LAPORAN RESMI

PRAKTIKUM DESAIN WEB

**PERTEMUAN 5**

**CSS**



**NIM : 21104410079**

**NAMA : Muhammad Yusuf Ibrahim Ramadhani**

**JURUSAN : Teknik Informatika**

**KELAS : TI 3-B 2022**

**TGL. PRAKTEK : 9 Januari 2023**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ISLAM BALITAR**

**2021**

LEMBAR PERSETUJUAN

**MODUL V**

**TAILWIND CSS**

**NIM : 21104410079**

**NAMA : Muhammad Yusuf Ibrahim Ramadhani**

**JURUSAN : Teknik Informatika**

**KELAS : TI 3-B 2022**

**TGL. PRAKTEK : 9 Januari 2023**

**Disetujui,**

**Blitar, 16 Januari 2023**

**Dosen**

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

**NIDN.**

# https://anf.al/assets/blog-images/2021-03-06-tailwindcss-review-by-a-backend-developer/tailwindcss.png

# **MATERI DASAR**

# **Konsep CSS**

CSS adalah bahasa Cascading Style Sheet dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs. CSS dibuat dan dikembangkan oleh W3C (World Wide Web Consortium) pada tahun 1996 untuk alasan yang sederhana.

Dulu HTML tidak dilengkapi dengan tags yang berfungsi untuk memformat halaman. Anda hanya perlu menulis markup untuk situs. Tags, seperti <font>, diperkenalkan di HTML versi 3.2, dan ketika itu menyebabkan banyak masalah bagi developer. Karena website memiliki berbagai font, warna background, dan style, maka untuk menulis kembali (rewrite) kode memerlukan proses yang sangat panjang dan sulit. Oleh sebab itu, W3C membuat CSS untuk menyelesaikan masalah ini.

HTML dan CSS memiliki keterikatan yang erat. Karena HTML adalah bahasa markup (fondasi situs) dan CSS memperbaiki style (untuk semua aspek yang terkait dengan tampilan website), maka kedua bahasa pemrograman ini harus berjalan beriringan

# **Aturan Penulisan CSS**

Berikut aturan penulisan CSS:

1. Selector di tempatkan pada awal penulisan CSS. Dalam contoh diatas, body, p, h1 dan h2 adalah selector. Khusus untuk selector yang lebih dari 1 (seperti pada contoh selector terakhir: h1,h2) untuk memisahkan kedua selector tersebut, digunakan tanda koma (,)
2. Setelah penulisan selector, seluruh isi dari property dan valuenya (nilainya), berada di antara kurung kurawal “{“ dan “}”.
3. Diantara property satu dengan yang lainnya, dipisahkan dengan tanda titik koma (;).
4. Diantara property dengan value (nilai), dipisahkan dengan anda titik dua (:).
5. Untuk property yang memiliki 2 kata , spasi diantaranya digantikan dengan tanda hubung (-), contohnya: background-color dan border-left.
6. Untuk properti yang berada pada baris terakhir, kita boleh mengabaikan tanda ‘;’ sebagai tanda tutup, tetapi disarankan agar tetap menggunakan tanda ‘;’, karena bisa saja kita lupa menuliskan tanda titik koma pada saat menambahkan properti lainnya. Contoh property tanpa tanda ‘;’ dapat di lihat pada contoh “text-decoration: underline”.

## **Tailwind**

Menurut informasi pada website-nya, Tailwind CSS merupakan framework yang bersifat utility-first untuk membangun desain antarmuka khusus dengan cepat.

Berbasis utility artinya Tailwind cuma terdiri dari 100% utility class dan nggak ada class komponen seperti Navbar, Button, Card, Modal, dll. Komponen-komponen ini kita buat sendiri dengan class utility.

Kebanyakan framework CSS yang lain menyediakan sesuatu secara berlebihan, penyediaan segala jenis komponen yang telah didesain sebelumnya, seperti: buttons, cards, alerts, dan lain sebagainya. Komponen-komponen tersebut mungkin membantu dengan cepat di awal, tetapi masalah baru muncul ketika kita ingin membangun situs kita dengan desain khusus yang sesuai keinginan kita. Melakukan perubahan terhadap komponen-komponen tersebut membutuhkan effort dan waktu yang tidak sedikit. Akan tetapi, Tailwind CSS memiliki pendekatan yang berbeda. Daripada menyediakan komponen-komponen yang sudah didesain sebelumnya, Tailwind menyediakan low-level utility classes yang dapat kita gunakan untuk membangun desain yang khusus secara keseluruhan, tanpa harus keluar dari file HTML kita.

1. Responsif

Setiap utilitas Tailwind memiliki jenis-jenis responsive yang mempermudah untuk membangun interface responsive tanpa perlu menggunakan CSS khusus. Tailwind menggunakan prefix {screen}: intuitif, sehingga mudah untuk mengetahui kelas-kelas responsif yang ada dalam markup kita sekaligus menjaga nama kelas original dapat dikenali dan utuh.

1. Component-friendly

Meskipun kita dapat melakukan banyak hal hanya dengan menggunakan kelas utilitas, seiring dengan berkembangnya sebuah proyek, akan berguna untuk mengkodifikasikan pola yang umum menjadi abstraksi tingkat tinggi. Tailwind menyediakan alat untuk mengekstraksi kelas komponen dari pola utilitas yang berulang, yang memudahkan untuk memperbarui beberapa instances dalam sebuah komponen dari satu tempat.

Tailwind membiarkan kita untuk menyesuaikannya. Hal ini termasuk warna, ukuran border, bobot font, spacing utilities, breakpoints, shadows, dan masih banyak lagi. Tailwind ditulis dalam bentuk PostCSS dan dikonfigurasi dalam JavaScript. Tailwind lebih dari sekadar framework CSS, Tailwind merupakan sebuah mesin untuk membuat sistem desain.

# **PENGINSTALAN TAILWIND**

## **Install Tailwind CSS**

## **Hal yang perlu disiapkan**

1. Text Editor

Digunakan untuk menulis kode. Ada banyak pilihan, disini kita menggunakan Visual Studio Code.

1. Node.js

Digunakan untuk install library dan package, serta untuk menjalankan perintah npm (node package manage) dalam proses instalasi.

1. npm

Ada banyak pilihan lain, tapi kita menggunakan npm.

1. Extensi

Karena kita menggunakan vs code, ada beberapa extensi yang bisa digunakan untuk mempermudah kita nantinya.

* Live server

Live Server Adalah sebuah ekstensi VSCode yang digunakan untuk menjalankan server web lokal dalam proyek Anda. Ekstensi ini sangat berguna saat Anda ingin menjalankan dan menguji halaman web yang Anda buat tanpa harus mengkonfigurasi server web lokal secara manual. Setelah diinstal, ekstensi ini akan menambahkan opsi baru dalam menu konteks VSCode yang memungkinkan Anda untuk menjalankan server web lokal dengan satu klik saja.

* HTML CSS support

HTML CSS Support Adalah ekstensi VSCode yang digunakan untuk meningkatkan pengalaman penulisan kode HTML dan CSS di dalam editor. Ekstensi ini menyediakan fitur seperti syntax highlighting yang lebih baik, autocomplete, dan validasi sintaksis yang membantu Anda dalam menulis kode HTML dan CSS yang benar. Ekstensi ini juga menyediakan fitur lain seperti snippet, emmet dll yang membantu Anda dalam menyelesaikan kode dengan cepat.

* Tailwind CSS Intellisense

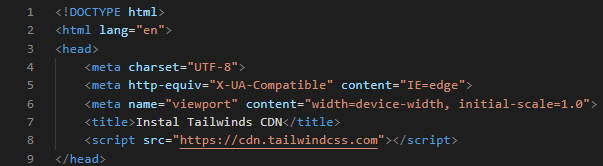
Berfungsi untuk mempermudah ketika kita kurang ingat dengan nama-nama kelas.

## **Instalasi Tailwind**

1. **Menggunakan link CDN**

Menggunakan Tailwind dari CDN tidak direkomendasikan untuk dipakai di production, karena kita tidak akan mendapatkan hasil build CSS yang optimal.

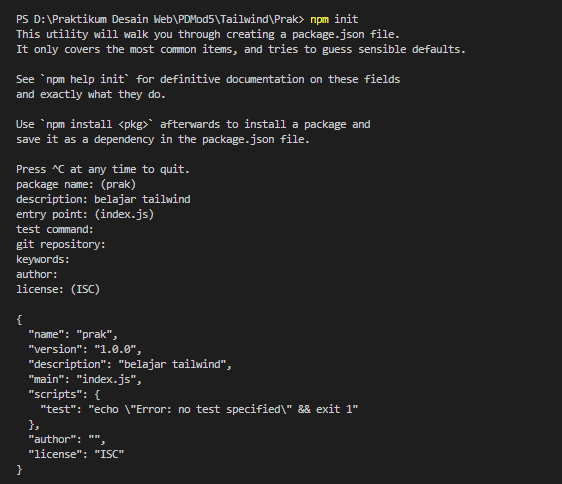
1. Siapkan file atau kode HTML baru pada proyek.
2. Kemudian di dalam tag <head> tambahkan link CSS dengan alamat CDN Tailwind.



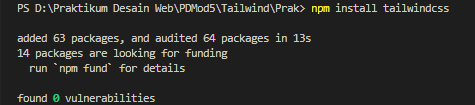
1. **Menggunakan NPM**
2. Pastikan anda telah menginstall node.js dan juga npm. Kita dapat mengeceknya dengan menggunakan cmd, klik windows+R lalu ketikkan cmd.

1. Pada proyek yang dibuat, masuk ke terminal ketikkan ‘npm init’. Perintah ini digunakan untuk membuat file package.json, dimana setelahnya kita diminta untuk memasukkan data dari proyek yang kita buat.



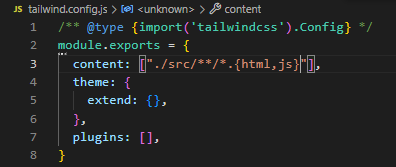
1. Setelah itu, kita memlakukan pengistalan dengan memasukkan perintah ‘npm install tailwindcss’. Ini akan menambahka tailwind css kedalam package.json dan membuat folder node\_module dialam proyek yang dibuat.



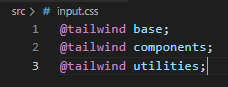
1. Setelah penginstalan selesai seperti diatas, kita perlu mengkonfigurasi tailwind css dengan memasukkan perintah ‘npx tailwind init’. Perintah ini akan membuat file baru bernama tailwind.config.js, yang berisi konfigurasi yang dibutuhkan tailwind.



1. Lakukan konfigurasi path konten pada tailwind.config.js

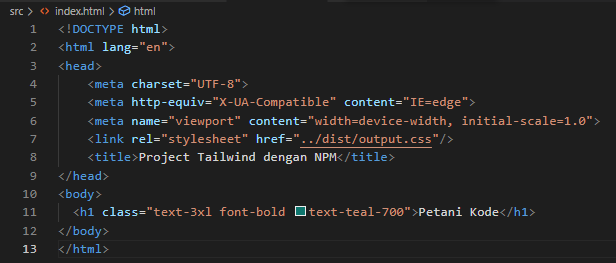


1. Setelah itu, buat folder baru dengan nama src di dalam project. Kemudian buat file CSS baru di dalam src dengan nama input.css dan isi seperti ini:

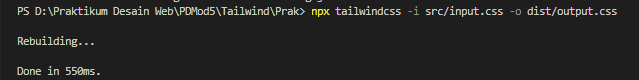


1. Sebagai contoh kita membuat file index.html di dalam folder src untuk menggunakan class-class dari Tailwind.

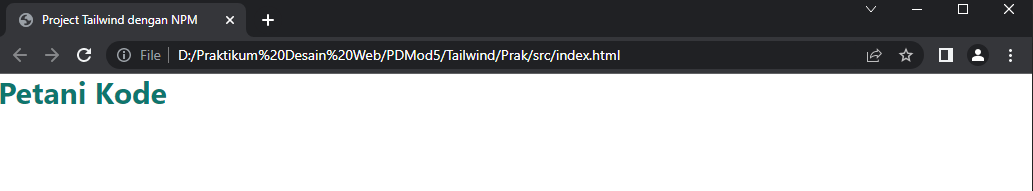
Isi file index.html seperti ini:



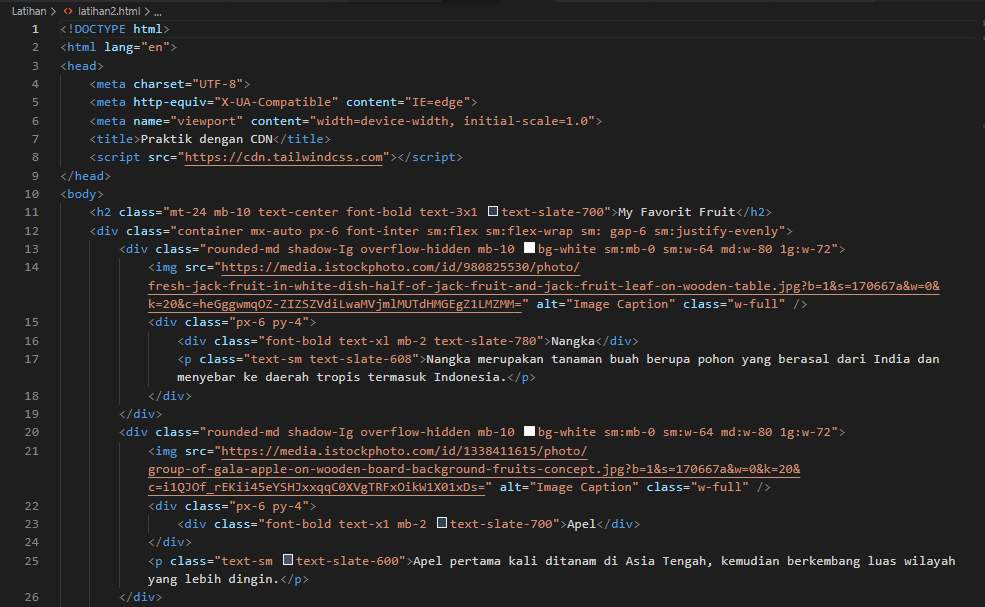
1. Pada gambar, terdapat kode ../dist/output.css. File ini tidak terdapat dalam project, maka kita perlu membuatnya dengan perintah ‘npx tailwindcss -i src/input.css -o dist/output.css’.

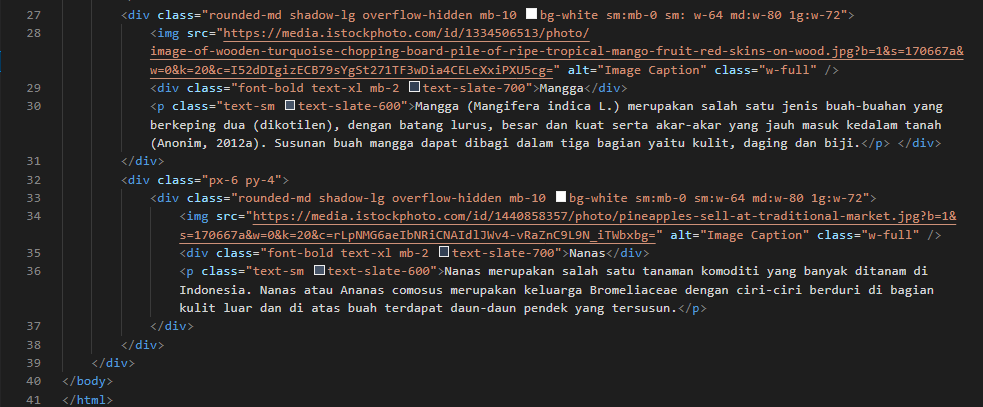


1. Outputnya:

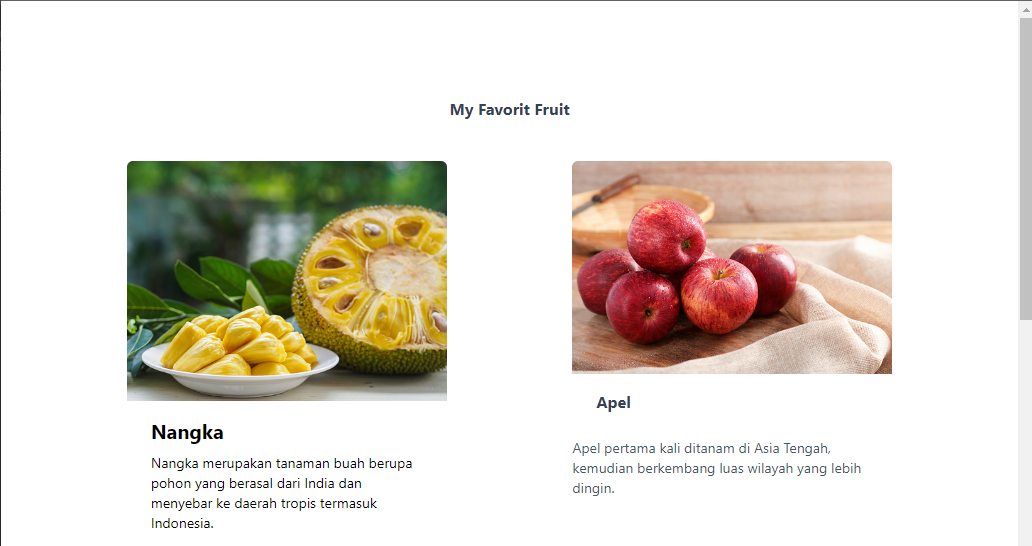


1. Dari gambar terlihat bahwa tailwind css sudah bisa digunakan.
2. **Implementasi dari Penginstalan**
3. **Menggunakan link CDN**

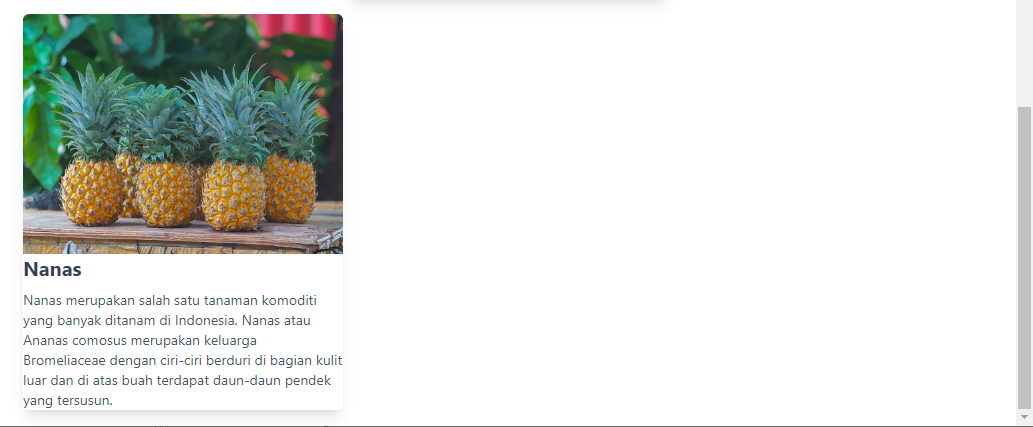




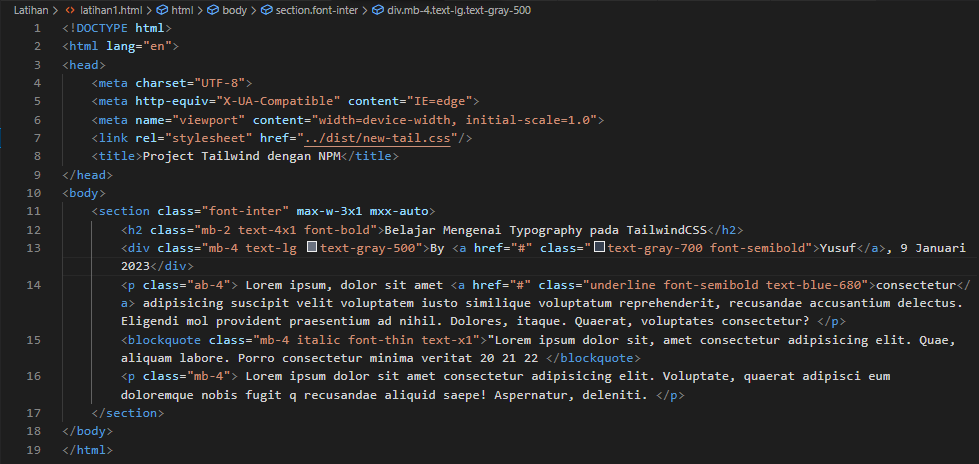
Output:



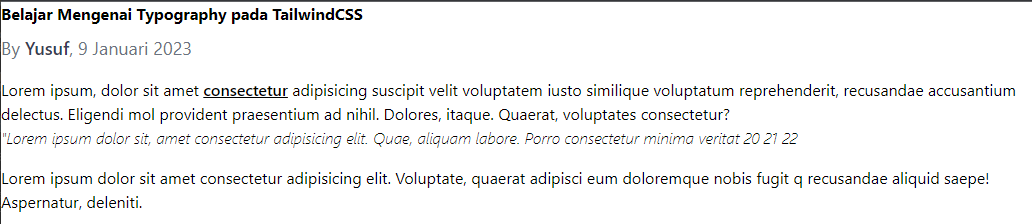




1. **Menggunakan NPM**

****

Output:

****