

# MATERI BASIC: CASCADING STYLE SHEETS (CSS)



**Untuk SMKS Al Badar Dangdeur**

Disusun oleh:

Ahadi, S.Kom

## Daftar Isi – Materi CSS Basic

Latar Belakang.....	i
Pengantar CSS.....	1
a. Apa itu CSS.....	1
b. Mengapa Disebut “Cascading”.....	1
Cara Menulis CSS.....	1
a. Inline CSS.....	1
b. Internal CSS.....	2
c. External CSS.....	2
Struktur Penulisan CSS.....	2
Jenis-Jenis Selektor CSS.....	3
a. Selektor Tag (Elemen).....	3
b. Selektor Class.....	3
c. Selektor ID.....	3
d. Selektor Universal.....	4
e. Selektor Turunan (Descendant Selector).....	4
f. Selektor Gabungan.....	4
Warna dan Latar Belakang (Background).....	4
Mengatur Teks dan Font.....	5
Mengatur Tata Letak (Layout).....	6
a. Box Model.....	6
b. Display.....	6
c. Float.....	6
Flexbox – Sistem Layout Modern.....	7
Desain Responsif dengan Media Query.....	7
Transisi dan Efek Hover.....	8
Animasi Dasar CSS.....	8
Proyek Praktik: Membuat Kartu Profil.....	9
Kesalahan Umum dalam CSS.....	11
Latihan Mandiri.....	11
Refleksi dan Diskusi.....	12
Rangkuman.....	12

## Latar Belakang

Dalam dunia digital modern, hampir setiap informasi yang kita akses disajikan melalui halaman web. Mulai dari berita, belanja online, hingga sistem administrasi sekolah — semuanya menggunakan teknologi berbasis web. Oleh karena itu, kemampuan untuk memahami dan membuat tampilan web menjadi keterampilan penting bagi pelajar SMK, khususnya dalam bidang coding dan teknologi informasi.

Sebelum mempelajari CSS, siswa telah mengenal HTML (HyperText Markup Language) sebagai bahasa untuk membangun struktur dasar sebuah halaman web. Namun, HTML hanya mampu menampilkan isi dalam bentuk polos — teks, gambar, dan tautan tanpa keindahan visual. Agar tampilan web menjadi menarik, mudah dibaca, dan memiliki gaya yang konsisten, dibutuhkan CSS (Cascading Style Sheets).

CSS berfungsi untuk memisahkan antara struktur dan tampilan. Artinya, HTML bertugas menulis “apa” yang muncul di halaman, sementara CSS mengatur “bagaimana” tampilannya. Dengan CSS, kita bisa mengatur warna, ukuran teks, jarak antar elemen, tata letak, hingga membuat animasi dan efek interaktif sederhana. Pemisahan ini juga membuat kode lebih rapi, efisien, dan mudah dikembangkan — prinsip yang sangat penting dalam dunia profesional.

Melalui pembelajaran CSS, siswa diharapkan mampu memahami logika desain antarmuka (UI) dan menerapkan prinsip estetika dalam konteks teknologi. Tidak hanya belajar menulis kode, tetapi juga belajar menciptakan pengalaman visual yang komunikatif dan fungsional bagi pengguna.

Selain itu, materi CSS juga menjadi dasar untuk mempelajari konsep lanjutan seperti framework front-end (misalnya Bootstrap atau Tailwind), serta menjadi fondasi sebelum siswa melangkah ke JavaScript interaktif dan pengembangan web dinamis.

Dengan memahami CSS, siswa tidak hanya mampu membuat halaman web yang “berfungsi”, tetapi juga mampu membuatnya “menarik dan profesional”. Itulah alasan mengapa CSS menjadi bagian penting dalam kurikulum Coding SMK dan menjadi jembatan antara logika pemrograman dan dunia desain web modern.

# 1. Pengantar CSS

## a. Apa itu CSS?

CSS (Cascading Style Sheets) adalah bahasa yang digunakan untuk mengatur **tampilan dan gaya visual halaman web**.

Jika HTML adalah kerangka atau struktur tubuh dari sebuah halaman, maka CSS adalah **pakaian dan warna** yang membuatnya indah dan enak dipandang.

Dengan CSS, kita bisa:

1. Mengubah warna, ukuran, dan jenis huruf.
2. Mengatur posisi elemen di halaman.
3. Menentukan jarak antar bagian.
4. Membuat efek transisi dan animasi.

Tanpa CSS, semua halaman HTML akan tampak polos seperti teks dokumen biasa.

## a. Kenapa Disebut “Cascading”?

Kata *Cascading* berarti “menurun seperti air terjun”.

Artinya, **aturan CSS yang ditulis terakhir akan menggantikan aturan sebelumnya** jika keduanya mengatur hal yang sama.

Ini penting untuk memahami bagaimana browser menentukan tampilan akhir elemen.

Contoh:

```
p { color: red; }  
p { color: blue; }
```

# 2. Cara Menulis CSS

CSS bisa digunakan dalam tiga cara utama:

## a. Inline CSS

CSS ditulis langsung di dalam elemen HTML menggunakan atribut style.

```
<p style="color: red; font-size: 20px;">Teks ini merah dan besar.</p>
```

Cocok untuk percobaan cepat, **tidak direkomendasikan** untuk proyek besar.

#### b. Internal CSS

CSS ditulis di bagian `<head>` dalam file HTML yang sama.

```
<head>
  <style>
    h1 {
      color: navy;
      text-align: center;
    }
  </style>
</head>
```

Cocok untuk halaman kecil atau latihan individu.

#### c. External CSS

CSS disimpan di file terpisah, lalu dihubungkan ke HTML dengan `<link>`.

```
<link rel="stylesheet" href="style.css">
```

Isi style.css

```
body {
  background-color: #f4f4f4;
  font-family: Arial, sans-serif;
}
```

Ini cara **paling profesional dan efisien**, karena satu file CSS bisa digunakan oleh banyak halaman web.

## 3. Struktur Penulisan CSS

Struktur umum CSS seperti ini:

```
selektor {  
  properti: nilai;  
}
```

Contoh:

```
p {  
  color: green;  
  font-size: 18px;  
}
```

Selektor = elemen HTML yang akan diubah.

Properti = aspek yang ingin diatur (warna, ukuran, dsb).

Nilai = hasil pengaturan yang diinginkan.

## 4. Jenis-Jenis Selektor CSS

### a. Selektor Tag (Elemen)

```
h1 { color: blue; }  
p { text-align: justify; }
```

### b. Selektor Class

Digunakan untuk banyak elemen sekaligus.

```
<p class="judul">Halo Dunia!</p>
```

```
.judul { font-weight: bold; }
```

### c. Selektor ID

Digunakan untuk satu elemen unik saja.

```
<div id="utama">Selamat Datang</div>
```

```
#utama { background-color: yellow; }
```

d. Selektor Universal

Mengatur semua elemen.

```
* { margin: 0; padding: 0; }
```

e. Selektor Turunan (Descendant Selector)

Mengatur elemen di dalam elemen lain.

```
div p { color: gray; }
```

f. Selektor Gabungan

Memilih beberapa elemen sekaligus.

```
h1, h2, h3 { color: navy; }
```

## 5. Warna dan Latar Belakang

CSS mendukung beberapa format warna:

- Nama warna: red, blue, black
- Kode heksadesimal: #ff0000
- RGB: rgb(255, 0, 0)
- RGBA: rgba(255, 0, 0, 0.5) (ada transparansi)

```
body {
  background-color: #f0f8ff;
  color: #333333;
}
```

Menambahkan gambar latar:

```
body {
  background-image: url("background.jpg");
  background-repeat: no-repeat;
  background-size: cover;
}
```

## 6. Mengatur Teks dan Font

```
p {
  font-family: 'Poppins', sans-serif;
  font-size: 16px;
  color: #333;
  text-align: justify;
  line-height: 1.5;
}
```

Properti penting:

- font-family → jenis huruf
- font-size → ukuran teks
- color → warna teks
- text-align → perataan teks
- line-height → jarak antarbaris

Contoh kombinasi:



```
h1 {  
  text-transform: uppercase;  
  letter-spacing: 2px;  
  text-shadow: 2px 2px 4px gray;  
}
```

## 7. Mengatur Tata Letak (Layout)

Tata letak menentukan posisi elemen di halaman web.

### a. Box Model

Setiap elemen di halaman dianggap sebagai kotak yang terdiri dari:

- content (isi)
- padding (jarak isi ke tepi)
- border (garis tepi)
- margin (jarak antar elemen)

```
div {  
  width: 200px;  
  padding: 10px;  
  border: 2px solid black;  
  margin: 20px;  
}
```

### b. Display

Mengontrol cara elemen ditampilkan.

```
p { display: block; }  
span { display: inline; }  
nav { display: flex; }
```

### c. Float

Memposisikan elemen di kanan atau kiri.

```
img {  
  float: right;  
  margin: 10px;  
}
```

## 8. Flexbox (Sistem Layout Modern)

Flexbox adalah sistem layout modern untuk mengatur posisi elemen secara fleksibel.

```
.container {  
  display: flex;  
  justify-content: center; /* posisi horizontal */  
  align-items: center;     /* posisi vertikal */  
  height: 300px;  
}
```

Kelebihan Flexbox:

- Mudah mengatur posisi tengah.
- Responsif untuk berbagai ukuran layar.
- Tidak butuh banyak float atau position.

## 9. Desain Responsif dengan Media Query

Agar tampilan web bagus di semua perangkat, gunakan media query.

```
@media (max-width: 768px) {
  body {
    background-color: lightgray;
  }

  h1 {
    font-size: 20px;
  }
}
```

Artinya: jika lebar layar  $\leq 768\text{px}$ , maka warna latar belakang menjadi abu-abu dan ukuran judul mengecil.

## 10. Transisi dan Efek Hover

```
button {
  background-color: #0099ff;
  color: white;
  padding: 10px 20px;
  transition: 0.3s;
}

button:hover {
  background-color: #0066cc;
}
```

Efek ini membuat tombol berubah warna secara halus saat kursor diarahkan.

## 11. Animasi Dasar

CSS juga bisa membuat animasi sederhana tanpa JavaScript.

```

@keyframes gerak {
  from { transform: translateX(0); }
  to { transform: translateX(100px); }
}

div {
  animation: gerak 2s infinite alternate;
}

```

Elemen akan bergerak bolak-balik setiap 2 detik.

## 12. Proyek Praktik: Membuat Kartu Profil

HTML

```

<div class="card">
  
  <h2>Rani Putri</h2>
  <p>Pelajar SMK yang gemar belajar pemrograman web.</p>
</div>

```

CSS

```
body {  
  background-color: #f7f7f7;  
  font-family: Arial, sans-serif;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
}  
  
.card {  
  background-color: white;  
  padding: 20px;  
  width: 250px;  
  border-radius: 10px;  
  box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
  text-align: center;  
  transition: 0.3s;  
}  
  
.card:hover {  
  transform: scale(1.05);  
}  
  
.card img {  
  width: 100px;  
  border-radius: 50%;  
}
```

Output



## Rani Putri

Pelajar SMK yang gemar belajar pemrograman web.

### 13. Kesalahan Umum dalam CSS

- Lupa menutup kurung { } atau titik koma ;.
- Menulis nama file CSS salah di <link>.
- Menggunakan inline CSS secara berlebihan.
- Menyalin kode tanpa memahami fungsinya.

### 14. Latihan Mandiri

- Buat halaman berjudul “Portfolio Sederhana”.
- Tambahkan tiga bagian: Tentang Saya, Hobi, dan Kontak.

- Gunakan warna dan font berbeda di setiap bagian.
- Gunakan efek hover untuk tautan atau tombol.
- Buat tampilannya tetap rapi di layar HP dan laptop.

## **15. Refleksi dan Diskusi**

- Bagian mana dari CSS yang paling menarik menurutmu?
- Mengapa CSS dan HTML harus dipisahkan dalam proyek web profesional?
- Bagaimana CSS memengaruhi pengalaman pengguna (User Experience)?

## **Rangkuman**

- CSS digunakan untuk mengatur tampilan halaman HTML.
- Tiga cara penulisan CSS: inline, internal, external.
- Konsep box model dan flexbox penting dalam tata letak.
- CSS modern mendukung efek interaktif, animasi, dan desain responsif.
- CSS yang rapi, terstruktur, dan efisien adalah dasar desain web profesional.