

# Bab 1 | Modul Pembelajaran HTML & CSS

## 1.1. Pengantar Website

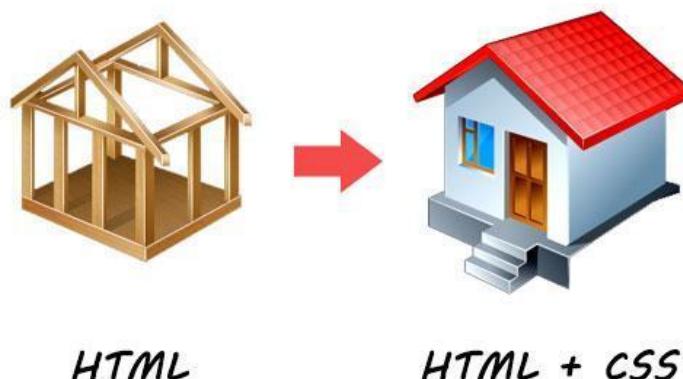
### 1.1.1. Apa itu Website?

Website adalah kumpulan informasi yang tersedia dalam sebuah komputer yang terhubung secara terus menerus melalui internet. Sebuah website dapat berisi informasi dalam bentuk apapun, seperti teks, gambar, audio, video, dan lain-lain.

### 1.1.2. Komponen Dasar Website

Untuk membangun website sederhana, kita memerlukan dua komponen utama:

- **HTML (HyperText Markup Language)**: Sebagai kerangka atau struktur website
- **CSS (Cascading Style Sheets)**: Sebagai pengatur tampilan dan gaya website



### 1.1.3. Web Server

Website biasanya berjalan pada aplikasi yang disebut Web Server. Ini adalah aplikasi yang digunakan untuk menyimpan dan mendistribusikan konten informasi web.

## 1.2. HTML as Skeleton

Pada bab ini kita akan mengenal dan mempelajari mengenai dasar-dasar HTML (*Hypertext Markup Language*). Namun sebelum itu, kita akan membahas mengenai situs web terlebih dahulu. Situs web merupakan kumpulan informasi yang tersedia dalam sebuah komputer yang terhubung secara terus menerus melalui internet. Sebuah situs web dapat berisi informasi dalam bentuk apapun, seperti teks, gambar, audio, video, dan lain-lain. Sebuah situs web biasanya berjalan pada aplikasi yang disebut Web Server. Ini adalah aplikasi yang digunakan untuk menyimpan dan mendistribusikan konten informasi web.

### 1.2.1. Pengenalan HTML

HTML (*Hypertext Markup Language*) adalah *markup language* standar yang sering digunakan untuk membuat halaman *website*. *Markup language* adalah bahasa komputer yang digunakan untuk menyusun hubungan antara berbagai bagian dokumen teks dengan bantuan simbol atau tag yang dimasukkan dalam dokumen. HTML mengatur dan mendefinisikan struktur, konten, dan tampilan halaman *website* dengan menggunakan tag (</>) untuk mendefinisikan elemen-elemen dasar. Tag-tag inilah yang akan memberitahu *browser* untuk menampilkan konten seperti teks, gambar, audio, video, dan lain-lain.

### 1.2.2. Struktur dasar HTML

Berikut ini adalah struktur dasar dari HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Document</title>
</head>
<body>
    <!-- Konten Website -->
</body>
</html>
```

1. <!DOCTYPE html> menginformasikan bahwa tipe dokumen tersebut adalah HTML.
2. <html> merupakan akar dari halaman HTML.
3. <head> merupakan tag yang berisi informasi tentang halaman HTML yang tidak ditampilkan di halaman web (meta-informasi).
4. <title> merupakan salah satu informasi yang dapat diisi di dalam tag ‘<head>’ yang menginformasikan judul yang ditampilkan pada tab browser.
5. <body> merupakan tag yang berisi informasi yang ditampilkan pada halaman web.

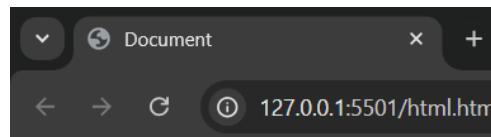
### 1.2.3. Elemen dan tag HTML

Elemen merupakan hal yang bertanggung jawab untuk membuat halaman web juga konten di dalamnya dan tag adalah kode yang membungkus elemen tersebut.

**contoh:**

```
<h1>Ini adalah judul</h1>
<p>ini adalah paragraf</p>
```

**Output:**



## Ini adalah judul

ini adalah paragraf

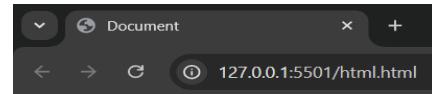
Contoh di atas merupakan salah satu elemen yang memiliki tag di awal dan di akhir untuk membungkus konten, namun ada juga elemen yang tidak memiliki konten dan tidak mempunyai tag di akhir yang biasa disebut *Empty Element* atau *Void Element*.

**contoh:**

```
<p>Tini adalah <br> makanan sapi</p>
<p>Jalan Sore</p>
<hr>
```

```
<p>Dengan Amoy</p>
```

### Output:



Tini adalah  
makanan sapi

Jalan Sore

---

Dengan Amoy

<br> dan <hr> merupakan salah satu contoh elemen yang tidak memiliki konten di dalamnya. '<br>' berfungsi untuk membuat baris baru di dalam HTML dan '<hr>' berfungsi untuk membuat garis datar memanjang secara horizontal.

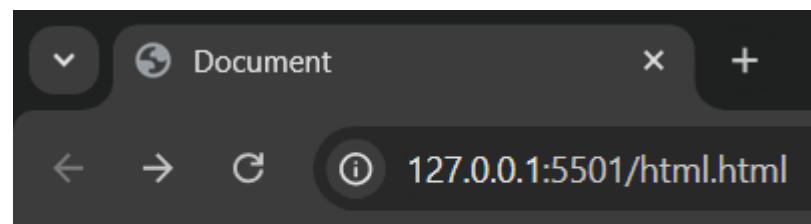
#### 1.2.4. Atribut HTML

Atribut HTML merupakan informasi tambahan dari sebuah elemen dan selalu berada pada tag awal.

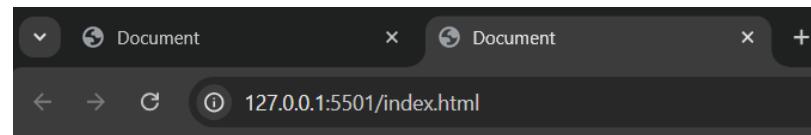
##### Contoh:

```
<a href="index.html" target="_blank">pindah halaman ke index.html</a>
```

### Output:



pindah halaman ke index.html



## ini adalah halaman index.html

Elemen '<a>' berfungsi untuk membuat tautan yang menghubungkan dari satu halaman ke halaman yang lain. *Href* berfungsi menentukan *URL*

(Uniform Resource Locator) yang dituju. Target berfungsi untuk mendefinisikan tempat di mana halaman tersebut akan dibuka, salah satu contohnya “\_blank” yang akan membuat tab baru untuk membuka halaman tersebut.

Ada beberapa contoh attribut lainnya seperti :

- **id="kontainer"** → identitas unik elemen.
- **class="box utama"** → bisa dipakai untuk CSS/JS.
- **style="color: red;"** → inline CSS.
- **title="Kotak Utama"** → teks tooltip saat hover.

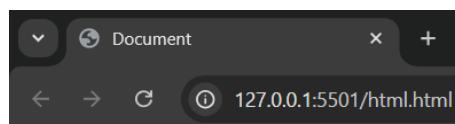
### 1.2.5. Format dan Semantik HTML

HTML menyediakan berbagai elemen untuk menambahkan format dan semantik ke dalam konten.

**Contoh:**

```
<h1>ini adalah h1</h1>
<h2>ini adalah h2</h2>
<h3>ini adalah h3</h3>
<h4>ini adalah h4</h4>
<h5>ini adalah h5</h5>
<h6>ini adalah h6</h6>
<strong>tebal</strong>
<em>miring</em>
```

**Output:**



**ini adalah h1**

**ini adalah h2**

**ini adalah h3**

**ini adalah h4**

**ini adalah h5**

**ini adalah h6**

**tebal miring**

1. `<h1>` hingga `<h6>` untuk judul tingkat berbeda.
2. `<strong>` untuk penekanan yang kuat (tebal).
3. `<em>` untuk teks yang ditekankan (miring).

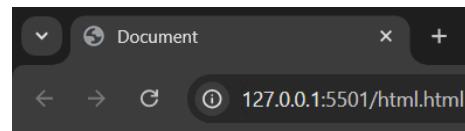
### 1.2.6. Daftar (*List*) HTML

*List* atau daftar dalam HTML merupakan kumpulan item atau poin yang dikelompokkan bersama. Ada 3 jenis *list* yang ditawarkan HTML, yaitu *Ordered List*, *Unordered List*, dan *Description List*.

**Contoh:**

```
<h4>Ordered List</h4>
<ol>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
</ol>
<h4>Unordered List</h4>
<ul>
    <li>Merah</li>
    <li>Biru</li>
    <li>Kuning</li>
</ul>
<h4>Description List</h4>
<dl>
    <dt>HTML</dt>
    <dd>- Hyper Text Markup Language</dd>
    <dt>CSS</dt>
    <dd>- Cascading Style Sheet</dd>
    <dt>JavaScript</dt>
    <dd>- JavaScript</dd>
</dl>
```

### **Output:**



#### **Ordered List**

1. HTML
2. CSS
3. JavaScript

#### **Unordered List**

- Merah
- Biru
- Kuning

#### **Description List**

HTML  
    - Hyper Text Markup Language  
CSS  
    - Cascading Style Sheet  
JavaScript  
    - JavaScript

### **1.2.7. Multimedia HTML**

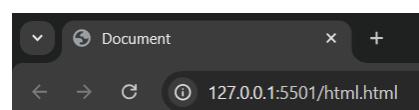
HTML memungkinkan untuk memasukkan berbagai jenis media ke dalam halaman web, seperti gambar, audio, dan video.

#### **Contoh gambar:**

```

```

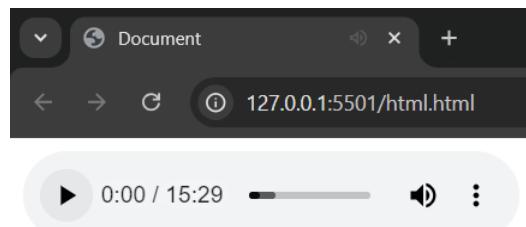
### **Output:**



**Contoh audio:**

```
<audio controls>
    <source src="assets/Audio/Littleroot town Pokemon omega
ruby.mp3"      type="audio/mpeg">
</audio>
```

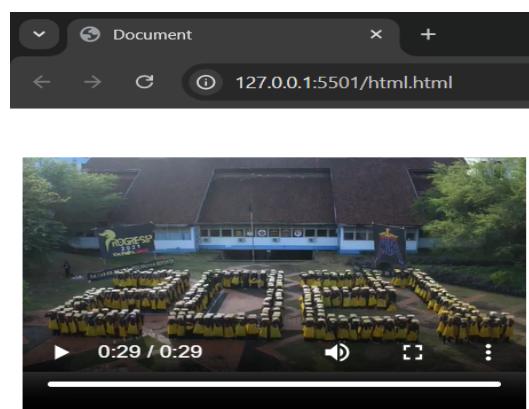
**Output:**



**Contoh Video:**

```
<video width="320" height="240" controls>
    <source src="assets/Video/WhatsApp Video 2022-01-10 at
21.36.41.mp4" type="video/mp4">
</video>
```

**Output:**



### 1.2.8. Formulir (*Form*) HTML

Formulir atau *form* dalam HTML merupakan elemen yang digunakan untuk mengumpulkan data yang dimasukkan oleh pengguna *website*.

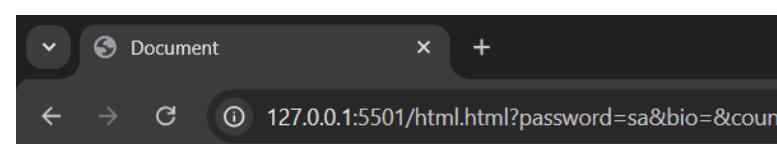
**Contoh:**

```

<form>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required>
    <br>
    <label>Gender:</label>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Male</label>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Female</label>
    <br>
    <label for="bio">Bio:</label>
    <textarea id="bio" name="bio" rows="4" cols="50"></textarea>
    <br>
    <label for="country">Country:</label>
    <select id="country" name="country">
        <option value="Sunda Empire">Sunda Empire</option>
        <option value="Gowa-Tallo">Gowa-Tallo</option>
        <option value="Sriwijaya">Sriwijaya</option>
    </select>
    <br>
    <input type="submit" value="Submit">
</form>

```

**Output:**



The screenshot shows a web browser window with the title "Document". The address bar displays the URL "127.0.0.1:5501/html.html?password=sa&bio=&country=Sunda%20Empire". Below the address bar is a form with the following fields:

- Password:** A text input field.
- Gender:** Two radio buttons labeled "Male" and "Female".
- Bio:** A large text area for entering bio information.
- Country:** A dropdown menu currently set to "Sunda Empire".
- Submit:** A submit button at the bottom of the form.

1. **<form>** : Elemen ini mendefinisikan formulir.

2. **<label>** : Elemen ini digunakan untuk menambahkan teks deskriptif untuk elemen formulir lainnya. Atribut "for" digunakan untuk menghubungkan label dengan elemen formulir.
3. **<input>** : Elemen ini digunakan untuk membuat berbagai jenis kontrol input, seperti:
  - **type="password"**: Membuat kotak input untuk kata sandi. Nilai yang dimasukkan akan disembunyikan.
  - **type="radio"**: Membuat tombol radio. Hanya satu tombol radio dalam grup dapat dipilih sekaligus.
  - **type="submit"**: Membuat tombol pengiriman formulir.
  - **required** : Membuat agar elemen wajib di isi
4. **<textarea>** : Elemen ini digunakan untuk membuat area teks multi baris. Atribut "rows" dan "cols" digunakan untuk menentukan ukuran area teks.
5. **<select>** : Elemen ini digunakan untuk membuat daftar dropdown. Atribut "name" digunakan untuk memberikan nama pada elemen.
6. **<option>** : Elemen ini digunakan untuk menentukan opsi dalam daftar dropdown.

### 1.2.9. Tabel HTML

Tabel merupakan elemen HTML yang memungkinkan untuk mengatur data dalam format yang terstruktur dan terorganisasi. Fungsi utama tabel HTML adalah membantu pembuat *website* menampilkan informasi dalam format *grid* yang dapat mempermudah pengguna memahami dan menginterpretasikan data.-

**Contoh:**

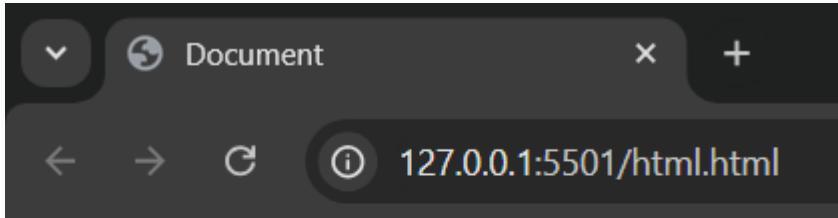
```
<table border="1">
<tr>
<th>Mahasiswa</th>
<th>Jenis Kelamin</th>
<th>Jurusan</th>
</tr>
```

```

<tr>
    <td>Adhe</td>
    <td>perempuan</td>
    <td>Matematika</td>
</tr>
<tr>
    <td>Cia</td>
    <td>Perempuan</td>
    <td>Ilmu Aktuaria</td>
</tr>
</table>

```

**Output:**



Mahasiswa	Jenis Kelamin	Jurusan
Adhe	perempuan	Matematika
Cia	Perempuan	Ilmu Aktuaria

1. **<table>** : Tag ini mendefinisikan sebuah tabel.
2. **<tr>** : Tag ini mendefinisikan sebuah baris (*row*) dalam tabel. Setiap baris akan berisi satu atau lebih sel.
3. **<th>** : Tag ini mendefinisikan sebuah sel kepala (*header*) dalam tabel. Sel kepala biasanya digunakan untuk judul kolom dan memiliki tampilan yang sedikit berbeda (misalnya, teks tebal dan rata tengah).
4. **<td>** : Tag ini mendefinisikan sebuah sel data (*data cell*) dalam tabel. Sel data berisi konten utama dari tabel, seperti nama, nilai, atau informasi lainnya.
5. **border="1"**: Atribut ini menambahkan *border* (garis tepi) pada tabel dengan ketebalan 1 piksel.

## 1.3. Pengenalan CSS

### 1.3.1. Apa itu CSS?

CSS adalah singkatan dari Cascading Style Sheets yang biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti HTML. Dengan menggunakan CSS, kita dapat memisahkan konten dari tampilan visualnya, sehingga memungkinkan kita untuk membuat laman website yang lebih menarik dan mudah diubah tampilannya.

### 1.3.2. Cara Menggunakan CSS

Kita dapat menambahkan CSS ke halaman dengan beberapa cara berbeda. Ada tiga cara dalam menggunakan CSS pada file HTML

#### 1. Inline

Untuk menggunakan inline CSS, kita dapat menambahkan aturan CSS langsung ke elemen HTML menggunakan atribut style.

Contoh:

```
<p style="color: blue;">Ini adalah teks berwarna biru</p>
```

Inline CSS memungkinkan kontrol spesifik untuk suatu elemen. Namun, cara ini dapat menyebabkan duplikasi kode dan kesulitan dalam pemeliharaan.

#### 2. Internal

Untuk menggunakan internal CSS, kita dapat menulis aturan CSS di dalam tag <style> di bagian <head> dokumen HTML.

Contoh:

```
<!DOCTYPE html>

<html>
<head>
    <style>
        p {
            color: green;
            font-size: 18px;
        }
    </style>
</head>
<body>
    <p>Ini adalah teks berwarna hijau dan ukuran font 18px.</p>
</body>
```

```
</html>
```

Internal CSS dapat memudahkan pengelolaan gaya/*styling* untuk seluruh halaman dari satu tempat. Akan tetapi gaya/*styling* hanya berlaku untuk halaman tersebut dan dapat meningkatkan ukuran file HTML.

### 3. External

Untuk menggunakan External CSS ,kita dapat Menulis aturan CSS dalam file terpisah dengan ekstensi .css dan menghubungkannya ke dokumen HTML menggunakan tag <link> di bagian <head>.

Contoh:

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

External CSS mempermudah pengelolaan gaya/*styling* di seluruh halaman web dari satu file dan membuat kode HTML lebih kecil, rapi, dan terorganisir.

#### 1.3.3. Styling Elemen HTML

Di dalam file CSS, kita dapat menggunakan elemen HTML dan mengatur gayanya menggunakan properti dan nilai. Berikut adalah contohnya:

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

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

#### 1.3.4. Kelas dan ID

Kita dapat memberikan kelas dan ID pada elemen HTML untuk memberikan gaya ke suatu elemen HTML secara spesifik. Berikut adalah contohnya:

```
<html>
<head>
    <title>Contoh Kelas dan ID</title>
    <style>
        /* Kelas */
        .highlight {
            color: yellow;
            background-color: black;
        }

        /* ID */
        #unique {
            color: red;
            font-size: 24px;
        }
    </style>
</head>
<body>
    <p class="highlight">Teks ini memiliki gaya dengan kelas "highlight".</p>
    <p class="highlight">Teks ini juga memiliki gaya dengan kelas "highlight".</p>
    <p id="unique">Teks ini memiliki gaya dengan ID "unique".</p>
</body>
</html>
```

Perlu diperhatikan bahwa sebuah ID harus bersifat unik dan hanya boleh digunakan sekali dalam satu dokumen. ID digunakan untuk mengidentifikasi elemen secara spesifik, dan memiliki lebih dari satu elemen dengan ID yang sama dapat menyebabkan masalah dalam pengaturan gaya dan pemrograman. Sebaiknya gunakan kelas untuk gaya yang diterapkan pada beberapa elemen.

#### 1.3.5. Konsep Dalam CSS

##### 1. The Cascade (Rule Order)

Konsep The Cascade menunjukkan bahwa aturan styling pada CSS diterapkan dari atas ke bawah. Artinya, jika ada beberapa aturan dengan selector yang sama, aturan yang terakhir ditulis akan diutamakan.

Contoh:

```
h1 {  
    color: green;  
}  
  
h1 {  
    color: red;  
}
```

## 2. Specificity

Specificity menentukan bahwa selector yang lebih spesifik akan menggesampingkan selector yang kurang spesifik.

Contoh:

```
body h1 {  
    color: green;  
}  
  
h1 {  
    color: red;  
}
```

## 3. Inheritance

Inheritance dalam CSS berarti bahwa beberapa properti akan diwariskan dari elemen induk ke elemen anak. Misalnya, jika font-family diatur pada elemen body, semua elemen di dalamnya akan mewarisi gaya tersebut.

Contoh:

```
<html>  
<head>  
    <title>Website Pertama</title>  
    <style>  
        body {  
            font-family: Arial;  
        }  
    </style>
```

```
</head>
<body>
  <h1>Halo Konoha</h1>
  <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit</p>
</body>
</html>
```

#### 4. Group Selector

Group Selector memungkinkan kita mengaplikasikan aturan yang sama ke beberapa elemen dengan menggabungkan selector menggunakan koma. Hal ini dapat mengurangi pengulangan kode.

Contoh:

```
h1, p {
  color: blue;
}
```

#### 1.3.6. 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 mengatur cara elemen ditempatkan dalam halaman. Nilai-nilai yang umum digunakan:
  - a. static (nilai default): Jenis ini adalah posisi bawaan untuk semua elemen web. Jika Kita tidak menentukan jenis posisi apa pun, elemen tersebut akan memiliki static position secara default.
  - b. relative: Elemen ditempatkan sesuai dengan posisi normalnya, tetapi dapat digeser relatif terhadap posisi aslinya menggunakan properti top, right, bottom, atau left. Sehingga memungkinkan Kita untuk memindahkan elemen tanpa mengubah posisi elemen lainnya.
  - c. absolute: Menggunakan absolute position berarti kita mengambil elemen keluar dari normal flow dan apabila tidak

ada elemen parent yang memiliki position:relative maka elemen yang memiliki position:absolute letak dan posisinya akan bergantung pada elemen dokumen atau tag html.

- d. fixed: Elemen dengan fixed position akan tetap berada di posisi yang sama, bahkan ketika kita men-scroll halaman.

Contoh penggunaan fixed:

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

- e. sticky: jenis ini adalah kombinasi dari posisi relative dan fixed. Elemen akan bergerak seperti biasa sampai mencapai titik tertentu, lalu akan "menempel" seperti fixed position.
- 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: Elemen mengambil satu baris penuh dan memulai baris baru setelahnya. Contoh: <div>, <p>.
  - b. inline: Elemen hanya menempati ruang yang diperlukan dan tidak memulai baris baru. Contoh: <span>, <a>.
  - c. inline-block: Menggabungkan sifat elemen block dan inline; elemen berada di baris yang sama namun dapat memiliki dimensi lebar dan tinggi.
  - d. flex: Membuat elemen mengikuti model tata letak flexbox, memungkinkan penataan fleksibel dari elemen dalam baris atau kolom.
  - e. grid: Membuat elemen mengikuti model tata letak grid, memungkinkan penataan elemen dalam baris dan kolom dengan struktur grid.

4. float: Properti ini digunakan untuk mengatur elemen mengapung ke sisi tertentu dari elemen lain, seperti mengapungkannya ke kiri atau kanan. Float sering digunakan untuk mengatur tata letak kolom dalam desain web.

Contoh 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;  
}
```

5. Box model: Box model di CSS menggambarkan struktur setiap elemen HTML sebagai kotak yang terdiri dari konten, padding, border, dan margin. Content adalah area di dalam kotak yang berisi konten elemen, padding memberikan jarak antara konten dan border, border mengelilingi padding dan konten, dan margin menciptakan ruang di luar border untuk memisahkan elemen dari elemen lain.



6. transform: Digunakan untuk melakukan transformasi seperti rotasi, pergeseran, dan perubahan skala pada elemen.

7. z-index: Properti z-index digunakan untuk mengatur urutan tumpukan (stacking order) elemen-elemen yang tumpang tindih di halaman untuk menentukan elemen mana yang akan berada di atas atau di bawah elemen lainnya ketika elemen-elemen tersebut saling bertumpuk. z-index hanya berfungsi pada elemen dengan properti position yang memiliki nilai selain static (yaitu relative, absolute, fixed, atau sticky).

Contoh:

```
.box1 {  
    position: relative; /* Elemen ini diatur relatif terhadap posisinya semula */  
    z-index: -1; /* Elemen ini berada di bawah elemen lain dengan z-index lebih tinggi */  
    background-color: lightblue;  
    width: 100px;  
    height: 100px;  
}  
.box2 {  
    position: relative;  
    z-index: 1; /* Elemen ini berada di atas elemen dengan z-index lebih rendah, seperti .box1 */  
    background-color: lightgray;  
    width: 100px;  
    height: 100px;  
}
```

#### 1.3.7. Responsif dengan Media Query

Responsif (atau responsiveness) 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 (portrait atau landscape), 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;
}
}
```

### 1.3.8. 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;
}
```

### 1.3.9. 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.

### 1.3.10. Flexbox

Flexbox (Flexible Box Layout) memudahkan pembuatan tata letak responsif dan fleksibel tanpa bergantung pada float atau positioning. Untuk menggunakan Flexbox, pertama-tama buat elemen container dengan display:

flex, lalu atur elemen anak di dalamnya. Contoh penggunaan Flexbox dalam HTML adalah sebagai berikut:

```
<!-- html -->
<div class="container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
</div>
```

```
/* css */
.container {
    display: flex;
    background-color: DodgerBlue;
}
.container > div {
    background-color: #f1f1f1;
    margin: 10px;
    padding: 20px;
    font-size: 30px;
}
```



Berikut beberapa property Flex yang dapat digunakan dalam container:

1. **flex-direction:** Menentukan arah penumpukan item dalam kontainer.
  - row: Horizontal dari kiri ke kanan.
  - row-reverse: Horizontal dari kanan ke kiri.
  - column: Vertikal dari atas ke bawah.
  - column-reverse: Vertikal dari bawah ke atas.
2. **flex-wrap:** Mengatur apakah item harus dibungkus ke baris baru.
  - wrap: Membungkus item ke baris baru.

- nowrap: Tidak membungkus item (semua item dalam satu baris).
  - wrap-reverse: Membungkus item ke baris baru secara terbalik.
3. justify-content: Menyelaraskan item di sepanjang sumbu utama
    - center: Menyelaraskan item di tengah.
    - flex-start: Menyelaraskan item ke awal sumbu utama.
    - flex-end: Menyelaraskan item ke akhir sumbu utama.
    - space-between: Ruang sama rata di antara item.
    - space-around: Ruang sama rata di sekitar item.
  4. align-items: Menyelaraskan item di sepanjang sumbu silang.
    - flex-start: Menyelaraskan item ke awal sumbu silang.
    - flex-end: Menyelaraskan item ke akhir sumbu silang.
    - center: Menyelaraskan item di tengah sumbu silang.

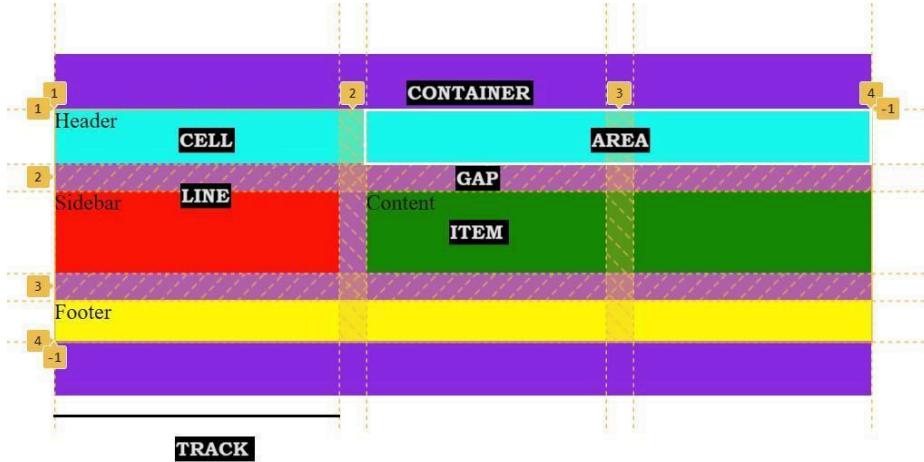
### 1.3.11. Grid

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

Beberapa 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-a
reas:
    '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;
}
```

