

React Styling

[CSS Inline](#)

[CSS dalam File Terpisah \(CSS External\)](#)

[Buat file CSS terpisah](#)

[CSS-in-JS](#)

[CSS Modules](#)

[CSS Layout](#)

[Latihan](#)

Terdapat beberapa cara untuk melakukan styling pada komponen React. Dalam tutorial ini, kita akan mempelajari cara umum yang digunakan dalam styling komponen React: CSS inline, CSS dalam file terpisah, dan penggunaan library CSS-in-JS. Kita dapat memilih cara yang paling sesuai dengan kebutuhan proyek kita.

CSS Inline

Penggunaan CSS inline cocok untuk kasus styling yang sederhana. Kita dapat menggunakan prop `style` pada elemen React untuk menentukan gaya inline. Berikut adalah contoh penggunaannya:

```
import React from 'react';
const MyComponent = () => {
  const inlineStyle = {
    backgroundColor: 'lightblue',
    color: 'white',
    padding: '10px',
  };

  return (
    <div style={inlineStyle}>
      Ini adalah komponen dengan CSS inline.
    </div>
  );
}

export default MyComponent;
```

CSS dalam File Terpisah (CSS External)

Cara lain untuk melakukan styling adalah dengan membuat file CSS terpisah dan menghubungkannya dengan komponen React. Berikut adalah langkah-langkahnya:

Buat file CSS terpisah

Misalnya, kita membuat file dengan nama `MyComponent.css` :

```
/* MyComponent.css */
.my-component {
  background-color: lightblue;
  color: white;
  padding: 10px;
}
```

Impor file CSS ke komponen React

```
import React from 'react';
import './MyComponent.css'; // Impor file CSS

const MyComponent = () => {
  return (
    <div className="my-component">
      Ini adalah komponen dengan CSS dalam file terpisah.
    </div>
  );
}

export default MyComponent;
```

CSS-in-JS

Library CSS-in-JS seperti "styled-components" memungkinkan kita untuk menulis CSS menggunakan JavaScript. Ini memungkinkan komponen React untuk memiliki gaya yang terkait dengannya. Berikut adalah contoh penggunaan "styled-components":

a. Instalasi "styled-components"

Jika kita belum menginstalnya, kita bisa melakukannya dengan perintah:

```
npm install styled-components
```

b. Gunakan "styled-components" dalam komponen

```
import React from 'react';
import styled from 'styled-components';

const StyledDiv = styled.div`
  background-color: lightblue;
  color: white;
  padding: 10px;
`;

const MyComponent = () => {
  return (
    <StyledDiv>
      Ini adalah komponen dengan "styled-components".
    </StyledDiv>
  );
}

export default MyComponent;
```

CSS Modules

CSS Modules adalah pendekatan dalam pengembangan web yang memungkinkan kita untuk membuat style CSS lokal untuk setiap komponen React secara terpisah. Ini membantu menghindari masalah tumpang tindih style antar komponen dan membuat kode CSS lebih modular dan mudah diatur. Setiap komponen memiliki lingkup sendiri untuk style CSS, yang berarti kita dapat menggunakan nama kelas yang sama di berbagai komponen tanpa khawatir dengan adanya konflik.

Berikut adalah contoh penggunaan CSS Modules dalam React:

1. Instalasi dan Konfigurasi

Pastikan kita telah menginstal paket CSS Modules. Kita dapat melakukannya dengan perintah:

```
npm install --save node-sass
```

Pada umumnya, kita juga harus mengonfigurasi proyek untuk mengenali file dengan ekstensi `.module.css` atau `.module.scss` sebagai file CSS Modules. Ini dapat dilakukan dengan menggunakan Create React App atau konfigurasi manual di proyek Anda.

2. Membuat file CSS Module

Buat file CSS Module, misalnya `MyComponent.module.css`. Ini adalah contoh isi dari file tersebut:

```
/* MyComponent.module.css */
.my-component {
  background-color: lightblue;
  color: white;
  padding: 10px;
}
```

3. Menggunakan CSS Module dalam Komponen React

Dalam komponen React, kita dapat mengimpor dan menggunakan kelas dari CSS Module tersebut. Berikut contoh komponen React yang menggunakan CSS Module:

```
import React from 'react';
import styles from './MyComponent.module.css'; // Import file CSS Module

function MyComponent() {
  return (
    <div className={styles['my-component']}>
      Ini adalah komponen dengan CSS Modules.
    </div>
  );
}

export default MyComponent;
```

Perhatikan bahwa kita menggunakan objek `styles` yang mengandung kelas CSS sebagai properti. Kita menggunakan notasi `styles['my-component']` untuk merujuk pada kelas CSS yang didefinisikan dalam file CSS Module.

CSS Layout

[https://www.figma.com/file/vMntCVVy5d2CUksNruYzWA/flexbox-\(Community\)?type=design&node-id=0-1&mode=design&t=AKSaJEXdMfsJzsFc-0](https://www.figma.com/file/vMntCVVy5d2CUksNruYzWA/flexbox-(Community)?type=design&node-id=0-1&mode=design&t=AKSaJEXdMfsJzsFc-0)

Ada beberapa cara dalam melakukan layouting pada CSS, beberapa diantaranya adalah normal flow (browser yang mengandalkan HTML sebagai default dalam mengontrol layout dari suatu page), display property (inline, block, inline-block, dll), flexbox, grid layout, float, dll. Pada kali ini kita akan membahas sedikit bagaimana flexbox bekerja. Flexbox adalah model tata letak yang dirancang untuk mengatur elemen dalam satu dimensi (entah itu horizontal atau vertikal) terutama untuk mengelola tata letak item dalam satu baris atau satu kolom. Flexbox memperlakukan satu dimensi sebagai sumbu utama (main axis) dan yang lainnya sebagai sumbu lintang (cross axis). Kita dapat mengontrol distribusi, penyesuaian ukuran, dan perataan elemen dalam sumbu utama.

Latihan

- Case study: **spotify — sidebar**
- Task
 - Mengidentifikasi komponen yang dibutuhkan
 - Layouting
 - Styling

<https://www.figma.com/file/Ws78LuySZlxNYAp6VObWC6/Spotify-Web-UI-Community?fuid=1277638794361508518>