LAPORAN RESMI

PRAKTIKUM DESAN WEB

**MODUL**

**TAILWIND CSS**



**NIM : 21104410066**

**NAMA : Muhammad Fahmi Mujib**

**JURUSAN : Teknik Informatika**

**KELAS : TI 3B**

**TGL. PRAKTEK : Senin, 16 Januari 2023**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ISLAM BALITAR**

**2023**

LEMBAR PERSETUJUAN

**MODUL**

**TAILWIND CSS**

**NIM : 21104410066**

**NAMA : Muhammad Fahmi Mujib**

**JURUSAN : Teknik Informatika**

**KELAS : TI 3B**

**TGL. PRAKTEK : Senin, 16 Januari 2023**

**Disetujui,**

**Blitar, 16 Januari 2023**

**Dosen**

**Mohammad Faried Rahmat, S.ST., M.Tr.T**

**NIDN.**

# Tailwind CSS



**Tailwind CSS** adalah framework CSS open source. Tailwind CSS bekerja dengan memindai semua file HTML, komponen JavaScript, dan template lain untuk nama kelas, menghasilkan gaya yang sesuai, lalu menuliskannya ke file CSS statis. Tailwind CSS bersifat cepat, fleksibel dan andal tanpa runtime.

Informasi selengkapnya bisa dilihat pada halaman resmi TailwindCSS [**disini**](https://tailwindcss.com/).

# Instalasi Tailwind CSS

**Instalasi Tailwind CSS** bisa dilakukan dengan beberapa cara, yaitu:

1. Tailwind CLI
2. Menggunakan PostCSS
3. Framework Guides
4. Menggunakan CDN

Pada modul kali ini akan dijelaskan instalasi Framework **Tailwind CSS** menggunakan CLI dan CDN. Sebelum masuk ke Langkah instalasi berikut ini beberapa tools yang akan digunakan dalam instalasi dan penggunaan Tailwind CSS:

1. **Teks Editor**

* Visual Studio Code

(<https://code.visualstudio.com/>)

* Sublime Text

(<https://www.sublimetext.com/>)

1. **Javascript runtime environment**

* NodeJS

(<https://nodejs.org/en/>)

1. **Git** (<https://git-scm.com/>)
2. **Plugin Teks Editor Visual Studio Code** (*Opsional*)

* Live Server

(<https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>)

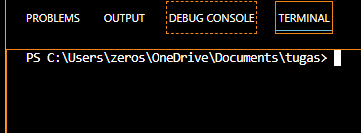
* HTML CSS Support

(<https://marketplace.visualstudio.com/items?itemName=ecmel.vscode-html-css>)

* **Instalasi Tailwind CLI**

1. **Buka Terminal pada teks editor atau git, masuk ke folder project website.**

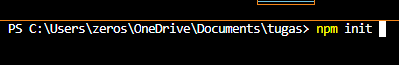
Sebelum Langkah instalasi pastikan nodeJS sudah terinstall pada device yang akan digunakan. Jika belum silahkan lakukan instalasi nodeJS.



Gambar 1 Terminal Visual Studio Code

1. **Membuat file package.json**

Gunakan perintah **npm init** untuk membuat file package.json



Gambar 2 npm init

Selanjutnya tekan enter pada setiap Langkah dan isi pada bagian yang diperlukan. Setelah perintah **npm init** dijalankan file package.json berhasil dibuat.



Gambar 3 package.json

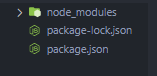
1. **Install tailwindcss via npm (nodeJS)**

Jika sudah masuk pada terminal Kembali dan masukkan perintah **npm install -D tailwindcss**



Gambar 4 install tailwindcss

Jika sudah terinstall maka akan terdapat folder dan file baru yaitu node\_modules dan package-lock.json.

****

Gambar 5 Node Modules - Package-lock.json

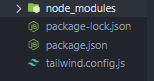
1. **Buat file tailwind.config.js**

Selanjutnya buat file **tailwind.config.js** dengan cara memasukkan perintah **npx tailwindcss init** pada terminal.



Gambar 6 buat file tailwind.config.js

Jika sudah maka akan muncul file baru bernama **tailwind.config.js**



Gambar 7 tailwind.config.js

1. **Konfigurasi template path (*Opsional*)**

Tambahkan semua path ke file template pada file tailwind.config.js

**Module.exports = {**

**content: [“./src/\*\*/\*.{html,js}”],**

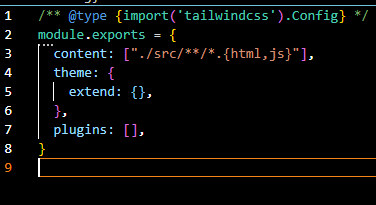
**theme: {**

**extend:{},**

**},**

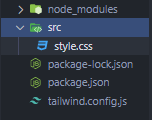
**plugins: [],**

**}**

/

Gambar 8 tambahkan path

1. **Membuat folder src dan konfigurasi file style.css**



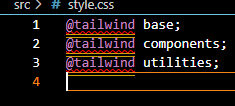
Gambar 9 folder src dan file style.css

Buat folder **src** dan di dalamnya buat file dengan nama **style.css** yang berisikan variable Tailwind

**@tailwind** base;

**@tailwind** components;

**@tailwind** utilities;



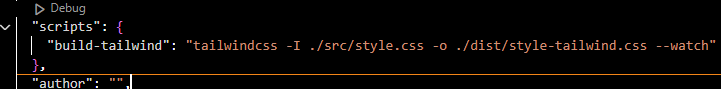
Gambar 10 tambahkan variabel tailwind

1. **Atur cara me-build tailwind pada project**

Atur **script** pada file **package.json** dan buat seperti pada kode di bawah ini.

**“build-tailwind”: “tailwindcss -i ./src/style.css -o ./dist/style-tailwind.css --watch”**

* **Build-tailwind**: nama perintah untuk menjalankan tailwind
* **./src/styles.css**: nama file css yang telah dibuat sebelumnya
* **./dist/style-tailwind.css**: nama file css tailwind



Gambar 11 package.json edit script

1. **Build tailwind**

Selanjutnya jalankan perintah **npm run build-tailwind** untuk mem-build Tailwind pada project.



Gambar 12 build tailwind

**HOMEPAGE SEKOLAH KODING**

1. **Bagian 1**

Menggunakan framework TailwindCSS sangat memudahkan developer dalam pembuatan sebuah website. Dengan memanfaatkan utilities class yang disediakan developer mampu membuat web dengan lebih praktis tanpa perlu menuliskan sintaks CSS secara menual.

Di tailwind sering dijumpai istilah sumbu X dan Y seperti pada contoh class my-2 atau px-2. Dimana X merujuk ke sumbu horizontal (kanan-kiri) dan Y merujuk ke sumbu vertical (atas-bawah).

1. **Bagian 2**

Konsep **responsive** pada tailwind menggunakan Breakpoint prefix. Konsep ini akan sering digunakan ketika ingin membuat tampilan website yang berbeda pada saat ukuran tertentu.

Untuk mengatur lebar pada point tertentu biasanya digunakan sintaks **breakpoint:size** atau sebagai contoh **md:w-4/6**.

Konsep **hover** adalah aksi dimana cursor terletak di atas elemen. Dengan tailwind penggunaan metode hover sangat mudah, yaitu dengan menggunakan sintaks **hover:aksi** atau sebagai contoh **hover:bg-red-500**.

Konsep **fokus** adalah aksi dimana elemen tertentu sedang digunakan. Dengan tailwind penggunaan metode fokus hampir sama denganhover, yaitu dengan menggunakan sintaks **focus:aksi** atau sebagai contoh **focus:outline-non**

1. **Bagian 3**

Tailwind mendukung developer untuk melakukan kustomisasi pada class ataupun membuat class sendiri. Untuk membuat kustomisasi pada tailwind dapat dilakukan pada file **tailwind.config.js**. Sebagai contoh untuk kustomisasi warna tambahan pada tailwind dapat dilakukan pada baris **theme.**

/\*\*@type{import('tailwindcss').Config}\*/

module.exports = {

    content: ['./src/\*\*/\*.{html,js}'],

    theme: {

        extend: {

            // *Kustomisasi Tailwind*

            colors: {

                mycolor: {

                    DEFAULT: '#7b99b9',

                    dark: '#517dad',

                },

            },

        },

    },

    plugins: [],

};

Pada baris kode diatas memasukkan warna kustom pada baris kode extend theme. Di letakkan pada baris extend karena disini kita hanya ingin menambahkan keyword warna baru pada projek website kita. Jika sudah rebuild ulang tailwind untuk melakukan perbaruan pada tailwind.Untuk penggunaannya sama dengan warna pada tailwind biasa. Sebagai contoh perhatikan baris kode dengan keyword mycolor seperti yang telah dibuat di atas.

<button class="mt-2 w-full text-center bg-mycolor hover:bg-mycolor-dark text-white p-2 text-large rounded-md">Daftar</button>

1. **Bagian 4**

Dalam menggunakan tailwind mungkin saja developer menemui kesulitan dimana harus menuliskan banyak kode untuk setiap elemen. Sebagai contoh untuk membuat **button** harus menambahkan beberapa utilities class tailwind. Dengan menggunakan konsep fungsi **apply** pada tailwind akan memudahkan developer dalam menulis kode agar lebih optimal.

Penggunaan konsep **apply** pada tailwind saya kira hampir mirip seperi pada pembuatan class dan css pada umumnya. Penulisan **apply** tailwind terjadi pada file main css pada project atau satu file pada import variable pada tailwindnya. Sebagai contoh perhatikan penjelasan berikut.

Jika ingin membuat button tanpa menggunakan konsep apply maka akan menggunakan beberapa class pada satu buah element. Dengan menggunakan **apply** baris kode tersebut bisa terlihat lebih rapi.

<a href="" class="text-sm text-white py-1 px-6 rounded-md bg-red-400">Google</a>

                        <a href="" class="text-sm text-white py-1 px-6 rounded-md bg-gray-400">Github</a>

                        <a href="" class="text-sm text-white py-1 px-6 rounded-md bg-blue-400">Twitter</a>

Pada file utama css buat sebuah selector yang akan digunakan sebagai nama keyword class lalu masukkan keyword **@apply** dan diikuti nama class tailwind yang digunakan sebelumnya pada bagian deklarator. Dalam penggunaan **apply** sebisa mungkin hindari penggunaan nama class yang terlalu spesifik. Sebagai contoh perhatikan baris kode berikut.

Jika sudah cara penggunaannya sama seperti halnya penggunaan class tailwind pada umumnya. Sebagai contoh pada baris class sebelumnya baris kode terlihat sangat Panjang, dan setelah penggunaan metode **apply** baris kode akan terlihat lebih rapi.

1. **Bagian 5**

Penggunaan **group hover ke child** sama seperti konsep css position relative dan absolute. Dimana kedua metode tersebut saling berhubungan. Sama halnya konsep **group hover to child** pada parent element harus menerapkan class **group** untuk menandai bahwa class tersebut berada pada parent dari elemen HTML. Dan pada child class nya untuk menggunakan hover yaitu dengan menggunakan class **group-hover:aksi**. Sebagai contoh **group-hover:text-gray-900**. Karena property pada tailwind masih terbatas, maka jika ingin menambahkan sebuah property baru bisa ditambahkan pada file **tailwind.config.js** pada bagian variants lalu extend.

Sebagai contoh:

/\*\*@type{import('tailwindcss').Config}\*/

module.exports = {

    content: ['./src/\*\*/\*.{html,js}'],

    theme: {

        extend: {

            // *Kustomisasi Tailwind*

            colors: {

                mycolor: {

                    DEFAULT: '#7b99b9',

                    dark: '#517dad',

                },

            },

        },

        variants:{

            extends:{

                divideColor:['group-hover']

            }

        }

    },

    plugins: [],

};

Setiap melakukan penambahan atau kustomisasi pada tailwind lakukan re build untuk melihat perubahan yang terjadi.

1. **Bagian 6**

Responsive menu mempunyai konsep yang sama pada responsive pada tailwind yang sudah dijelaskan. Namun pada responsive bagian menu terdapat beberapa trik untuk mengakali tampilan website yang berbeda pada setiap device. Dengan penggunaan sintaks yang sama dan diikuti dengan pengaturan display pada elemen HTML hal itu dapat digunakan untuk mengatur tampilan website antara device satu dengan lainnya.