

2.1.6 CSS Framework TailwindCSS

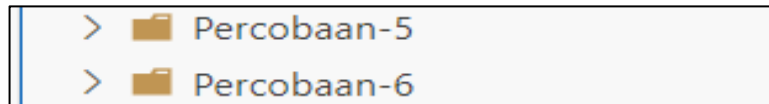
Tailwind CSS adalah sebuah kerangka (*framework*) CSS yang dirancang untuk mempermudah pengembangan antarmuka pengguna dengan menggunakan pendekatan "*utility-first*". Pendekatan ini memungkinkan pengembang untuk menentukan gaya dan tata letak dalam markup HTML mereka dengan menggunakan kelas-kelas utilitas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

1. *Utility-First*: Tailwind mempromosikan pendekatan *utility-first*, di mana Anda menggunakan kelas-kelas kecil untuk menerapkan gaya secara langsung pada elemen HTML. Contohnya, untuk memberi elemen warna latar belakang, Anda bisa menggunakan kelas "*bg-{color}*".
2. Modular dan *Konfigurable*: Tailwind dirancang agar mudah diadaptasi sesuai dengan kebutuhan proyek Anda. Anda dapat mengonfigurasi tema, menyesuaikan kelas-kelas utilitas, dan menonaktifkan bagian-bagian tertentu yang tidak Anda butuhkan.
3. Tidak Ada Gaya Bawaan: Tailwind tidak memberikan gaya bawaan, sehingga Anda membangun gaya dari nol berdasarkan kebutuhan proyek Anda. 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 Anda 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. Penting untuk diingat bahwa Tailwind CSS tidak sesuai untuk setiap proyek. 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 “Percobaan-6”



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



- c. Buka <https://tailwindcss.com/>, klik “Get Started” pilih Play CDN



- d. Copy dan paste-kan 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).

```
<nav class="bg-neural-900 p-4">
  <div class="container mx-auto">
    <div class="flex p-3 px-16 justify-between">
      <a href="#" class="text-white text-lg font-bold">Rasya</a>
      <div class="flex space-x-4">
        <a href="#" class="text-white">home</a>
        <a href="#" class="text-white">about</a>
        <a href="#" class="text-white">services</a>
        <a href="#" class="text-white">contact</a>
      </div>
    </div>
  </div>
</nav>
```

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

```
<div class="relative bg-cover bg-center h-screen" style="background-image: url('img/wallpaper.jpg');">
  <div class="absolute inset-0 bg-black opacity-50"></div>
  <div class="absolute inset-0 flex items-center justify-center">
    <div class="text-center text-white">
      <h1 class="text-4xl font-bold mb-4">Hello, I'm Rasya Rahmat Syaban</h1>
      <p class="text-lg">
        <button class="bg-transparent hover:bg-gray-200 text-white font-semibold py-2 px-4 border border-white
        hover:border-transparent hover:text-black rounded">Join With Me</button>
      </p>
    </div>
  </div>
</div>
```

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

```
<section class="p-16">
  <div class="container mx-auto">
    <div class="flex flex-col md:flex-row items-center justify-center">
      <div class="flex flex-col md:mb-0 md:mr-6">
        
      </div>
      <div>
        <h2 class="text-3xl font-semibold mb-4">About Me</h2>
        <p class="text-gray-600">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
        <p class="text-gray-600 mt-4">Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
      </div>
    </div>
  </div>
</section>
```

- d. Tambahkan kode berikut untuk membuat *my services*

```
<section class="p-16 bg-gray-200">
  <div class="container mx-auto">
    <h2 class="text-3xl font-semibold text-center mb-8">My Services</h2>
    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8">
      <div class="bg-white p-6 rounded-lg shadow-md flex items-center">
        <div class="bg-[#572DFF] rounded-full p-3">
          <svg class="text-white w-6 h-6" fill="none" stroke="currentColor" stroke-linecap="round" stroke-join="round"
          stroke-width="2" viewBox="0 0 24 24">
            <path d="M12 206.48 2 2 6.48 2 1254.48 10 10 10 4.48 10 10517.52 2 12 22m-1 15h-1v-2h2v22m0-4h-1v7h2v6z"></path>
          </svg>
        </div>
        <div class="ml-4">
          <h3 class="text-lg font-semibold">Service 1</h3>
          <p class="text-gray-600">Description of Service 1 goes here.</p>
        </div>
      </div>
      <div class="bg-white p-6 rounded-lg shadow-md flex items-center">
        <div class="bg-[#572DFF] rounded-full p-3">
          <svg class="text-white w-6 h-6" fill="none" stroke="currentColor" stroke-linecap="round" stroke-join="round"
          stroke-width="2" viewBox="0 0 24 24">
            <path d="M12 206.48 2 2 6.48 2 1254.48 10 10 10 4.48 10 10517.52 2 12 22m-1 15h-1v-2h2v22m0-4h-1v7h2v6z"></path>
          </svg>
        </div>
      </div>
    </div>
  </div>
```

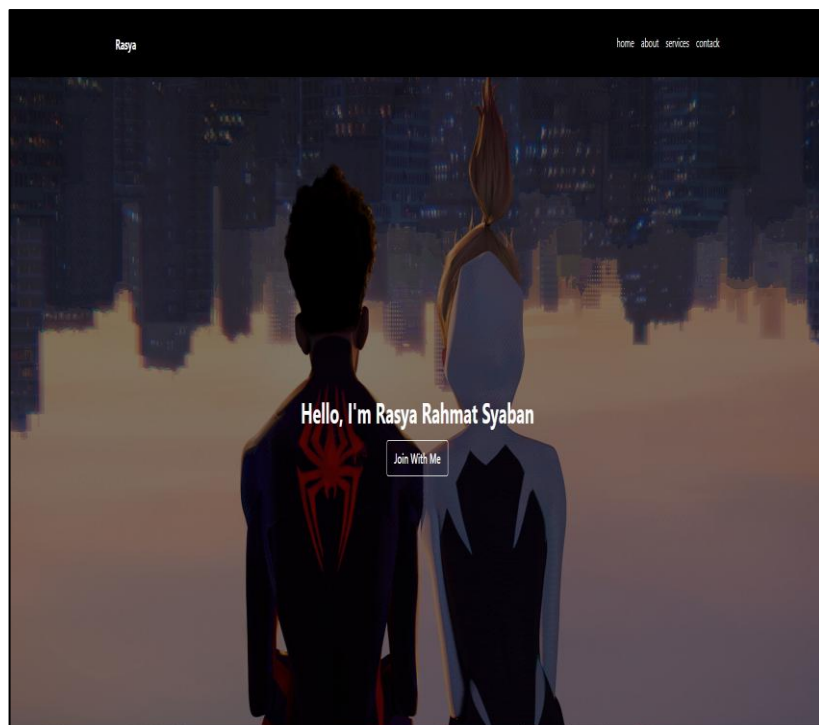

4.1.6 CSS Framework TailwindCSS\

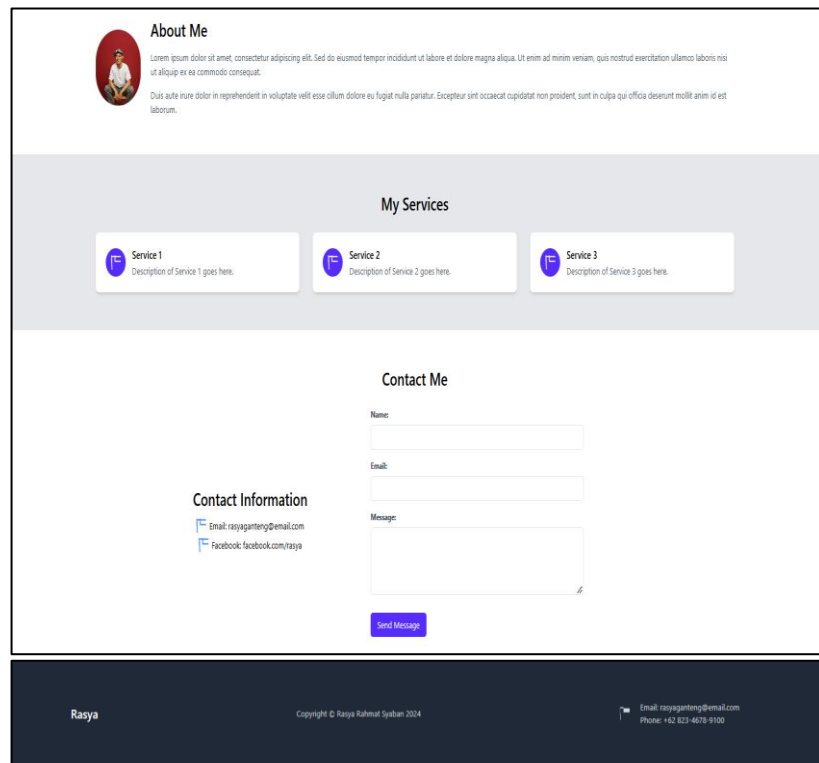
1. Tailwind CSS



Pada gambar diatas menampilkan hasil kode menggunakan *framework* TailwindCSS. Kode diatas di tulis menggunakan *tag* “<h1>” dengan kelas “*text-3xl font bold underline*” yang berguna untuk menentukan ukuran text dan juga *style* dari teks tersebut, yang menggunakan gaya tebal dan memiliki garis bawah.

2. Web Tailwind





Gambar di atas merupakan hasil kode yang dirancang menggunakan *framework* TailwindCSS untuk membuat antarmuka responsif dan menarik. Pada bagian navigasi, *tag* “<nav>” dengan kelas “bg-black p-4” menciptakan navigasi dengan latar belakang hitam dan memiliki *padding*. Di dalamnya, terdapat elemen “<div class=“container mx-auto”>” untuk memusatkan konten. Struktur navigasi menggunakan “<div class=“flex p-3 px-16 justify-between”>”, yang mengatur elemen agar berjajar menggunakan *flexbox*. Teks “Rasya” ditampilkan dalam “<a>” dengan kelas “text-white text-lg font-bold”, sementara tautan navigasi seperti “home”, “about”, dan lainnya ditempatkan dalam elemen “<div class=“flex space-x-4”>”, memberikan jarak horizontal di antara tautan. Kemudian pada bagian selanjutnya menggunakan *tag* “<div>” dengan kelas “relative bg-cover bg-center h-screen” untuk menampilkan gambar latar belakang yang mencakup seluruh layar. *Inline style* “background-image:

`url('img/wallpaper.jpg');`” menentukan gambar yang digunakan. Lapisan transparan hitam dibuat dengan “`<div class=`*absolute inset-0 bg-black opacity-50*`>`”, sementara teks "Hello, I'm Rasya Rahmat Syaban" ditampilkan di tengah halaman menggunakan elemen “`<h1>`” dengan kelas “`text-4xl font-bold mb-4`”. Kemudian pada bagian selanjutnya dimulai dengan tag “`<section>`” dengan kelas “`p-16`”, memberikan *padding* di sekitar konten. Elemen “`<div class=`*container mx-auto flex flex-col md:flex-row items-center justify-center*`>`” mengatur elemen secara fleksibel, baik dalam satu kolom di perangkat kecil atau dua kolom di perangkat lebih besar. Foto profil ditampilkan menggunakan tag “``” dengan kelas “`w-32 h-32 rounded-full object-cover`”, sedangkan deskripsi tentang diri berada dalam elemen “`<div>`” berisi “`<h2>`” dan paragraf teks. Kemudian pada bagian “*My Services*”, bagian ini menggunakan *grid layout* yang dibuat dengan “`<div class=`*grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-8*`>`”. Setiap layanan (*service*) diwakili oleh elemen “`<div>`” dengan kelas “`bg-white p-6 rounded-lg shadow-md flex items-center`”. Ikon berbentuk lingkaran ungu dengan SVG digunakan untuk representasi visual, diikuti dengan judul dan deskripsi layanan dalam elemen “`<div>`” lainnya. Kemudian pada bagian “Contact me”, bagian ini dimulai dengan tag “`<h2>`” dengan kelas “`text-3xl font-semibold text-center mb-8`” untuk menampilkan judul di tengah. Elemen “`<form>`” memungkinkan pengguna mengirim pesan. *Input* nama, *email*, dan pesan berada dalam elemen “`<div>`” dengan label masing-masing, diikuti dengan tombol kirim berbentuk persegi panjang dengan warna ungu. Kemudian pada bagian terakhir halaman menggunakan “`<footer>`” dengan kelas “`bg-gray-800 text-white p-16`”. Footer mencakup nama "Rasya", teks hak cipta, dan informasi kontak seperti *email* dan telepon.

BAB V

PENUTUP

5.1 Kesimpulan

5.1.1 *Basic Tag HTML Part 1*

Dari hasil percobaan sebelumnya, menunjukkan bahwa struktur dasar HTML terdiri dari berbagai elemen seperti *tag* HTML, *head*, dan *body* yang menentukan tampilan halaman *web*. Penggunaan *tag heading*, paragraf, dan *formatting* memberikan variasi dan penekanan pada teks, sedangkan *tag komentar* membantu pengembang untuk mencatat informasi penting dalam kode tanpa memengaruhi tampilan. *Hyperlink* memungkinkan navigasi antar halaman dan ke URL *eksternal*, sedangkan *tag gambar* menyediakan cara untuk menampilkan media visual dengan atribut yang mendukung. Semua elemen ini berkontribusi pada pembentukan halaman *web* yang interaktif dan informatif.

5.1.2 *Basic Tag HTML Part 2*

Dari percobaan sebelumnya, dapat disimpulkan bahwa HTML memiliki elemen-elemen dasar seperti *list*, tabel, dan *form input*. *List* terbagi menjadi *unordered list* dan *ordered list*, yang masing-masing menggunakan *tag* “” dan “” serta diisi dengan elemen “”. Tabel dibangun dengan baris dan kolom menggunakan *tag* “<tr>”, “<td>”, dan “<th>”, yang dapat diatur ukurannya. *Form input* menyediakan berbagai tipe *input* untuk data pengguna, seperti *text*, *email*, *number* dan lainnya, serta atribut *id* dan *name* untuk identifikasi dan pengumpulan data. Dengan elemen-elemen ini, dapat memungkinkan pembuatan struktur halaman yang interaktif dan fungsional.

5.1.3 *HTML Layout*

Dari percobaan ini, dapat disimpulkan bahwa materi HTML Layout ini adalah penggunaan *semantic tag* seperti “<nav>”, “<section>”, “<article>”, dan “<footer>” membantu memberi makna yang jelas pada struktur halaman. Setiap elemen dibungkus dalam *div* dengan atribut *id* untuk identifikasi yang spesifik, yang memudahkan penerapan gaya atau

interaksi dengan CSS atau JavaScript. Elemen-elemen seperti “<h1>”, “<h2>”, “”, dan “<p>” membantu membentuk konten dan navigasi dengan lebih terstruktur. Struktur ini membuat halaman lebih rapi, mudah dipahami, dan meningkatkan aksesibilitas bagi pengguna maupun mesin pencari.

5.1.4 CSS (*Cascading Style Sheets*)

Dari percobaan sebelumnya, dijelaskan penggunaan HTML untuk menyusun elemen-elemen teks dan CSS untuk mengatur tampilannya. Melalui *class* dan *selector*, setiap elemen dapat diberi gaya khusus seperti warna latar, ukuran teks, dan tata letak fleksibel menggunakan *flexbox*. Properti seperti *padding*, *margin*, dan *border* menambah struktur visual yang rapi, sementara *background-image* digunakan untuk memperindah tampilan dengan gambar latar.

5.1.5 CSS *Part 2 Flex & Grid*

Dari percobaan sebelumnya, dapat disimpulkan bahwa materi HTML *Layout* ini mengajarkan cara menggunakan *layout flexbox* dan *grid* untuk mengatur elemen-elemen di halaman *web* dengan cara yang lebih efisien dan responsif. Penggunaan elemen-elemen seperti “<h2>”, “<div class=’grid-container’>”, dan “<div class=’grid-item’>” membantu dalam menyusun konten dengan tata letak yang rapi dan terstruktur. Penerapan properti CSS seperti “*display: flex;*”, “*grid-template-columns;*”, dan “*grid-gap;*”. Struktur ini membuat halaman lebih teratur, mudah dinavigasi, dan responsif terhadap berbagai ukuran layar.

5.1.6 CSS *Framework TailwindCSS*

Dari percobaan sebelumnya, dapat disimpulkan bahwa modul CSS *Framework TailwindCSS* ini mengajarkan cara memanfaatkan kelas-kelas bawaan untuk mempercepat proses pengembangan *web* tanpa harus menulis CSS secara manual. Penggunaan kelas seperti “*bg-black;*”, “*flex;*”, “*grid;*”, “*text-white;*”, dan “*rounded-lg;*” mempermudah pembuatan elemen yang responsif, terstruktur, dan estetis. Modul ini juga menunjukkan fleksibilitas TailwindCSS dalam mengatur *layout*, warna, dan elemen interaktif seperti

tombol dan formulir. Dengan pendekatan berbasis *utility class*, TailwindCSS memungkinkan pembuatan halaman *web* yang modern, responsif, dan efisien.