

Tanggal Asistensi :	
Tanda Tangan	Nilai

**MODUL 5**  
***CSS Part 2 Flex & Grid***

**Nama : Imam Agil Aiman**

**Nim : F55123066**

**Kelas : B**

### 2.1.5 CSS Part 2 Flex & Grid

*Flex & Grid* adalah dua metode tata letak moderen dalam CSS yang dirancang untuk mengatur elemen *web* secara efisien. *Flexbox* berfokus pada tata letak satu dimensi (baris atau kolom) dan cocok untuk elemen yang fleksibel, seperti menu navigasi atau baris produk. Dengan properti seperti *justify-content* dan *align-items*, *Flexbox* mempermudah pengaturan posisi dan ruang antar elemen.

Sementara itu, *Grid Layout* dirancang untuk tata letak dua dimensi, memungkinkan pengaturan elemen dalam baris dan kolom secara bersamaan. Dengan properti seperti *grid-template-columns* dan *gap*, *Grid* sangat cocok untuk desain kompleks seperti halaman utama atau *dashboard*. CSS *Grid Layout* berupa teknik *layout* dua dimensi yang memungkinkan pengaturan elemen dalam bentuk baris (*rows*) dan kolom (*columns*). *Grid* memberikan fleksibilitas tinggi untuk membuat tata letak kompleks dengan lebih sederhana dibandingkan metode sebelumnya. CSS *Grid* mendukung berbagai properti untuk mengatur jumlah kolom, baris, ukuran masing-masing, jarak antar elemen *gap*, serta lokasi elemen dalam *grid*.

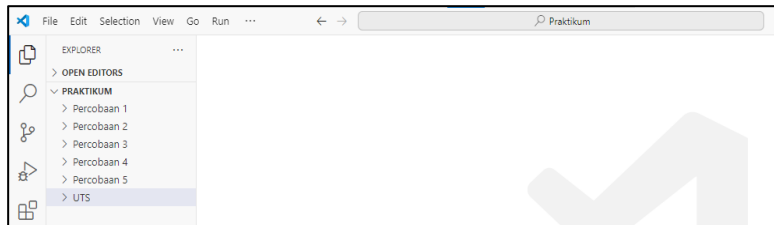
*Flexbox* ideal untuk elemen dinamis, sedangkan *Grid* unggul dalam tata letak terstruktur. Keduanya sering digunakan bersama untuk menciptakan desain *web* yang responsif dan moderen. Dan metode *layout* satu dimensi yang dirancang untuk menyusun elemen secara fleksibel dalam sumbu horizontal atau vertikal. *Flexbox* memberikan kontrol penuh atas perataan, distribusi ruang, dan urutan elemen, baik untuk layar yang besar maupun kecil. Konsep ini sangat berguna dalam membuat tata letak responsif karena sifatnya yang adaptif terhadap perubahan ukuran layar, propertinya memungkinkan pengembang untuk mengatur tata letak elemen.

Properti filter CSS berfungsi untuk menambahkan properti *Grid* pada CSS berfungsi untuk mengatur *layout* pada halaman *web*. Properti *display* berfungsi untuk mengatur bagaimana suatu elemen ditampilkan. Nilai *flex* pada properti *display* berfungsi untuk mengatur elemen-elemen menjadi sejajar kesamping dengan rapi (Putri and Aknuranda, 2022).

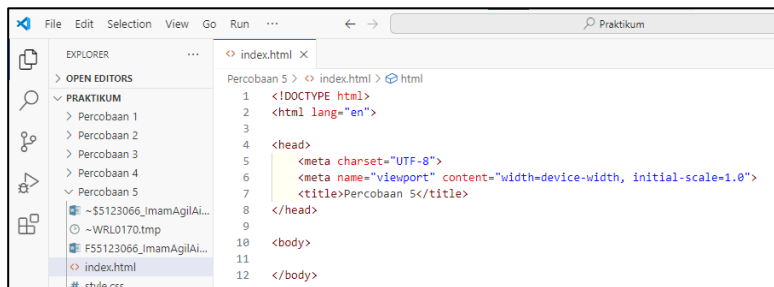
### 3.1.5 CSS Part 2 Flex & Grid

#### 1. Project

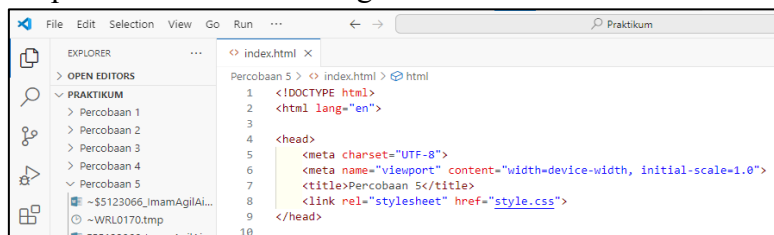
- a. Pada folder utama, buat folder baru dengan nama “Pertemuan-5”



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

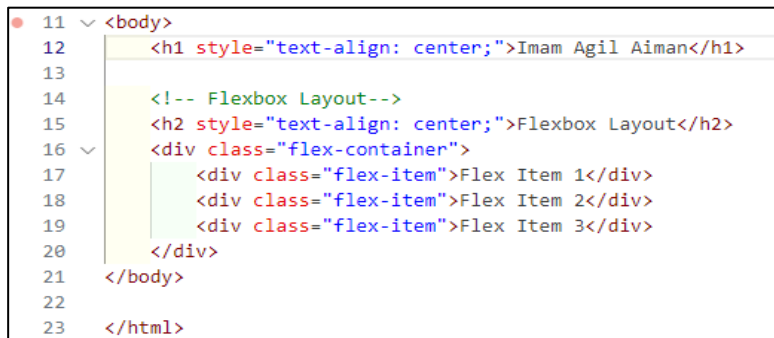


- c. Buat *file* baru dengan nama “style.css” dan tambahkan kode ini pada “index.html” di bagian *head*.



#### 2. Flexbox layout

- a. Tambahkan kode seperti berikut pada *file* HTML yang telah dibuat.



- b. Tambahkan juga kode seperti berikut pada *file* CSS yang telah dibuat.

```
Percobaan 5 > # style.css > .flex-item
1  h1 {
2      padding: 10px;
3  }
4
5  /* Flexbox */
6  .flex-container {
7      display: flex;
8      justify-content: space-evenly;
9      align-items: center;
10     background-color: whitesmoke;
11     padding: 20px;
12     border: 2px solid #333;
13 }
14
15 .flex-item {
16     background-color: #008f97;
17     color: white;
18     padding: 20px;
19     flex: 1;
20     margin: 5px;
21     text-align: center;
22     border-radius: 5px;
23 }
```

### 3. Grid layout

- a. Tambahkan lagi kode seperti berikut pada *file* HTML yang sama untuk membuat bagian *grid*.

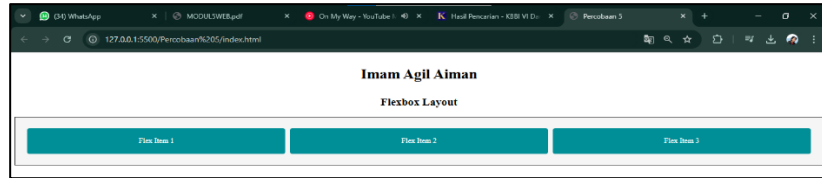
```
22  <!-- Grid Layout -->
23  <h2 style="text-align: center;">Grid Layout</h2>
24  <div class="grid-container">
25      <div class="grid-item">Grid 1</div>
26      <div class="grid-item">Grid 2</div>
27      <div class="grid-item">Grid 3</div>
28      <div class="grid-item">Grid 4</div>
29  </div>
30  </body>
```

- b. Tambahkan lagi kode seperti berikut pada *file* CSS nya untuk mengatur bagian *grid*.

```
25  /* Grid */
26  .grid-container {
27      display: grid;
28      grid-template-columns: repeat(3, 1fr);
29      grid-gap: 10px;
30      background-color: whitesmoke;
31      padding: 20px;
32      border: 2px solid #333;
33  }
34
35  .grid-item {
36      background-color: #008f97;
37      color: white;
38      padding: 20px;
39      text-align: center;
40      border-radius: 5px;
41  }
```

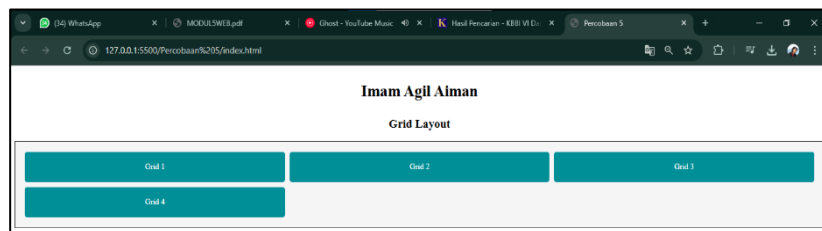
#### 4.1.5 CSS Part 2 Flex & Grid

##### 1. Flexbox layout



Pada percobaan pertama kali ini, berdasarkan gambar diatas, *layout* halaman untuk menampilkan beberapa elemen yang diatur menggunakan *flexbox*, elemen ditampilkan sejajar secara horizontal dengan ruang yang merata. *Style* “h1” “padding: 10px;” untuk memberikan jarak internal, *class* “*.flex-container*” yang didalamnya ada “*display: flex;*” untuk mengatur elemen agar memungkinkan tata letak elemen anaknya “*.flex-item*” menggunakan properti *flexbox*. Didalam “*.flex-item*” ada “*flex: 1;*” untuk menjadikan setiap elemen anak fleksibel, membagi ruang yang tersedia secara merata di dalam *flex container*.

##### 2. Grid layout



Pada percobaan *grid container* digunakan untuk elemen yang disusun dalam beberapa kolom dan baris, *class* “*.grid-container*” ada “*display: grid;*” untuk mengubah elemen menjadi sebuah *grid container*, yang memungkinkan elemen anaknya menggunakan sistem *grid*, “*grid-template-columns: repeat(3, 1fr);*” untuk membagi area *grid* menjadi 3 kolom yang memiliki lebar yang sama “*grid-gap: 10px;*” agar memberikan jarak sebesar 10px di antara kolom dan baris elemen. *Class* “*background-color: #008f97;*” untuk warna latar belakang biru kehijauan untuk setiap elemen *grid*. “*.grid-item*” ada “*text-align: center;*” untuk memusatkan teks secara horizontal.

## 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 memungkinkan 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 terhadap 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 estetik, 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 estetik. *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 efisien.