|  |  |
| --- | --- |
| **Hari/Tanggal:** | |
| **Paraf** | **Nilai** |

**MODUL V**

**CSS *PART* 2 *FLEX* & *GRID***

**NAMA : Angelina Febriani Madesen**

**NIM : F55123084**

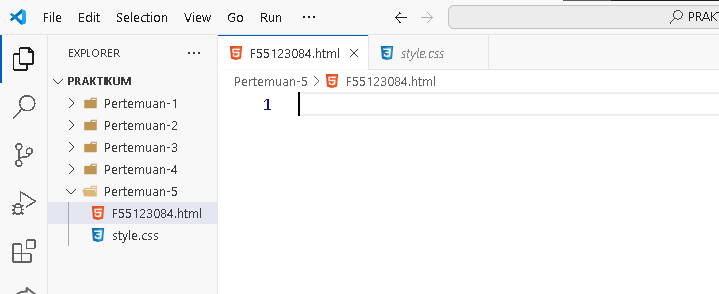
1. **CSS PART 2 *FLEX & GRID***

CSS (*Cascading Style Sheets*) adalah bahasa pemrograman yang digunakan dalam pengembangan *web* untuk mengontrol tampilan dan gaya visual dari halaman-halaman *web*. 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. Dengan CSS, pengembang dapat menciptakan desain yang menarik dan terstruktur, serta mengatur bagaimana halaman *web* merespons berbagai perangkat dan ukuran layar. CSS juga memungkinkan pemisahan antara konten (HTML) dan presentasi (gaya), sehingga perubahan pada tampilan dapat dilakukan tanpa mengubah struktur dasar halaman *web*. Ini membuat CSS menjadi alat yang sangat penting dalam menciptakan pengalaman pengguna yang konsisten dan menarik di seluruh *web*.

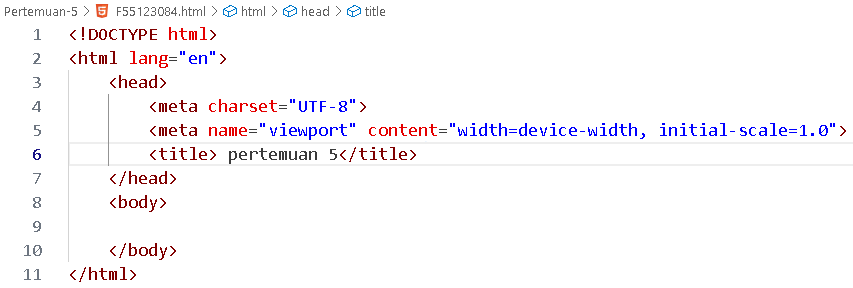
Selain itu, CSS juga mengadopsi prinsip *cascading*, yang berarti aturan gaya dapat ditentukan dengan tingkat kejelasan tertentu. Ini memungkinkan pengembang untuk menggabungkan aturan gaya dari berbagai sumber dan menentukan prioritasnya. Selain itu, CSS mendukung konsep warisan di mana aturan yang didefinisikan pada tingkat lebih tinggi dapat diterapkan pada elemen-elemen yang lebih dalam dalam struktur dokumen. Dengan kata lain, CSS memberikan fleksibilitas yang luar biasa dalam mengendalikan tampilan halaman *web* dan memastikan bahwa situs *web* dapat tampak baik dan berfungsi dengan baik di semua perangkat dan layar.

Salah satu komponen dasar dari CSS adalah prinsip *selectors*, yang digunakan untuk menargetkan elemen HTML berdasarkan tipe, kelas, *ID*, atau atribut spesifik. Dengan *selector* yang fleksibel ini, CSS dapat memodifikasi berbagai aspek visual, seperti warna, *font*, tata letak, dan animasi, yang sangat berperan dalam membuat halaman *web* lebih interaktif dan menarik secara visual. Dengan berbagai fitur yang fleksibel dan canggih, CSS menjadi pondasi dalam menciptakan situs *web* yang estetis, fungsional, dan ramah pengguna.

1. **CSS PART 2 *FLEX & GRID***
2. *Project*
3. Pada folder utama, buat folder baru dengan nama “Percobaan-5”



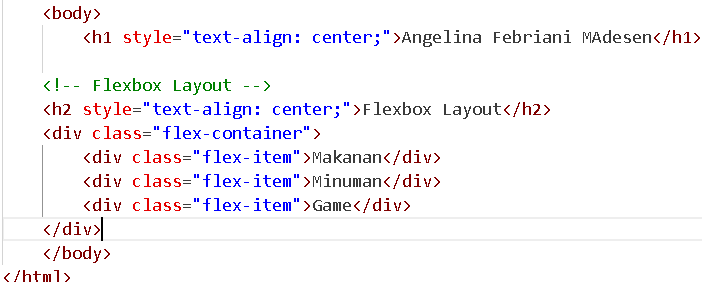
1. Pada folder “Pertemuan-5”, buat *file* baru dengan format nama “NIM.html” dan tambahkan struktur dasar HTML 5 ke dalamnya.



1. Buat *file* baru dengan nama “*Style*.css” dan hubungkan file nya dengan file HTML yang telah diuat sebelumnya.



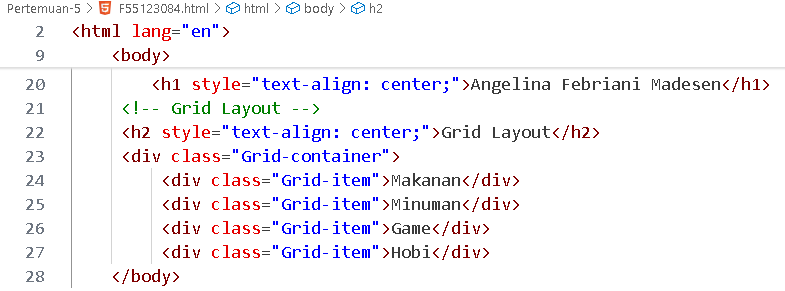
1. *Flexbox Layout*
2. Tambahkan kode seperti berikut pada file HTML yang telah dibuat.



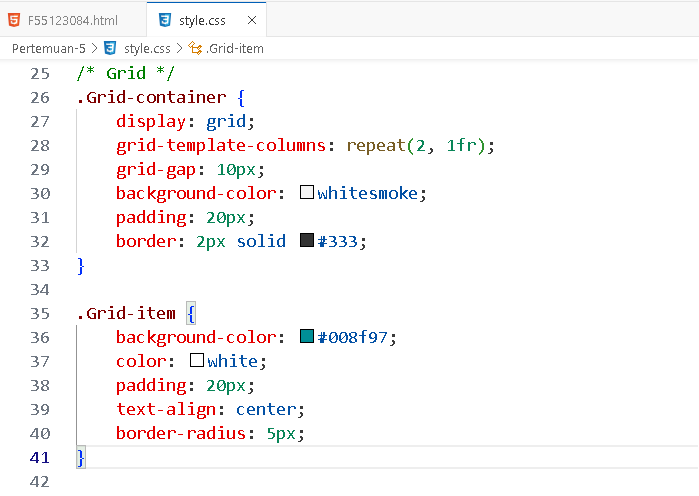
1. Tambahkan juga kode seperti beerikut ini pada file CSS yang telah dibuat.



1. *Grid Layout*
2. Tambahkan lagi kode seperti berikut ini pada file HTML yang sama untuk membuat again grid.



1. Tambahkan lagi kode seperti berikut ini pada file CSS nya untuk mengatur bagian grid.



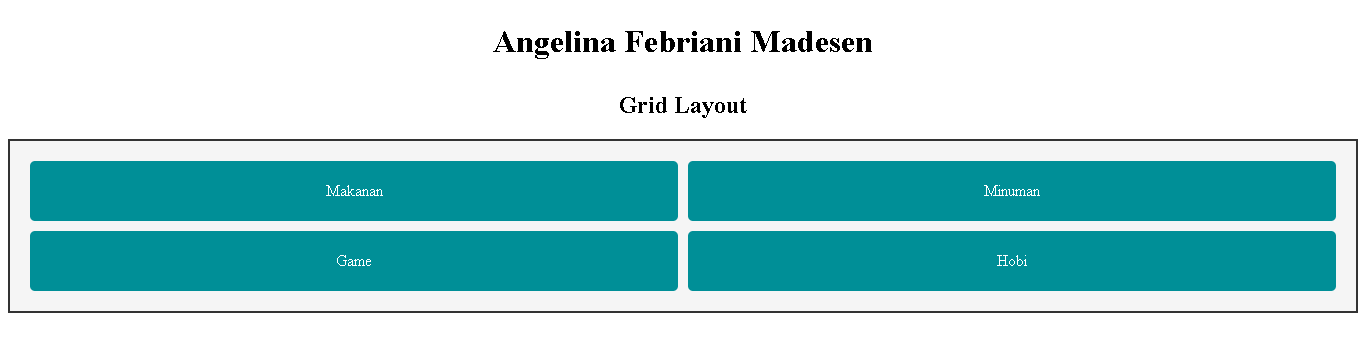
**4.1.5 CSS PART 2 FLEX & GRID**

1. *Flexbox layout*



Gambar di atas merupakan hasil dari implementasi percobaan *Flexbox Layout* yang dimana menuncukkan tata letak yang berisi tiga elemen berbentuk kotak disusun secara horizontal di dalam sebuah kontainer dengan label "Makanan", "Minuman", dan "Game", serta jarak merata antar elemen-elemen. Properti *Flexbox* seperti *justify-content* dan *align-items* digunakan untuk memusatkan elemen secara horizontal dan vertikal, sehingga menghasilkan tata letak yang seimbang.

1. *Grid layout*



Gambar di atas merupkan hasil dari implementasi percobaan *Grid Layout* pada gambar menunjukkan tata letak yang terstruktur, yang di mana empat elemen berbentuk persegi Panjang disusun dalam dua baris dan dua kolom yang dimana secara simetris atantar jaraknya. Teknik *CSS Grid* ini digunakan untuk mengatur elemen dalam bentuk kisi, dengan properti seperti *grid*-*template*-*columns* dan *grid*-*gap* yang memungkinkan distribusi elemen yang rapi dengan jarak yang konsisten.

* + 1. **CSS PART 2 *FLEX & GRID***

Dari percobaan yang sudah kita buat sebelumnya mengenai *Flexbox Layout* danjuga *Grid Layout* yang dimana kita dapat mengetahui bagaimana penggunaan kedua *layout* tersebut yang dimana dapat mempermudah kita untuk memisahkan antara *padding* dan juga *content* secara simetris. Selain itu juga dapat memisahkan antar *gap*-nya secara simetris antara satu dengan yang lain.