

Bab 2 | Pengenalan CSS

2.1 Apa itu CSS?

CSS adalah singkatan dari Cascading Style Sheets. Ini adalah bahasa yang digunakan untuk mengatur tampilan dan gaya elemen-elemen HTML di halaman web. Dengan menggunakan CSS, kita dapat memisahkan presentasi dari struktur konten, memungkinkan kita membuat halaman yang lebih menarik dan mudah diubah tampilannya.

2.2 Menghubungkan CSS dengan HTML

Kita dapat menghubungkan file CSS eksternal dengan dokumen HTML kita menggunakan elemen `<link>` dalam bagian `<head>` dokumen HTML. Contoh:

```
<!DOCTYPE html>
<html>
<head>
  <title>Judul Halaman</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <!-- Konten HTML di sini -->
</body>
</html>
```

2.3 Styling Elemen HTML

Dalam file CSS, kita dapat memilih elemen HTML dan mengatur gayanya menggunakan properti dan nilai. Contoh penggunaan:

```
/* style.css */
p {
  color: blue;
  font-size: 16px;
  font-family: Arial, sans-serif;
}

h1 {
  color: green;
  text-align: center;
}
```

2.4 Kelas dan ID

Kita dapat memberikan kelas dan ID pada elemen untuk meng gaya elemen secara spesifik.

```
/* style.css */
.nama-kelas {
    font-weight: bold;
}

#id-khusus {
    color: red;
}
```

2.5 Tata Letak dan Posisi

Untuk mengatur tata letak dan posisi elemen di CSS, ada banyak properti yang bisa Anda gunakan. Di bawah ini adalah beberapa properti yang umum digunakan untuk mengatur tata letak dan posisi:

1. position: Properti ini mengendalikan cara elemen ditempatkan dalam halaman. Nilai-nilai yang umum digunakan adalah:
 - a. static (nilai default): Elemen ditempatkan sesuai alur dokumen.
 - b. relative: Elemen ditempatkan sesuai dengan posisi relatif terhadap posisi normalnya.
 - c. absolute: Elemen ditempatkan sesuai dengan posisi relatif terhadap elemen yang berada dalam konteks posisi.
 - d. fixed: Elemen tetap pada posisi layar, tidak peduli pergeseran halaman.
2. top, right, bottom, left: Digunakan bersamaan dengan position (kecuali static) untuk mengatur posisi tepat elemen.
3. display: Properti ini mengendalikan tampilan elemen. Beberapa nilai yang umum digunakan adalah:
 - a. block: Membuat elemen menjadi blok, yang akan menempati satu baris penuh.
 - b. inline: Membuat elemen menjadi inline, sehingga akan menempati sebagian baris.
 - c. inline-block: Menggabungkan sifat inline dan block.
 - d. flex: Membuat elemen mengikuti model tata letak flexbox.
 - e. grid: Membuat elemen mengikuti model tata letak grid.
4. float: Properti ini digunakan untuk mengatur elemen mengapung ke sisi tertentu dari elemen lain, seperti mengapungkannya ke kiri atau kanan. Ini sering digunakan untuk mengatur tata letak kolom dalam desain web.

5. margin dan padding: Properti ini mengendalikan jarak di sekitar elemen (margin) dan di dalam elemen (padding), yang mempengaruhi tata letak dan posisi elemen.
6. transform: Digunakan untuk melakukan transformasi seperti rotasi, pergeseran, dan perubahan skala pada elemen.

flexbox

```
.flex-container {  
  display: flex;  
  flex-direction: row; /* Elemen-elemen akan ditampilkan dalam satu baris */  
  justify-content: space-between; /* Jarak yang meratakan elemen */  
}
```

grid

```
.grid-container {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr); /* Membuat 3 kolom dengan lebar yang sama */  
  grid-gap: 10px; /* Jarak antara sel-sel grid */  
}
```

float dan clear

```
.left-floated {  
  float: left; /* Mengapungkan elemen ke kiri */  
}  
  
.right-floated {  
  float: right; /* Mengapungkan elemen ke kanan */  
}  
  
.clearfix::after {  
  content: "";  
  clear: both; /* Membersihkan elemen yang mengapung di sekitarnya */  
  display: table;  
}
```

z-index

```
.element1 {  
  z-index: 1; /* Lebih tinggi dalam tumpukan tampilan */  
}  
  
.element2 {  
  z-index: 2; /* Lebih tinggi dari element1 dalam tumpukan tampilan */  
}
```

```
/* style.css */  
.container {  
  width: 80%;  
  margin: 0 auto;  
  padding: 20px;  
}  
  
#header {  
  position: fixed;  
  top: 0;  
  left: 0;  
}
```

2.6 Responsif dengan Media Query

Responsif (atau responsivitas) dalam desain web merujuk pada kemampuan sebuah situs web atau aplikasi web untuk menyesuaikan tampilannya dengan baik pada berbagai perangkat dan ukuran layar yang berbeda. Media Query adalah salah satu teknik dalam CSS yang digunakan untuk mencapai responsivitas ini.

Media Query adalah aturan CSS yang memungkinkan kita mengubah tampilan dan tata letak elemen HTML berdasarkan karakteristik perangkat yang sedang digunakan, seperti lebar layar, orientasi (potrait atau landscape), resolusi, dan lainnya. Media Query memungkinkan kita memberikan instruksi khusus kepada browser untuk menerapkan gaya CSS tertentu hanya jika kondisi yang ditentukan dalam query terpenuhi.

Contoh penggunaan Media Query untuk membuat tampilan responsif:

```

/* Gaya dasar untuk semua Layar */
p {
    font-size: 16px;
}

/* Mengubah ukuran font pada Layar dengan Lebar kurang dari 600px */
@media (max-width: 600px) {
    p {
        font-size: 14px;
    }
}

/* Mengubah tata Letak pada Layar dengan Lebar lebih dari 1200px */
@media (min-width: 1200px) {
    p {
        margin: 20px;
    }
}

```

2.7 Transisi dan Animasi

Transisi CSS memungkinkan kita mengubah tampilan elemen HTML secara halus ketika ada perubahan dalam properti tertentu, seperti warna, ukuran, atau posisi. Kita dapat mengendalikan bagaimana perubahan tersebut terjadi dengan mengatur durasi, fungsi waktu (timing function), dan sifat perubahan (property) yang terlibat.

```

a {
    transition: color 0.3s ease;
}

a:hover {
    color: red;
}

```

Animasi CSS memungkinkan kita membuat pergerakan atau perubahan lebih kompleks pada elemen HTML dengan menggabungkan beberapa langkah atau frame animasi. Kita mendefinisikan keyframes yang berisi nilai-nilai properti yang berubah dari satu frame ke frame berikutnya, dan kemudian menerapkan animasi ini pada elemen HTML.

```
@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

.animated-element {
  animation: spin 2s linear infinite;
}
```

2.8 Pseudo-class dan Pseudo-element

Pseudo-class adalah istilah yang digunakan untuk memilih elemen HTML berdasarkan keadaan atau karakteristik tertentu yang tidak dapat diwakili oleh selektor kelas atau ID biasa. Pseudo-class diidentifikasi dengan tanda titik dua (:) diikuti oleh nama pseudo-class. Beberapa contoh pseudo-class yang umum digunakan:

1. `:hover`, Memilih elemen saat mouse mengarah ke atas elemen tersebut.
2. `:active`, Memilih elemen saat elemen tersebut sedang ditekan (klik).
3. `:focus`, Memilih elemen saat elemen tersebut memiliki fokus (biasanya digunakan untuk elemen formulir).
4. `:first-child`, Memilih elemen yang menjadi anak pertama dari elemen lain dalam hierarki.

Pseudo-element memungkinkan kita membuat dan memodifikasi bagian khusus dari elemen HTML, seperti bagian awal, akhir, atau elemen yang tertentu dalam elemen tersebut. Pseudo-element diidentifikasi dengan dua tanda titik dua (::) diikuti oleh nama pseudo-element. Beberapa contoh pseudo-element yang umum digunakan:

1. `::before`, Membuat elemen sebelum konten dalam elemen yang dipilih.
2. `::after`, Membuat elemen setelah konten dalam elemen yang dipilih.
3. `::first-line`, Memilih baris pertama dalam sebuah elemen teks.
4. `::first-letter`, Memilih huruf pertama dalam sebuah elemen teks.

```
/* style.css */
a:hover {
  text-decoration: underline;
}

p::first-line {
```

```
font-weight: bold;
}
```

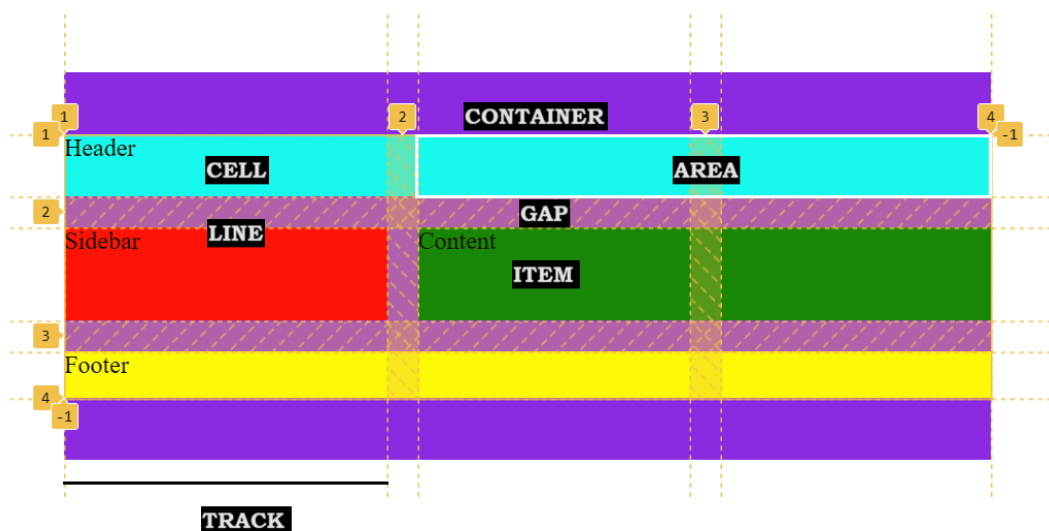
2.9 Grid

Grid dibuat untuk mendefinisikan sistem layout berbentuk grid dalam 2 dimensi (baris dan kolom).

Terminologi dalam grid:

1. Grid container, yaitu elemen pembungkus grid.
2. Grid item, yaitu elemen-elemen yang berada (1 level) di dalam grid container.
3. Grid line, yaitu garis horizontal (kolom) atau vertikal (baris) yang memisahkan grid menjadi beberapa bagian dan ditandai dengan angka.
4. Grid cell, yaitu perpotongan / pertemuan antara baris dan kolom di dalam grid.
5. Grid area, yaitu kumpulan lebih dari satu grid cell yang membentuk kotak.
6. Grid track, yaitu ukuran / jarak antara 2 grid line, bisa horizontal (kolom) atau vertikal (baris).
7. Grid gap, yaitu jarak antar grid track / cell.

Contoh:



```
<!-- grid.html -->
<div class="container">
  <div class="header">Header</div>
  <div class="sidebar">Sidebar</div>
  <div class="content">Content</div>
  <div class="footer">Footer</div>
</div>
```

```
/* style.css */
.container{
  width: 600px;
  margin: 50px auto;
  padding: 40px 0;
  display: grid;
  background-color: blueviolet;
  grid-template-areas:
    'header header header'
    'sidebar content content'
    'footer footer footer';
  grid-gap: 50px;
}
.header{
  height: 40px;
  grid-area: header;
  background-color: aqua;
}
.sidebar{
  height: 60px;
  grid-area : sidebar;
  background-color: red;
}
.content{
  height: 60px;
  grid-area: content;
  background-color: green;
}
.footer{
  height: 30px;
  grid-area: footer;
  background-color: yellow;
}
```