

CSS Frameworks: Tailwind CSS

CSS Frameworks

- Koleksi aturan style yang telah dibuat sebelumnya.
- Membantu mempercepat pengembangan web dan memastikan tampilan yang konsisten di berbagai peramban.
- Dapat digunakan untuk membangun halaman web yang responsif dan menarik dalam waktu singkat.
- Beberapa CSS Frameworks juga menyediakan komponen JavaScript yang dapat digunakan untuk membuat interaksi yang lebih kompleks.

Apa itu Tailwind CSS?

Tailwind CSS adalah framework CSS utility-first yang sangat dapat dikustomisasi, memungkinkan pengembang untuk membangun desain dengan cepat tanpa menulis CSS dari awal.

- **Utility-First:** Menyediakan utility class untuk styling cepat.
- **Responsif:** Desain responsif dengan mudah menggunakan kelas bawaan.
- **Customizable:** Mudah disesuaikan sesuai kebutuhan proyek.

Instalasi Tailwind CSS

```
npm init vite@latest vite-app  
cd vite-app  
npm install  
  
npm install -D tailwindcss postcss autoprefixer  
npx tailwindcss init -p
```

Prasyarat: project telah diinisialisasi dengan Vite.

PostCSS digunakan untuk mengkompilasi utility class menjadi CSS.

Autoprefixer digunakan untuk menambahkan vendor prefix pada CSS.

Install extension Tailwind CSS IntelliSense untuk VSCode.

Instalasi Tailwind CSS

```
/* ./styles.css */  
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

@tailwind base : Menambahkan aturan CSS dasar.

@tailwind components : Menambahkan aturan CSS untuk komponen.

@tailwind utilities : Menambahkan utility class.

File Konfigurasi Tailwind CSS

```
/** @type {import('tailwindcss').Config} */  
export default {  
  content: ['./index.html'],  
  theme: {  
    extend: {},  
  },  
  plugins: [],  
}
```

File Konfigurasi Tailwind CSS

- Properti `content` adalah array yang menentukan file mana yang akan Tailwind cari untuk nama kelas.
- Ketika Anda membangun CSS Anda, Tailwind akan menggunakan konfigurasi ini untuk menentukan kelas mana yang harus disertakan dalam file CSS akhir.
- Ini adalah bagian dari mode "Just-In-Time" (JIT) Tailwind, yang dapat secara signifikan mengurangi ukuran file CSS akhir Anda dengan hanya menyertakan kelas yang benar-benar digunakan.

File Konfigurasi Tailwind CSS

- Properti `theme` adalah objek yang memungkinkan Anda untuk menyesuaikan konfigurasi default Tailwind.
- Properti `extend` di dalam `theme` digunakan untuk meng-extend konfigurasi default berdasarkan properti.
- Properti `plugins` adalah array di mana Anda dapat menyertakan plugin Tailwind CSS apa pun yang ingin Anda gunakan.
- Plugin dapat digunakan untuk meng-extend set default utility class Tailwind, menambahkan fitur CSS baru, atau membuat penyesuaian lainnya.

Core Concepts Tailwind CSS

Pendekatan Utility-First

Tailwind CSS dibangun di atas paradigma utility-first, yang berarti Anda mendesain elemen menggunakan utility class yang melayani tujuan desain spesifik. Pendekatan ini mendorong komposisi desain langsung dalam markup Anda daripada menulis CSS custom.

Desain Responsif

Tailwind menggunakan pendekatan mobile-first dengan varian utility responsif. Anda dapat menerapkan style yang berbeda pada berbagai break point dengan menambahkan modifikasi responsif seperti `md:`, `lg:`.

Core Concepts Tailwind CSS

PurgeCSS untuk Produksi

Karena Tailwind menghasilkan banyak utility class, ini terintegrasi dengan PurgeCSS untuk menghapus style yang tidak digunakan untuk build produksi, secara signifikan mengurangi ukuran berkas akhir.

Mode Just-In-Time (JIT)

Mode JIT Tailwind mengkompilasi CSS Anda sesuai permintaan saat Anda membuat template Anda, membuat style hanya untuk utility yang Anda gunakan. Ini menghasilkan waktu build yang lebih cepat dan ukuran berkas yang lebih kecil.

Core Concepts Tailwind CSS

Kustomisasi dan Konfigurasi

Tailwind sangat bisa disesuaikan melalui berkas konfigurasi (`tailwind.config.js`). Anda dapat mendefinisikan palet warna, spasi, font, break point, dan lainnya, menyesuaikan kerangka kerja sesuai kebutuhan proyek Anda.

Menggabungkan Utility Class dengan Direktif `@apply`

Tailwind memungkinkan penggunaan direktif `@apply` untuk menggabungkan beberapa utility class menjadi kelas CSS custom di stylesheet Anda, menggabungkan kekuatan CSS tradisional dengan fleksibilitas utility-first.

Sistem Plugin

Tailwind mendukung sistem plugin yang kuat, memungkinkan Anda menambahkan custom utility, komponen, atau bahkan plugin pihak ketiga untuk meng-extend kemampuan kerangka kerja.

Utility Class yang Sering Digunakan

Spacing:

- `p-{ukuran}` : Padding. Ukuran berkisar dari `p-0` (tanpa padding) hingga `p-64` (padding besar), termasuk ukuran pecahan seperti `p-0.5`.
- `m-{ukuran}` : Margin. Serupa dengan padding, margin berkisar dari `m-0` hingga `m-64`.
- `{arah}-{jenis}-{ukuran}` : Untuk arah tertentu, seperti `mt-4` (margin atas), `pr-2` (padding kanan).

Utility Class yang Sering Digunakan

```
<!-- All sides padding -->
<div class="p-4">...</div>
<!-- Padding top -->
<div class="pt-4">...</div>
<!-- Padding left and right -->
<div class="px-4">...</div>

<!-- All sides margin -->
<div class="m-4">...</div>
<!-- Margin bottom -->
<div class="mb-4">...</div>
<!-- Horizontal centering -->
<div class="mx-auto">...</div>
```

Utility Class yang Sering Digunakan

Tipografi (Typography):

- `text-{ukuran}` : Ukuran font. Contoh termasuk `text-sm`, `text-lg`, `text-xl`.
- `font-{berat}` : Berat font, seperti `font-bold`, `font-light`.
- `text-{warna}` : Warna teks. Gunakan sistem warna Tailwind, seperti `text-gray-700`, `text-blue-500`.

```
<p class="text-lg">...</p>  
<p class="font-bold">...</p>  
<p class="text-blue-500">...</p>
```

Utility Class yang Sering Digunakan

Latar Belakang (Backgrounds):

- `bg-{warna}` : Warna latar belakang, serupa dengan warna teks, mis. `bg-red-500`, `bg-green-300`.
- `bg-opacity-{persentase}` : Kekeruhan latar belakang, seperti `bg-opacity-50` untuk 50% opacity.

```
<!-- Background color -->  
<div class="bg-red-500">...</div>  
<!-- Background opacity -->  
<div class="bg-opacity-50">...</div>
```

Utility Class yang Sering Digunakan

Tata Letak (Layout):

- `flex` , `inline-flex` : Tata letak Flexbox.
- `grid` : Tata letak CSS Grid.
- `container` : Memusatkan konten dan mengatur lebar maksimum berdasarkan ukuran layar.

```
<!-- Flex container -->  
<div class="flex">...</div>  
<!-- Grid container -->  
<div class="grid">...</div>  
<!-- Centered container -->  
<div class="container">...</div>
```


Utility Class yang Sering Digunakan

Utility Class Flexbox dan Grid:

- `justify-{nilai}`, `items-{nilai}` : Untuk menyelaraskan item flex atau grid, seperti `justify-center`, `items-end`.
- `gap-{ukuran}` : Spacing antara item grid atau flex.

```
<!-- Flexbox alignment -->  
<div class="flex justify-center items-end">...</div>  
<!-- Grid gap -->  
<div class="grid gap-4">...</div>
```

Utility Class yang Sering Digunakan

Kontainer Flex Dasar:

```
<div class="flex justify-center items-center">  
  <div>Item 1</div>  
  <div>Item 2</div>  
</div>
```

Arah Flex:

```
<div class="flex flex-col">  
  <div>Item 1</div>  
  <div>Item 2</div>  
</div>
```

Utility Class yang Sering Digunakan

Membungkus Item Flex:

```
<div class="flex flex-wrap">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>
```

Utility Class yang Sering Digunakan

Tata Letak Grid Dasar:

```
<div class="grid grid-cols-3 gap-4">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>
```

Kolom Grid Responsif:

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3">  
  <div>Item 1</div>  
  <div>Item 2</div>  
  <div>Item 3</div>  
</div>
```

Utility Class yang Sering Digunakan

Area Template Grid:

```
<div  
  class="grid grid-areas['header header header' 'sidebar content content' 'footer footer footer']"  
>  
  <div class="area-header">Header</div>  
  <div class="area-sidebar">Sidebar</div>  
  <div class="area-content">Konten</div>  
  <div class="area-footer">Footer</div>  
</div>
```

Utility Class Flexbox

- `flex` : Mendefinisikan kontainer flex.
- `flex-row`, `flex-row-reverse` : Menetapkan arah flex horizontal.
- `flex-col`, `flex-col-reverse` : Menetapkan arah flex vertikal.
- `flex-wrap`, `flex-wrap-reverse`, `flex-nowrap` : Mengontrol pembungkusan item flex.
- `justify-{start|end|center|between|around|evenly}` : Penyelarasan horizontal item flex.
- `items-{start|end|center|baseline|stretch}` : Penyelarasan vertikal item flex.
- `space-x-{ukuran}`, `space-y-{ukuran}` : Mengontrol penjarakan antara item flex.

Utility Class Grid

- `grid` : Mendefinisikan kontainer grid.
- `grid-cols-{nomor}` : Menentukan jumlah kolom dalam grid.
- `grid-rows-{nomor}` : Menentukan jumlah baris dalam grid.
- `grid-flow-{row|col}` : Mengontrol arah item diletakkan dalam grid.
- `gap-{ukuran}` , `gap-x-{ukuran}` , `gap-y-{ukuran}` : Mengontrol penjarakan antara item grid.
- `grid-auto-cols-{auto|minmax|mincontent|maxcontent}` , `grid-auto-rows-{...}` : Menentukan ukuran baris dan kolom grid yang dibuat secara implisit.

Utility Class yang Sering Digunakan

Borders:

- `border`, `border-{ukuran}` : Lebar border, `border-2` untuk border 2px.
- `border-{warna}` : Warna border, seperti `border-blue-500` .
- `rounded-{ukuran}` : Radius border untuk sudut bulat, seperti `rounded-lg` .

```
<!-- Red border -->
<div class="border border-red-500">...</div>
<!-- Blue border -->
<div class="border border-blue-300">...</div>
<!-- Border width and color -->
<div class="border border-2 border-blue-500">...</div>
```


Utility Class yang Sering Digunakan

```
<!-- Small rounded corners -->  
<div class="rounded">...</div>  
<!-- Larger rounded corners -->  
<div class="rounded-lg">...</div>
```

```
<!-- Top corners rounded -->  
<div class="rounded-t-lg">...</div>  
<!-- Bottom right corner rounded -->  
<div class="rounded-br-xl">...</div>
```

```
<!-- Circular image -->  

```

```
<!-- No rounded corners -->  
<div class="rounded-none">...</div>
```

Utility Class yang Sering Digunakan

Lebar dan Tinggi (Width and Height):

- `w-{ukuran}` dan `h-{ukuran}` : Menetapkan lebar dan tinggi, dengan nilai seperti `w-1/2` (lebar 50%), `h-full` (tinggi 100%).

```
<!-- Width and height -->  
<div class="w-1/2 h-full">...</div>
```

Utility Class yang Sering Digunakan

Positioning:

- `absolute`, `relative`, `fixed` : Utility Class penempatan.
- `top-{ukuran}`, `right-{ukuran}` : Menempatkan elemen relatif terhadap blok pembungkusnya.

```
<!-- Relative positioning -->  
<div class="relative">...</div>  
<!-- Absolute positioning -->  
<div class="absolute top-0 right-0">...</div>
```

Utility Class yang Sering Digunakan

Visibilitas dan Tampilan (Visibility and Display):

- `hidden`, `block`, `inline-block` : Mengontrol tampilan elemen.
- `opacity-{persentase}` : Utility Class opacity, seperti `opacity-75` .

```
<!-- Hide element -->
<div class="hidden">...</div>
<!-- Display block -->
<div class="block">...</div>
<!-- Opacity -->
<div class="opacity-75">...</div>
```

Utility Class yang Sering Digunakan

Desain Responsif (Responsive Design):

- Menambahkan kelas dengan prefiks `sm:`, `md:`, `lg:`, `xl:`, `2xl:` untuk desain responsif, seperti `md:text-left`, `lg:p-8`.

```
<!-- Text alignment changes on medium devices -->  
<div class="text-center md:text-left">...</div>  
<!-- Padding changes on large devices -->  
<div class="p-4 lg:p-8">...</div>
```

Palet Warna Default

Tailwind CSS menyediakan rangkaian palet warna default yang dapat Anda gunakan langsung dalam kelas Anda. Setiap warna di Tailwind hadir dalam berbagai nuansa, biasanya mulai dari **50** (terang) hingga **900** (gelap).

Palet Warna Default

1. **Gray:** dari `gray-50` hingga `gray-900`
2. **Red:** dari `red-50` hingga `red-900`
3. **Yellow:** dari `yellow-50` hingga `yellow-900`
4. **Green:** dari `green-50` hingga `green-900`
5. **Blue:** dari `blue-50` hingga `blue-900`
6. **Indigo:** dari `indigo-50` hingga `indigo-900`
7. **Purple:** dari `purple-50` hingga `purple-900`
8. **Pink:** dari `pink-50` hingga `pink-900`

Palet Warna Default

Selain itu, Tailwind juga mencakup warna lain:

- **Hitam (Black):** Direpresentasikan sebagai `black` .
- **Putih (White):** Direpresentasikan sebagai `white` .
- **Blue Gray, Cool Gray, True Gray, Warm Gray:** Ini adalah variasi dari tone abu-abu.
- **Orange, Amber, Lime, Emerald, Teal, Cyan, Light Blue, Violet, Fuchsia, Rose:** Ini menawarkan nuansa tambahan.

Ukuran Font

- `text-xs` : Ukuran teks ekstra kecil.
- `text-sm` : Ukuran teks kecil.
- `text-base` : Ukuran teks dasar (biasanya ukuran font default).
- `text-lg` : Ukuran teks besar.
- `text-xl` : Ukuran teks ekstra besar.
- `text-2xl` : 2 ukuran teks besar.
- `text-3xl` : 3 ukuran teks besar.
- `text-4xl` : 4 ukuran teks besar.
- `text-5xl` : 5 ukuran teks besar.

Berat Font

- `font-thin` : Berat font 100.
- `font-extralight` : Berat font 200.
- `font-light` : Berat font 300.
- `font-normal` : Berat font 400 (berat font default).
- `font-medium` : Berat font 500.
- `font-semibold` : Berat font 600.
- `font-bold` : Berat font 700.
- `font-extrabold` : Berat font 800.
- `font-black` : Berat font 900.

Mengkonfigurasi Tailwind CSS

Berkas `tailwind.config.js` dalam Tailwind CSS adalah berkas konfigurasi sentral di mana Anda dapat menyesuaikan dan meng-extend Tailwind untuk memenuhi kebutuhan Anda.

Mengkonfigurasi Tailwind CSS

- **theme** : adalah tempat Anda mendefinisikan sistem desain Anda. Ini termasuk mengatur warna, font, breakpoint, dan lainnya.
- **variants** : mengontrol varian (seperti hover, focus, active, dll.) yang dihasilkan untuk setiap utilitas.
- **plugins** : memungkinkan Anda menambahkan plugin pihak ketiga atau plugin custom.
- **content** : digunakan untuk menentukan lokasi template dan komponen sehingga Tailwind dapat menghilangkan style yang tidak digunakan dalam build produksi.

Mengkonfigurasi Tailwind CSS

Atribut Kunci di Bagian `theme`

- `screens` : Tentukan breakpoint custom untuk desain responsif.
- `colors` : Sesuaikan palet warna dengan pilihan warna Anda sendiri.
- `spacing` : Tetapkan nilai custom untuk margin, padding, gap, dll.
- `fontFamily` : Tentukan tumpukan font Anda.
- `fontSize` : Tetapkan ukuran font custom.
- `fontWeight` : Sesuaikan opsi berat font.
- `lineHeight` : Tentukan nilai tinggi baris.

Mengkonfigurasi Tailwind CSS

Atribut Kunci di Bagian `theme`

- `letterSpacing` : Sesuaikan skala spasi huruf.
- `borderWidth` : Tetapkan lebar perbatasan custom.
- `borderColor` : Sesuaikan warna perbatasan.
- `borderRadius` : Tentukan ukuran radius perbatasan untuk sudut bulat.
- `width` , `height` : Tetapkan lebar dan tinggi custom.
- `minWidth` , `minHeight` , `maxWidth` , `maxHeight` : Tentukan dimensi minimum dan maksimum.
- `extend` : Extend theme default Tailwind tanpa menyimpannya.

Mengkonfigurasi Tailwind CSS

Extending Theme:

- Sesuaikan default theme atau extend dengan nilai baru.
- Contoh: meng-extend palet warna.

```
module.exports = {  
  theme: {  
    extend: {  
      colors: {  
        'custom-blue': '#4B9CD3',  
      },  
    },  
  },  
}
```

Mengkonfigurasi Tailwind CSS

Breakpoints:

- Tentukan breakpoint custom untuk desain responsif.
- Contoh: Menambahkan breakpoint baru.

```
module.exports = {  
  theme: {  
    extend: {  
      screens: {  
        '2xl': '1440px',  
      },  
    },  
  },  
}
```


Mengkonfigurasi Tailwind CSS

Font custom:

- Tentukan font custom.
- Contoh: Menambahkan font Google.

```
module.exports = {  
  theme: {  
    extend: {  
      fontFamily: {  
        sans: ['Inter', 'sans-serif'],  
      },  
    },  
  },  
}
```

Mengkonfigurasi Tailwind CSS

Varian:

- Kontrol varian mana yang dihasilkan untuk setiap utility.
- Contoh: Mengaktifkan varian `focus` untuk warna latar belakang.

```
module.exports = {  
  variants: {  
    extend: {  
      backgroundColor: ['focus'],  
    },  
  },  
}
```

Mengkonfigurasi Tailwind CSS

Plugin:

```
const plugin = require('tailwindcss/plugin')

module.exports = {
  plugins: [
    plugin(function ({ addUtilities }) {
      const newUtilities = {
        '.custom-utilities': {
          'background-color': '#4B9CD3',
        },
      },
      addUtilities(newUtilities)
    }),
  ],
}
```

Mengkonfigurasi Tailwind CSS

Direktif `@apply` dalam Tailwind CSS adalah fitur yang memungkinkan Anda menerapkan kelas utilitas pada kelas CSS kustom dalam stylesheet Anda.

Hal ini membantu dalam menggabungkan beberapa kelas utilitas menjadi satu kelas kustom, membuat HTML Anda lebih rapi dan lebih mudah dikelola, terutama ketika kombinasi yang sama dari kelas utilitas digunakan berulang kali di seluruh proyek Anda.

Cara Kerja `@apply`

1. **Mendefinisikan Kelas Kustom:** Anda membuat kelas CSS kustom dalam stylesheet Anda (biasanya dalam berkas CSS global).
2. **Menerapkan Utilitas Tailwind:** Dalam kelas kustom ini, Anda menggunakan direktif `@apply` untuk menerapkan kelas utilitas Tailwind. Ini secara efektif "menyuntikkan" gaya dari kelas utilitas ke dalam kelas kustom Anda.
3. **Menggunakan Kelas Kustom di HTML:** Anda kemudian dapat menggunakan kelas kustom ini di HTML Anda, memanfaatkan gaya gabungan dari utilitas yang diterapkan.

Contoh

Misalkan Anda sering menggunakan kombinasi kelas utilitas Tailwind untuk tombol - seperti `bg-blue-500`, `text-white`, `py-2`, `px-4`, dan `rounded`. Alih-alih mengulangi utilitas ini di setiap tombol, Anda dapat membuat kelas tombol kustom:

Di berkas CSS Anda:

```
.btn-blue {  
  @apply bg-blue-500 text-white py-2 px-4 rounded;  
}
```

Kemudian, di HTML Anda:

```
<button class="btn-blue">Klik saya</button>
```

Hal yang Perlu Diperhatikan

- **Gunakan Secara Hemat:** Penggunaan berlebihan dapat menyebabkan berkas CSS yang besar. Ini terbaik digunakan untuk pola yang sering diulang.
- **Tidak Dinamis:** Tidak seperti menggunakan utilitas langsung di HTML, perubahan pada kelas `@apply` memerlukan rekompilasi CSS Anda.
- **Keterbatasan:** `@apply` tidak dapat digunakan dengan kelas kondisional atau varian (seperti `hover:` atau `md:`). Ini masih perlu diterapkan langsung di HTML.

Singkatnya, `@apply` adalah direktif yang bermanfaat untuk mengelompokkan kelas utilitas Tailwind ke dalam kelas CSS kustom, sehingga menyederhanakan alur kerja Anda dan menjaga HTML Anda tetap rapi.