

Pengembangan Aplikasi Web

TI2109

By:
Participant Handbook

Course Version: 2023

Universitas Mikroskil, Copyright ©2023

COURSE OVERVIEW

COURSE OVERVIEW

Mata kuliah ini, mahasiswa akan mempelajari tentang bagaimana membangun dan mengembangkan aplikasi web mulai dari front end, back-end server, aplikasi dan basis data sampai pembuatan RestAPI dengan memanfaatkan React Js, Node Js dan Express Js.

COURSE GOALS

Capaian Pembelajaran Lulusan yang dibebankan:

- Mahasiswa mampu memahami dan menerapkan konsep-konsep dasar dalam pengembangan web menggunakan React Js.
- Mahasiswa Mampu mengimplementasikan teknik routing dan styling dalam pengembangan web menggunakan React Js.
- Mahasiswa mampu membangun dan mengelola server menggunakan Express Js.
- Mahasiswa mampu mengintegrasikan API dalam pengembangan web menggunakan React Js.

COURSE OBJECTIVES

Capaian Pembelajaran Mata Kuliah:

- Mahasiswa mampu memahami dan menerapkan konsep-konsep ke dalam HTML.
- Mahasiswa mampu memilih element dan semantik yang sesuai pada HTML.
- Mahasiswa mampu menerapkan konsep tipografi dan multimedia pada halaman HTML.
- Mahasiswa mampu memilih CSS format yang sesuai untuk diterapkan ke dalam HTML.
- Mahasiswa mampu merencanakan penggunaan GIT untuk kolaborasi dalam proses pengembangan aplikasi.
- Mahasiswa mampu menerapkan konsep dari layout dan grid pada CSS.
- Mahasiswa mampu membangun sebuah halaman web yang responsive secara design dengan menggunakan media queries.
- Mahasiswa mampu menerapkan penggunaan dari list untuk pembuatan daftar pada halaman web.
- Mahasiswa mampu membangun tabel dan form untuk proses pendataan pada halaman web.
- Mahasiswa mampu menyusun web dengan menerapkan konsep javascript.



UNIT 2

KOMPONEN JS DAN JSX

UNIT OVERVIEW

Pada minggu kedua, kita akan mengeksplorasi konsep fundamental dalam pengembangan aplikasi React, yaitu "Komponen dan JSX". React memungkinkan kita untuk memecah User Interface (UI) menjadi komponen - komponen yang dapat digunakan kembali, sehingga mempermudah pengembangan aplikasi yang skalabel dan interaktif.

UNIT OBJECTIVES

Pada minggu kedua ini, mahasiswa akan fokus pada praktik langsung cara menerapkan konsep komponen berbasis JSX pada aplikasi react. Berikut adalah capaian yang diharapkan setelah menyelesaikan topik ini:

- Memahami Konsep Komponen: Mahasiswa akan mampu menjelaskan konsep komponen dalam pengembangan dengan React dan mengapa komponen sangat penting dalam pengembangan aplikasi web.
- Mengidentifikasi dan Memanfaatkan Fitur React Komponen: Mahasiswa dapat memahami keunggulan dan manfaat penggunaan komponen dalam membangun antarmuka pengguna yang dinamis.
- Memahami Konsep Jsx: Mahasiswa dapat mendefinisikan elemen JSX dalam kode React serta dapat memahami perbedaan antara JSX dan HTML konvensional.
- Memahami Konsep Dasar Styling Komponen : Mahasiswa dapat memisahkan kode React komponen dengan styling React komponen.
- Menyusun Komponen : Mahasiswa mampu menyusun komponen dengan baik dalam proyek React untuk membangun antarmuka pengguna yang kompleks dan dinamis.

UNIT CONTENTS

Lesson 1: Komponen dan Jsx	5 - 6
Lesson 2: Styling dasar Komponen React	6 - 7
Lesson 3: Menyusun Komponen React	7 - 9

PRE LAB

Pre lab, berisi pertanyaan mendasar terkait teori materi yang sedang diajarkan.

QUESTION

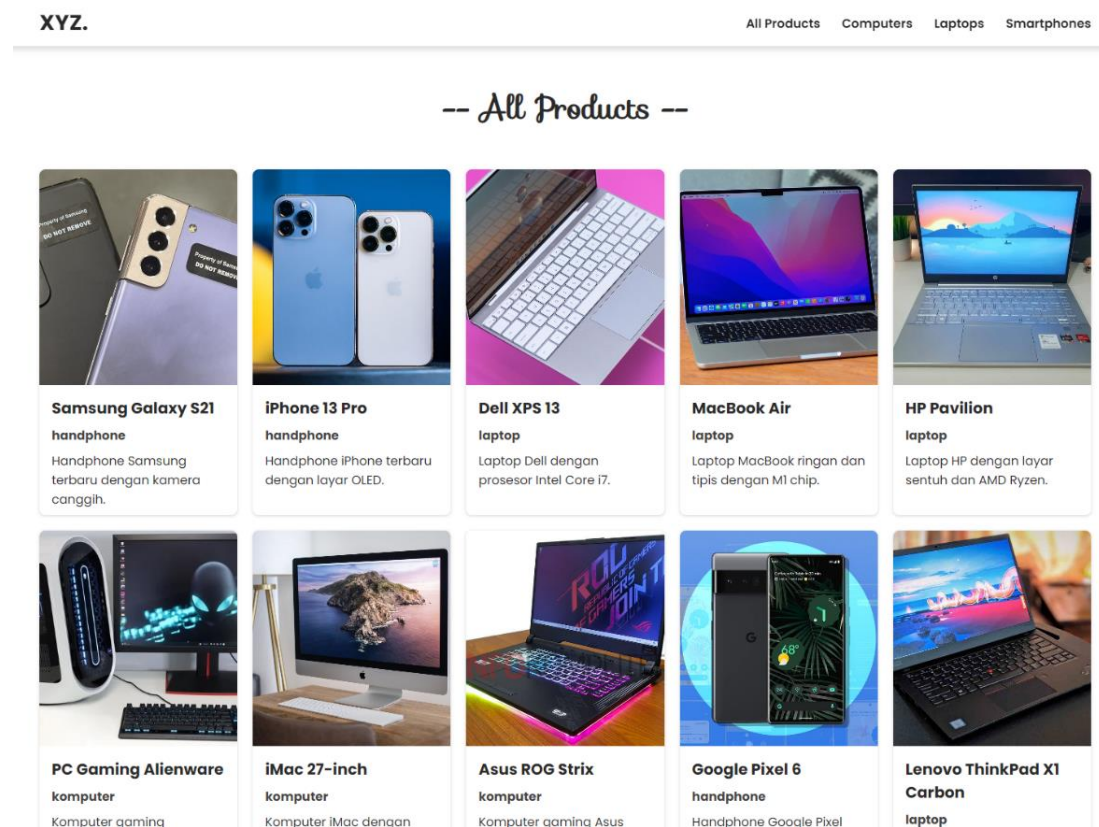
1. Apa yang dimaksud dengan komponen dalam React, dan mengapa konsep ini penting dalam pengembangan aplikasi React?
2. Apa itu JSX (JavaScript XML), dan bagaimana JSX digunakan untuk mendefinisikan elemen-elemen dalam tampilan antarmuka pengguna React?
3. Apa perbedaan antara komponen fungsional dan komponen kelas dalam React? Bagaimana cara Anda memilih antara keduanya dalam pengembangan aplikasi?
4. Apa itu props (property) dalam React, dan bagaimana Anda mengirim data antara komponen menggunakan props?
5. Bagaimana menggabungkan CSS atau Style dalam komponen React?

Apa peran dari penyusunan komponen dalam pengembangan aplikasi React, dan mengapa ini penting untuk membangun aplikasi yang skalabel dan mudah dipelihara?

CONTENT LESSON

CASE STUDY / PROJECT

Toko XYZ adalah sebuah toko yang menjual produk - produk elektronik. Pemilik toko berencana untuk membangun platform e-commerce mini yang memungkinkan customernya untuk melihat daftar produk yang mereka jual. Bantulah pemilik toko XYZ untuk membangun website yang bisa menampilkan daftar produk ke customer.



IDENTIFICATION CONCEPT OF PROBLEM / PROJECT

Identifikasi Proyek Minggu 2: Pengembangan E-Commerce Mini untuk Toko XYZ

Setelah memahami kasus yang diberikan, jelas bahwa tujuan akhir dari proyek ini adalah untuk membangun sebuah aplikasi web yang menampilkan produk yang dijual oleh Toko XYZ.

LESSON 01: KOMPONEN JS DAN JSX

Pada bagian ini, kita akan memahami dua konsep penting dalam pengembangan aplikasi web dengan React, yaitu "Komponen" dan "JSX" (JavaScript XML). Kedua konsep ini merupakan fondasi utama dalam membangun antarmuka pengguna yang dinamis dan responsif menggunakan React.

Komponen adalah bagian terkecil dari aplikasi React yang dapat digunakan kembali. Mereka adalah blok pembangun dasar dalam pengembangan aplikasi React. Komponen dapat berisi elemen HTML, fungsi, dan logika yang bersifat mandiri. Dengan komponen, Anda dapat mengorganisasi UI Anda menjadi bagian-bagian yang lebih kecil dan mudah dikelola.

JSX adalah ekstensi dari JavaScript yang memungkinkan kita untuk menulis kode HTML dalam JavaScript. Ini membuat pembuatan antarmuka pengguna menjadi lebih mudah dan intuitif dalam React. Dengan JSX, Anda dapat membuat struktur tampilan seperti elemen HTML, dan React akan mengubahnya menjadi elemen-elemen DOM yang sesuai.

```
1  import React from 'react';
2
3  // Contoh komponen sederhana
4  function Header(props) {
5    return (
6      <header>
7        <h1>{props.title}</h1>
8      </header>
9    );
10 }
11
12 function App() {
13   return (
14     <div>
15       {/* Menggunakan komponen Header dengan prop title */}
16       <Header title="Selamat Datang di Aplikasi React" />
17       <p>Selamat datang di aplikasi React yang sederhana!</p>
18     </div>
19   );
20 }
21
22 export default App;
23
```

Pada kode di atas, kita telah membuat dua komponen: Header dan App. Komponen Header digunakan dalam komponen App dengan prop title. JSX memungkinkan kita untuk menuliskan elemen HTML dalam komponen dengan lebih mudah dan jelas.

Dengan konsep Komponen dan JSX, kita dapat membangun aplikasi React yang terstruktur dan mudah dikelola. Ini adalah fondasi penting dalam pengembangan dengan React.

LESSON 02 : STYLING DASAR KOMPONEN REACT

Dalam pengembangan aplikasi React, styling adalah aspek penting untuk membuat antarmuka pengguna (UI) menarik dan sesuai dengan desain yang diinginkan. Salah satu cara untuk mengelola styling komponen React adalah dengan menggunakan CSS eksternal. Dalam metode ini, Anda akan membuat file CSS terpisah untuk setiap komponen dan menghubungkannya dengan komponen tersebut.

Buat file CSS eksternal yang akan digunakan untuk mengggaya komponen. Misalnya, beri nama file ini Header.css

```
1  /* Header.css */
2  .header {
3      background-color: #333;
4      color: #fff;
5      padding: 10px;
6      text-align: center;
7  }
8
```

Buat komponen React yang akan menggunakan CSS ini. Misalnya, beri nama file ini Header.js:

```
1  // Header.js
2  import React from 'react';
3  import './Header.css'; // Menghubungkan file CSS eksternal
4
5  function Header(props) {
6      return (
7          <header className="header"> {/* Menggunakan class CSS */}
8              <h1>{props.title}</h1>
9          </header>
10     );
11 }
12
13 export default Header;
14
```

Gunakan komponen Header dalam komponen lain, seperti App.js:

```
1 // App.js
2 import React from 'react';
3 import Header from './Header'; // Mengimpor komponen Header
4
5 function App() {
6   return (
7     <div>
8       <Header title="Selamat Datang di Aplikasi React" />
9       <p>Selamat datang di aplikasi React yang sederhana!</p>
10    </div>
11  );
12 }
13
14 export default App;
15
```

LESSON 03 : MENYUSUN KOMPONEN

Atomic Design adalah sebuah metodologi yang digunakan dalam desain antarmuka pengguna (UI) untuk memecah komponen menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola. Dalam pengembangan aplikasi React, konsep Atomic Design dapat diterapkan untuk menyusun komponen dengan struktur yang terorganisir dan reusabilitas yang tinggi. Berikut pengantar mengenai Atomic Design dalam konteks React:

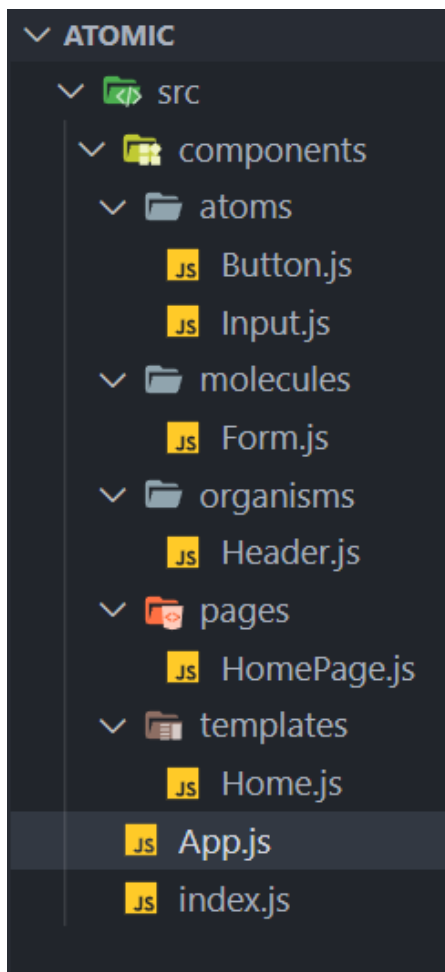
Mengapa Menggunakan Atomic Design?

Modularitas yang Tinggi: Atomic Design membagi komponen menjadi tingkatan yang berbeda, mulai dari atom (komponen paling dasar) hingga organisme (komponen yang lebih besar). Ini membuat komponen lebih mudah dikelola dan digunakan kembali.

Reusabilitas: Dengan Atomic Design, Anda dapat dengan mudah menggunakan kembali komponen yang ada dalam berbagai bagian aplikasi Anda, meningkatkan efisiensi pengembangan.

Konsistensi: Metodologi ini mempromosikan konsistensi desain dan struktur komponen di seluruh aplikasi, menghasilkan UI yang lebih terstruktur dan profesional.

Buat direktori (folder) untuk struktur Atomic Design Anda. Biasanya, ada lima tingkatan: atom, molekul, organisme, template, dan halaman. Contoh struktur direktori dapat seperti ini:



Buat komponen dalam setiap tingkatan sesuai dengan tujuannya. Misalnya, berikut adalah komponen sederhana untuk atom "Button" dan molekul "Form":

```
1 // Button.js (Atom)
2 import React from 'react';
3
4 function Button(props) {
5   return <button>{props.label}</button>;
6 }
7
8 export default Button;
9
```

Buat komponen dalam setiap tingkatan sesuai dengan tujuannya. Misalnya, berikut adalah komponen sederhana untuk atom "Button" dan molekul "Form":


```
1 // Form.js (Molekul)
2 import React from 'react';
3 import Button from '../atoms/Button';
4
5 function Form() {
6   return (
7     <form>
8       <input type="text" placeholder="Nama" />
9       <input type="email" placeholder="Email" />
10      <Button label="Submit" />
11    </form>
12  );
13 }
14
15 export default Form;
16
```

Dalam contoh ini, kita memiliki komponen atom Button dan molekul Form. Molekul Form menggunakan komponen atom Button sebagai bagian dari dirinya.

Gunakan komponen-komponen tersebut dalam komponen organisme, template, dan halaman sesuai kebutuhan aplikasi Anda.

Penggunaan Atomic Design memungkinkan Anda untuk menyusun komponen React dengan cara yang terstruktur dan mudah dikelola. Dengan pemisahan yang jelas antara atom, molekul, organisme, template, dan halaman, Anda dapat membangun antarmuka pengguna yang konsisten, modular, dan reaktif dengan lebih efisien.

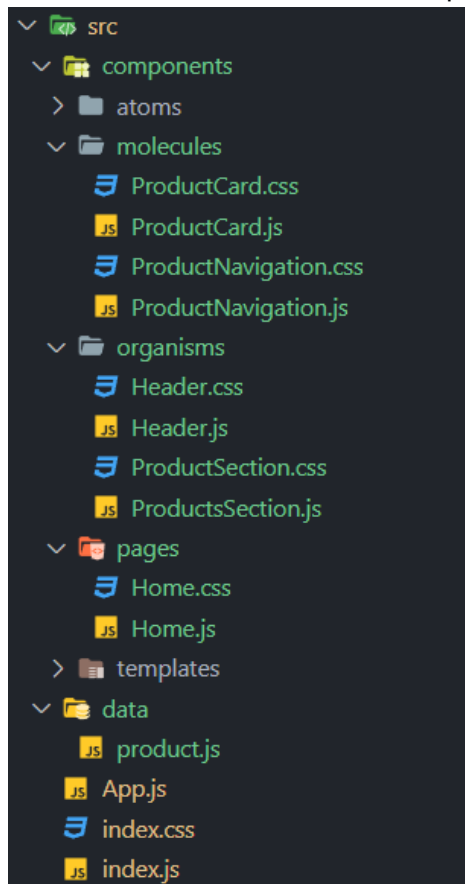
SOLUTION

Setelah memahami konsep komponen, jsx, styling dan penyusunan komponen, berikutnya kita akan mulai mengerjakan E-Commerce Mini Toko XYZ.

INSTRUCTION

Silahkan ikuti langkah-langkah berikut untuk menyelesaikan permasalahan yang kalian buat:

1. Buatlah project react dengan menggunakan create-react-app sesuai yang diajarkan pada pertemuan 1.
2. Buka project react menggunakan vscode dan jalankan Aplikasi React.
3. Perhatikan penjelasan dosen untuk struktur web app react.
4. Setelah semua siap, implementasikan struktur folder, dan file berikut :
Buat struktur folder dan file berikut pada project anda :



5. Implementasikan css berikut ke tiap file css yang tersedia :

index.css

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Sofia&display=swap");

html {
  scroll-behavior: smooth;
}
```

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}

body {
  font-family: "Poppins", sans-serif;
  color: rgb(41, 41, 41);
  height: 100vh;
}
```

Home.css

```
main {
  max-width: 1366px;
  width: 100%;
  margin: 0 auto;
  padding-top: 60px;
}
```

Header.css

```
header {
  background-color: red;
  display: flex;
  justify-content: center;
  box-shadow: rgba(64, 62, 62, 0.24) 0px 3px 8px;
  background-color: #fff;
  height: 60px;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
}

.inner-header {
  display: flex;
  background-color: #fff;
  max-width: 1366px;
  align-items: center;
  justify-content: space-between;
  width: 100%;
}

.inner-header .logo {
  font-size: 28px;
  cursor: pointer;
}
```

ProductNavigation.css

```
nav ul {
  list-style: none;
  display: flex;
  align-items: center;
  gap: 28px;
}

nav ul li a {
  text-decoration: none;
  font-size: 16px;
  font-weight: 600;
  color: rgb(36, 36, 36);
}

nav ul li a:hover {
  border-bottom: 2px solid rgb(36, 36, 36);
}
```

ProductSection.css

```
#products .title-section {
  text-align: center;
  margin: 48px 0;
  font-family: "Sofia", cursive;
  font-size: 40px;
}

.product-container {
  display: grid;
  grid-template-columns: auto auto auto auto auto;
  gap: 20px;
}
```

ProductCard.css

```
.product-card {
  background-color: #fff;
  box-shadow: rgba(0, 0, 0, 0.16) 0px 1px 4px;
  border-radius: 8px;
  overflow: hidden;
  padding-bottom: 8px;
  cursor: pointer;
}

.product-card img {
  height: 280px;
  width: 100%;
  min-width: 200px;
  object-fit: cover;
}
```

```

}

.product-card .title {
  font-size: 20px;
}

.product-card .category {
  color: rgb(67, 66, 66);
  font-size: 16px;
}

.product-card .title,
.product-card .description,
.product-card .category {
  padding: 0 16px;
  margin-top: 8px;
  overflow: hidden;
}

```

6. Implementasikan code berikut pada tiap file komponen react yang ada :

index.js

```

1  import React from "react";
2  import ReactDOM from "react-dom/client";
3  import App from "./App";
4  import "./index.css";
5
6  const root = ReactDOM.createRoot(document.getElementById("root"));
7  root.render(<App />);
8

```

App.js

```

1  import React from "react";
2  import Home from "./components/pages/Home";
3
4  const App = () => {
5    return <Home />;
6  };
7
8  export default App;
9

```

Home.js

```
1 import React from "react";
2 import Header from "../organisms/Header";
3 import ProductsSection from "../organisms/ProductsSection";
4 import "./Home.css";
5
6 const Home = () => {
7   return (
8     <>
9       <Header />
10      <main>
11        <ProductsSection />
12      </main>
13    </>
14  );
15 };
16
17 export default Home;
18
```

Header.js

```
1 import React from "react";
2 import ProductNavigation from "../molecules/ProductNavigation";
3 import "./Header.css";
4
5 const Header = () => {
6   return (
7     <header>
8       <div className="inner-header">
9         <h1 className="logo">
10           XYZ<span>.</span>
11         </h1>
12         <ProductNavigation />
13       </div>
14     </header>
15   );
16 };
17
18 export default Header;
19
```

ProductSection.js

```
1 import React from "react";
2 import ProductCard from "../molecules/ProductCard";
3 import allProducts from "../data/product";
4 import "../ProductSection.css";
5
6 const ProductsSection = () => {
7   return (
8     <section id="products">
9       <h2 className="title-section">-- All Products --</h2>
10      <div className="product-container">
11        {allProducts.map((product) => {
12          return (
13            <ProductCard
14              key={product.id}
15              cover="not-available.jpeg"
16              title={product.nama_product}
17              description={product.deskripsi}
18              category={product.kategori}
19            />
20          );
21        })}
22      </div>
23    </section>
24  );
25 };
26
27 export default ProductsSection;
28
```

ProductNavigation.js



```
1  import React from "react";
2  import "../ProductNavigation.css";
3
4  const ProductNavigation = () => {
5    return (
6      <nav>
7        <ul>
8          <li>
9            <a href="#products">All Products</a>
10          </li>
11          <li>
12            <a href="#">Computers</a>
13          </li>
14          <li>
15            <a href="#">Laptops</a>
16          </li>
17          <li>
18            <a href="#">Smartphones</a>
19          </li>
20        </ul>
21      </nav>
22    );
23  };
24
25  export default ProductNavigation;
```


ProductCard.js

```
1 import React from "react";
2 import "../ProductCard.css";
3
4 const ProductCard = (props) => {
5   return (
6     <div className="product-card">
7       <img src={props.cover ?? "not-available.jpeg"} alt="product" />
8       <h4 className="title">{props.title}</h4>
9       <h5 className="category">{props.category}</h5>
10      <p className="description">{props.description}</p>
11    </div>
12  );
13 };
14
15 export default ProductCard;
```

7. Untuk data product bisa di ambil dari sini [data products](#)
8. Lihat hasil source code di atas dengan memastikan koding di atas telah **disimpan** dan aplikasi react telah dijalankan dengan **npm start**

EXERCISE

EXERCISE OBJECTIVES

Pada latihan ini, mahasiswa diharapkan mampu untuk:

- Membuat website sederhana dengan memanfaatkan fitur Komponen, JSX, Styling dasar dan Penyusunan Komponen pada React.

TASK 1: ANDA DIMINTA UNTUK MENGHAPUS NAVIGASI ALL PRODUCTS DAN SECTION ALL PRODUCTS, KEMUDIAN MENAMBAHKAN 3 SECTION BARU UNTUK TIAP KATEGORI PRODUK, TIAP SECTION HANYA MENAMPILKAN KATEGORI PRODUK YANG SESUAI DENGAN KATEGORINYA, KETIKA NAVIGASI PADA HEADER DI KLIK WEBSITE SECARA OTOMATIS BERGULIR KE ARAH SECTION YANG DI PILIH. PADA BAGIAN INI ANDA BELAJAR:

- Mapping Over Arrays: Cara menggunakan metode map untuk membuat daftar elemen JSX dari array, yang merupakan teknik umum untuk menampilkan daftar item di React.
- React Components: Konsep dasar dari komponen React dan bagaimana membuat komponen fungsional menggunakan sintaks fungsi.
- Styling dasar : Cara memberikan styling pada komponen React dengan menggunakan CSS standar.
- Atomic Design : Cara menyusun komponen react agar komponen bisa reusable dan mudah di maintenance


Silahkan lengkapi koding di atas dan buat perubahannya.

Hasil Akhir :


XYZ.

ComputersLaptopsSmartphones


Computers




PC Gaming Alienware komputer
Komputer gaming Alienware dengan kartu grafis NVIDIA.




iMac 27-inch komputer
Komputer iMac dengan layar retina 5K.




Asus ROG Strix komputer
Komputer gaming Asus ROG dengan prosesor intel Core i9.




Custom Gaming PC komputer
Komputer gaming kustom dengan berbagai pilihan komponen.




Custom Workstation PC komputer
Komputer workstation kustom untuk tugas-tugas berat.




Custom Home Theater PC komputer
Komputer khusus untuk hiburan di rumah.




HP OMEN Gaming PC komputer
Komputer gaming HP OMEN dengan RGB lighting.



Alienware Aurora Gaming PC komputer
Komputer gaming Alienware Aurora dengan prosesor AMD Ryzen.




MSI Gaming Desktop komputer
Komputer gaming MSI dengan kartu grafis NVIDIA RTX.




Custom Streaming PC komputer
Komputer khusus untuk streaming dan konten kreatif.


Laptops




Dell XPS 13 laptop
Laptop Dell dengan prosesor intel Core i7.




MacBook Air laptop
Laptop MacBook ringan dan tipis dengan M1 chips.




HP Pavilion laptop
Laptop HP dengan layar sentuh dan AMD Ryzen.




Lenovo ThinkPad X1 Carbon laptop
Laptop Lenovo dengan layar 4K dan keyboard ergonomis.




Acer Predator Helios 300 laptop
Laptop gaming Acer Predator dengan NVIDIA GeForce GTX.




HP Envy x360 laptop
Laptop HP Envy dengan layar sentuh dan AMD Ryzen 7.




Microsoft Surface Laptop 4 laptop
Laptop Microsoft dengan layar PixelSense dan pilihan intel/AMD.




Lenovo Yoga C940 laptop
Laptop Lenovo Yoga dengan layar sentuh 2-in-1.



Asus ZenBook Pro Duo laptop
Laptop Asus ZenBook Pro Duo dengan layar OLED sekunder.




Apple MacBook Pro 16-inch laptop
Laptop Apple MacBook Pro dengan layar besar dan performa tinggi.




Acer Aspire 5 laptop
Laptop Acer Aspire dengan harga terjangkau dan performa baik.


Smartphones




Samsung Galaxy S21 handphone
Handphone Samsung terbaru dengan kamera canggih.




iPhone 13 Pro handphone
Handphone iPhone terbaru dengan layar OLED.




Google Pixel 6 handphone
Handphone Google Pixel dengan kamera AI.




OnePlus 9 handphone
Handphone OnePlus dengan layar Fluid AMOLED.




Xiaomi Mi 11 handphone
Handphone Xiaomi Mi dengan prosesor Snapdragon.




Sony Xperia 1 III handphone
Handphone Sony Xperia dengan layar 4K HDR OLED.



Google Pixel 5a handphone
Handphone Google Pixel 5a dengan kamera kualitas tinggi.



OnePlus 9 Pro handphone
Handphone OnePlus 9 Pro dengan kamera Hasselblad.



Xiaomi Redmi Note 10 handphone
Handphone Xiaomi Redmi dengan baterai tahan lama.



UNIVERSITAS
MIKROSKIL