**PRAKTIKUM 12**

**Fundamental ReactJS**

**A. TUJUAN PEMBELAJARAN**

1. Mampu memahami fundamental React JS

2. Mampu memahami ReactDOM Render

3. Mampu memahami konsep Lanjutan React JS

**B. DASAR TEORI**

**1. Slicing Component**

Slicing adalah kegiatan memiliki tujuan untuk mengkonversi component tampilan design web menjadi bagian-bagian yang kecil. Bagian bagian yang kecil ini adalah bagian-bagian yang akan disusun di component utama sebuah web. Pada react kita bisa melakukan slicing component yang ada di fungsi utama yaitu App, component yang ada di fungsi utama tersebut kemudian di pecah menjadi bagian kecil-kecil menggunakan fungsi.

|  |  |
| --- | --- |
| Before Slicing | After Slicing |
| import logo from './logo.svg';  import './App.css';  function App() {    return (     <div className="Parentbox">      <div className="Foto">        <img src="sneakers.jpg"/>      </div>      <div className="Deskripsi">        <p className="Cate">SNEAKER</p>        <h1 className="Title">Pinky Swagg</h1>        <p className="Price">IDR 73.309.399</p>        <p className="Info">        Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.        </p>      </div>     </div>    );  }  export default App; | import logo from './logo.svg';  import './App.css';  function App() {    return (     <div className="Parentbox">       </div>    );  }  function FotoProduk(){    return (      <div className="Foto">      <img src="sneakers.jpg"/>    </div>     );  }  function ProdukInfo(){    const category = "SNEAKER";    return (      <div>       <div className="Deskripsi">         <p className="Cate">{category}</p>         <h1 className="Title">Pinky Swagg</h1>         <p className="Price">IDR 73.309.399</p>         <p className="Info">         Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.         </p>       </div>      </div>     );  }  export default App; |

**2. Props**

Props adalah argumen yang diteruskan ke komponen React. Props diteruskan ke komponen melalui atribut HTML. Kemudian apa yang bisa di passing atau dikirim dalam props? Yang bisa dikirim dalam props bisa berupa data, variables, state function dan bahkan sebuah kelas. Jadi initinya Props adalah suatu cara untuk mengirim dan mengakses data dari komponent lain secara langsung.

**3. Event**

Event adalah kejadian yang terjadi di halaman web. Kejadian yang dimaksud di sini seperti aktivitas yang dikerjakan pada halaman web, Misalnya seperti:

* User melakukan scroll;
* User melakukan klik pada elemen tertentu;
* Halaman web di-load;
* Form di-submit;
* dan sebagainya.

Kita bisa melakukan aksi tertentu untuk merespon event tersebut. Misalnya Saat elemen <button> diklik, kita bisa jalankan fungsi untuk melakukan sesuatu

**4. Argument**

Parameter, dalam pemanggilan fungsi, adalah argumen fungsi. Argumen JavaScript diteruskan oleh nilai: Fungsi hanya mengetahui nilai, bukan lokasi argumen. Jika suatu fungsi mengubah nilai argumen, maka tidak mengubah nilai asli parameter

|  |
| --- |
| Contoh penggunaah argument didalam fungsi |
| **function functionName(argument1, argument2, argument3) {**  **// code to be executed**  **}** |

**5. Conditional Rendering**

Conditional Rendering pada React berfungsi sama halnya dengan conditional operator pada Javascript. Gunakan JavaScript operator seperti if atau confitional operator untuk membuat representasi elemen dari state saat ini, dan React akan memperbarui UI sesuai dengan state tersebut.

function UserGreeting(props) {

  return <h1>Welcome back!</h1>;

}

function GuestGreeting(props) {

  return <h1>Please sign up.</h1>;

}

function Greeting(props) {

  const isLoggedIn = props.isLoggedIn;

  if (isLoggedIn) {

  return <GuestGreeting />;

  }

    return <UserGreeting />;

}

const root = ReactDOM.createRoot(document.getElementById('root'));

// Try changing to isLoggedIn={true}:

root.render(<Greeting isLoggedIn={false} />);

**5. Lists dan Keys**

const numbers = [1, 2, 3, 4, 5];

const doubled = numbers.map((number) => number \* 2);

console.log(doubled);

Kode diatas akan me-log [2,4,6,8,10] ke dalam konsol.

Jika melihat kode di diatas, kita menggunakan fungsi map() untuk mengambil list numbers dan menggandakan nilainya. Kita akan menaruh list baru yang dikembalikan oleh map() ke dalam sebuah variabel doubled dan me-log-nya.

**C. TUGAS PENDAHULUAN**

1. Apakah yang dimaksud dengan slicing component, props, event, argument dan conditional rendering?

2. Buatlah contoh penggunaan slicing component, props, event, argument dan conditional rendering

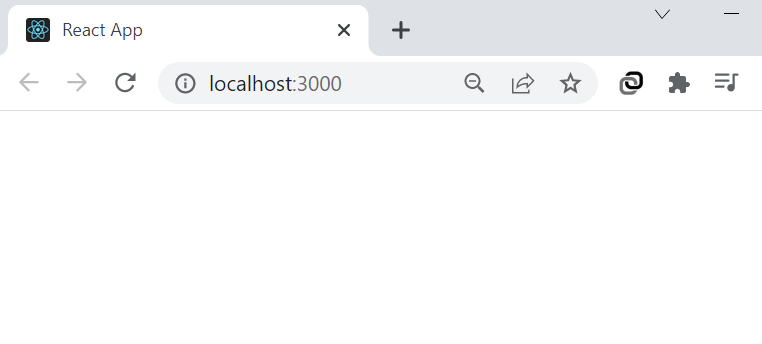
**D. PERCOBAAN**

1. Design WEB Part 2 | Slicing Component

Kemudian kita akan membuat data produk yang kita letakkan di dalam function App menjadi lebih dinamis. Dengngan melakukan Slicing component, lalu component yang tadi berada di functionn App, kita slicing menjadi beberapa component menggunakan function seperti berikut

|  |  |
| --- | --- |
| Before Slicing | After Slicing |
| import logo from './logo.svg';  import './App.css';  function App() {    return (     <div className="Parentbox">      <div className="Foto">        <img src="sneakers.jpg"/>      </div>      <div className="Deskripsi">        <p className="Cate">SNEAKER</p>        <h1 className="Title">Pinky Swagg</h1>        <p className="Price">IDR 73.309.399</p>        <p className="Info">        Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.        </p>      </div>     </div>    );  }  export default App; | import logo from './logo.svg';  import './App.css';  function App() {    return (     <div className="Parentbox">       </div>    );  }  function FotoProduk(){    return (      <div className="Foto">      <img src="sneakers.jpg"/>    </div>     );  }  function ProdukInfo(){    const category = "SNEAKER";    return (      <div>       <div className="Deskripsi">         <p className="Cate">{category}</p>         <h1 className="Title">Pinky Swagg</h1>         <p className="Price">IDR 73.309.399</p>         <p className="Info">         Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.         </p>       </div>      </div>     );  }  export default App; |

Maka hasilnya akan seperti ini



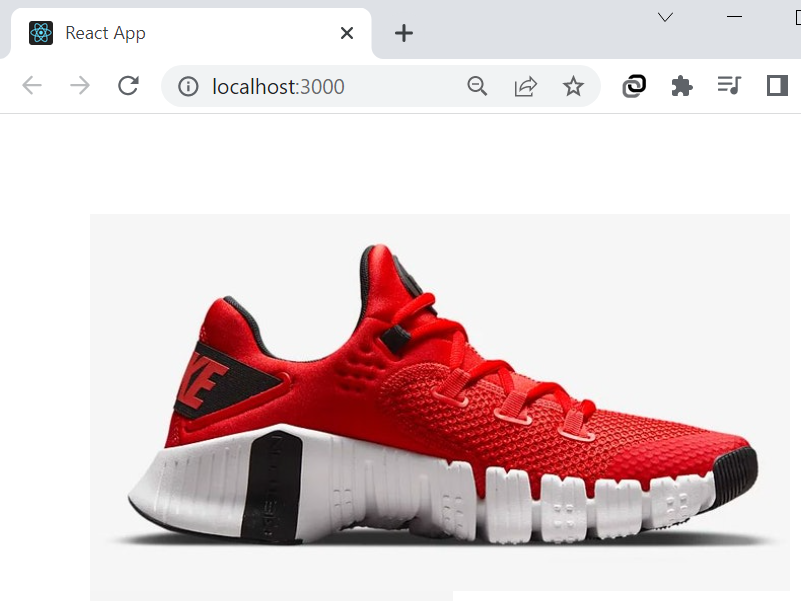
2. Props Component

Dengan menggunakan props ini kita akan melemparkan data dari component satu dengan yang lainnya

Contoh dari script sebelumnya kita bisa mendefiniksikan component FotoProduk yang sebelumnya sudah kita buat seperti contoh berikut

|  |  |
| --- | --- |
| App component | Component FotoProduk |
| function App() {    return (     <div className="Parentbox">      <FotoProduk/>     </div>    );  } | function FotoProduk(){    return (      <div className="Foto">      <img src="sneakers.jpg"/>    </div>     );  } |

Maka component Foto produk yang diteruskan ke Component App akan dirender oleh Function App di root. Hasilnya seperti berikut.



Lakukan hal yang sama pada component ProdukInfo dengan meletakkannya dibawah FotoProduk, seperti berikut.

function App() {

  return (

   <div className="Parentbox">

    <FotoProduk/>

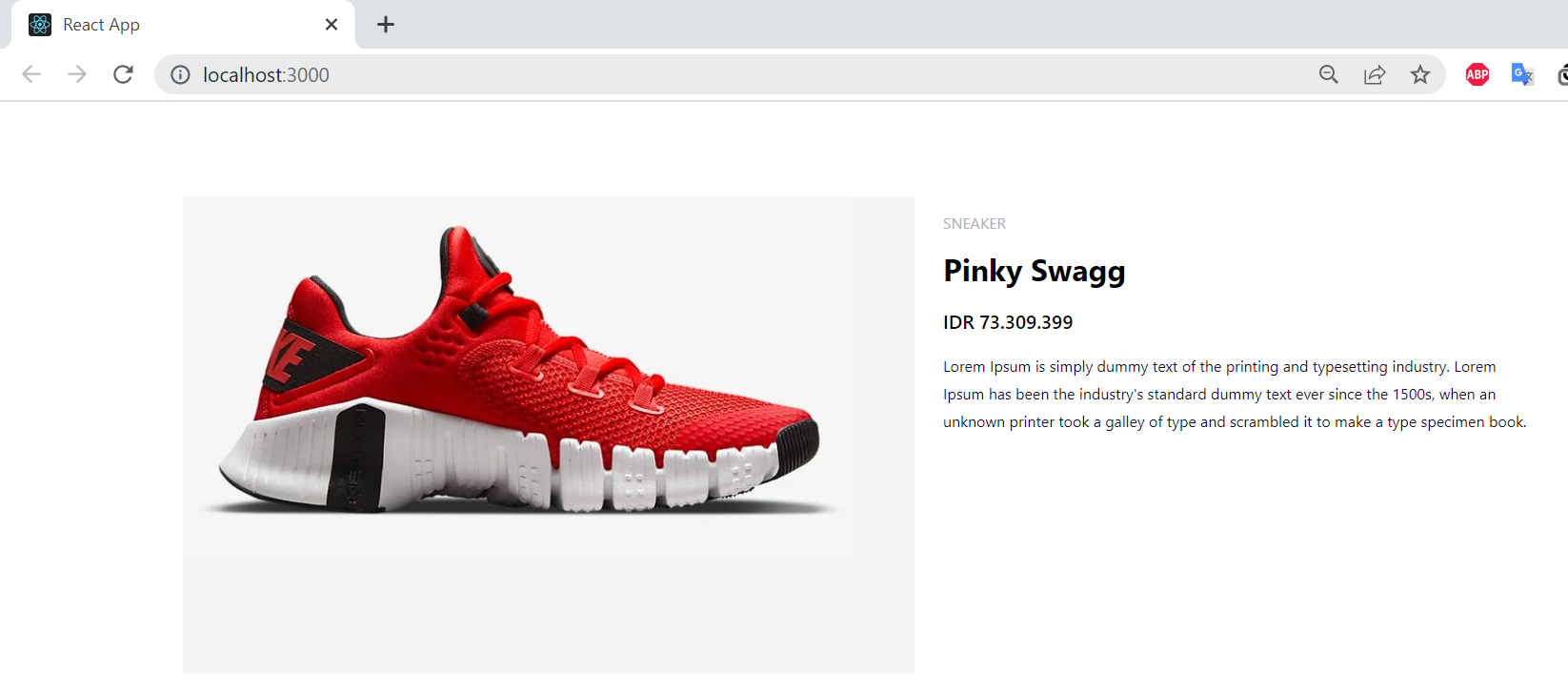
    <ProdukInfo/>

   </div>

  );

}

Dan hasilnya akan seperti ini.



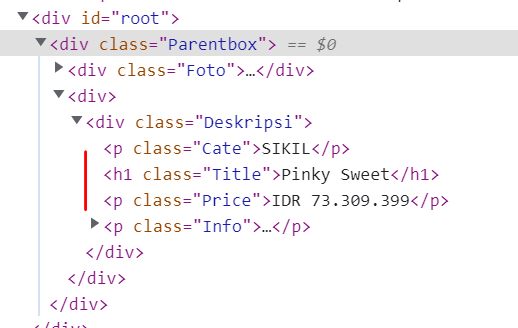
Kemudian kita tambahkan opsi props pada component ProdukInfo(props), seperti berikut ini.

|  |  |
| --- | --- |
| Component ProdukInfo | Component App |
| function ProdukInfo(props){    const {category} = props;  //object distrupsion    return (      <div>       <div className="Deskripsi">         <p className="Cate">{category}</p>         <h1 className="Title">Pinky Swagg</h1>         <p className="Price">IDR 73.309.399</p>         <p className="Info">         Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.         </p>       </div>      </div>     );  } | function App() {    return (     <div className="Parentbox">      <FotoProduk/>      <ProdukInfo category="SIKIL"/>     </div>    );  } |

Beikutnya lakukan hal yang sama pada Judul produk seperti berikut ini.

|  |  |
| --- | --- |
| Component ProdukInfo | Component App |
|  | function App() {    return (     <div className="Parentbox">      <FotoProduk/>      <ProdukInfo category="SIKIL" judul="Pinky Sweet"/>     </div>    );  } |

Dan jika kita inpect elemen hasilnya maka judul dan kategori tetap terlihat ada pada Deskripsi / Component ProdukInfo



3. Memberikan Event

Pada component ProdukInfo tambahkan button Add to Cart menggunakan tag <a> dan tambahkan styling untuk tombol add too cart di App.css .

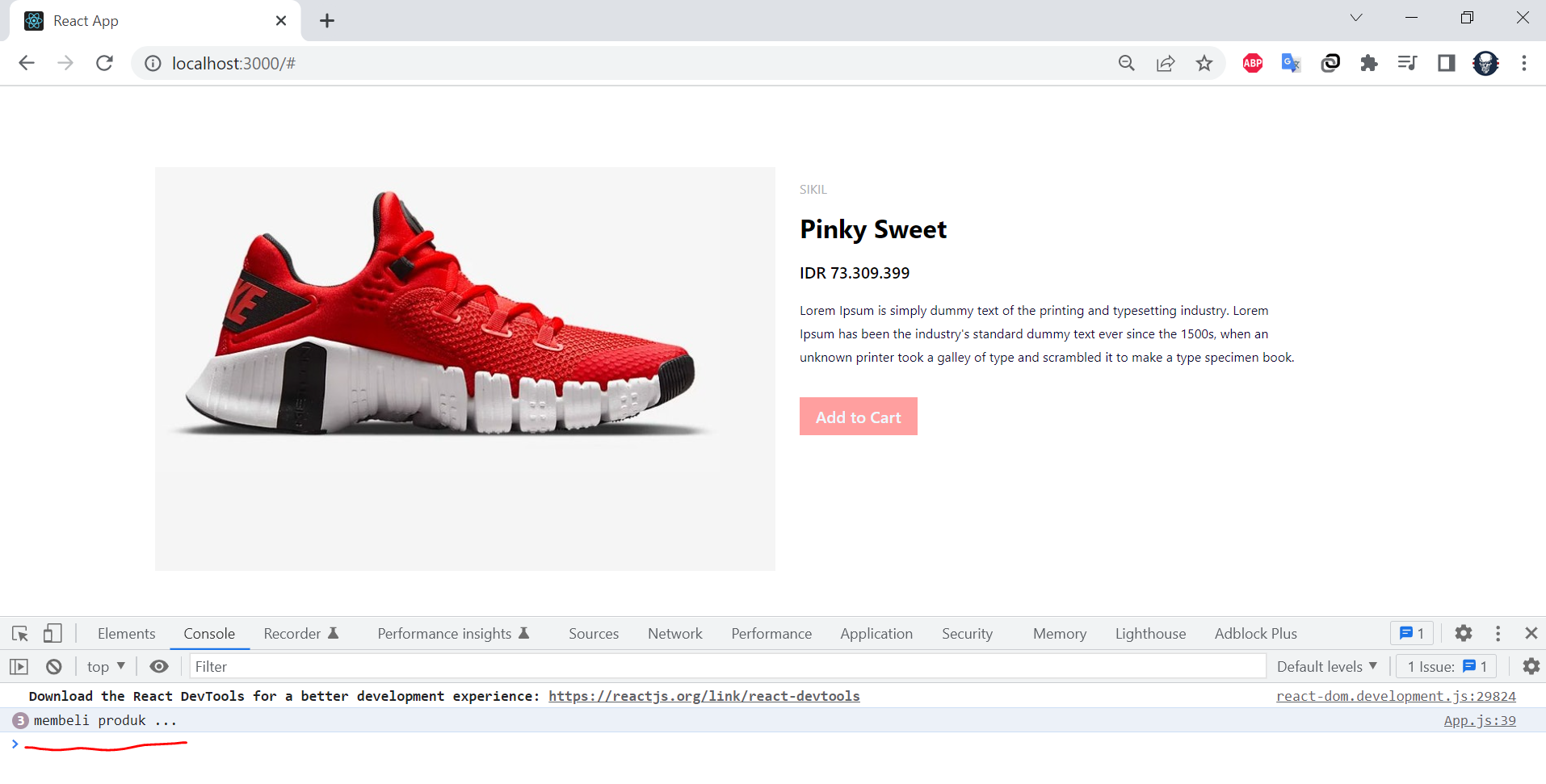
Script add to cart pada

<a onClick={Tambahcart} href="#">Add to Cart</a>

|  |  |
| --- | --- |
| Component Add to Cart | Css styling fo Add to Cart |
| function Tambahcart(){    return console.log("membeli produk ...")  } | .Deskripsi a{    margin-top: 20px;    color: aliceblue;    font-weight: 500;    background: rgb(255, 159, 159);    text-decoration: none;    display: inline-block;    padding: 10px 20px 10px 20px;    font-size: 20px;    border-radius: 4x;    transition: all 0.8s;  }  .Deskripsi a:hover{    color: #ffffff;    background: rgb(221, 135, 135);    transition: all 0.8s;  } |

|  |
| --- |
| Script lengkap pada component produkInfo |
| function ProdukInfo(props){    const {category, judul} = props; //object distrupsion    return (      <div>       <div className="Deskripsi">         <p className="Cate">{category}</p>         <h1 className="Title">{judul}</h1>         <p className="Price">IDR 73.309.399</p>         <p className="Info">         Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.         </p>         <a onClick={Tambahcart} href="#">Add to Cart</a>       </div>      </div>     );  } |

Maka hasilnya akan seperti ini, dan jika button add to cart di klik maka di console akan terlihat event , ketika produk tersebut dibeli



4. Passing Arguments to Event Handlers

kita juga bisa memberikan argument pada event handlers TambahCart, yaitu argument berupa id produk, jadi ketika button Add to Cart di klik maka id dari produk tersebut akan dibawa oleh event handlers.

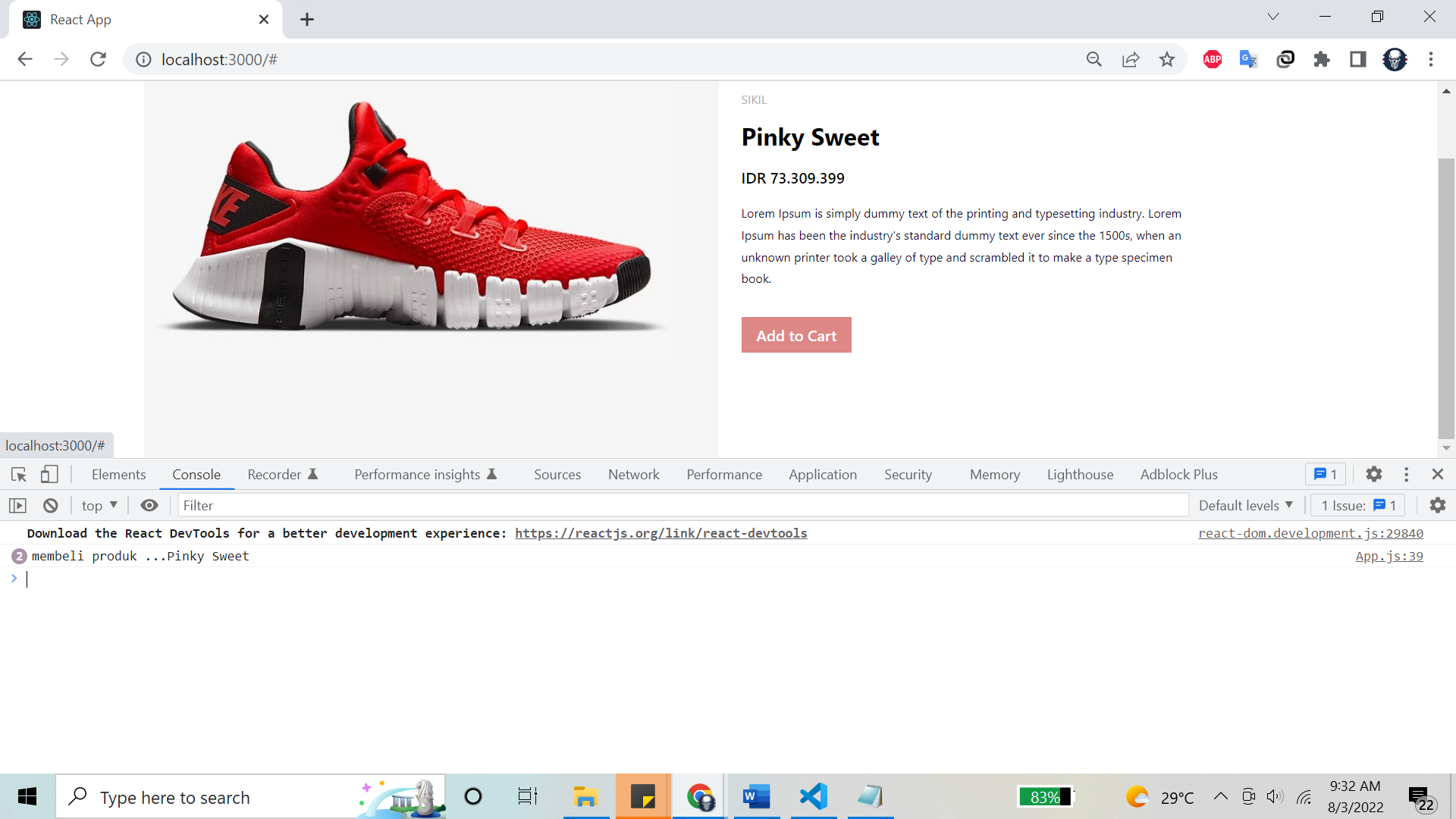
Tambahkan argument pada event onclick yang ada pata tag <a> di button Add to Cart, seperti berikut.

|  |  |
| --- | --- |
| Event onclick sebelum di tambahkan argument | Event onclick setelah di tambahkan argument |
| <a onClick={Tambahcart} href="#">Add to Cart</a> | <a onClick={(e) => Tambahcart(judul, e)} href="#">Add to Cart</a> |

Tambahkan parameter (e) pada fungsi tambah cart.

|  |  |
| --- | --- |
| Fungsi Tambahcart sebelum ditambahkan parameter (e) | Fungsi Tambahcart setelah ditambahkan parameter (e) |
| function Tambahcart(){    return console.log("membeli produk ...")  } | function Tambahcart(e){    return console.log("membeli produk ..." + e)  } |

maka hasilnya akan seperti ini, jika tombol Add to Cart di klik pada consol akan di tampilkan id atau produk yang baru saja klik.



5. Conditional Rendering

Kita juga bisa membuat sebuah kondisi pada info produk yaitu diskon, dengan menggunakan operator kondisional if seperti berikut.

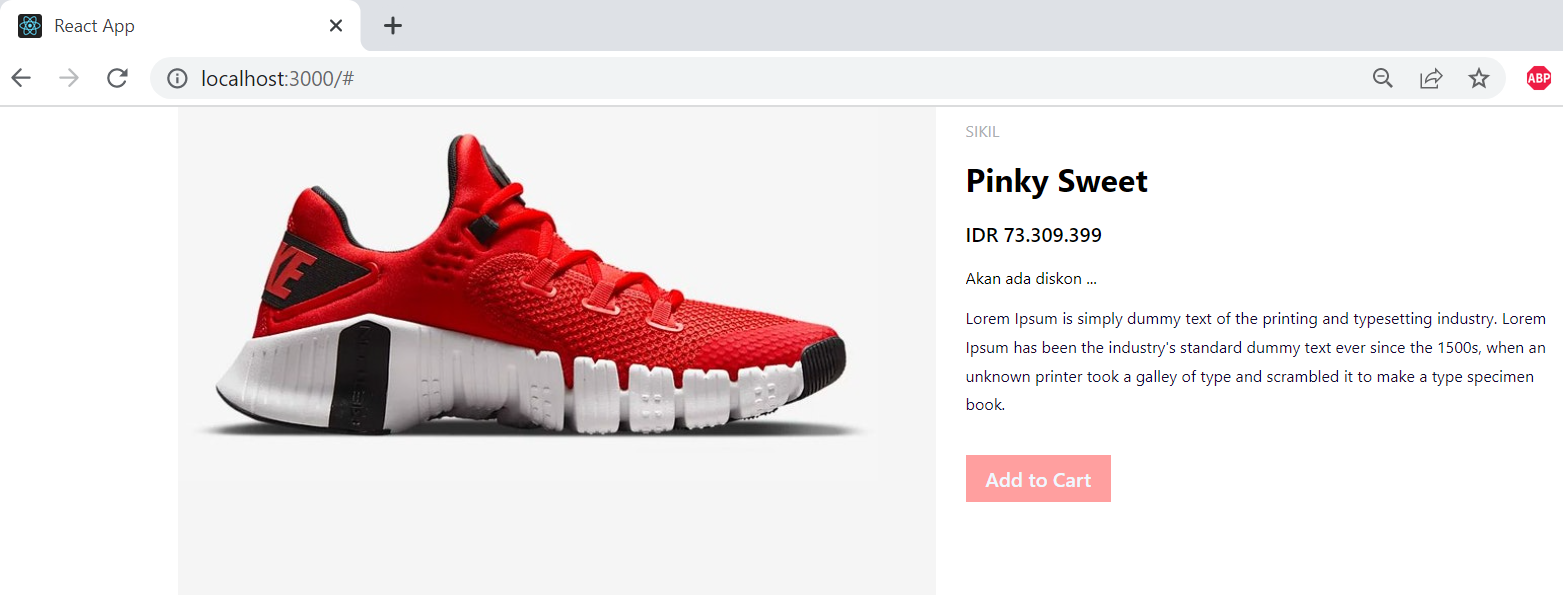
Pertama buat sebuah fungsi jika ada diskon seperti berikut

|  |
| --- |
| Fungsi checkDiscount |
| function CheckDiscount(props){    const {isDiscount} = props;    if(isDiscount == "yes"){      return(        <p>Diskon 50% off</p>      );    }    else if(isDiscount == "coming"){      return(        <p>Akan ada diskon ...</p>      );    }    else{      return(        <p>Belum ada diskon</p>      );    }  } |
| Kemudian tambahkan component isdiscount di bawah price, samakan seperti script dibawah. |
|  |

Kemudian tambahkan parameter isDiscount dengan value sesuai kondisi yang ada pada fungsi isDiscount.

|  |
| --- |
| Fungsi App |
| function App() {    return (     <div className="Parentbox">      <FotoProduk/>      <ProdukInfo isDiscount="coming" category="SIKIL" judul="Pinky Sweet"/>     </div>    );  } |

Maka hasilnya akan seperti berikut.



6. Map Array

Kita juga bisa melakukan looping data menggunakan fungsi map yang ada pada react, jadi pada produk info kita tambahkan list keuntungan jika mebeli produk tersebut. Refrensi [https://reactjs.org/docs/lists-and-keys.html#basic-list-component](https://reactjs.org/docs/lists-and-keys.html" \l "basic-list-component)

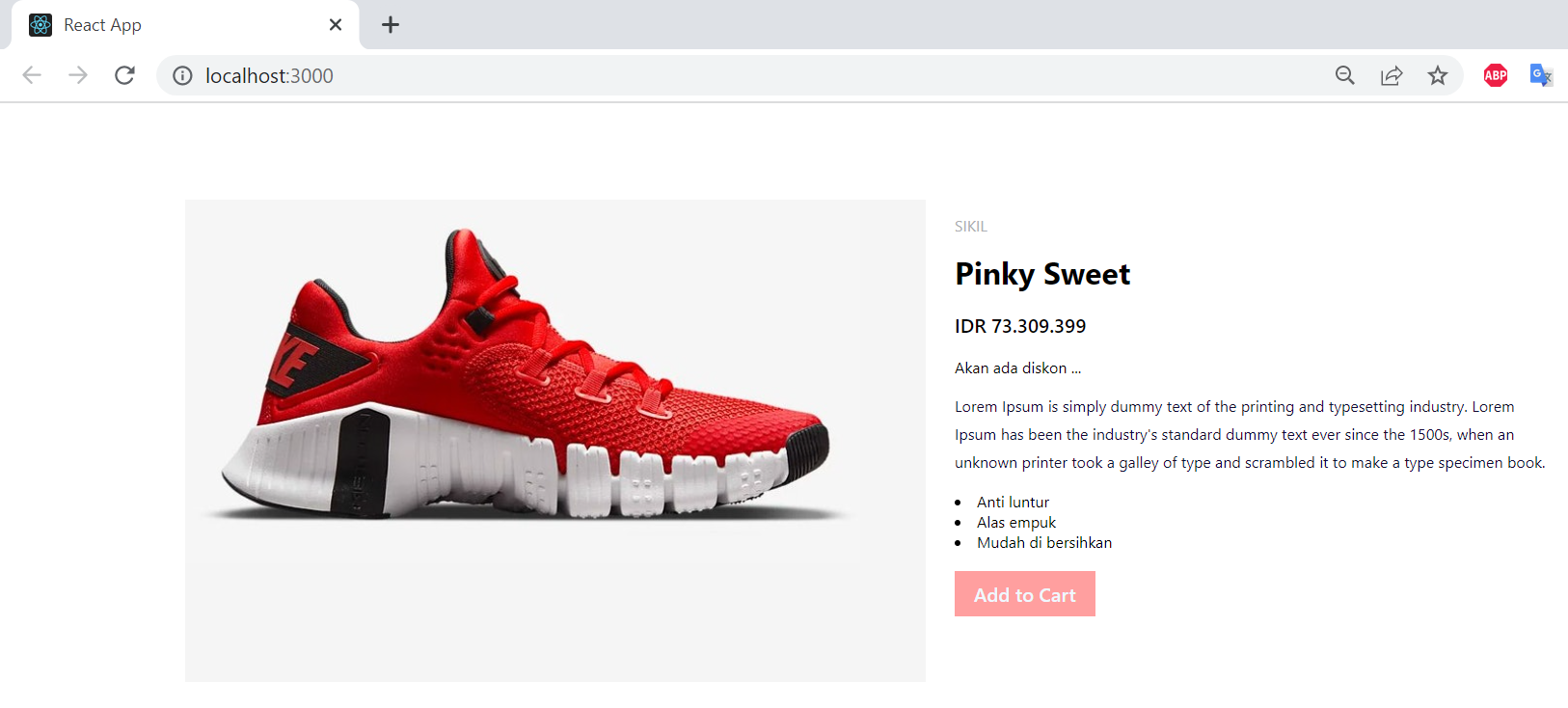
|  |
| --- |
| Buat data array kemudian data tersebut di loping menggunakan fungsi map seperti berikut. |
| const benefits = ["Anti luntur","Alas empuk","Mudah di bersihkan"]    const listbenefits = benefits.map((itembenefits) =>        <li>          {itembenefits}        </li>    ); |

|  |
| --- |
| Kemudian data yang sudah di loping di deklarasikan di produk info dibawah deskripsi produk |
| {listbenefits} |

Beikut adalah script lengkap pada fungsi ProdukInfo

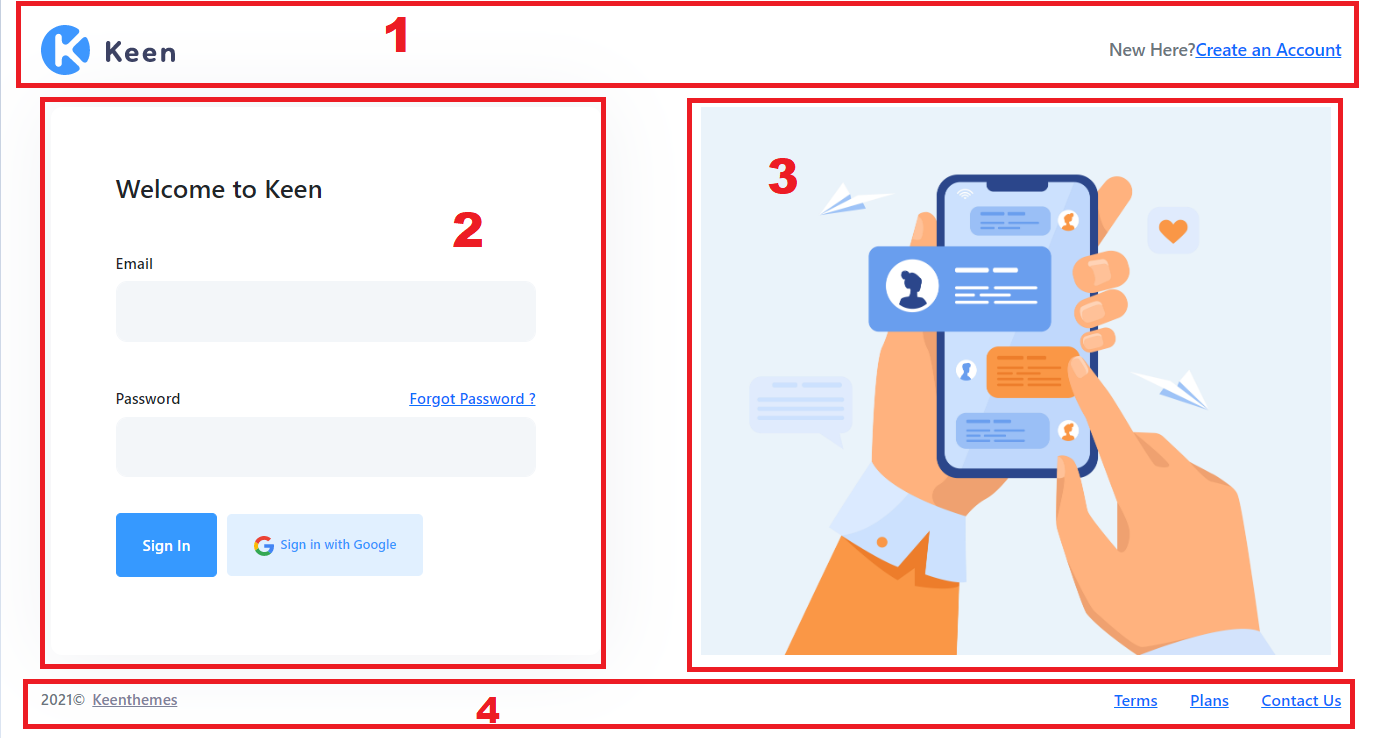
|  |
| --- |
| function ProdukInfo(props){    const {isDiscount,category, judul} = props; //object distrupsion      const benefits = ["Anti luntur","Alas empuk","Mudah di bersihkan"]    const listbenefits = benefits.map((itembenefits) =>        <li>          {itembenefits}        </li>    );      return (      <div>       <div className="Deskripsi">         <p className="Cate">{category}</p>         <h1 className="Title">{judul}</h1>         <p className="Price">IDR 73.309.399</p>         <CheckDiscount isDiscount={isDiscount}/>         <p className="Info">         Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.         </p>         {listbenefits}         <a onClick={(e) => Tambahcart(judul, e)} href="#">Add to Cart</a>       </div>      </div>     );  } |

Maka data array yang kita looping akan tampil seperti berikut.



**F. TUGAS**

1 lakukan slicing 4 component dari desain web berikut kedalam react.



|  |
| --- |
| App.js |
| import logo from './logo.svg';  import './App.css';  import 'bootstrap/dist/css/bootstrap.min.css';  function App() {    return (        <div className="d-flex flex-column flex-root">       <div className="login login-2 login-signin-on d-flex flex-column flex-column-fluid bg-white position-relative overflow-hidden" id="kt\_login">            <div className="login-header py-10 flex-column-auto">              <div className="container d-flex flex-column flex-md-row align-items-center justify-content-center justify-content-md-between">                    <a href="#" className="flex-column-auto py-5 py-md-0">                      <img src="https://preview.keenthemes.com/keen/theme/demo5/dist/assets/media/logos/logo-6.svg" alt="logo" className="h-50px" />                  </a>                    <span className="text-muted font-weight-bold font-size-h4">New Here?                  <a href="javascript:;" id="kt\_login\_signup" className="text-primary font-weight-bolder">Create an Account</a></span>              </div>          </div>            <div className="login-body d-flex flex-column-fluid align-items-stretch justify-content-center">              <div className="container row">                  <div className="col-lg-6 d-flex align-items-center">                        <div className="login-form login-signin">                            <form className="form w-xxl-550px rounded-lg p-20" novalidate="novalidate" id="kt\_login\_signin\_form">                                <div className="pb-13 pt-lg-0 pt-5">                                  <h3 className="font-weight-bolder text-dark font-size-h4 font-size-h1-lg">Welcome to Keen</h3>                              </div>                                  <div className="form-group">                                  <label className="font-size-h6 font-weight-bolder text-dark">Email</label>                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg" type="text" name="username" autocomplete="off" />                              </div>                                  <div className="form-group">                                  <div className="d-flex justify-content-between mt-n5">                                      <label className="font-size-h6 font-weight-bolder text-dark pt-5">Password</label>                                      <a href="javascript:;" className="text-primary font-size-h6 font-weight-bolder text-hover-primary pt-5" id="kt\_login\_forgot">Forgot Password ?</a>                                  </div>                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg" type="password" name="password" autocomplete="off" />                              </div>                                  <div className="pb-lg-0 pb-5">                                  <button type="button" id="kt\_login\_signin\_submit" className="btn btn-primary font-weight-bolder font-size-h6 px-8 py-4 my-3 mr-3">Sign In</button>                                  <button type="button" className="btn btn-light-primary font-weight-bolder px-8 py-4 my-3 font-size-lg">                                  <span className="svg-icon svg-icon-md">                                        <svg xmlns="http://www.w3.org/2000/svg" width="20" height="20" viewBox="0 0 20 20" fill="none">                                          <path d="M19.9895 10.1871C19.9895 9.36767 19.9214 8.76973 19.7742 8.14966H10.1992V11.848H15.8195C15.7062 12.7671 15.0943 14.1512 13.7346 15.0813L13.7155 15.2051L16.7429 17.4969L16.9527 17.5174C18.879 15.7789 19.9895 13.221 19.9895 10.1871Z" fill="#4285F4" />                                          <path d="M10.1993 19.9313C12.9527 19.9313 15.2643 19.0454 16.9527 17.5174L13.7346 15.0813C12.8734 15.6682 11.7176 16.0779 10.1993 16.0779C7.50243 16.0779 5.21352 14.3395 4.39759 11.9366L4.27799 11.9466L1.13003 14.3273L1.08887 14.4391C2.76588 17.6945 6.21061 19.9313 10.1993 19.9313Z" fill="#34A853" />                                          <path d="M4.39748 11.9366C4.18219 11.3166 4.05759 10.6521 4.05759 9.96565C4.05759 9.27909 4.18219 8.61473 4.38615 7.99466L4.38045 7.8626L1.19304 5.44366L1.08875 5.49214C0.397576 6.84305 0.000976562 8.36008 0.000976562 9.96565C0.000976562 11.5712 0.397576 13.0882 1.08875 14.4391L4.39748 11.9366Z" fill="#FBBC05" />                                          <path d="M10.1993 3.85336C12.1142 3.85336 13.406 4.66168 14.1425 5.33717L17.0207 2.59107C15.253 0.985496 12.9527 0 10.1993 0C6.2106 0 2.76588 2.23672 1.08887 5.49214L4.38626 7.99466C5.21352 5.59183 7.50242 3.85336 10.1993 3.85336Z" fill="#EB4335" />                                      </svg>                                    </span>Sign in with Google</button>                              </div>                            </form>                        </div>                        <div className="login-form login-signup">                            <form className="form w-xxl-550px rounded-lg p-20" novalidate="novalidate" id="kt\_login\_signup\_form">                                <div className="pb-13 pt-lg-0 pt-5">                                  <h3 className="font-weight-bolder text-dark font-size-h4 font-size-h1-lg">Sign Up</h3>                                  <p className="text-muted font-weight-bold font-size-h4">Enter your details to create your account</p>                              </div>                                  <div className="form-group">                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg font-size-h6" type="text" placeholder="Fullname" name="fullname" autocomplete="off" />                              </div>                                  <div className="form-group">                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg font-size-h6" type="email" placeholder="Email" name="email" autocomplete="off" />                              </div>                                  <div className="form-group">                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg font-size-h6" type="password" placeholder="Password" name="password" autocomplete="off" />                              </div>                                  <div className="form-group">                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg font-size-h6" type="password" placeholder="Confirm password" name="cpassword" autocomplete="off" />                              </div>                                  <div className="form-group">                                  <div className="checkbox-inline">                                      <label className="checkbox">                                      <input type="checkbox" name="agree" />                                      <span></span>I Agree the                                      <a href="#" className="ml-1">terms and conditions</a>.</label>                                  </div>                              </div>                                  <div className="form-group d-flex flex-wrap pb-lg-0 pb-3">                                  <button type="button" id="kt\_login\_signup\_submit" className="btn btn-primary font-weight-bolder font-size-h6 px-8 py-4 my-3 mr-4">Submit</button>                                  <button type="button" id="kt\_login\_signup\_cancel" className="btn btn-light-primary font-weight-bolder font-size-h6 px-8 py-4 my-3">Cancel</button>                              </div>                            </form>                        </div>                        <div className="login-form login-forgot">                            <form className="form w-xxl-550px rounded-lg p-20" novalidate="novalidate" id="kt\_login\_forgot\_form">                                <div className="pb-13 pt-lg-0 pt-5">                                  <h3 className="font-weight-bolder text-dark font-size-h4 font-size-h1-lg">Forgotten Password ?</h3>                                  <p className="text-muted font-weight-bold font-size-h4">Enter your email to reset your password</p>                              </div>                                  <div className="form-group">                                  <input className="form-control form-control-solid h-auto p-6 rounded-lg font-size-h6" type="email" placeholder="Email" name="email" autocomplete="off" />                              </div>                                  <div className="form-group d-flex flex-wrap pb-lg-0">                                  <button type="button" id="kt\_login\_forgot\_submit" className="btn btn-primary font-weight-bolder font-size-h6 px-8 py-4 my-3 mr-4">Submit</button>                                  <button type="button" id="kt\_login\_forgot\_cancel" className="btn btn-light-primary font-weight-bolder font-size-h6 px-8 py-4 my-3">Cancel</button>                              </div>                            </form>                        </div>                    </div>                  <div className="col-lg-6 bgi-size-contain bgi-no-repeat bgi-position-y-center bgi-position-x-center min-h-150px mt-10 m-md-0 gambar"></div>                  </div>            </div>            <div className="login-footer py-10 flex-column-auto">              <div className="container d-flex flex-column flex-md-row align-items-center justify-content-center justify-content-md-between">                  <div className="font-size-h6 font-weight-bolder order-2 order-md-1 py-2 py-md-0">                      <span className="text-muted font-weight-bold mr-2">2021©</span>                      <a href="https://keenthemes.com/keen" target="\_blank" className="text-dark-50 text-hover-primary">Keenthemes</a>                  </div>                  <div className="font-size-h5 font-weight-bolder order-1 order-md-2 py-2 py-md-0">                      <a href="#" className="text-primary">Terms</a>                      <a href="#" className="text-primary ml-10">Plans</a>                      <a href="#" className="text-primary ml-10">Contact Us</a>                  </div>              </div>          </div>        </div>  </div>        );  }    export default App; |

|  |
| --- |
| App.css |
| .gambar {    background-image: url("../public/assets/img/6531.jpg");   } |

|  |
| --- |
| Index.html |
| <!DOCTYPE html>  <html lang="en">    <head>          <!-- Google Tag Manager -->          <script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start': new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0], j=d.createElement(s),dl=l!='dataLayer'?'&amp;l='+l:'';j.async=true;j.src= '/assets/../../www.googletagmanager.com/gtm5445.html?id='+i+dl;f.parentNode.insertBefore(j,f); })(window,document,'script','dataLayer','GTM-5FS8GGP');</script>          <!-- End Google Tag Manager -->          <meta charset="utf-8" />          <title>Keen | Login Page 2</title>          <meta name="description" content="Login page example" />          <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />          <!--begin::Fonts-->          <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Poppins:300,400,500,600,700" />          <!--end::Fonts-->          <!--begin::Page Custom Styles(used by this page)-->          <link href="/assets/theme/demo5/dist/assets/css/pages/login/login-21036.css?v=2.1.1" rel="stylesheet" type="text/css" />          <!--end::Page Custom Styles-->          <!--begin::Global Theme Styles(used by all pages)-->          <link href="/assets/theme/demo5/dist/assets/plugins/global/plugins.bundle1036.css?v=2.1.1" rel="stylesheet" type="text/css" />          <link href="/assets/theme/demo5/dist/assets/plugins/custom/prismjs/prismjs.bundle1036.css?v=2.1.1" rel="stylesheet" type="text/css" />          <link href="/assets/theme/demo5/dist/assets/css/style.bundle1036.css?v=2.1.1" rel="stylesheet" type="text/css" />          <!--end::Global Theme Styles-->          <!--begin::Layout Themes(used by all pages)-->          <!--end::Layout Themes-->          <link rel="shortcut icon" href="https://preview.keenthemes.com/keen/theme/demo5/dist/assets/media/logos/favicon.ico" />          <!-- Hotjar Tracking Code for keenthemes.com -->          <script>(function(h,o,t,j,a,r){ h.hj=h.hj||function(){(h.hj.q=h.hj.q||[]).push(arguments)}; h.\_hjSettings={hjid:1070954,hjsv:6}; a=o.getElementsByTagName('head')[0]; r=o.createElement('script');r.async=1; r.src=t+h.\_hjSettings.hjid+j+h.\_hjSettings.hjsv; a.appendChild(r); })(window,document,'https://static.hotjar.com/c/hotjar-','.js?sv=');</script>      </head>    <body id="kt\_body" class="header-mobile-fixed subheader-enabled aside-enabled aside-fixed aside-secondary-enabled page-loading">          <!-- Google Tag Manager (noscript) -->          <noscript>              <iframe src="https://www.googletagmanager.com/ns.html?id=GTM-5FS8GGP" height="0" width="0" style="display:none;visibility:hidden"></iframe>          </noscript>          <!-- End Google Tag Manager (noscript) -->          <!--begin::Main-->      <div id="root"></div>        <script>var HOST\_URL = "https://preview.keenthemes.com/keen/theme/tools/preview";</script>          <!--begin::Global Config(global config for global JS scripts)-->          <script>var KTAppSettings = { "breakpoints": { "sm": 576, "md": 768, "lg": 992, "xl": 1200, "xxl": 1200 }, "colors": { "theme": { "base": { "white": "#ffffff", "primary": "#1BC5BD", "secondary": "#E5EAEE", "success": "#1BC5BD", "info": "#6993FF", "warning": "#FFA800", "danger": "#F64E60", "light": "#F3F6F9", "dark": "#212121" }, "light": { "white": "#ffffff", "primary": "#1BC5BD", "secondary": "#ECF0F3", "success": "#C9F7F5", "info": "#E1E9FF", "warning": "#FFF4DE", "danger": "#FFE2E5", "light": "#F3F6F9", "dark": "#D6D6E0" }, "inverse": { "white": "#ffffff", "primary": "#ffffff", "secondary": "#212121", "success": "#ffffff", "info": "#ffffff", "warning": "#ffffff", "danger": "#ffffff", "light": "#464E5F", "dark": "#ffffff" } }, "gray": { "gray-100": "#F3F6F9", "gray-200": "#ECF0F3", "gray-300": "#E5EAEE", "gray-400": "#D6D6E0", "gray-500": "#B5B5C3", "gray-600": "#80808F", "gray-700": "#464E5F", "gray-800": "#1B283F", "gray-900": "#212121" } }, "font-family": "Poppins" };</script>          <!--end::Global Config-->          <!--begin::Global Theme Bundle(used by all pages)-->          <script src="/assets/theme/demo5/dist/assets/plugins/global/plugins.bundle1036.js?v=2.1.1"></script>          <script src="/assets/theme/demo5/dist/assets/plugins/custom/prismjs/prismjs.bundle1036.js?v=2.1.1"></script>          <script src="/assets/theme/demo5/dist/assets/js/scripts.bundle1036.js?v=2.1.1"></script>          <!--end::Global Theme Bundle-->          <!--begin::Page Scripts(used by this page)-->          <script src="/assets/theme/demo5/dist/assets/js/pages/custom/login/login1036.js?v=2.1.1"></script>          <!--end::Page Scripts-->      </body>      <!--end::Body--> |

**G. LATIHAN**

Carilah Template Landing page Singgle page kemudian lakukan slicing kedalam react.

**H. LAPORAN RESMI**

Kumpulkan hasil latihan dan tugas di atas. Tambahkan analisa dalam laporan resmi.