|  |  |
| --- | --- |
| **Tanggal Asistensi :** | |
| **Tanda Tangan** | **Nilai** |
|  |  |

**MODUL 6**

**CSS *Framework* TailwindCSS**

**Nama : Imam Agil Aiman**

**Nim : F55123066**

**Kelas : B**

**2.1.6 CSS *Framework* TailwindCSS**

Tailwind CSS adalah sebuah kerangka (*framework*) CSS yang dirancang untuk mempermudah pengembangan antarmuka pengguna dengan menggunakan pendekatan *"utility-first"*. Pendekatan ini memungkinkan pengembang untuk langsung menentukan gaya dan tata letak dalam *markup* HTML mereka dengan menggunakan kelas-kelas utilitas yang disediakan oleh Tailwind. Beberapa karakteristik dari Tailwind CSS:

1. *Utility-First*: Tailwind mempromosikan pendekatan *utility-first*, di mana pengguna menggunakan kelas-kelas kecil untuk menerapkan gaya secara langsung pada elemen HTML. Contohnya, untuk memberi elemen warna latar belakang, bisa menggunakan kelas “*bg*-{*color*}”.

2. Modular dan *Konfigurable*: Tailwind dirancang agar mudah diadaptasi sesuai dengan kebutuhan proyek pengguna. Pengguna dapat mengonfigurasi tema, menyesuaikan kelas-kelas utilitas, dan menonaktifkan bagian-bagian tertentu yang tidak pengguna butuhkan.

3. Tidak Ada Gaya Bawaan: Tailwind tidak memberikan gaya bawaan, sehingga pengguna membangun gaya dari nol berdasarkan kebutuhan proyek pengguna. Ini memberikan fleksibilitas yang tinggi tetapi juga menuntut keputusan desain yang lebih besar.

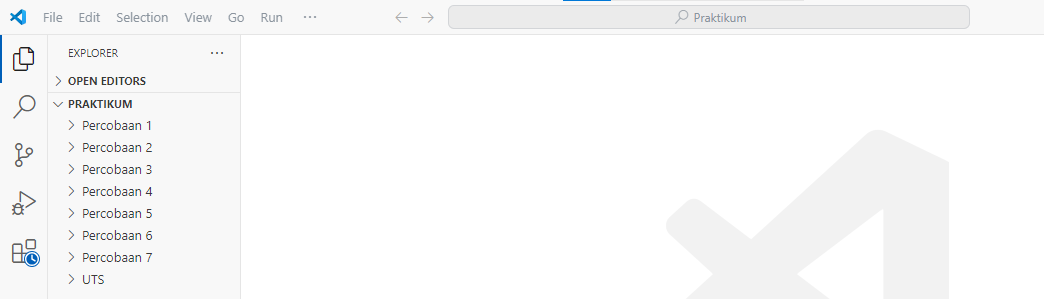
4. Responsif: Tailwind menyediakan kelas-kelas utilitas untuk responsif secara mudah, memungkinkan pengguna mengonfigurasi tata letak dan gaya yang responsif tanpa menulis CSS khusus untuk setiap *breakpoint*.

5. Dokumentasi yang Kuat: Tailwind memiliki dokumentasi yang sangat baik dengan contoh-contoh yang jelas dan informasi yang rinci. Ini membuatnya mudah bagi pengembang untuk memahami dan menggunakan kelas-kelas utilitas yang disediakan.

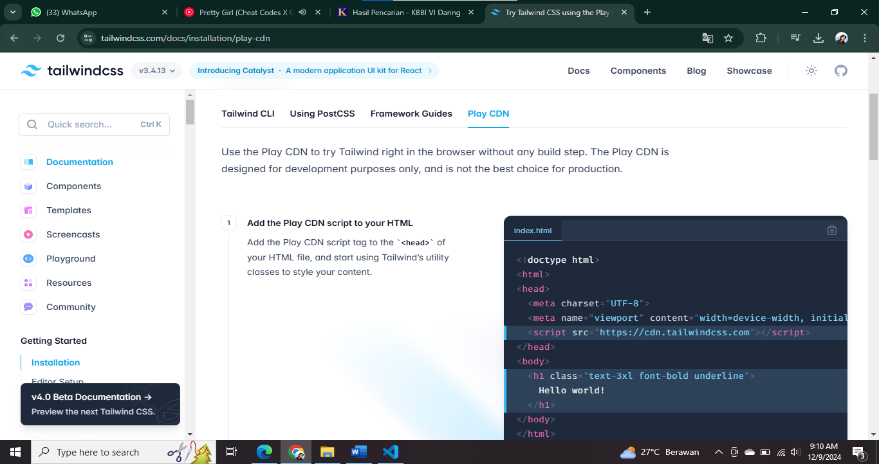
Beberapa pengembang lebih suka menggunakan pendekatan tradisional dengan menggunakan CSS *preprocessor* atau kerangka CSS lain yang lebih besar. Keputusan untuk menggunakan Tailwind atau tidak tergantung pada preferensi dan kebutuhan proyek tertentu.

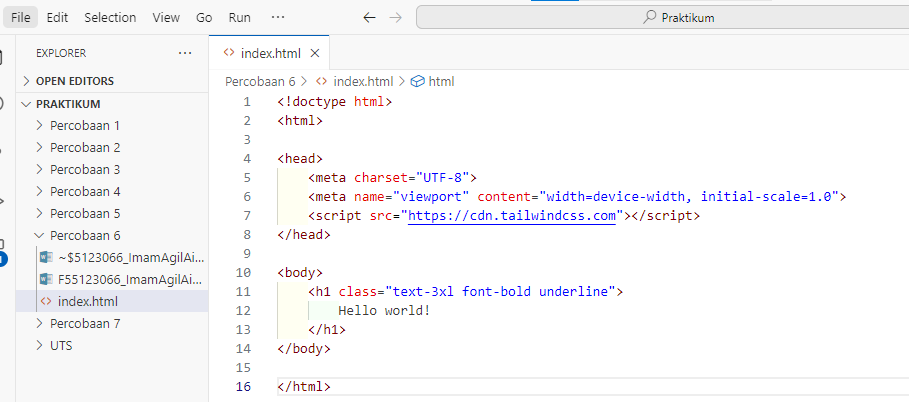
**3.1.6 CSS *Framework* TailwindCSS**

1. *Project*

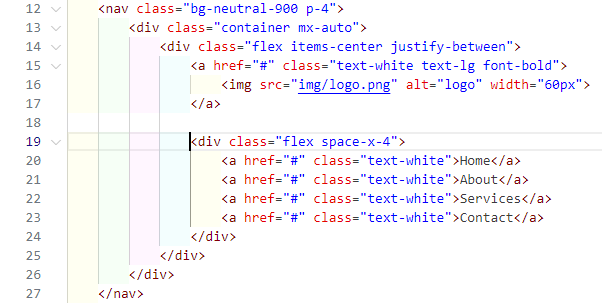
a. Pada folder utama, buat folder baru dengan nama “Pertemuan-6”

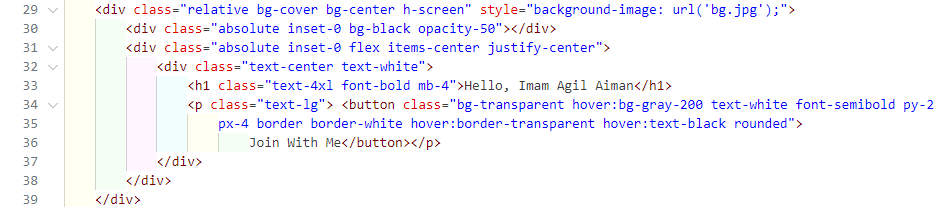
b. Pada folder “Pertemuan-6”, buat *file* baru dengan nama “index.html” dan tambahkan struktur dasar HTML 5 kedalamnya.

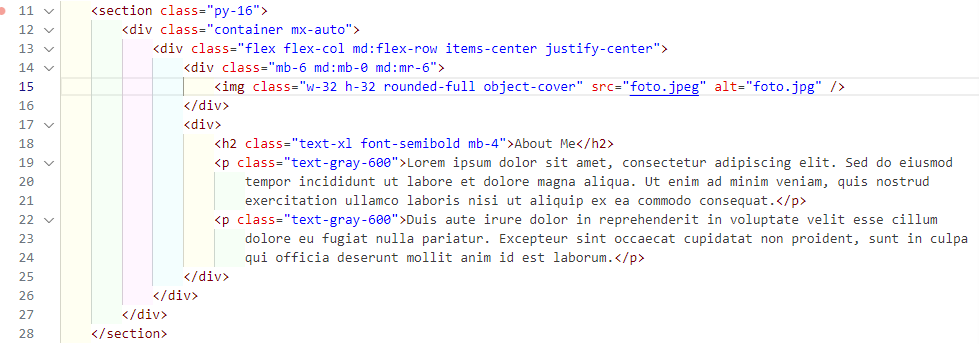
c. Buka [*https://tailwindcss.com/*](https://tailwindcss.com/), klik “*Get Started*” pilih *Play* CDN.

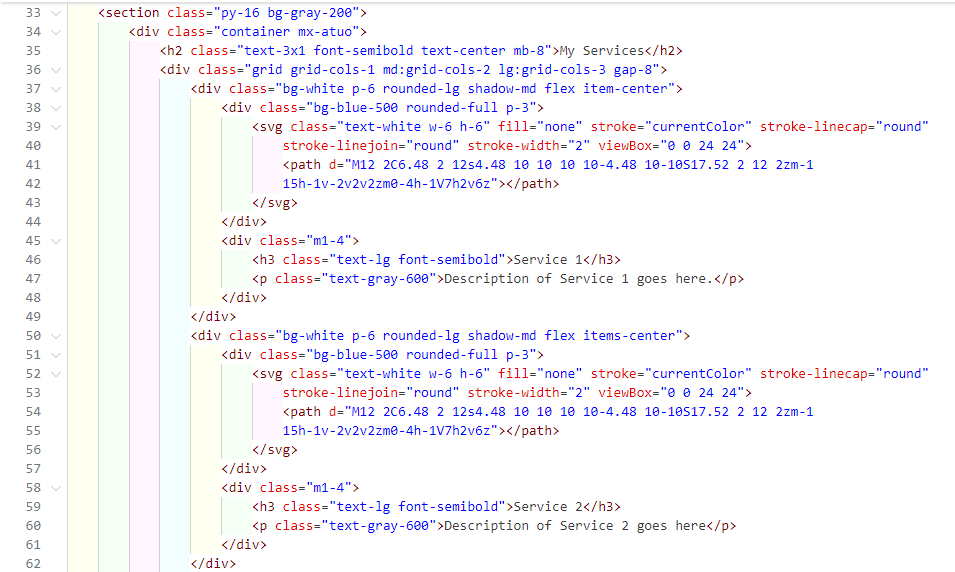
d. *Copy* dan pastekan pada *file* “index.html”

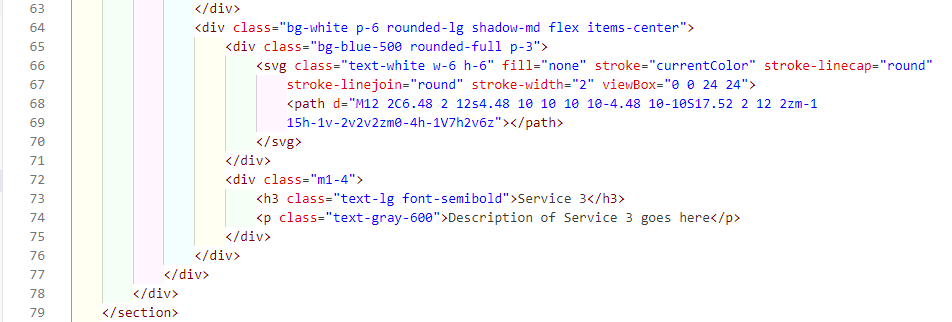
2. *Web* Tailwind

****a. Pada *file* “index.html” tambahkan kode seperti berikut untuk membuat *navbar* (jangan lupa menyiapkan logo, jika tidak ada *tag img* ganti dengan teks nama).

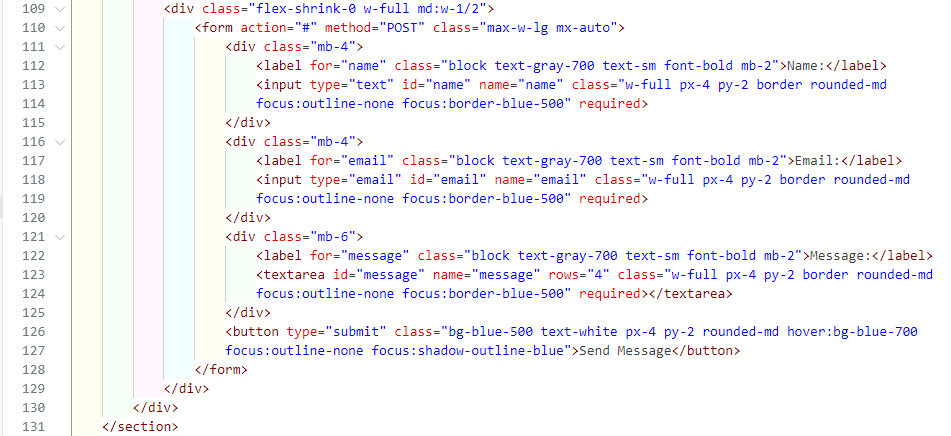
b. Tambahkan kode berikut untuk membuat *hero section* (siapkan satu buah *image* boleh cari di [*https://unsplash.com/*](https://unsplash.com/))

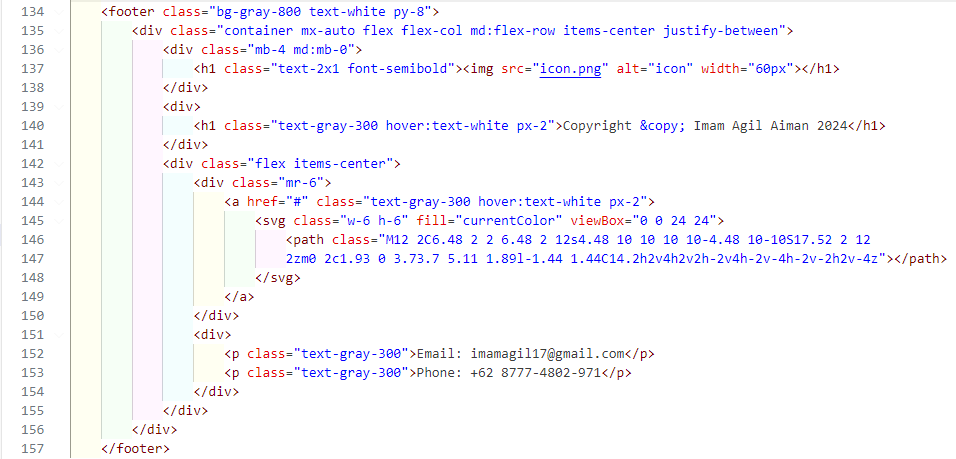
c. Tambahkan kode berikut untuk membuat *about me* (siapkan foto kalian)

f. Tambahkan kode berikut untuk membuat *my services*

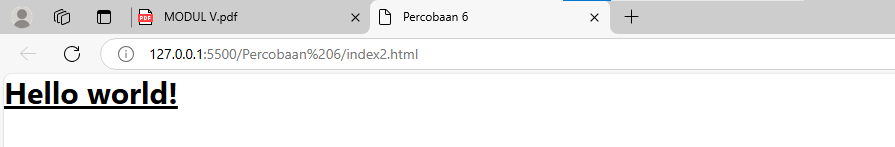


g. Tambahkan kode berikut untuk menambahkan *contact me*

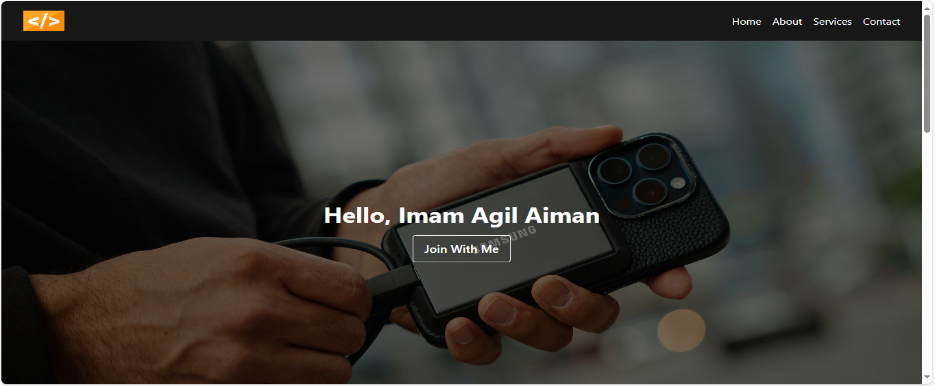


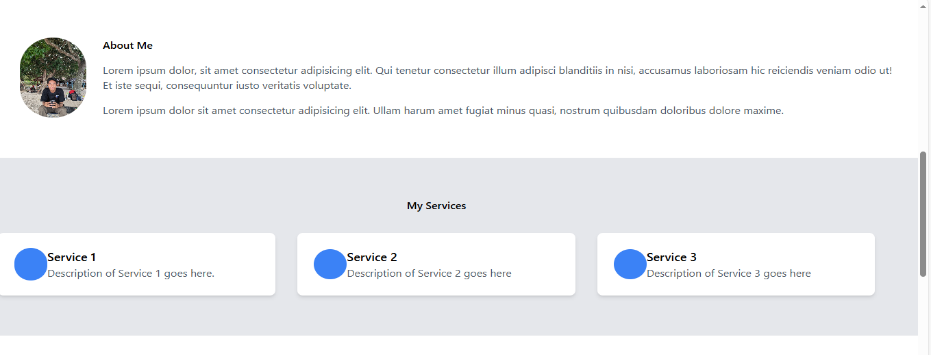
h. Tambahkan kode berikut untuk membuat *footer*

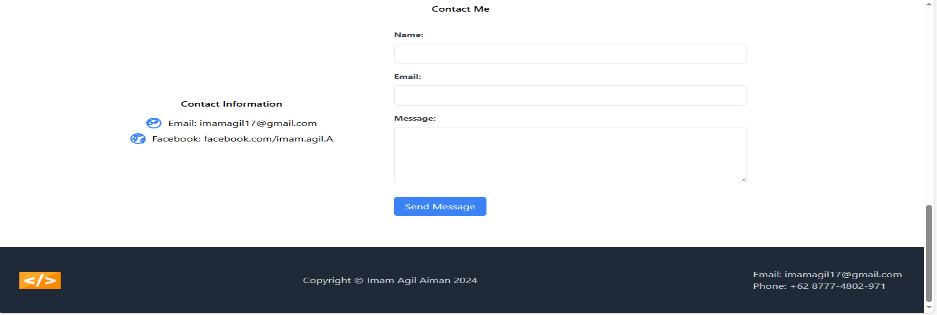
**4.1.6 CSS *Framework* TailwindCSS**

1. Tailwind CSS

Gambar diatas contoh sederhana dengan menggunakan gaya *framework* CSS bernama Tailwind, *tag* *“<script src="https*://cdn.tailwindcss.com*"></script>”* untuk membuat Tailwind, dan *“<h1 class="text*-3xl *font-bold underline">”* untuk mengatur ukuran, ketebalan, dan garis bawha teks.

 2. *Web* Tailwind





Percobaan diatas merupakan tampilan Tailwind yang telah ditambahkan kode mulai dari tampilan, *about me*, *my services*, *contact me*, dan *footer*. Dengan mengikuti perintah yang berada di modul 6, tampilan akhirnya akan menjadi seperti gambar diatas dengan menggunakan kelas-kelas utilitas dari Tailwind.

**BAB V   
PENUTUP**

**5.1 Kesimpulan**

**5.1.1 *Basic tag* HTML *Part* 1**

Kesimpulan dari modul 1 adalah untuk mengenalkan *tag-tag* dasar pada HTML yang digunakan untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di internet melalui layanan *web* dengan menggunakan berbagai macam *tag-tag* yang sesuai dengan kebutuhan seperti teks, gambar, *link*, *header*, *image*, dan lain-lain

**5.1.2 *Basic Tag* HTML *Part* 2**

Kesimpulan mengenai konsep dasar HTML *part* 2 yang mencakup *bullet list*, tabel, dan *form*. Dalam HTML, membuat daftar dengan menggunakan daftar tak berurutan(*unordered list*)dan daftar berurutan (*ordered list*). Tabel dalam HTML menggunakan beberapa elemen penting, yaitu : *table row*, *table header*, dan *table* data, dengan kombinasi elemen tersebut dapat membuat tabel yang rapi. Formulir yang memungkin pengguna memasukkan data, elemen-elemen dalam *form*, seperti : “*<input>”*, “*<textarea>”*, dan “*<select>”*.

**5.1.3 HTML *Layout***

Penggunaan *semantic layout* dalam HTML, seperti *“<nav>”*, *“<main>”*, *“<article>”*, dan *“<footer>”*, memberikan konteks dan makna yang lebih jelas pada struktur halaman, hal ini membantu pembaca layar untuk memahami konten dan fungsi setiap bagian, meningkatkan *Search Engine Optimization* (SEO) halaman. Sedangkan *layout* seperti *“<div>”* tidak memberikan arti atau tujuan khusus. Artinya *layout* lebih sulit diinterpretasikan oleh mesin pencari. *Division* merupakan kontainer blok yang sering digunakan untuk mengelompokkan elemen-elemen dalam sebuah bagian. Biasanya *“<div>”* dipakai dengan CSS untuk mengatur tata letak atau gaya dari sekelompok elemen, tapi *“<div>”* tidak memberikan informasi atau makna yang spesifik.

**5.1.4 CSS *(Cascading Style Sheets)***

Ketika menggunakan CSS untuk mendesain dan mengatur tata letak elemen-elemen dalam halaman *web*, dapat memberikan gaya visual pada HTML, dan membuat halaman *web* lebih menarik, mudah dibaca, dan responsif terhapat berbagai perangkat. CSS memungkinkan pengaturan berbagai elemen visual, seperti warna, ukuran, *font, margin, padding, dan border*. CSS sangat penting dalam pengembangan *web modern*, memungkinkan pengembang untuk menciptakan desain yang estetis, fungsional, dan responsif.

**5.1.5 CSS *Part* 2 *Flex* & *Grid***

CSS *Flexbox* dan *Grid* adalah dua teknik tata letak yang sangat kuat dalam pengembangan *web* moderen. Dengan memahami kapan dan bagaimana menggunakan masing-masing alat ini akan memungkinkan untuk menciptakan desain yang responsif, fleksibel, dan estetis. *Flexbox* menawarkan kesederhanaan dan fleksibilitas untuk tata letak satu dimensi, sedangkan *Grid* memberikan kontrol yang lebih besar untuk tata letak dua dimensi yang kompleks. Kombinasi keduanya digunakan untuk mendapatkan tata letak yang optimal dan efesien.

**5.1.6 CSS *Framework* TailwindCSS**

Tailwind CSS adalah salah satu *framework* CSS moderen yang berbasis utilitas. *Framework* ini memberikan kumpulan kelas siap pakai untuk mempermudah pengembangan antarmuka *web* yang responsif dan konsisten. *Framework* ini juga sangat cocok bagi pengembang yang ingin menciptakan desain yang unik tanpa bergantung pada gaya bawaan (*predefined components*) dan dengan pengendalian penuh atas tampilan.