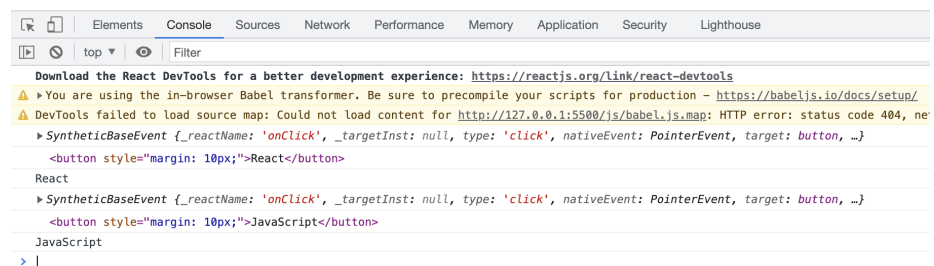
</html>

```

## ii. Hasil

### Belajar React

React JavaScript



## iii. Penjelasan

1. Script di atas adalah contoh kode React yang menampilkan dua tombol, masing-masing dengan pesan yang berbeda, di dalam komponen MyApp. Setiap kali tombol diklik, fungsi `handleButtonClick` akan dipanggil, yang kemudian akan mencetak beberapa nilai dari objek event yang dihasilkan, termasuk node dimana event tersebut terjadi dan nilai HTML dari node tersebut.
2. Pada baris ke-6, method `handleButtonClick` di-bind ke instance kelas Tombol dengan menggunakan arrow function yang menghindari penggunaan `this` dan memastikan bahwa method tersebut selalu terikat dengan instance kelas Tombol. Kemudian, method tersebut dipanggil di dalam event `onClick` pada baris ke-15.
3. Kemudian, pada baris ke-23, kelas `MyApp` di-render ke dalam elemen dengan `id="root"`. Komponen `MyApp` hanya berisi judul dan dua instance komponen Tombol. Setiap instance Tombol memiliki properti pesan yang berisi teks yang akan ditampilkan pada saat diklik.

## k. Menurunkan (Passing Down) Event Handler

### i. Kode Program

slide35.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    class Tombol extends React.Component {
      render() {
        return (
          <button
onClick={this.props.onTombolClick}>Click Me</button>
        )
      }
    }

    class MyApp extends React.Component {
      handleClick = () => {
        alert("Alert dari MyApp");
      }

      render() {
        return (
          <div>
            <h1>Belajar React</h1>
            <Tombol
onTombolClick={this.handleClick} />
          </div>
        )
      }
    }

ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);

  </script>
</body>
```

```
</html>
```

## ii. Hasil

### Belajar React

Click Me

127.0.0.1:5500 says  
Alert dari MyApp

OK

## iii. Penjelasan

1. Kode di atas merupakan contoh penggunaan props untuk melewati sebuah fungsi (`handleClick`) dari komponen induk (`MyApp`) ke komponen anak (`Tombol`). Properti `onTombolClick` pada komponen `Tombol` akan menerima fungsi `handleClick` sebagai nilainya. Kemudian, pada saat tombol di dalam komponen `Tombol` diklik, fungsi yang diterima dari properti `onTombolClick` akan dieksekusi.
2. Dalam contoh ini, saat tombol di dalam komponen `Tombol` diklik, fungsi `handleClick` pada komponen `MyApp` akan dieksekusi dan memunculkan sebuah alert dengan pesan "Alert dari MyApp".

## I. Event Handler di Functional Component

### i. Kode Program

slide41.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B   *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

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

    const MyApp = () => {
      return (
        <div>
          <h1>Belajar React</h1>
          <Tombol pesan="Belajar
React">React</Tombol>
          <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
        </div>
      )
    }

ReactDOM.createRoot(document.getElementById('root')).ren
der(<MyApp />);

  </script>
</body>

</html>
```



ii. Hasil

**Belajar React**

React JavaScript

127.0.0.1:5500 says  
Belajar JavaScript

OK

**Belajar React**

React JavaScript

127.0.0.1:5500 says  
Belajar React

OK

iii. Penjelasan

1. Kode di atas adalah contoh pembuatan event dan event handler di functional component React:
2. Di dalam function component, event handler ditulis sebagai fungsi biasa. Dalam contoh di atas, fungsi handleClick() di baris 20-22 bertindak sebagai event handler untuk komponen Tombol.

## 2. Latihan

### a. Membuat Class Component

#### i. Kode Program

latihan1.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    const Tombol = (props) => {
      const handleButtonClick = (event) => {
        console.log(event); // event object
        console.log(event.target); // node
object tempat event terjadi
        console.log(event.target.innerHTML); //
ambil isi HTML dari node object. Untuk form, biasanya
ambil nialai value
      }

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

    const MyApp = () => {
      return (
        <div>
          <h1>Belajar React</h1>
          <Tombol pesan="Belajar
React">React</Tombol>
          <Tombol pesan="Belajar
JavaScript">JavaScript</Tombol>
        </div>
      )
    }
  </script>
</body>
</html>
```

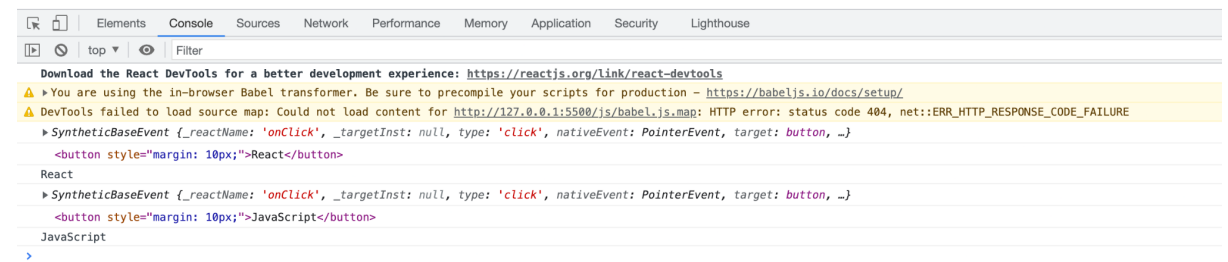
```
ReactDOM.createRoot(document.getElementById('root')).render(<MyApp />);
```

```
    </script>  
</body>  
  
</html>
```

## ii. Hasil

### Belajar React

React JavaScript



## b. Membuat Class Component

### i. Kode Program

latihan2.html

```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM      : 2041720002 *** -->
<!-- *** Kelas   : TI-3B *** -->

<!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>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="../js/react.development.js"></script>
  <script
src="../js/react-dom.development.js"></script>
  <script src="../js/babel.js"></script>
  <script type="text/babel">

    const Tombol = (props) => {
      return (
        <button
onClick={props.onTombolClick}>Click Me</button>
      )
    }

    const MyApp = () => {
      const handleClick = () => {
        alert("Alert dari MyApp");
      }

      return (
        <div>
          <h1>Belajar React</h1>
          <Tombol onClick={handleClick}
/>
        </div>
      )
    }

    ReactDOM.createRoot(document.getElementById('root')).ren
der(<MyApp />);

  </script>
</body>

</html>
```

ii. Hasil

## Belajar React

Click Me

127.0.0.1:5500 says  
Alert dari MyApp

OK