

Modul Ajar Web Programming Dasar (HTML & CSS)

1. Tujuan Pembelajaran

2. Materi dan Pembahasan

2.1. Pengenalan HTML

2.2. Pengenalan CSS

3. Langkah-langkah Praktikum

4. Latihan

1.1. Tujuan Pembelajaran HTML

Setelah mempelajari bagian ini, peserta diharapkan mampu:

- Memahami struktur dasar dokumen HTML.
- Menggunakan berbagai elemen HTML untuk membuat konten web.
- Membuat halaman web sederhana menggunakan HTML.
- Memahami konsep dasar tag, atribut, dan elemen dalam HTML.

2.1.1. Apa itu HTML?

HTML (HyperText Markup Language) adalah bahasa standar yang digunakan untuk membuat halaman web. HTML mendefinisikan struktur dan konten dari sebuah halaman web. Dengan HTML, Anda dapat membuat teks, gambar, tautan, dan elemen multimedia lainnya muncul di browser web Anda. HTML bukanlah bahasa pemrograman; ini adalah bahasa markup yang menggunakan serangkaian 'tag' untuk memberi tahu browser bagaimana menampilkan konten.

2.1.2. Struktur Dasar Dokumen HTML

Setiap dokumen HTML memiliki struktur dasar yang terdiri dari beberapa elemen penting. Berikut adalah contoh struktur dasar dokumen HTML:

```
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Judul Halaman Saya</title>
</head>
<body>
```

```
<h1>Selamat Datang di Halaman Web Saya</h1>
<p>Ini adalah paragraf pertama saya.</p>
</body>
</html>
```

Penjelasan Kode:

- `<!DOCTYPE html>` : Deklarasi ini mendefinisikan bahwa dokumen ini adalah dokumen HTML5.
- `<html lang="id">` : Elemen `<html>` adalah elemen root dari setiap halaman HTML. Atribut `lang="id"` menunjukkan bahwa bahasa utama dokumen adalah Bahasa Indonesia.
- `<head>` : Bagian `<head>` berisi metadata tentang halaman HTML, seperti judul halaman, set karakter, dan tautan ke file CSS atau JavaScript eksternal. Konten di dalam `<head>` tidak ditampilkan langsung di browser.
 - `<meta charset="UTF-8">` : Menentukan set karakter untuk dokumen, memastikan teks ditampilkan dengan benar.
 - `<meta name="viewport" content="width=device-width, initial-scale=1.0">` : Mengatur tampilan halaman agar responsif di berbagai perangkat.
 - `<title>Judul Halaman Saya</title>` : Menentukan judul halaman yang akan muncul di tab browser atau jendela.
- `<body>` : Bagian `<body>` berisi semua konten yang terlihat oleh pengguna di browser, seperti teks, gambar, tautan, dan lainnya.
 - `<h1>Selamat Datang di Halaman Web Saya</h1>` : Elemen `<h1>` digunakan untuk judul utama (heading 1).
 - `<p>Ini adalah paragraf pertama saya.</p>` : Elemen `<p>` digunakan untuk paragraf teks.

2.1.3. Tag-tag HTML Umum

HTML memiliki banyak tag yang digunakan untuk berbagai tujuan. Berikut adalah beberapa tag HTML yang paling umum digunakan:

1. Heading (Judul)

Digunakan untuk membuat judul atau sub-judul. Ada enam tingkat heading, dari `<h1>` (terbesar) hingga `<h6>` (terkecil).

```
<h2>Ini adalah Sub-Judul</h2>
<h3>Ini adalah Sub-Sub-Judul</h3>
<h4>Ini adalah Sub-Sub-Sub-Judul</h4>
<h5>Ini adalah Sub-Sub-Sub-Sub-Judul</h5>
<h6>Ini adalah Sub-Sub-Sub-Sub-Sub-Judul</h6>
```

2. Paragraf (`<p>`)

Digunakan untuk membuat paragraf teks.

```
<p>Ini adalah contoh paragraf. Kita bisa menambahkan teks panjang di sini.</p>  
<p>Setiap tag p akan membuat paragraf baru.</p>
```

3. Tautan (`<a>`)

Digunakan untuk membuat hyperlink yang mengarahkan pengguna ke halaman lain atau bagian lain dari halaman yang sama.

```
<a href="https://www.google.com/search?q=google">Google</a>  
<a href="halaman-lain.html">Pergi ke Halaman Lain</a>  
<a href="#bagian-bawah">Lompat ke Bagian Bawah</a>
```

- `href` : Atribut ini menentukan URL tujuan tautan.

4. Gambar (``)

Digunakan untuk menyisipkan gambar ke dalam halaman web. Tag ini adalah *self-closing* (tidak memiliki tag penutup).

```

```

- `src` : Atribut ini menentukan jalur (path) ke file gambar.
- `alt` : Atribut ini memberikan teks alternatif untuk gambar, yang akan ditampilkan jika gambar tidak dapat dimuat atau untuk pembaca layar (aksesibilitas).
- `width` dan `height` : Atribut ini menentukan lebar dan tinggi gambar dalam piksel.

5. Daftar (List)

Ada dua jenis daftar utama:

- **Daftar Tak Berurut (`` - Unordered List)**: Menggunakan simbol (bullet points).

```
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
</ul>
```

- **Daftar Berurut (- Ordered List)**: Menggunakan angka atau huruf.

```
<li>Langkah Pertama</li>
<li>Langkah Kedua</li>
<li>Langkah Ketiga</li>
</ol>
```

- ****: Digunakan untuk setiap item dalam daftar.

6. Penekanan Teks

- ****: Untuk teks yang penting (biasanya tebal).
- ****: Untuk teks yang ditekankan (biasanya miring).

```
<p>Ini adalah teks <strong>penting</strong> dan ini adalah teks <em>yang ditekankan</em>.</p>
```

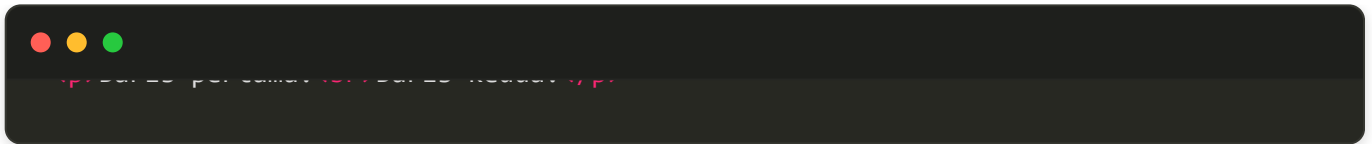
7. Garis Horizontal (<hr>)

Digunakan untuk membuat garis horizontal sebagai pemisah konten. Tag ini juga *self-closing*.

```
<p>Konten di atas garis.</p>
<hr>
<p>Konten di bawah garis.</p>
```

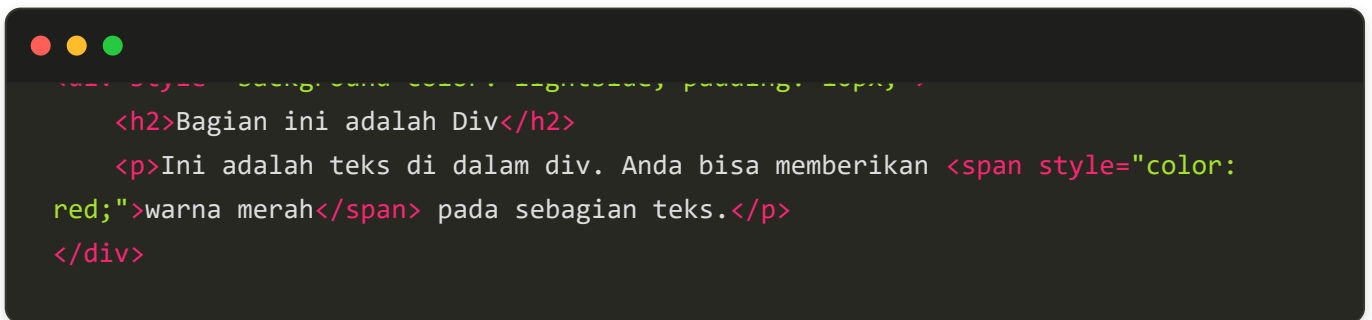
8. Baris Baru (
)

Digunakan untuk membuat baris baru dalam teks tanpa memulai paragraf baru. Tag ini juga *self-closing*.



9. Div (`<div>`) dan Span (``)

- **<div>** : Elemen *block-level* yang digunakan sebagai wadah generik untuk mengelompokkan elemen-elemen lain. Biasanya digunakan untuk tata letak halaman.
- **** : Elemen *inline-level* yang digunakan sebagai wadah generik untuk mengelompokkan bagian kecil dari teks atau elemen *inline* lainnya.



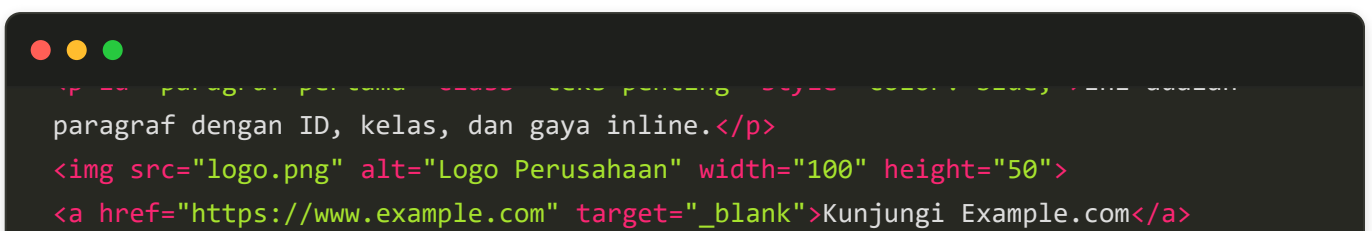
2.1.4. Atribut HTML

Atribut memberikan informasi tambahan tentang elemen HTML. Atribut selalu ditentukan di tag pembuka dan biasanya datang dalam pasangan nama/nilai seperti: `nama="nilai"`.

Beberapa atribut umum yang sering digunakan:

- **id** : Menentukan ID unik untuk elemen. ID harus unik dalam satu halaman HTML.
- **class** : Menentukan satu atau lebih nama kelas untuk elemen. Kelas dapat digunakan oleh CSS untuk menata gaya elemen dengan kelas yang sama, atau oleh JavaScript untuk memanipulasi elemen dengan kelas yang sama.
- **style** : Menentukan gaya *inline* untuk elemen (misalnya, warna, ukuran font, dll.). Penggunaan atribut **style** secara langsung tidak disarankan untuk styling yang kompleks; lebih baik menggunakan CSS eksternal atau internal.
- **href** : Digunakan pada tag **<a>** untuk menentukan URL tujuan tautan.
- **src** : Digunakan pada tag **** untuk menentukan jalur gambar.
- **alt** : Digunakan pada tag **** untuk teks alternatif gambar.

Contoh Penggunaan Atribut:



- `target="_blank"` : Atribut ini pada tag `<a>` akan membuka tautan di tab atau jendela browser baru.

2.1.5. Semantic HTML

Semantic HTML adalah penggunaan elemen HTML untuk memperkuat makna atau tujuan informasi di dalam halaman web, bukan hanya untuk presentasi. Dengan menggunakan elemen semantik, Anda membantu browser, mesin pencari, dan teknologi bantu (seperti pembaca layar) untuk memahami struktur dan konten halaman Anda dengan lebih baik.

Beberapa elemen semantik HTML5 yang umum:

- `<header>` : Mewakili bagian pengantar atau navigasi untuk sebuah dokumen atau bagian.
- `<nav>` : Berisi tautan navigasi.
- `<main>` : Berisi konten utama yang unik untuk dokumen.
- `<article>` : Berisi konten independen yang dapat didistribusikan secara mandiri (misalnya, posting blog, berita).
- `<section>` : Mengelompokkan konten terkait, biasanya dengan judul.
- `<aside>` : Berisi konten yang secara tidak langsung terkait dengan konten utama (misalnya, sidebar).
- `<footer>` : Mewakili bagian footer untuk sebuah dokumen atau bagian.

Contoh Penggunaan Semantic HTML:

```
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contoh Semantic HTML</title>
</head>
<body>

  <header>
    <h1>Judul Situs Web Saya</h1>
    <nav>
      <ul>
        <li><a href="#">Beranda</a></li>
        <li><a href="#">Tentang Kami</a></li>
        <li><a href="#">Kontak</a></li>
      </ul>
    </nav>
```

```
</header>

<main>
  <section>
    <h2>Tentang HTML Semantik</h2>
    <p>HTML semantik membantu dalam strukturisasi konten web.</p>
  </section>

  <article>
    <h3>Posting Blog Terbaru</h3>
    <p>Ini adalah isi dari postingan blog.</p>
  </article>
</main>

<aside>
  <h4>Informasi Tambahan</h4>
  <p>Ini adalah sidebar dengan informasi terkait.</p>
</aside>

<footer>
  <p>&copy; 2025 Modul Web Dasar</p>
</footer>

</body>
</html>
```

Penggunaan elemen semantik tidak hanya membuat kode Anda lebih mudah dibaca dan dipelihara, tetapi juga meningkatkan SEO (Search Engine Optimization) dan aksesibilitas situs web Anda.

1.2. Tujuan Pembelajaran CSS

Setelah mempelajari bagian ini, peserta diharapkan mampu:

- Memahami peran CSS dalam desain web.
- Menggunakan berbagai metode untuk menyertakan CSS dalam dokumen HTML.
- Menggunakan selektor CSS dasar untuk menargetkan elemen HTML.
- Mengaplikasikan properti CSS untuk mengubah tampilan elemen.
- Memahami konsep Box Model dalam CSS.

2.2.1. Apa itu CSS?

CSS (Cascading Style Sheets) adalah bahasa *stylesheet* yang digunakan untuk mendeskripsikan presentasi dokumen yang ditulis dalam bahasa markup seperti HTML. CSS memungkinkan Anda untuk mengontrol tampilan dan tata letak halaman web, termasuk warna, font, tata letak, dan banyak lagi. Dengan CSS, Anda

dapat memisahkan konten (HTML) dari presentasi (CSS), yang membuat pengembangan web lebih efisien dan mudah dikelola.

2.2.2. Cara Menyertakan CSS

Ada tiga cara utama untuk menyertakan CSS dalam dokumen HTML:

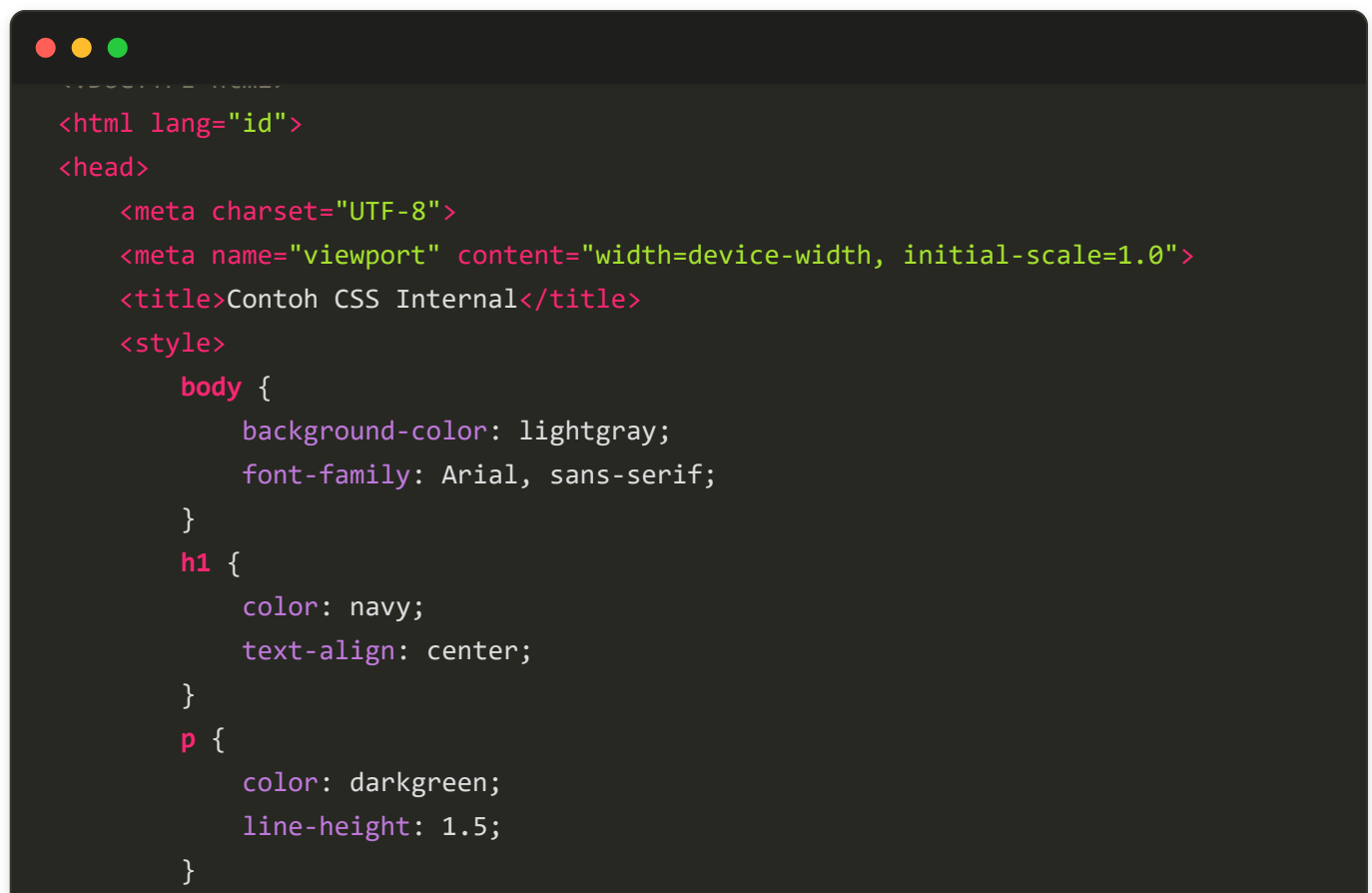
1. CSS Inline

CSS inline digunakan untuk menerapkan gaya unik pada satu elemen HTML. Ini dilakukan dengan menggunakan atribut `style` langsung di dalam tag HTML. Meskipun mudah untuk penggunaan cepat, metode ini tidak disarankan untuk styling yang kompleks karena mencampur konten dengan presentasi dan sulit untuk dikelola.



2. CSS Internal (Embedded CSS)

CSS internal digunakan untuk menerapkan gaya pada satu halaman HTML. Kode CSS ditempatkan di dalam tag `<style>` di bagian `<head>` dokumen HTML. Metode ini berguna jika Anda memiliki gaya unik untuk satu halaman saja.




```
</style>
</head>
<body>
  <h1>Selamat Datang</h1>
  <p>Ini adalah contoh penggunaan CSS internal.</p>
</body>
</html>
```

3. CSS Eksternal

CSS eksternal adalah metode yang paling umum dan disarankan untuk menyertakan CSS. Gaya disimpan dalam file `.css` terpisah (misalnya, `style.css`) dan dihubungkan ke dokumen HTML menggunakan tag `<link>` di bagian `<head>`. Metode ini memungkinkan Anda untuk menggunakan kembali gaya yang sama di banyak halaman HTML, membuat situs web lebih mudah dikelola dan diperbarui.

File `index.html` :

```
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Contoh CSS Eksternal</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Selamat Datang</h1>
  <p>Ini adalah contoh penggunaan CSS eksternal.</p>
  <button>Klik Saya</button>
</body>
</html>
```

File `style.css` :

```
body {
  background-color: #f0f0f0;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

h1 {
```

```
    color: #333;
    text-align: center;
    margin-top: 50px;
}

p {
    color: #666;
    font-size: 16px;
    text-align: justify;
    padding: 0 20px;
}

button {
    background-color: #007bff;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    display: block;
    margin: 20px auto;
}

button:hover {
    background-color: #0056b3;
}
```

- `rel="stylesheet"` : Menentukan hubungan antara dokumen HTML dan file yang ditautkan (dalam hal ini, stylesheet).
- `href="style.css"` : Menentukan lokasi (path) file CSS eksternal.

2.2.3. Selektor CSS

Selektor CSS digunakan untuk "menemukan" atau "memilih" elemen HTML yang ingin Anda gayakan. Setelah elemen dipilih, Anda dapat menerapkan properti CSS ke elemen tersebut.

Berikut adalah beberapa jenis selektor CSS yang umum:

1. Selektor Elemen (Type Selector)

Memilih semua elemen dari jenis tertentu.

```
color: blue;
font-family: sans-serif;
}

h1 {
  font-size: 32px;
  text-decoration: underline;
}
```

2. Selektor ID (#id)

Memilih elemen dengan atribut `id` tertentu. ID harus unik dalam satu halaman HTML, sehingga selektor ID akan menargetkan hanya satu elemen.

```
background-color: #f8f8f8;
padding: 20px;
}

#logo {
  width: 150px;
  border-radius: 50%;
}
```

3. Selektor Kelas (.class)

Memilih semua elemen dengan atribut `class` tertentu. Kelas dapat digunakan kembali pada banyak elemen, sehingga selektor kelas dapat menargetkan beberapa elemen.

```
background-color: green;
color: white;
padding: 10px 15px;
border: none;
}

.warning-text {
  color: red;
}
```

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

4. Selektor Universal (`*`)

Memilih semua elemen HTML di halaman.

```
margin: 0;
padding: 0;
box-sizing: border-box;
}
```

5. Selektor Atribut (`[attribute]`)

Memilih elemen berdasarkan atributnya.

```
color: orange;
}

input[type="text"] {
  border: 1px solid gray;
  padding: 5px;
}
```

6. Selektor Kombinator

- **Descendant Selector (spasi):** Memilih elemen yang merupakan turunan dari elemen lain.

```
color: purple;
}

/* Akan menargetkan semua paragraf di dalam div */
```

- **Child Selector (`>`):** Memilih elemen yang merupakan anak langsung dari elemen lain.

```
list-style-type: square;
}
/* Akan menargetkan item daftar yang merupakan anak langsung dari ul */
```

- **Adjacent Sibling Selector (+)**: Memilih elemen yang langsung mengikuti elemen lain.

```
margin-top: 0;
}
/* Akan menargetkan paragraf yang langsung mengikuti h1 */
```

- **General Sibling Selector (~)**: Memilih elemen yang merupakan saudara dari elemen lain (tidak harus langsung).

```
background-color: yellow;
}
/* Akan menargetkan semua paragraf yang merupakan saudara dari h1 */
```

2.2.4. Properti CSS Umum

Setelah memilih elemen dengan selektor, Anda dapat menerapkan berbagai properti CSS untuk mengubah tampilannya. Setiap properti memiliki nama dan nilai, dipisahkan oleh titik dua (:), dan diakhiri dengan titik koma (;).

Berikut adalah beberapa properti CSS yang paling umum digunakan:

1. Warna (Color)

- **color**: Mengatur warna teks.
- **background-color**: Mengatur warna latar belakang elemen.

```
color: #333; /* Warna teks abu-abu gelap */
background-color: #f9f9f9; /* Warna latar belakang abu-abu muda */
}
```

```
h1 {  
  color: rgb(255, 0, 0); /* Warna teks merah menggunakan RGB */  
}  
  
body {  
  background-color: lightblue; /* Warna latar belakang biru muda */  
}
```

2. Teks (Text)

- **font-family** : Mengatur jenis font (misalnya, Arial, Helvetica, sans-serif).
- **font-size** : Mengatur ukuran font (misalnya, 16px, 1.2em, 120%).
- **font-weight** : Mengatur ketebalan font (misalnya, normal, bold, 400, 700).
- **text-align** : Mengatur perataan teks (left, right, center, justify).
- **text-decoration** : Menambahkan dekorasi teks (underline, overline, line-through, none).
- **line-height** : Mengatur tinggi baris.
- **letter-spacing** : Mengatur jarak antar huruf.
- **word-spacing** : Mengatur jarak antar kata.

```
--- {  
  font-family: 'Times New Roman', serif;  
  font-size: 24px;  
  font-weight: bold;  
  text-align: center;  
  text-decoration: underline;  
}  
  
p {  
  line-height: 1.6;  
  letter-spacing: 0.5px;  
}
```

3. Model Kotak (Box Model)

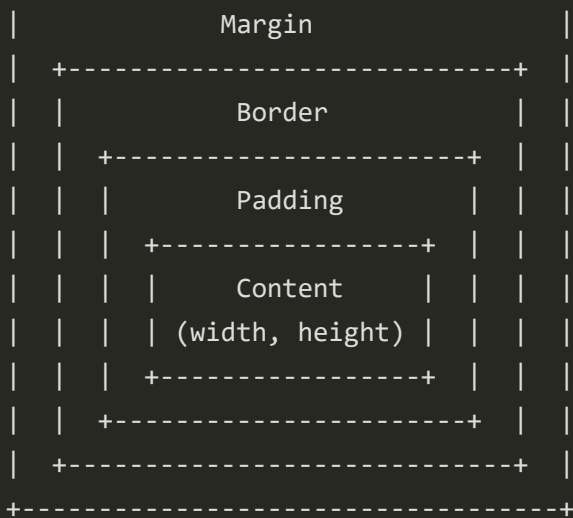
Setiap elemen HTML dianggap sebagai kotak. Model kotak CSS adalah konsep fundamental yang menjelaskan bagaimana elemen-elemen ini dirender di browser. Ini terdiri dari:

- **width** dan **height** : Mengatur lebar dan tinggi konten elemen.
- **padding** : Ruang di antara konten elemen dan batasnya (border). Padding bersifat transparan.

- **border** : Garis di sekitar padding dan konten. Anda dapat mengatur ketebalan, gaya, dan warna border.
- **margin** : Ruang di luar border elemen. Margin menciptakan ruang di antara elemen-elemen lain.

```
width: 200px;
height: 100px;
padding: 20px; /* Padding 20px di semua sisi */
border: 2px solid black; /* Border 2px, solid, warna hitam */
margin: 10px; /* Margin 10px di semua sisi */
background-color: lightgreen;
}
```

Penjelasan Box Model:



- **Content Box**: Area di mana konten sebenarnya (teks, gambar, dll.) berada. Ukurannya ditentukan oleh **width** dan **height**.
- **Padding Box**: Area di sekitar content box. Padding membersihkan area di sekitar konten. Anda dapat mengatur padding untuk setiap sisi secara terpisah (**padding-top** , **padding-right** , **padding-bottom** , **padding-left**) atau menggunakan shorthand (**padding: 10px 20px 30px 40px;** untuk top, right, bottom, left).
- **Border Box**: Area di sekitar padding box. Border adalah garis yang mengelilingi padding dan konten. Anda dapat mengatur gaya (**border-style**), lebar (**border-width**), dan warna (**border-color**) border.

- **Margin Box:** Area di luar border box. Margin membersihkan area di luar border. Margin juga dapat diatur untuk setiap sisi secara terpisah (`margin-top` , `margin-right` , `margin-bottom` , `margin-left`) atau menggunakan shorthand (`margin: 10px 20px;` untuk top/bottom dan left/right).

Properti `box-sizing` : Properti ini memungkinkan Anda untuk menyertakan padding dan border dalam total lebar dan tinggi elemen. Nilai default adalah `content-box` , yang berarti `width` dan `height` hanya berlaku untuk konten. Jika Anda mengatur `box-sizing: border-box;` , maka `width` dan `height` akan mencakup padding dan border.

```
/* Contoh border box */
width: 200px;
height: 100px;
padding: 20px;
border: 2px solid blue;
margin: 10px;
background-color: lightcoral;
box-sizing: border-box; /* Padding dan border termasuk dalam width/height */
}
```

4. Tampilan (Display)

Properti `display` adalah salah satu properti CSS yang paling penting untuk mengontrol tata letak. Ini menentukan bagaimana elemen ditampilkan.

- `block` : Elemen mengambil lebar penuh yang tersedia dan selalu dimulai pada baris baru (misalnya, `<div>` , `<p>` , `<h1>`).
- `inline` : Elemen hanya mengambil lebar yang dibutuhkan oleh kontennya dan tidak memulai baris baru (misalnya, `` , `<a>` , ``).
- `inline-block` : Mirip dengan `inline` , tetapi Anda dapat mengatur lebar dan tinggi (misalnya, tombol navigasi).
- `none` : Elemen tidak ditampilkan sama sekali.

```
/* Contoh */
display: block; /* Mengubah span menjadi elemen block */
}

a {
display: inline-block;
width: 100px;
}
```



```
    height: 30px;
    background-color: yellow;
}

.sembunyi {
    display: none;
}
```

5. Posisi (Position)

Properti `position` digunakan untuk mengontrol penempatan elemen di halaman web.

- `static` : (Default) Elemen diposisikan sesuai aliran normal dokumen.
- `relative` : Elemen diposisikan relatif terhadap posisi normalnya. Properti `top`, `right`, `bottom`, `left` dapat digunakan.
- `absolute` : Elemen diposisikan relatif terhadap elemen induk terdekat yang diposisikan (bukan `static`). Jika tidak ada induk yang diposisikan, itu relatif terhadap `<body>`.
- `fixed` : Elemen diposisikan relatif terhadap viewport dan tetap di tempatnya bahkan saat halaman digulir.
- `sticky` : Mirip dengan `relative` sampai posisi scroll mencapai ambang batas tertentu, lalu menjadi `fixed`.

```
.kotak-relative {
    position: relative;
    top: 20px;
    left: 30px;
    border: 1px solid red;
}

.kotak-absolute {
    position: absolute;
    top: 50px;
    right: 10px;
    background-color: orange;
}

.header-fixed {
    position: fixed;
    top: 0;
    width: 100%;
    background-color: #eee;
}
```

```
padding: 10px;
}
```

6. Flexbox (Fleksibel Box Layout)

Flexbox adalah modul tata letak satu dimensi yang dirancang untuk mendistribusikan ruang di antara item dalam antarmuka dan kemampuan perataan yang kuat. Ini sangat berguna untuk membuat tata letak yang responsif.

```
<div class="container flex">
  <div class="item">Item 1</div>
  <div class="item">Item 2</div>
  <div class="item">Item 3</div>
</div>
```

```
.container.flex {
  display: flex; /* Mengaktifkan flexbox */
  justify-content: space-around; /* Mendistribusikan item secara merata dengan
ruang di sekitarnya */
  align-items: center; /* Menyelaraskan item secara vertikal di tengah */
  border: 2px solid blue;
  height: 150px;
}

.item {
  background-color: lightgray;
  padding: 15px;
  margin: 5px;
  border: 1px solid #333;
}
```

7. Grid (Grid Layout)

CSS Grid Layout adalah sistem tata letak dua dimensi yang memungkinkan Anda untuk mendesain antarmuka pengguna yang kompleks dengan baris dan kolom. Ini sangat kuat untuk tata letak halaman secara keseluruhan.

```
<div class="header">Header</div>
<div class="sidebar">Sidebar</div>
<div class="content">Content</div>
<div class="footer">Footer</div>
</div>
```

```
display: grid; /* Mengaktifkan grid */
grid-template-columns: 1fr 3fr; /* Dua kolom: satu bagian, tiga bagian */
grid-template-rows: auto 1fr auto; /* Baris otomatis, satu bagian, otomatis */
gap: 10px; /* Jarak antar grid item */
height: 300px;
border: 2px solid green;
}

.header {
  grid-column: 1 / -1; /* Membentang dari kolom pertama hingga terakhir */
  background-color: #f0f0f0;
  padding: 10px;
}

.sidebar {
  background-color: #e0e0e0;
  padding: 10px;
}

.content {
  background-color: #d0d0d0;
  padding: 10px;
}

.footer {
  grid-column: 1 / -1;
  background-color: #f0f0f0;
  padding: 10px;
}
```

3. Langkah-langkah Praktikum

3.1. Praktikum HTML: Membuat Halaman Profil Sederhana

Tujuan: Membuat halaman web sederhana yang menampilkan informasi profil pribadi menggunakan elemen-elemen HTML dasar.

Alat yang Dibutuhkan:

- Editor teks (misalnya, VS Code, Sublime Text, Notepad++)
- Browser web (misalnya, Google Chrome, Mozilla Firefox)

Langkah-langkah:

1. **Buat Folder Proyek:**

Buat folder baru di komputer Anda dengan nama `modul-html-css`. Di dalam folder ini, buat sub-folder bernama `praktikum-html`.

2. **Buat File HTML:**

Di dalam folder `praktikum-html`, buat file baru dan beri nama `profil.html`. Buka file ini dengan editor teks Anda.

3. **Tulis Struktur Dasar HTML:**

Salin dan tempel kode struktur dasar HTML berikut ke dalam file `profil.html`:

```
<!-- Struktur Dasar HTML -->
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Profil Saya</title>
</head>
<body>

</body>
</html>
```

4. **Tambahkan Judul Utama:**

Di dalam tag `<body>`, tambahkan judul utama untuk halaman profil Anda:

```
<h1>Profil Pribadi</h1>
</body>
```

5. Tambahkan Informasi Dasar:

Di bawah judul, tambahkan paragraf untuk nama, pekerjaan, dan deskripsi singkat tentang diri Anda. Gunakan tag `<p>`.

```
...  
<h1>Profil Pribadi</h1>  
<p>Nama: [Nama Lengkap Anda]</p>  
<p>Pekerjaan: [Pekerjaan Anda]</p>  
<p>Halo! Saya adalah seorang [Pekerjaan Anda] yang tertarik pada  
pengembangan web. Saya suka belajar hal-hal baru dan menciptakan sesuatu  
yang bermanfaat.</p>  
</body>
```

6. Tambahkan Daftar Hobi:

Buat daftar hobi Anda menggunakan daftar tak berurut (`` dan ``).

```
...  
<h1>Profil Pribadi</h1>  
<p>Nama: [Nama Lengkap Anda]</p>  
<p>Pekerjaan: [Pekerjaan Anda]</p>  
<p>Halo! Saya adalah seorang [Pekerjaan Anda] yang tertarik pada  
pengembangan web. Saya suka belajar hal-hal baru dan menciptakan sesuatu  
yang bermanfaat.</p>  
  
<h2>Hobi</h2>  
<ul>  
  <li>Membaca Buku</li>  
  <li>Bermain Game</li>  
  <li>Memasak</li>  
  <li>Belajar Coding</li>  
</ul>  
</body>
```

7. Tambahkan Tautan ke Media Sosial (Opsional):

Jika Anda ingin, tambahkan tautan ke profil media sosial Anda menggunakan tag `<a>`.

```
...  
<!-- ... kode sebelumnya ... -->
```

```
<h2>Hubungi Saya</h2>
<p>Anda bisa menemukan saya di:
  <a href="https://linkedin.com/in/nama-anda"
target="_blank">LinkedIn</a> |
  <a href="https://github.com/nama-anda" target="_blank">GitHub</a>
</p>
</body>
```

8. Simpan dan Buka di Browser:

Simpan file `profil.html`. Kemudian, buka file tersebut di browser web Anda (Anda bisa menyeret file ke jendela browser atau klik kanan -> Open with -> pilih browser).

Anda akan melihat halaman profil sederhana Anda. Selamat, Anda telah berhasil membuat halaman web pertama Anda!

3.2. Praktikum CSS: Menata Gaya Halaman Profil

Tujuan: Menerapkan gaya CSS pada halaman profil HTML yang telah dibuat sebelumnya untuk mempercantik tampilannya.

Alat yang Dibutuhkan:

- Editor teks (misalnya, VS Code, Sublime Text, Notepad++)
- Browser web (misalnya, Google Chrome, Mozilla Firefox)
- File `profil.html` dari praktikum sebelumnya.

Langkah-langkah:

1. Buat File CSS:

Di dalam folder `modul-html-css` (folder utama proyek Anda), buat file baru dan beri nama `style.css`. Pastikan file ini berada di folder yang sama dengan `profil.html` atau sesuaikan path-nya.

2. Hubungkan CSS ke HTML:

Buka file `profil.html` Anda. Di dalam tag `<head>`, tambahkan baris berikut untuk menghubungkan file CSS eksternal:

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Profil Saya</title>
<link rel="stylesheet" href="style.css"> <!-- Tambahkan baris ini -->
</head>
```

3. Atur Gaya Dasar Body:

Buka file `style.css` dan tambahkan kode berikut untuk mengatur gaya dasar halaman:

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 20px;  
    background-color: #f4f4f4;  
    color: #333;  
}
```

4. Gaya Judul Utama (`<h1>`):

Tambahkan gaya untuk judul utama agar lebih menonjol:

```
h1 {  
    color: #0056b3;  
    text-align: center;  
    margin-bottom: 30px;  
    border-bottom: 2px solid #0056b3;  
    padding-bottom: 10px;  
}
```

5. Gaya Sub-Judul (`<h2>`):

Atur gaya untuk sub-judul:

```
h2 {  
    color: #007bff;  
    margin-top: 40px;  
    margin-bottom: 15px;  
}
```

6. Gaya Paragraf (`<p>`):

Perbaiki tampilan paragraf:

```
line-height: 1.6;
margin-bottom: 10px;
}
```

7. Gaya Daftar (`` dan ``):

Berikan gaya pada daftar hobi:

```
list-style-type: square;
margin-left: 20px;
margin-bottom: 20px;
}

li {
margin-bottom: 5px;
}
```

8. Gaya Tautan (`<a>`):

Perindah tampilan tautan:

```
color: #007bff;
text-decoration: none;
}

a:hover {
text-decoration: underline;
}
```

9. Tambahkan Kontainer untuk Konten (Opsional, tapi disarankan):

Untuk tata letak yang lebih baik, bungkus semua konten `<body>` Anda dengan `<div>` dengan kelas `container` di `profil.html`:

```
<div class="container">
```



```
<h1>Profil Pribadi</h1>
<p>Nama: [Nama Lengkap Anda]</p>
<!-- ... semua konten lainnya ... -->
</div>
</body>
```

Kemudian, tambahkan gaya untuk `.container` di `style.css` :

```
.container {
  max-width: 800px;
  margin: 30px auto;
  background-color: #fff;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

10. Simpan dan Lihat Hasil:

Simpan kedua file (`profil.html` dan `style.css`). Buka kembali `profil.html` di browser Anda. Anda akan melihat halaman profil Anda kini memiliki tampilan yang jauh lebih menarik dengan gaya CSS yang diterapkan.

4. Latihan

4.1. Latihan HTML

Buatlah sebuah halaman web sederhana dengan ketentuan sebagai berikut:

1. **Judul Halaman:** "Biodata Diri"
2. **Judul Utama** (`<h1>`): "Biodata [Nama Lengkap Anda]"
3. **Informasi Pribadi:** Gunakan paragraf (`<p>`) untuk menampilkan informasi berikut:
 - Nama Lengkap
 - Tempat, Tanggal Lahir
 - Alamat
 - Email
 - Nomor Telepon
4. **Pendidikan:** Buat daftar berurut (``) untuk riwayat pendidikan Anda (SD, SMP, SMA, Perguruan Tinggi).
5. **Pengalaman Kerja (Opsional):** Buat daftar tak berurut (``) untuk pengalaman kerja Anda (jika ada).

6. **Gambar Profil:** Sisipkan gambar profil Anda menggunakan tag ``. Pastikan Anda menyertakan atribut `alt` dan mengatur `width` serta `height` yang sesuai.
7. **Tautan:** Tambahkan tautan ke situs web favorit Anda atau profil media sosial (misalnya, Instagram, Twitter) menggunakan tag `<a>`. Pastikan tautan terbuka di tab baru.
8. Gunakan tag `<hr>` untuk memisahkan bagian-bagian konten.

4.2. Latihan CSS

Berikan gaya pada halaman "Biodata Diri" yang telah Anda buat di latihan HTML dengan ketentuan sebagai berikut:

1. **Gunakan CSS Eksternal:** Buat file `style-biodata.css` dan hubungkan ke halaman HTML Anda.
2. **Body:**
 - `font-family`: Pilih font yang mudah dibaca (misalnya, Arial, sans-serif).
 - `background-color`: Berikan warna latar belakang yang lembut (misalnya, `lightgray` atau `#f0f0f0`).
 - `margin` dan `padding`: Atur sesuai kebutuhan agar konten tidak terlalu menempel pada tepi browser.
3. **Container:** Bungkus semua konten di dalam `<body>` dengan `<div>` ber-kelas `container`.
 - `max-width`: Atur lebar maksimum (misalnya, `800px`).
 - `margin`: `auto` untuk menengahkan container.
 - `background-color`: `white` atau warna terang lainnya.
 - `padding`: Berikan padding yang cukup (misalnya, `20px` atau `30px`).
 - `box-shadow`: Tambahkan sedikit bayangan untuk efek kedalaman.
 - `border-radius`: Berikan sudut membulat pada container.
4. **Judul Utama (`<h1>`):**
 - `color`: Warna yang menarik.
 - `text-align`: `center`.
 - `text-transform`: `uppercase` atau `capitalize`.
5. **Paragraf (`<p>`):**
 - `line-height`: Atur tinggi baris agar teks lebih mudah dibaca (misalnya, `1.6`).
 - `margin-bottom`: Berikan jarak antar paragraf.
6. **Daftar (`` dan ``):**
 - `list-style-type`: Ubah gaya bullet/nomor sesuai keinginan (misalnya, `circle` untuk `ul`, `upper-roman` untuk `ol`).
 - `margin-left`: Atur indentasi daftar.
7. **Gambar Profil (``):**
 - `display`: `block`.
 - `margin`: `auto` untuk menengahkan gambar.

- `border-radius` : Buat gambar menjadi lingkaran atau sudut membulat (misalnya, `50%` untuk lingkaran).
- `border` : Tambahkan border pada gambar.

8. Tautan (`<a>`):

- `color` : Ubah warna tautan.
- `text-decoration` : `none` (hilangkan garis bawah).
- `a:hover` : Tambahkan efek `text-decoration: underline;` saat kursor diarahkan ke tautan.

Selamat mengerjakan! Jika Anda mengalami kesulitan, jangan ragu untuk merujuk kembali ke materi yang telah dijelaskan.