

**MODUL V**  
***CSS PART 2 FLEX & GRID***

### 2.1.5 CSS Part 2 Flex & Grid

*Flex & Grid* adalah dua metode tata letak modern 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.

Selain itu, salah satu kelebihan *Flexbox* adalah kemampuannya dalam mendistribusikan ruang secara proporsional, baik secara horizontal maupun vertikal. Hal ini sangat membantu ketika mengatur elemen-elemen dengan ukuran yang bervariasi namun tetap ingin menjaga keselarasan antar elemen tersebut. Misalnya, saat membuat daftar kartu produk yang memiliki deskripsi dengan panjang teks berbeda, *Flexbox* memastikan setiap kartu tetap terlihat rapi tanpa memengaruhi tata letak keseluruhan.

Di sisi lain, CSS *Grid* memberikan fleksibilitas lebih dalam mengatur tata letak kompleks dengan area tertentu yang telah ditentukan. Misalnya, *Grid* memungkinkan pembagian halaman menjadi *header*, *sidebar*, konten utama, dan *footer* dengan mudah menggunakan properti seperti *grid-template-areas*. Hal ini mengurangi kebutuhan penggunaan *float*, *positioning*, atau bahkan kombinasi elemen *wrapper* yang biasanya digunakan pada metode tata letak tradisional. Dengan *Grid*, desain menjadi lebih efisien dan mudah diatur ulang untuk berbagai ukuran layar.

*Flexbox* ideal untuk elemen dinamis, sedangkan *Grid* unggul dalam tata letak terstruktur. Keduanya sering digunakan bersama untuk menciptakan desain *web* yang responsif dan modern. Kolaborasi antara *Flexbox* dan *Grid* dapat menghasilkan desain yang sangat dinamis dan responsif.

### 3.1.5 CSS Part 2 Flex & Grid

#### 1. Project

- a. Buat *folder* dengan nama “Percobaan-5”



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

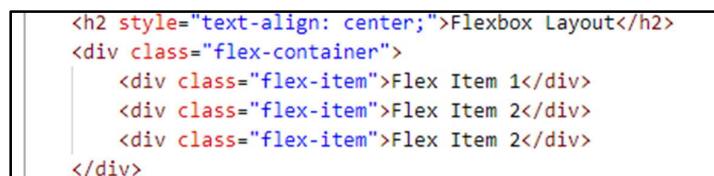


- c. Buat *file* baru dengan nama “*style.css*” dan hubungkan *file* nya dengan *file* HTML yang telah dibuat sebelumnya.



#### 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.

```
h1 {
  padding: 10px;
}

.flex-container {
  display: flex;
  justify-content: space-around;
  align-items: center;
  background-color: #whitesmoke;
  padding: 20px;
  border: 2px solid #333;
}

.flex-item {
  background-color: #008f97;
  color: white;
  padding: 20px;
  flex: 1;
  margin: 5px;
  text-align: center;
  border-radius: 5px;
}
```

### 3. Grid layout

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

```
<!-- Grid Layout -->
<h2 style="text-align: center;">Grid Layout</h2>
<div class="grid-container">
  <div class="grid-item">Grid 1</div>
  <div class="grid-item">Grid 2</div>
  <div class="grid-item">Grid 3</div>
  <div class="grid-item">Grid 4</div>
</div>
```

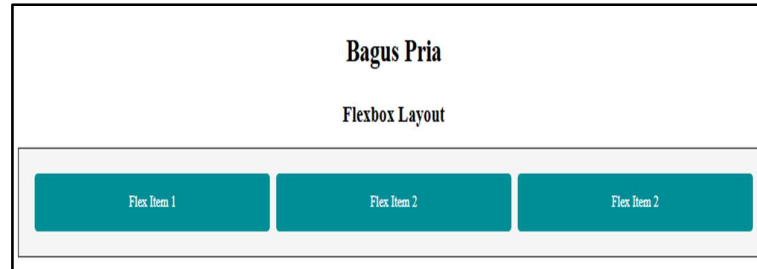
- b. Tambahkan lagi kode seperti berikut pada *file* CSS nya untuk mengatur bagian *grid*.

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  grid-gap: 10px;
  background-color: #whitesmoke;
  padding: 20px;
  border: 2px solid #333;
}

.grid-item {
  background-color: #008f97;
  color: white;
  padding: 20px;
  text-align: center;
  border-radius: 5px;
}
```

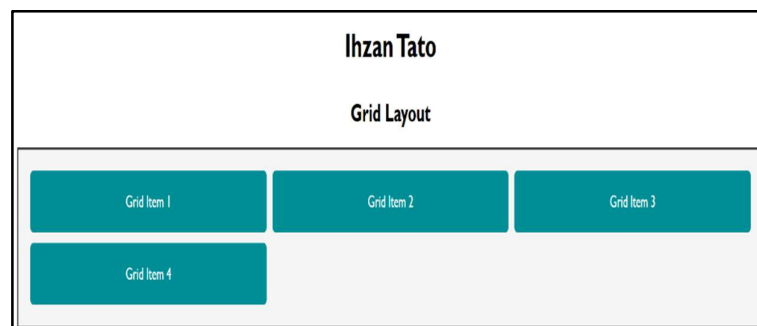
#### 4.1.5 CSS Part 2 Flex & Grid

##### 1. Flexbox layout



Pada gambar diatas merupakan hasil dari kode *Flexbox layout* . pada program diatas awalnya hanya *list* biasa saja namun dengan menambahkan *style* pada *file* "Style.css" maka tampilan yang tadinya hanya *list* biasa dapat terlihat lebih menarik. Kelas "*flex-container*" menggunakan properti "*display: flex*" untuk mengaktifkan *flexbox*, dengan "*justify-content: space-evenly*" untuk mendistribusikan elemen secara merata di sepanjang baris, sementara "*align-items: center*" menyelaraskan elemen secara vertikal

##### 2. Grid layout



Pada tampilan kali ini hampir sama dengan tampilan sebelumnya hanya saja pada kode ini memberikan batas 3 kolom saja kolom selanjutnya akan ditampilkan dibawah dan akan begitu seterusnya . Kelas "*grid-container*" diatur dengan properti "*display: grid*" untuk mengaktifkan *grid*, dan "*grid-template-columns: repeat(3, 1fr)*" membagi tata letak menjadi tiga kolom yang memiliki lebar yang sama.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

##### 5.1.1 Basic Tag HTML Part 1

Kesimpulan dari modul ini ialah mempelajari *tag basic* atau dasar dari HTML yang berisi mulai dari pemahaman mengenai pengertian *tag*, Langkah-langkah pembuatan suatu halaman *web* yang terstruktur mulai dari cara pemberian judul, sub-judul, *formatting text*, hingga cara berpindah dari halaman Utama kehalaman yang lain maupun ke situ atau url yang ada di *browser*.

##### 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

Kesimpulan dari modul ini adalah mempelajari dasar-dasar *layout* HTML dan berbagai elemen semantik HTML5 yang digunakan untuk mendefinisikan bagian-bagian halaman *web*, seperti “<header>“, “<nav>“, “<section>“, “<article>“, “<aside>“, dan “<footer>“. Modul ini juga mencakup teknik pembuatan *layout* dengan HTML dan CSS, serta penggunaan *framework* modern untuk menghasilkan tampilan *web* yang responsif dan dinamis. Meskipun *layout* berbasis tabel HTML pernah menjadi pilihan sederhana, penggunaan tabel untuk *layout* halaman kini tidak disarankan, karena tidak sefleksibel *layout* berbasis CSS yang lebih cocok untuk *web modern*. modul ini menekankan pentingnya menggunakan CSS untuk memastikan tata letak halaman yang rapi dan terstruktur.

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

Pada modul ini kita mempelajari tentang css yang berguna untuk membuat tampilan dari *web* yang kita buat menjadi lebih menarik. CSS memungkinkan pengembang *web* untuk mengatur elemen-elemen HTML, seperti teks, gambar, dan tata letak, dengan mengatur properti seperti warna, ukuran teks, jarak antar elemen, dan banyak lagi. Sehingga tampilan *web* kita lebih enak dilihat.

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

*Flexbox* dan *Grid* adalah dua metode tata letak dalam CSS yang saling melengkapi untuk menciptakan desain *web* yang efisien dan responsif. *Flexbox* unggul dalam tata letak satu dimensi, seperti mengatur elemen secara horizontal atau vertikal, sehingga cocok untuk elemen dinamis seperti menu atau daftar produk. Sebaliknya, *Grid* dirancang untuk tata letak dua dimensi, memungkinkan pengaturan elemen dalam baris dan kolom secara bersamaan, sehingga ideal untuk desain kompleks seperti halaman utama. Dengan kombinasi *Flexbox* dan *Grid*, pengembang dapat menghasilkan desain *web* yang fleksibel, terstruktur, dan mudah disesuaikan untuk berbagai ukuran layar.