LAPORAN RESMI

PRAKTIKUM DESAIN WEB

**MODUL VI**

CSS TAILWIND



**NIM : 21104410091**

**NAMA : BETA NURUL AWWALIN**

**JURUSAN : TEKNIK INFORMATIKA**

**KELAS : TI 3B 2022**

**TGL. PRAKTEK : 9 JANUARI 2023**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ISLAM BALITAR**

**2023**

LEMBAR PERSETUJUAN

**MODUL VI**

**CSS TAILWIND**

**NIM : 21104410091**

**NAMA : BETA NURUL AWWALIN**

**JURUSAN : TEKNIK INFORMATIKA**

**KELAS : TI 3B 2022**

**TGL. PRAKTEK : 9 JANUARI 2023**

**Disetujui,**

**Blitar, 14 Januari 2023**

**Dosen**

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

**NIDN. BAB I**

**PENDAHULUAN**

**1.1 CSS Tailwind**

Tailwind CSS merupakan framework yang bersifat utility-first untuk membangun desain antarmuka khusus dengan cepat.

**Kelebihan CSS Tailwind**

* **Lebih sederhana dibandingkan framework CSS lainnya**

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.

* **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.

* **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.

* **Didesain untuk disesuaikan**

Apabila logis untuk disesuaikan, 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.

**BAB II**

**TUGAS PRAKTIKUM**

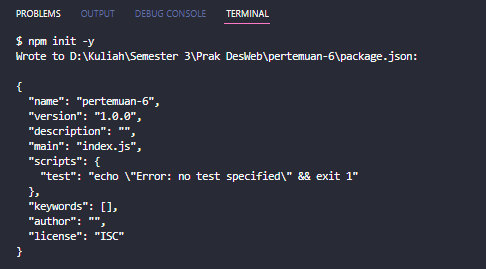
**2.1 Instalasi Tailwind**

Sebelum menginstal tailwind ada beberapa software yang perlu dipersiapkan diantaranya:

* Web Browser : Google Chrome
* Code Editor : Visual Studio Code
* VSCose Extension :
  + Tailwind CSS IntelliSense
  + Live Preview
  + Post CSS Languange Support
* Package Manager : NPM ( Node.js )
* Terminal

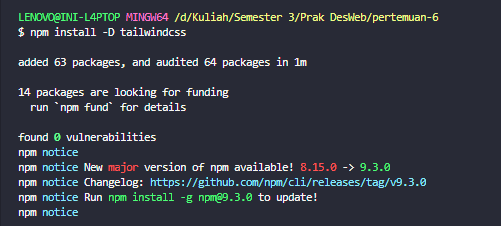
Untuk menginstall tailwind bisa menggunakan beberapa cara, namun disini saya menggunakan Tailwind CLI.

1. Instal terlebih dahulu extention yang diperlukan sebaimana terlampir diatas.
2. Buka terminal, bisa dengan shourcut *ctrl + shif + `*
3. Setelah itu tuliskan **npm init -y**, disini berguna untuk menginisialisasi aplikasi dengan init dan -y digunakan untuk mengisinya sebagai nilai default.



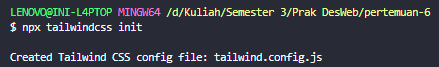
Selain itu dengan menggunakan npm init, pada sebelah kiri kita dibuatkan oleh npm nya file package.json, dan pada package ini kita dapat menginstall dependensi/ modul-modul yang akan digunakan pada aplikasi.

1. Saatnya menginstall tailwind dengan perintah **npm -D tailwindcss**

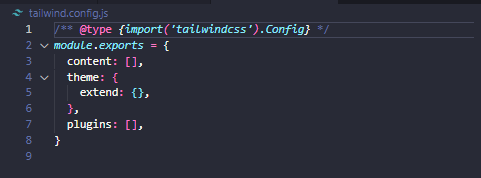
****

Disini bisa kita perhatikan pada kiri, terdapat folder node-modules yang berisi file-file yang dibutuhkan untuk jalanin tailwind. Dan pada file package.json kita sudah memiliki devDependencless karena kita telah mengetik -D.

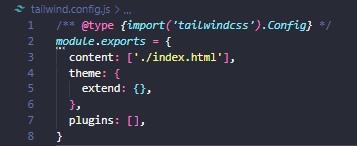
1. Selanjutkan menjalankan perintah **npx tailwindcss init,** ini untuk menginisiasi tailwindnya supaya kita memiliki konfigurasi tailwind.



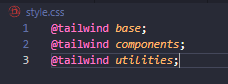
Hasilnya file tailwind.confid.js sudah dibuat, dan pada file ini berisi konfigurasinya.



1. Selanjutnya kita perlu mengkonfigurasi path atau lokasi template kita yang akan memiliki class-class tailwind. Dengan kita akan mengisi pada bagian content di tailwind.config.js, sebagai berikut.

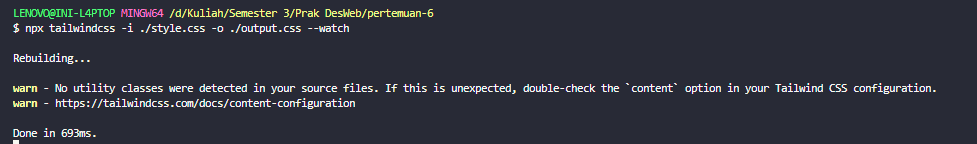


1. Selanjutnya kita buat file style.css, yang didalamnya berisi sebagai berikut.



Tiga itu disebut directive atau fungsi yang dimiliki oleh tailwindCSS yang ditulis berdasarkan postCSS. Dan ketiga hal tersebut bermaksud sebelum tailwind dijalankan maka akan melewati ketiga directive tersebut atau bisa disebut three layers.

1. Sekarang kita akan menjalankan build, dimana kita akan menjalankan proses CSS akhir yang akan digunakan oleh web kita. Dengan perintah



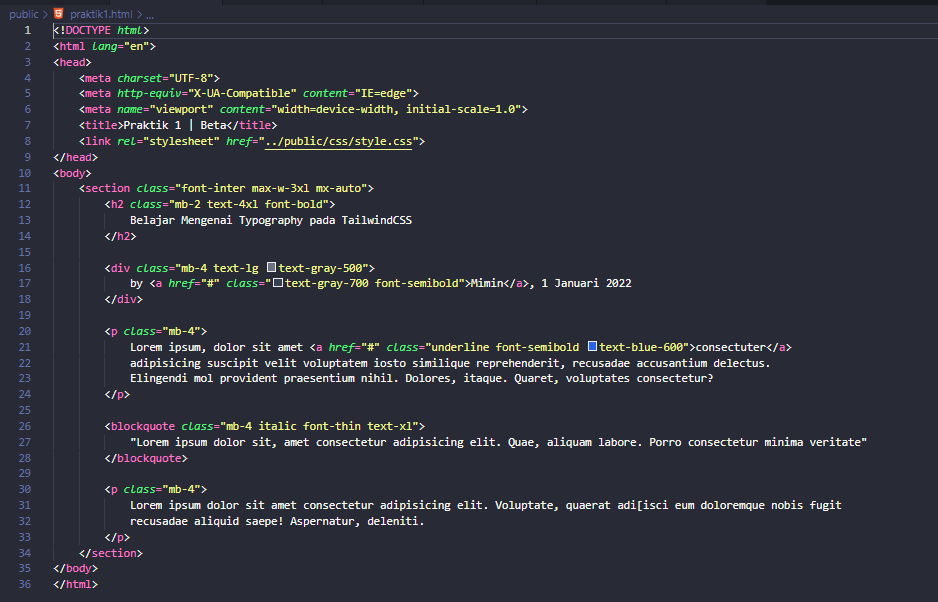
Dan hasilnya akan membuat file output.css yang berisi css yang akan kita gunakan.

1. Sekarang tailwind bisa digunakan. Tinggal menambahkan link rel pada file index.

**2.2 Praktik**

1. **Praktik 1**

**SourceCode :**

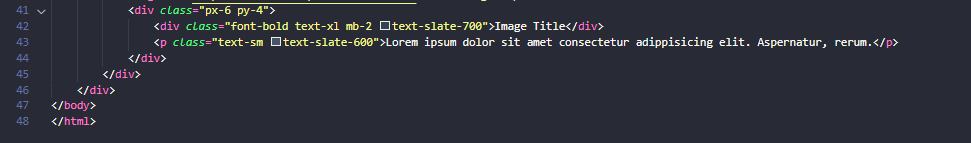
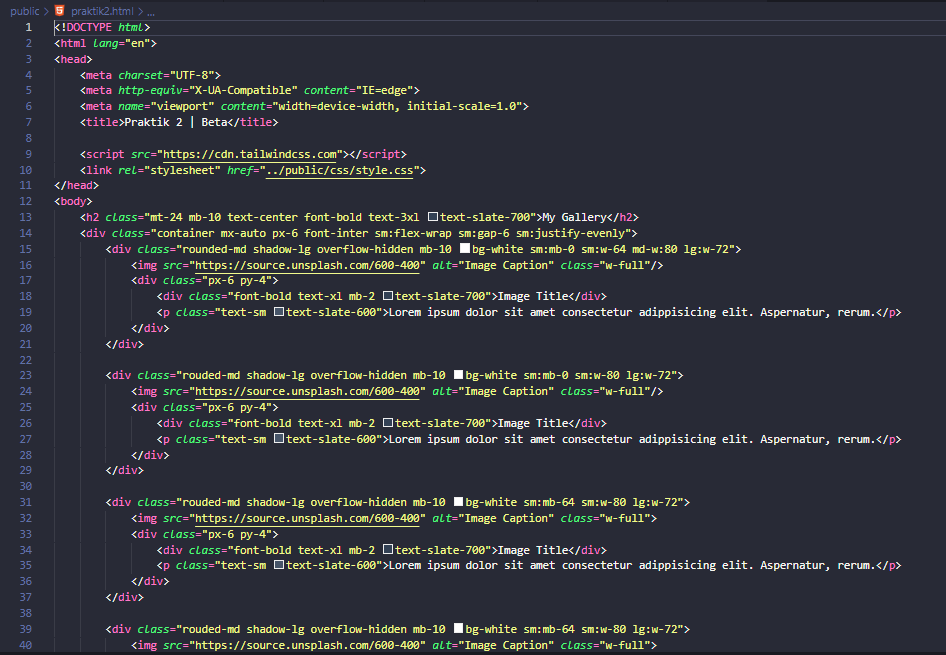
****

**Hasil Tampilan :**

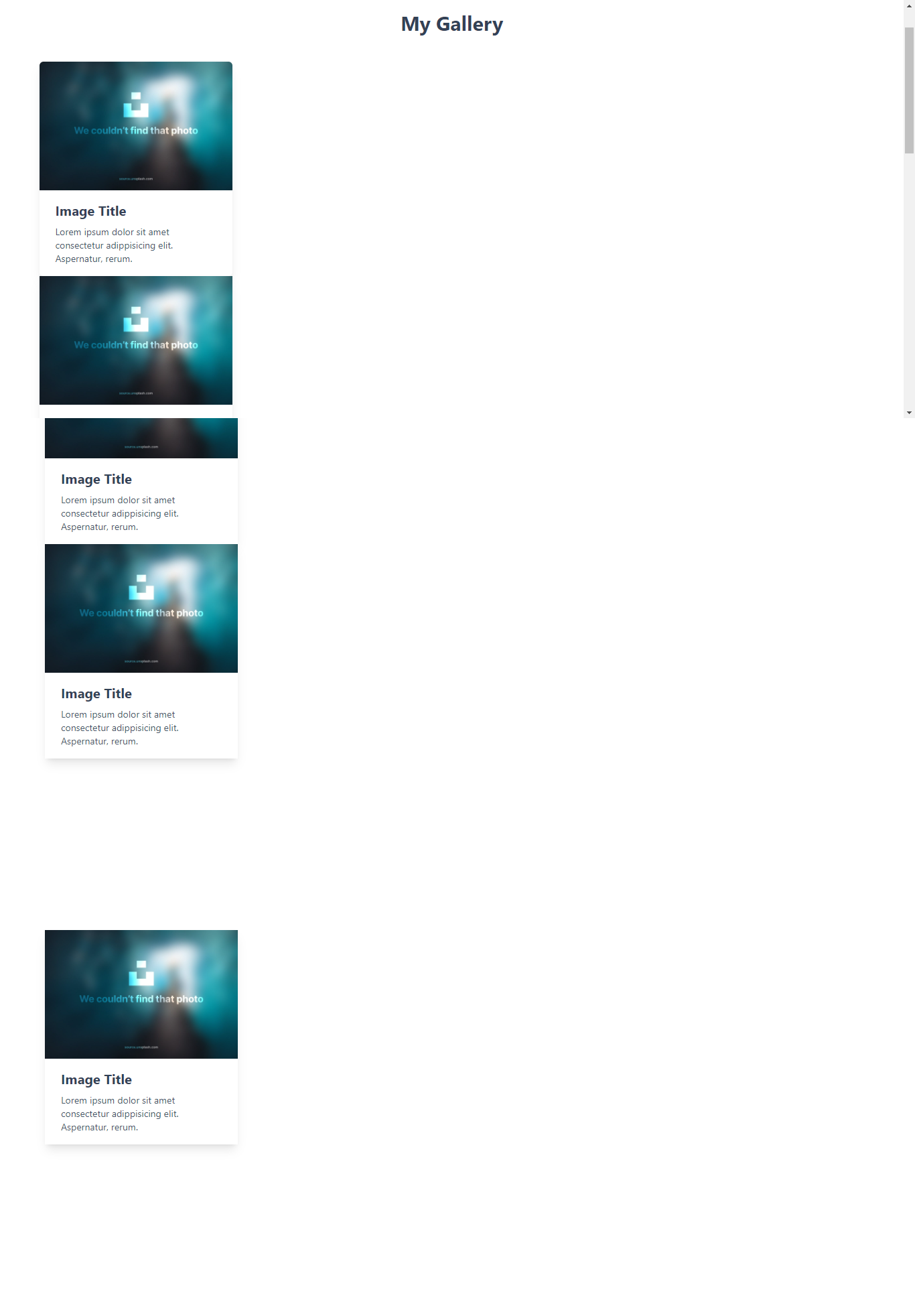
****

1. **Praktik 2**

**SourceCode :**

****

**Hasil Tampilan :**

****