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

*Flex* dan *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.

*Flexbox* ideal untuk elemen dinamis, sedangkan *grid* unggul dalam tata letak terstruktur. Keduanya sering digunakan bersama untuk menciptakan desain *web* yang responsif dan modern.

*Flexbox* dan *grid* memiliki perbedaan mendasar, tetapi keunggulan keduanya dapat saling melengkapi dalam pembuatan desain *web*. *Flexbox* lebih fokus pada distribusi ruang dalam satu arah, sehingga fleksibel untuk menyusun elemen yang perlu disesuaikan dengan konten. Contoh penggunaan *flexbox* adalah untuk membuat *header* yang berisi logo, menu navigasi, dan tombol aksi yang tetap terlihat rapi di berbagai ukuran layar. Di sisi lain, CSS *grid* memberikan kendali penuh untuk membuat tata letak yang lebih kompleks dengan pengaturan area. Misalnya, *grid* memudahkan pembuatan tata letak *grid* responsif untuk galeri foto, di mana Anda bisa mendefinisikan ukuran kolom atau baris secara eksplisit. Selain itu, properti seperti *grid*-*template*-*areas* memberikan cara intuitif untuk mengatur posisi elemen berdasarkan nama area yang telah didefinisikan.

Ketika digunakan bersama, *flexbox* dan *grid* memungkinkan pengembang menciptakan tata letak yang responsif dan beragam. Misalnya, Anda dapat menggunakan *grid* untuk membuat kerangka utama sebuah halaman dan *flexbox* untuk menata detail elemen kecil di dalamnya. Ini memberikan kombinasi terbaik antara fleksibilitas dan struktur.

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

1. *Project*
2. Buat folder baru dengan nama “Percobaan-5”



1. Pada folder “Percobaan-5”, buat *file* baru dengan nama “NIM.html” dan tambahkan struktur dasar HTML5 ke dalamnya.



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



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



1. Pada *file* CSS tambahkan kode berikut.



1. *Grid layout*
2. Tambahkan lagi kode seperti berikut pada *file* HTML yang sama untuk membuat bagian *grid*.

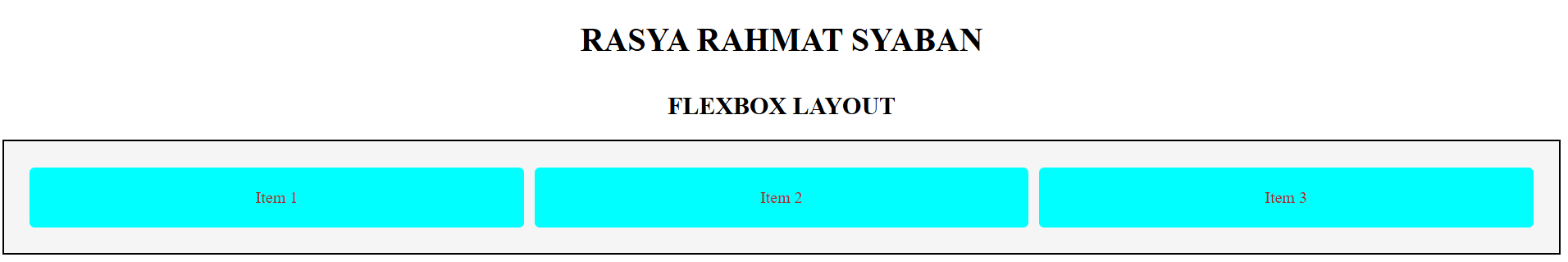


1. Tambahkan lagi kode seperti berikut pada *file* CSS nya untuk mengatur bagian *grid*.



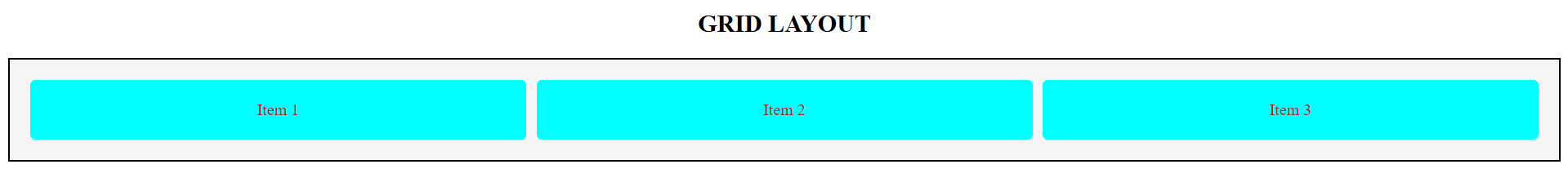
### **4.1.5 CSS *Part 2 Flex* & *Grid***

1. *Flexbox layout*



Pada kode di atas, terdapat dua elemen judul "*<h1>*" dan "*<h2>*", yang keduanya memiliki atribut “*style="text-align: center;"*” untuk memusatkan teks di halaman. Judul pertama adalah nama "RASYA RAHMAT SYABAN", dan subjudulnya adalah "*FLEXBOX LAYOUT*". Setelah itu, terdapat elemen "*<div class="flex-container">*" yang digunakan untuk mengelompokkan elemen-elemen lainnya menggunakan *layout* *flexbox*. Di dalam elemen ini terdapat tiga elemen "*<div class="flex-item">*", yang masing-masing berisi teks "Item 1", "Item 2", dan "Item 3". Elemen-elemen ini diatur dalam satu baris, dengan jarak yang merata berkat pengaturan *flexbox*. Pada bagian CSS, elemen "*<h1>*" ditambahkan “*padding: 10px;*” untuk memberi ruang di sekitar teks. Kemudian, untuk elemen dengan kelas "*flex-container*", properti “*display: flex;*” digunakan untuk mengubah *layout* menjadi *flexbox*. “*justify-content: space-evenly;*” mengatur agar elemen-elemen di dalamnya memiliki jarak yang merata, sementara “*align-items: center;*” memastikan elemen-elemen tersebut berada di tengah secara vertikal. Latar belakang elemen ini diubah menjadi warna abu-abu terang (*whitesmoke*), dan diberi “*padding: 20px;*” serta garis tepi hitam. Untuk elemen dengan kelas "*flex-item*", latar belakangnya diberi warna aqua, dan teks diwarnai coklat. “*padding: 20px;*” memberi ruang di dalam elemen, sedangkan “*flex: 1;*” memastikan setiap item memiliki lebar yang sama dan mengisi ruang yang tersedia. Dengan “*margin: 5px;*”, ada jarak antar elemen, dan “*border-radius: 5px;*” memberi sudut yang membulat pada setiap elemen.

1. *Grid layout*



Pada kode di atas, terdapat elemen "*<h2>*" dengan teks "*GRID LAYOUT*" yang memiliki atribut “*style="text-align: center;"*” untuk memusatkan teks di halaman. Setelah itu, terdapat elemen "*<div class="grid-container">*" yang digunakan untuk mengelompokkan elemen-elemen lainnya menggunakan *layout* *grid*. Di dalam elemen ini terdapat tiga elemen "*<div class="grid-item">*", yang masing-masing berisi teks "Item 1", "Item 2", dan "Item 3". Elemen-elemen ini diatur dalam *grid*, dengan jarak antar elemen yang teratur berkat pengaturan *grid Layout*. Pada bagian CSS, elemen "*grid-container*" diberi beberapa properti: *“display: grid;*” yang mengubah *layout* menjadi *grid*. Selanjutnya, “*grid-template-columns: repeat(3, 1fr);*” mengatur agar *grid* memiliki tiga kolom dengan lebar tiap kolomnya sama. Properti “*grid-gap: 10px;*” memberikan jarak sebesar 10 piksel antar elemen dalam *grid*. Latar belakang elemen ini diubah menjadi warna abu-abu terang (*whitesmoke*), dan diberi “*padding: 20px;*” serta garis tepi hitam. Untuk elemen dengan kelas "*grid-item*", latar belakangnya diberi warna aqua, dan teks diwarnai coklat. “*padding: 20px;*” memberi ruang di dalam elemen, dan “*text-align: center;*” memastikan teks berada di tengah elemen. Selain itu, “*border-radius: 5px;*” memberi sudut yang membulat pada setiap elemen.

# 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 simpulkan 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* “*<ul>”* dan “*<ol>”* serta diisi dengan elemen “*<li>”*. 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>”*, “*<ul>”*, 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.