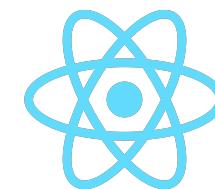


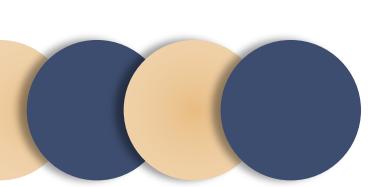


Pemrograman Sisi Cleint

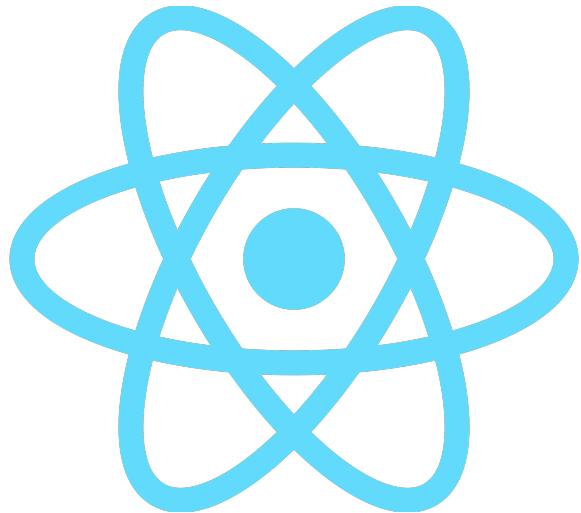
ReactJS Props dan List Map

Muhammad Syaifur Rohman, S.Kom, M.CS





Definisi



FYI

Props (singkatan dari "properties") adalah mekanisme untuk mengirim data dari satu komponen ke komponen lain sebagai parameter. Mereka adalah cara untuk membuat komponen React menjadi dinamis dan dapat digunakan kembali dengan cara yang berbeda.

Konsep Props



```
// Komponen induk
function ParentComponent() {
  return <ChildComponent name="John" />;
}
```

Send Props

```
// Komponen anak
function ChildComponent(props) {
  return <p>Hello, {props.name}!</p>;
}
```

Props

get Props

Mengirim Props: Anda dapat mengirim props dari komponen induk ke komponen anak dengan menyertakan properti di elemen JSX.

Menerima Props: Di komponen anak, Anda bisa mengakses props yang diterima melalui parameter fungsi.



Konsep Props

Latihan: Buat Komponen Table dan Komponen baris. Pass dari Table props lalu di tampilkan di Komponen baris

Konsep Props

Passing beberapa data melalui props memudahkan pembuatan komponen. Code yang dihasilkan bisa lebih sederhana

```
● ● ●  
function Table(){  
  return (  
    <div>  
      <table>  
        <tr>  
          <Cell isi="Judul 1" />  
          <Cell isi="Judul 2" />  
          <Cell isi="Judul 3" />  
        </tr>  
        <tr>  
          <Cell isi="Isi 1" />  
          <Cell isi="Isi 1" />  
          <Cell isi="Isi 1" />  
        </tr>  
        <tr>  
          <Cell isi="Isi 2" />  
          <Cell isi="Isi 2" />  
          <Cell isi="Isi 2" />  
        </tr>  
        <tr>  
          <Cell isi="Isi 3" />  
          <Cell isi="Isi 3" />  
          <Cell isi="Isi 3" />  
        </tr>  
      </table>  
    </div>  
  )  
}
```

```
● ● ●  
function Cell(props){  
  return (  
    <td>{props.isi}</td>  
  )  
}
```

TRY... 

TRY... 



Konsep Props

Latihan: Komponen Parent dengan sub sub child

Konsep Props

```
● ● ●  
function Table(){  
  return (  
    <div>  
      <table>  
        <Baris  
          isi1="Judul 1"  
          isi2="Judul 2"  
          isi3="Judul 3"  
        />  
  
        <Baris  
          isi1="Isian 1"  
          isi2="Isian 2"  
          isi3="Isian 3"  
        />  
  
        <Baris  
          isi1="Isian 1"  
          isi2="Isian 2"  
          isi3="Isian 3"  
        />  
  
        <Baris  
          isi1="Isian 1"  
          isi2="Isian 2"  
          isi3="Isian 3"  
        />  
  
      </table>  
    </div>  
  )  
}
```

```
● ● ●  
function Baris(props){  
  return (  
    <tr>  
      <Cell isi={props.isi1} />  
      <Cell isi={props.isi2} />  
      <Cell isi={props.isi3} />  
    </tr>  
  );  
}
```

TRY... 

```
● ● ●  
function Cell(props){  
  return (  
    <td>{props.isi}</td>  
  )  
}
```

TRY... 

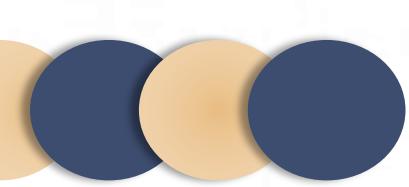
Spesific Props



```
function Landing(){
  return (
    <Header />
    <Deskripsi
      judul="Mengejar Doa Restu Ayang"
      penulis="sifulan wa fulin"
      deskripsiSingkat="Cerita pemuda yang rela belajar untuk pujaan hati"
    />
    <Footer />
  )
}
```



```
function Deskripsi({judul, penulis, deskripsiSingkat}){
  return (
    <div>
      <h1>{judul}</h1>
      <div>
        <b>{penulis}</b>
      </div>
      <div>
        {deskripsiSingkat}
      </div>
    </div>
  )
}
```



List Map

Intro

Di dalam pengembangan aplikasi web, seringkali kita perlu menampilkan daftar elemen yang berasal dari data dinamis, seperti daftar menu, daftar produk, atau daftar pengguna. Dalam ReactJS, kita dapat menggunakan metode `.map()` untuk melakukan iterasi melalui array data dan membuat komponen React untuk setiap item data.

Penggunaan

Ketika kita memiliki sebuah komponen React yang menerima props untuk menampilkan data, kita dapat menggunakan `.map()` di dalam komponen tersebut untuk membuat elemen UI untuk setiap item data yang diterima melalui props.

Why

Penggunaan `.map()` memungkinkan kita untuk membuat komponen React secara dinamis berdasarkan data yang diberikan. Tanpa menggunakan `.map()`, kita harus menulis kode secara manual untuk setiap item data, yang akan menjadi tidak efisien dan sulit dipelihara terutama jika data berubah atau berkembang.

Konsep Props



```
function MenuKopi() {
  const menu = [
    { nama: 'Espresso', deskripsi: 'Kopi pekat dengan rasa yang khas', harga: 'Rp 20.000' },
    { nama: 'Cappuccino', deskripsi: 'Kopi dengan campuran susu dan foam susu', harga: 'Rp 25.000' },
    { nama: 'Latte', deskripsi: 'Kopi dengan susu dan sedikit foam susu', harga: 'Rp 22.000' },
    { nama: 'Mocha', deskripsi: 'Kopi dengan campuran susu dan cokelat', harga: 'Rp 27.000' },
    { nama: 'Americano', deskripsi: 'Kopi hitam dengan air panas', harga: 'Rp 18.000' }
  ];

  return (
    <div>
      <h1>Menu Kopi</h1>
      {menu.map((item, index) => (
        <MenuKopiItem key={index} nama={item.nama} deskripsi={item.deskripsi} harga={item.harga} />
      ))}
    </div>
  );
}
```

List Map

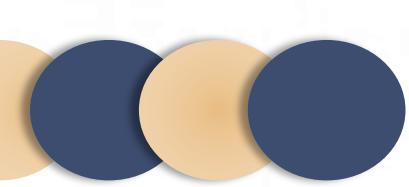
Konsep List Map



```
function MenuKopiItem({ nama, deskripsi, harga }) {  
  return (  
    <div>  
      <h2>{nama}</h2>  
      <p>{deskripsi}</p>  
      <p>Harga: {harga}</p>  
    </div>  
  );  
}
```

TRY...


A circular button with a black border and a light green background. The word "TRY..." is written in white capital letters. A hand cursor icon with a pointing finger is positioned over the bottom right corner of the button.



Referensi

1. Eran Krinsbruner, **A Frontend Web Developer's Guide to Testing**, 2022
2. Adam Boduch , Roy Derks , Mikhail Sakhniuk, **React and React Native - Fourth Edition**, 2022
3. Maya Shavin , Raymond Camden, **Frontend Development Projects with Vue.js 3 - Second Edition**, 2023
4. Waweru Mwaura, **End-to-End Web Testing with Cypress**, 2021
5. <https://roadmap.sh/frontend>
6. <https://jsonplaceholder.typicode.com/posts>



Thanks

Muhammad Syaifur Rohman, S.Kom, M.CS

