

TUGAS PERTEMUAN 4

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. Mengakses File Babel

i. Kode Program

akses_file_babel.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>

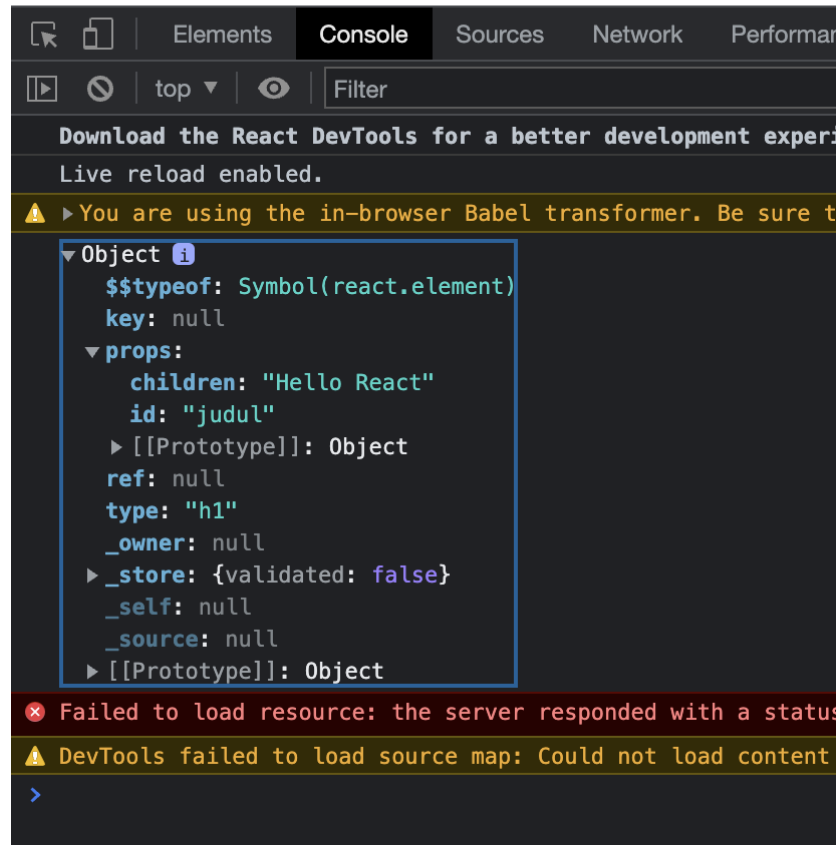
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const myElement = <h1 id='judul'>Hello
React</h1>;
    console.log(myElement);

ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil

Hello React



iii. Penjelasan

1. babel.js: transpiler yang mengubah sintaks JSX (JavaScript XML) menjadi JavaScript yang dapat dijalankan di browser
2. Pada bagian `<script type="text/babel">`, terdapat kode yang digunakan untuk membuat elemen HTML dengan menggunakan ReactJS. Kode ini membuat elemen H1 dengan teks "Hello React" dan ID "judul". Hasilnya disimpan dalam variabel myElement.
3. Selanjutnya, dengan menggunakan library ReactDOM, elemen myElement akan di-render ke dalam elemen HTML dengan ID "root" di dalam dokumen HTML. Hasilnya akan ditampilkan di browser.

b. Mengakses File Babel

i. Kode Program

akses_file_babel2.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>

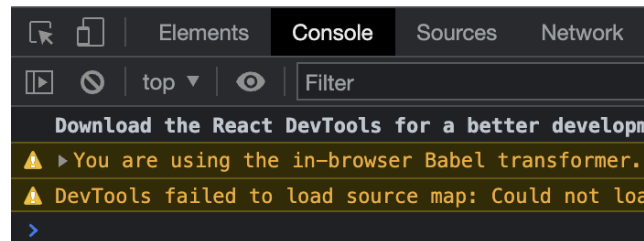
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const myElement = (
      <ul>
        <li><a
href='https://id.wikipedia.org/wiki/Espresoa'>Espresso</
a></li>
        <li><a
href='https://id.wikipedia.org/wiki/Kapucinoa'>Cappucino
</a></li>
        <li><a
href='https://id.wikipedia.org/wiki/Mokaa'>Moccacino</a>
</li>
      </ul>
    );

ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil

- [Espresso](#)
- [Cappucino](#)
- [Moccacino](#)



iii. Penjelasan

1. Pada bagian `<script type="text/babel">`, terdapat kode yang digunakan untuk membuat elemen HTML dengan menggunakan ReactJS. Kode ini membuat elemen unordered list (ul) dengan beberapa elemen list item (li) yang masing-masing memiliki satu link dengan teks yang berbeda. Hasilnya disimpan dalam variabel `myElement`.
2. Selanjutnya, dengan menggunakan library ReactDOM, elemen `myElement` akan di-render ke dalam elemen HTML dengan ID "root" di dalam dokumen HTML. Hasilnya akan ditampilkan di browser. Jadi, pada akhirnya, tampilan yang akan muncul di browser adalah daftar minuman kopi dengan link yang dapat diakses untuk membuka informasi tentang masing-masing minuman tersebut.

c. Aturan Dasar Penulisan JSX

i. Kode Program

aturan_penulisan_jsx0.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const myElement = (
      <p>Belajar JavaScript</p>
      <p>Belajar React</p>
    );
    // const myElement = (
    //   <div>
    //     <p>Belajar JavaScript</p>
    //     <p>Belajar React</p>
    //   </div>
    // );

ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. Hasil

```
<script type="text/babel">
  const myElement = (
    <p>Belajar JavaScript</p>
    <p>Belajar React</p>
  );
  // const myElement = (
```

JSX expressions must have one parent element.

iii. Penjelasan

1. Blok JSX hanya bisa memiliki 1 tag terluar (1 root element atau 1 top element). Jika ada 2 tag yang saling berdampingan, hasilnya akan error:

d. Aturan Dasar Penulisan JSX

i. Kode Program

aturan_penulisan_jsx.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    // const myElement = (
    //   <p>Belajar JavaScript</p>
    //   <p>Belajar React</p>
    // );
    const myElement = (
      <div>
        <p>Belajar JavaScript</p>
        <p>Belajar React</p>
      </div>
    );

ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. Hasil


```
<html lang="en">
  ><head> ... </head>
  ▼<body cz-shortcut-listen="true">
    ▼<div id="root">
      ▼<div>
        ... <p>Belajar JavaScript</p> == $0
          <p>Belajar React</p>
        </div>
      </div>
    </div>
```

iii. Penjelasan

1. Kode di atas adalah solusi dari masalah di atas dengan menambah satu tag <div> sebagai root

e. Aturan Dasar Penulisan JSX

i. Kode Program

aturan_penulisan_jsx2.html

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

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

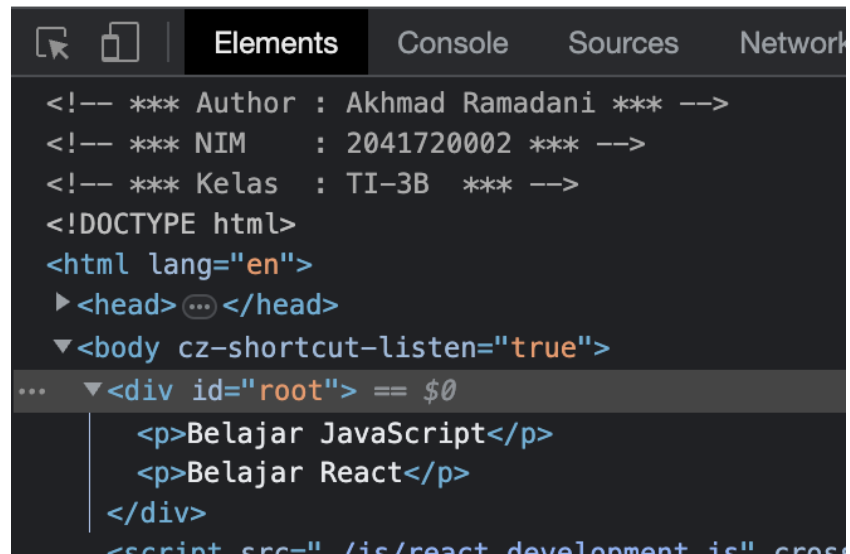
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>React Uncover</title>
</head>

<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const myElement = (
      <React.Fragment>
        <p>Belajar JavaScript</p>
        <p>Belajar React</p>
      </React.Fragment>
    );

    ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. Hasil



```
<!-- *** Author : Akhmad Ramadani *** -->
<!-- *** NIM : 2041720002 *** -->
<!-- *** Kelas : TI-3B *** -->
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body cz-shortcut-listen="true">
    ... <div id="root"> == $0
      <p>Belajar JavaScript</p>
      <p>Belajar React</p>
    </div>
    <script src="/is/react-development-is" croc
```

iii. Penjelasan

1. Kode di atas, digunakan untuk menghilangkan div yang tidak penting pada kode sebelumnya. Sehingga, terlihat perbedaannya pada hasil yaitu hilangnya div parent pada element p.

f. Aturan Dasar Penulisan JSX

i. Kode Program

aturan_penulisan_jsx3.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    // const myElement = <h1 class="judul">Belajar
React</h1>

    // const myElement = <h1
className="judul">Belajar React</h1>

    const myElement = (
      <div>
        <label for="username">Username: </label>
        <input type="text" id="username"
name="username" />
      </div>
    );

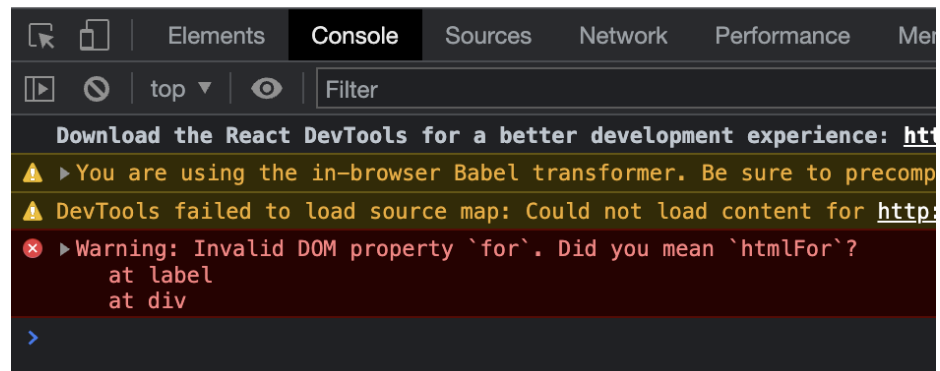
    // const myElement = (
    //   <div>
    //     <label htmlFor="username">Username:
</label>
    //     <input type="text" id="username"
name="username" />
    //   </div>
    // );

ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. Hasil

Username:



iii. Penjelasan

1. Pada bagian yang di-comment out, terdapat dua contoh kode yang menghasilkan elemen h1 dengan class "judul". Perlu diperhatikan bahwa pada ReactJS, class harus ditulis sebagai className. Oleh karena itu, contoh pertama harus diganti menjadi contoh kedua agar sesuai dengan sintaks ReactJS.
2. Pada contoh ketiga, kode yang ditulis menghasilkan sebuah div yang berisi label dan input untuk memasukkan username. Namun, perlu diperhatikan bahwa pada ReactJS, properti for pada label harus ditulis sebagai htmlFor, sehingga label terkait dengan input-nya. Oleh karena itu, perlu mengubah tag label for menjadi label htmlFor.

g. Statement di Dalam JSX

i. Kode Program

statement_dalam_jsx.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let myText = "Belajar React";

    const myElement = (
      <div>
        <h1>{myText}</h1>
        <p>10 + 5 = {10 + 5}</p>
        <span title={myText}>React
Uncover</span>
      </div>
    );

    ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil

Belajar React

$10 + 5 = 15$

React Uncover

iii. Penjelasan

1. Pada elemen `h1`, diberikan properti `{myText}`. Properti ini akan menampilkan isi variabel `myText` di dalam elemen `h1` tersebut.
2. Pada elemen `p`, diberikan ekspresi `{10 + 5}`. Ekspresi ini akan dihitung dan menampilkan hasilnya (15) di dalam elemen `p` tersebut.
3. Pada elemen `span`, diberikan properti `title={myText}`. Properti ini akan menampilkan isi variabel `myText`.

h. Penulisan Atribut style

i. Kode Program

styling.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const myElement = <h1 style={{color: 'violet',
backgroundColor: 'black'}}>Belajar React</h1>;

ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



iii. Penjelasan

1. Membuat elemen h1 dengan properti style yang ditulis dalam bentuk objek. Style tersebut akan mengubah warna teks menjadi violet dan latar belakang menjadi hitam.

i. Percabangan Kondisi

i. Kode Program

percabangan.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let nilai = 80;
    let pesan;
    if(nilai > 70){
      pesan = "Selamat, anda lolos";
    }else{
      pesan = "Maaf, silakan coba lagi";
    }

    const myElement = <h1>{pesan}</h1>;

    ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



Selamat, anda lolos

iii. Penjelasan

1. Membuat sebuah elemen h1 yang berisi pesan yang bergantung pada nilai dari variabel "nilai". Jika nilai lebih besar dari 70, maka variabel "pesan" akan diisi dengan "Selamat, anda lolos". Namun, jika nilai kurang dari atau sama dengan 70, maka variabel "pesan" akan diisi dengan "Maaf, silakan coba lagi".
2. Kemudian, elemen h1 tersebut akan di-render ke dalam elemen HTML dengan ID "root" di dalam dokumen HTML menggunakan library ReactDOM. Hasilnya akan ditampilkan di browser.
3. Jadi, pada akhirnya, tampilan yang akan muncul di browser akan berisi elemen h1 dengan teks "Selamat, anda lolos" atau "Maaf, silakan coba lagi" tergantung pada nilai variabel "nilai".

j. Percabangan Kondisi

i. Kode Program

percabangan2.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let nilai = 80;

    const myElement = <h1>{(nilai > 70) ? 'Selamat,
anda lolos' : 'Maaf, silakan coba lagi'}</h1>;

ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



Selamat, anda lolos

iii. Penjelasan

1. Sama dengan penjelasan di atas, namun kali ini menggunakan ternary operation.

k. Percabangan Kondisi

i. Kode Program

percabangan3.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let nilai = 60;

    const myElement = (nilai > 70) && <h1>Selamat,
anda lolos</h1>;

ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



Selamat, anda lolos

iii. Penjelasan

1. Dalam kasus tertentu, kita hanya ingin menampilkan sebuah element jika suatu kondisi terpenuhi, selain itu jangan

tampilkan apapun. Untuk logika ini, bisa juga dibuat menggunakan short circuit " && "

I. Percabangan Kondisi

i. Kode Program

kondisi.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>

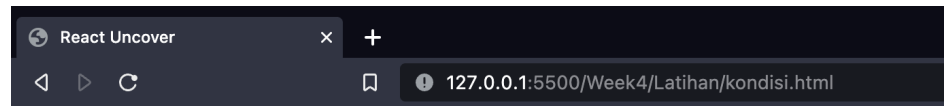
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">

      const myElement = (
        <div>
          <div />
          <div></div>
          <div>{false}</div>
          <div>{null}</div>
          <div>{undefined}</div>
          <div>{true}</div>
        </div>
      );

  ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. Hasil



iii. Penjelasan

1. JSX tidak akan menampilkan nilai boolean true, false, null dan undefined ke dalam web browser

m. Array Processing

i. Kode Program

array.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"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let kopis = ['Espresso', 'Cappuccino',
'Moccacino'];

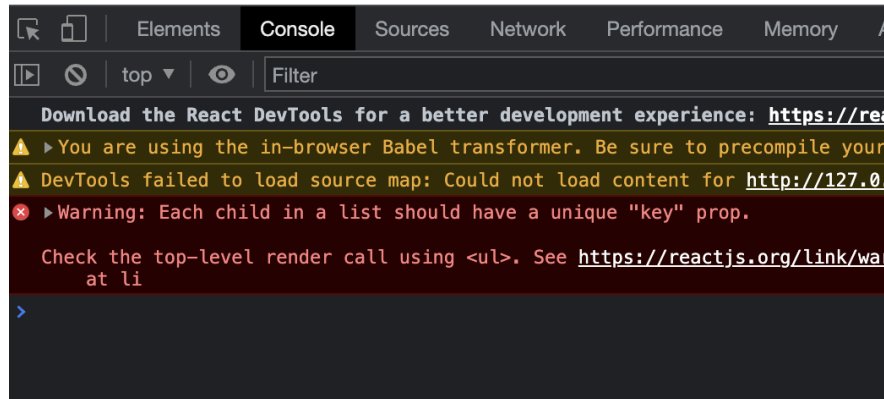
    const myElement = (
      <ul>
        {kopis.map(kopi => <li>{kopi}</li>)}
      </ul>
    );

    ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil

- Espresso
- Cappuccino
- Moccacino



iii. Penjelasan

1. Membuat sebuah elemen unordered list (ul) yang berisi item-item kopi. Item-item tersebut diambil dari variabel "kopis" yang berupa sebuah array.
2. Untuk menampilkan item-item tersebut dalam elemen ul, digunakan fungsi map pada array "kopis". Fungsi map akan menghasilkan array baru yang berisi elemen li yang memiliki teks berupa nama kopi. Kemudian, elemen ul dan elemen li-nya akan di-render ke dalam elemen HTML dengan ID "root" di dalam dokumen HTML menggunakan library ReactDOM. Hasilnya akan ditampilkan di browser.
3. Jadi, pada akhirnya, tampilan yang akan muncul di browser adalah sebuah elemen unordered list (ul) yang berisi item-item kopi, di mana setiap item diwakili oleh sebuah elemen list (li) dengan teks nama kopi.

n. Array Processing

i. Kode Program

array2.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>

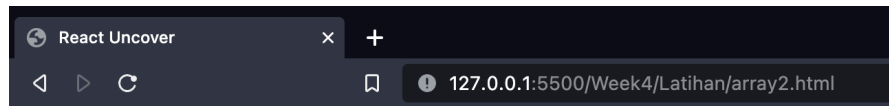
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let kopis = ['Espresso', 'Cappuccino',
'Moccacino'];

    const myElement = (
      <ul>
        {kopis.map((kopi, i) => <li
key={i}>{kopi}</li>)}
      </ul>
    );

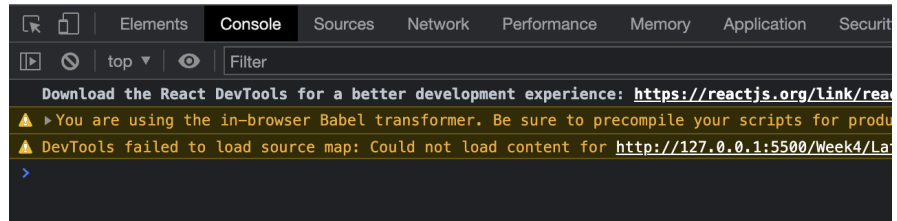
    ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



- Espresso
- Cappuccino
- Moccacino



iii. Penjelasan

1. Sama dengan penjelasan di atas. Namun, kali ini diberikan nilai key pada setiap item list untuk menghindari warning dan tentunya untuk memberi pembeda pada setiap list.

o. Array Processing

i. Kode Program

array3.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>

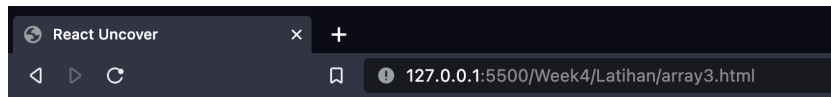
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    let kopis = ['Espresso', 'Cappuccino',
'Moccacino'];

    const myElement = (
      <ul>
        {kopis.map((kopi) => <li
key={kopi}>{kopi}</li>)}
      </ul>
    );

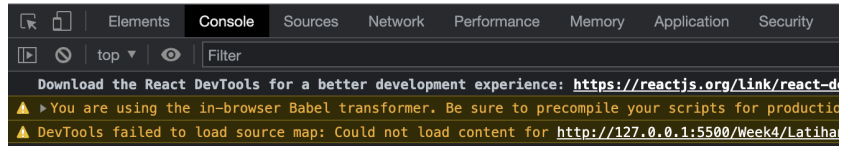
    ReactDOM.createRoot(document.getElementById('root')).ren
der(myElement);
  </script>
</body>

</html>
```

ii. Hasil



- Espresso
- Cappuccino
- Moccacino



iii. Penjelasan

1. Key juga bisa diambil dari nama value itu sendiri, asalkan tiap value memiliki nilai yang berbeda.

p. **JavaScript Event**

i. **Kode Program**

event.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>

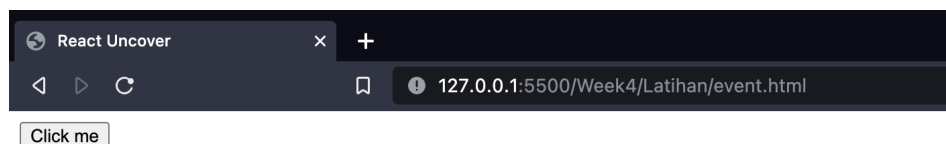
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">

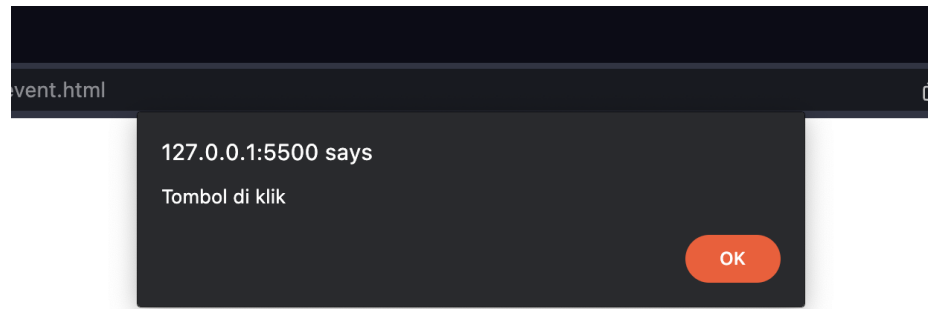
    const myElement = (
      <button onClick={
        () => {
          alert('Tombol di klik');
        }
      }>
        Click me </button>
    );

ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. **Hasil**





iii. Penjelasan

1. Pada kode JSX, prop onClick diisi dengan sebuah arrow function. Ketika tombol di-klik, maka arrow function tersebut akan dijalankan, sehingga akan muncul sebuah alert dengan pesan "Tombol di klik".

q. **JavaScript Event**

i. **Kode Program**

event2.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>

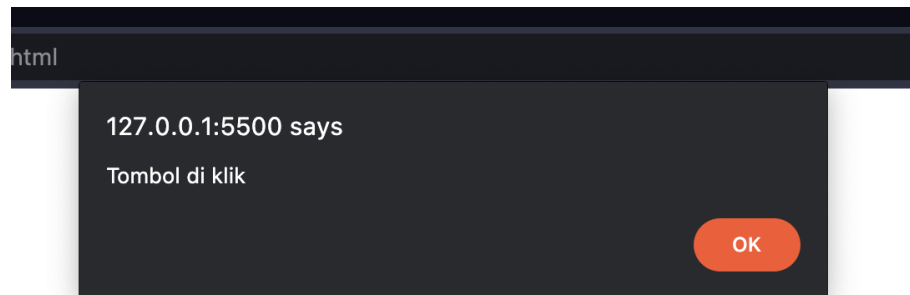
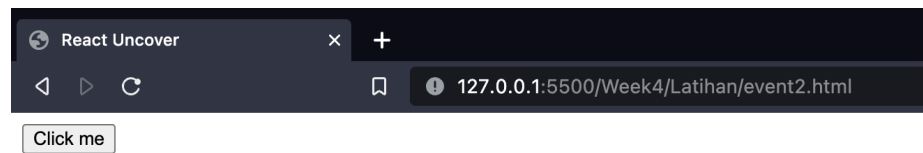
<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const onClickHandler = () => {
      alert('Tombol di klik');
    };

    const myElement = (
      <button onClick={
        onClickHandler
      }>
        Click me </button>
    );

    ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>

</html>
```

ii. **Hasil**



iii. Penjelasan

1. Sama dengan penjelasan di atas. Namun, kali ini arrow function ditaruh pada sebuah fungsi onClickHandler. Kemudian, tag button memiliki onClick yang memanggil fungsi onClickHandler

2. Soal

a. 1

i. Kode Program

soal.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>
  <style>
    table {
      border: 1px solid black;
    }

    tr {
      border: 1px solid black;
      padding: 5px;
    }

    td {
      border: 1px solid black;
      padding: 5px;
    }

    th {
      border: 1px solid black;
      padding: 5px;
    }
  </style>
</head>

<body>
  <div id="root"></div>
  <script src="./js/react.development.js"
crossorigin></script>
  <script src="./js/react-dom.development.js"
crossorigin></script>
  <script src="./js/babel.js"></script>
  <script type="text/babel">
    const mahasiswa = [
      {
        nama: "Eka",
        umur: 19,
        jurusan: "Teknik Informatika"
      },
      {
        nama: "Lisa",
        umur: 18,
```

```

        jurusan: "Sistem Informasi"
      },
      {
        nama: "Rudi",
        umur: 19,
        jurusan: "Teknik Elektro"
      }
    ];

    const myElement = (
      <div>
        <h1>Data Mahasiswa</h1>
        <table>
          <thead >
            <tr>
              <th>Nama</th>
              <th>Umur</th>
              <th>Jurusan</th>
            </tr>
          </thead>
          <tbody>
            {
              mahasiswas.map((mahasiswa,
i) => {
                return (
                  <tr key={i}>

<td>{mahasiswa.nama}</td>

<td>{mahasiswa.umur}</td>

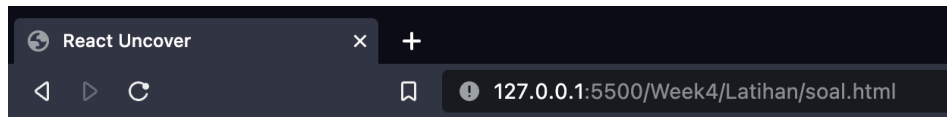
<td>{mahasiswa.jurusan}</td>

                  </tr>
                );
              })
            }
          </tbody>
        </table>
      </div>
    );

    ReactDOM.createRoot(document.getElementById('root')).render(myElement);
  </script>
</body>
</html>

```

ii. Hasil



Data Mahasiswa

Nama	Umur	Jurusan
Eka	19	Teknik Informatika
Lisa	18	Sistem Informasi
Rudi	19	Teknik Elektro

iii. Penjelasan

1. Kode di atas membuat sebuah komponen React yang menampilkan tabel data untuk daftar mahasiswa. Daftar mahasiswa didefinisikan sebagai sebuah array objek yang berisi nama, usia, dan jurusan mereka.
2. Variabel `myElement` mendefinisikan struktur JSX dari komponen tersebut, yang mencakup elemen `<h1>` dan elemen `<table>` dengan baris header dan tubuh data. Baris data dibuat menggunakan metode `.map()` dari array mahasiswa, yang membuat array baru elemen `<tr>` dengan elemen `<td>` anak yang berisi data untuk setiap mahasiswa. Atribut `key` ditambahkan pada setiap baris untuk membantu React mengoptimalkan penggambaran tabel.