

CSS

Cascading Style Sheets

Introduction | Selectors | Box Model

Introduction to CSS

What is CSS, how it works, and how to use it

Apa itu CSS?

CSS = Cascading Style Sheets

CSS adalah bahasa yang menjelaskan BAGAIMANA elemen HTML ditampilkan di layar

- **Cascading** = Gaya bisa saling menimpa berdasarkan aturan prioritas
- **Style** = Tampilan visual: warna, font, jarak, tata letak
- **Sheets** = File (.css) yang berisi aturan gaya

Tanpa CSS, website hanya halaman putih polos dengan gaya default browser

Dengan CSS, control seluruh tampilan visual halaman web!

Cara Kerja CSS: HTML + CSS

HTML provides structure , CSS provides style (HOW IT LOOKS)

HTML (structure):

```
<!-- index.html -->
<h1>Welcome to My Site</h1>
<p>This is a paragraph.</p>
<button>Click Me</button>

<!-- Without CSS: plain black text
      on white background,
      default browser font,
      no spacing control -->
```

HTML + CSS (styled):

```
<!-- Same HTML, but now with CSS -->
<link rel="stylesheet" href="style.css">

/* style.css */
h1 { color: navy; font-size: 36px; }
p { color: gray; line-height: 1.6; }
button {
    background: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
}
```

Tiga Cara Menambahkan CSS

1. Inline CSS (Avoid!):

```
<!-- Style directly on element -->
<p style="color: red;
        font-size: 20px;">
  This text is red.
</p>

<h1 style="color: blue;">
  Blue heading
</h1>
```

Masalah:

- Sulit dipelihara
- Tidak bisa dipakai ulang
- Campuran konten + gaya
- Hanya untuk tes cepat

2. Internal CSS (OK):

```
<!-- Style in <head> section -->
<!DOCTYPE html>
<html>
<head>
  <style>
    h1 { color: blue; }
    p {
      color: gray;
      font-size: 18px;
    }
  </style>
</head>
<body>
  <h1>Hello</h1>
  <p>Styled paragraph</p>
</body>
</html>
```

3. External CSS (Best!):

```
<!-- index.html -->
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet"
        href="styles.css">
</head>
<body>
  <h1>Hello</h1>
  <p>Styled paragraph</p>
</body>
</html>

/* styles.css (separate file) */
h1 { color: blue; }
p { color: gray; font-size: 18px; }
```

CSS Syntax

```
/* CSS Syntax Structure */
```

```
selector {  
    property: value;  
    property: value;  
}
```

```
/* Real Example: */
```

```
h1 {  
    color: blue;           /* text color */  
    font-size: 32px;       /* text size */  
    font-weight: bold;     /* text weight */  
    text-align: center;    /* alignment */  
    margin-bottom: 20px;   /* space below */  
}
```

```
/* selector = WHAT to style      (h1)  
   property = WHAT to change     (color)  
   value    = WHAT to set        (blue)  
   { }      = declaration block  
   ;        = end of each rule   */
```

Multiple Selectors & Comments:

```
/* Apply same style to multiple elements */  
h1, h2, h3 {  
    color: navy;  
    font-family: Arial, sans-serif;  
}
```

```
/* This is a single-line comment */
```

```
/*  
    This is a  
    multi-line comment.  
    Use comments to explain  
    complex CSS rules.  
*/
```

```
/* You can comment out code: */
```

```
p {  
    color: red;  
    /* font-size: 20px; -- disabled */  
}
```

HTML + CSS Integration

index.html:

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Halaman Saya</title>
  <link rel="stylesheet"
        href="styles.css">
</head>
<body>
  <h1>Selamat Datang di Website Saya</h1>
  <p>Ini adalah paragraf teks.</p>
  <p>Ini paragraf lainnya.</p>
  <button>Klik Saya</button>
</body>
</html>
```

styles.css:

```
body {
  background-color: #f0f0f0;
  font-family: Arial, sans-serif;
  padding: 20px;
}

h1 {
  color: #333;
  font-size: 36px;
  border-bottom: 3px solid #4CAF50;
  padding-bottom: 10px;
}

p {
  color: #666;
  font-size: 16px;
  line-height: 1.6;
  margin-bottom: 15px;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 12px 24px;
  border: none;
  border-radius: 5px;
  font-size: 16px;
  cursor: pointer;
}
```

CSS Units & Color Values

CSS Units - Cara mengukur ukuran:

Unit	Type	Description	Example
px	Absolute	Fixed pixels	font-size: 16px;
%	Relative	% of parent element	width: 50%;
rem	Relative	Based on root font (16px)	font-size: 1.5rem; = 24px
em	Relative	Based on parent font size	padding: 2em;
vw/vh	Relative	% of browser window	height: 100vh; = full screen

Color Formats:

```
/* Named Colors */
color: red; color: blue; color: green;

/* Hexadecimal (#RRGGBB) */
color: #FF0000; /* Red (FF=255,00=0,00=0) */
color: #00FF00; /* Green */
color: #333; /* Short for #333333 (dark gray) */

/* RGB (Red 0-255, Green 0-255, Blue 0-255) */
color: rgb(255, 0, 0); /* Red */
color: rgb(100, 100, 100); /* Gray */

/* RGBA (with transparency 0.0 - 1.0) */
color: rgba(255, 0, 0, 0.5); /* 50% red */
background: rgba(0, 0, 0, 0.3); /* overlay */

/* HSL (Hue 0-360, Saturation %, Lightness %) */
color: hsl(0, 100%, 50%); /* Red */
color: hsl(120, 100%, 50%); /* Green */
```

Summary

Concept	What It Does	Example
CSS	Styles HTML appearance	color, size, spacing
Selector	Targets elements	<code>h1 {} .class {}</code>
Property	What to change	color, font-size, margin
Value	What to set it to	red, 16px, 20px
External CSS	Best practice	<code><link href="style.css"></code>
Internal CSS	OK for small pages	<code><style></code> in <code><head></code>
Inline CSS	Avoid!	<code>style="color:red"</code>

Key Point: Always use External CSS for real projects – keeps HTML clean and CSS reusable!

LESSON 2

CSS Selectors

Cara menargetkan elemen HTML tertentu untuk di-styling

What are Selectors?

Selektor adalah pola yang memberi tahu CSS ELEMEN MANA yang harus di-styling.

Seperti: "Hei CSS, cari elemen INI dan buat tampilannya seperti ITU"

```
/* Selektor adalah bagian SEBELUM kurung kurawal { } */
```

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

```
-----
```

```
SIAPA yang di-styling   APA yang dilakukan
```

```
/* Examples: */
```

```
p          { color: blue; }          /* Pilih SEMUA elemen <p> */
```

```
.highlight { background: yellow; }    /* Pilih elemen dengan class="highlight" */
```

```
#header    { font-size: 24px; }       /* Pilih elemen dengan id="header" */
```

Selektor Dasar + Contoh HTML

1. Element Selector:

```
/* CSS: targets ALL <p> elements */
p {
  color: blue;
  font-size: 16px;
}

<!-- HTML -->
<p>This is blue</p>
<p>This is also blue</p>
<p>ALL paragraphs are blue!</p>

<!-- Every <p> on the page gets the same style -->
```

2. Class Selector (.):

```
/* CSS: dot = class */
.highlight {
  background-color: yellow;
  padding: 5px;
}
.error { color: red; }

<!-- HTML: class attribute -->
<p class="highlight">Yellow bg</p>
<p>No style (no class)</p>
<p class="error">Red text</p>
<p class="highlight error">Both!</p>
<!-- Classes are REUSABLE and
elements can have MULTIPLE -->
```

3. ID Selector (#):

```
/* CSS: hash = ID */
#header {
  background-color: navy;
  color: white;
  padding: 20px;
}
#footer { font-size: 12px; }

<!-- HTML: id attribute -->
<div id="header">Header</div>
<p>Content here</p>
<div id="footer">Footer</div>

<!-- IDs must be UNIQUE!
Only ONE element per ID.
Prefer classes for styling -->
```

Class vs ID: When to Use Which?

Class (.class) – Gunakan untuk STYLING:

```
/* Classes can be reused many times */
.btn {
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
.btn-green { background: green; color: white; }
.btn-red   { background: red; color: white; }

<!-- HTML: reuse classes freely -->
<button class="btn btn-green">Save</button>
<button class="btn btn-red">Delete</button>
<button class="btn btn-green">Submit</button>
<!-- Same .btn class = same base style -->
```

ID (#id) – Gunakan untuk elemen UNIK:

```
/* IDs are for one-of-a-kind elements */
#main-nav {
  background: #333;
  padding: 15px;
}
#hero-section {
  height: 100vh;
  background: linear-gradient(blue, purple);
}

<!-- HTML: each id used ONCE only -->
<nav id="main-nav">...</nav>
<section id="hero-section">...</section>

<!-- If you need the same style on
multiple elements, use a CLASS!
IDs are mainly for JavaScript -->
```

Class vs ID: Kapan Pakai yang Mana?

	Class (.)	ID (#)
Bisa dipakai ulang?	Ya - elemen tak terbatas	Tidak - satu per halaman
Sintaks	.namaclass { }	#namaid { }
HTML	class="nama"	id="nama"
Terbaik untuk	Styling (CSS)	Targeting JavaScript

Combinator Selectors: Menargetkan Elemen Bersarang

Descendant (space) vs Child (>):

```
<!DOCTYPE html>
<html>
<head>
  <title>Descendant Selector</title>
  <style>
    /* DESCENDANT (spasi): level MANAPUN di dalam */
    div p {
      color: blue;
    }
  </style>
</head>
<body>

  <div>
    <p>Biru! </p>
    <section>
      <p>Biru Juga! </p>
    </section>
  </div>

  <p>TIDAK biru (di luar div)</p>

</body>
</html>
```

Adjacent (+) vs General Sibling (-):

```
<!DOCTYPE html>
<html>
<head>
  <title>Adjacent Selector</title>
  <style>
    /* ADJACENT (+): tepat setelahnya */
    h2 + p {
      font-size: 18px;
      font-weight: bold;
      color: blue;
    }
  </style>
</head>
<body>

  <h2>Judul</h2>
  <p>Di-styling! (tepat setelah h2)</p>
  <p>TIDAK di-styling (bukan tepat setelah)</p>

</body>
</html>
```

Attribute Selectors

Pilih elemen berdasarkan atribut HTML-nya – sangat berguna untuk styling form

```
/* CSS: Attribute Selectors */

/* Has any type attribute */
input[type] { border: 1px solid gray; }

/* Exact match */
input[type="text"] { border: 2px solid blue; }
input[type="email"] { border: 2px solid green; }
input[type="password"] { border: 2px solid red; }

/* Contains value */
a[href*="google"] { color: orange; }

/* Starts with */
a[href^="https"] { color: green; } /* secure */
a[href^="http"] { color: red; } /* not secure */

/* Ends with */
a[href$=".pdf"] { color: red; } /* PDF links */
a[href$=".zip"] { color: purple; } /* ZIP links */
```

HTML yang cocok:

```
<!-- These inputs get different borders -->
<input type="text" placeholder="Name">
<input type="email" placeholder="Email">
<input type="password" placeholder="Pass">

<!-- These links get different colors -->
<a href="https://google.com">Google</a>
<a href="https://github.com">GitHub</a>
<a href="http://old-site.com">Old Site</a>

<!-- File links -->
<a href="report.pdf">Download PDF</a>
<a href="files.zip">Download ZIP</a>
```

Pseudo-Classes: Styling Berdasarkan Status Elemen

Pseudo-classes style elements in a specific STATE (hover, focus, position, etc.)

Interactive States:

```
/* Link States (order: L-V-H-A) */
a:link    { color: blue; }    /* unvisited */
a:visited { color: purple; }  /* visited */
a:hover   { color: red; }     /* mouse over */
a:active  { color: orange; }  /* being clicked */

/* Button hover effect */
.btn {
  background: #4CAF50;
  color: white;
  padding: 10px 20px;
  transition: 0.3s; /* smooth! */
}
.btn:hover {
  background: #388E3C; /* darker green */
  transform: scale(1.05); /* slightly bigger */
}

/* Form input states */
input:focus { border: 2px solid blue; outline: none; }
input:disabled { background: #eee; opacity: 0.6; }
input:required { border-left: 4px solid red; }
```

Position-Based:

```
/* First, Last, Nth child */
li:first-child { font-weight: bold; }
li:last-child  { border-bottom: none; }
li:nth-child(2) { color: red; } /* 2nd item */

/* Zebra-striped table rows */
tr:nth-child(odd) { background: #f5f5f5; }
tr:nth-child(even) { background: #ffffff; }

<!-- HTML -->
<ul>
  <li>First (bold)</li>
  <li>Second (red)</li>
  <li>Third</li>
  <li>Last (no border-bottom)</li>
</ul>

/* Exclusion */
p:not(.intro) { color: gray; }
/* All <p> EXCEPT those with class="intro" */

/* Empty elements */
div:empty { display: none; }
```

Pseudo-Classes: Styling Berdasarkan Status Elemen

Pseudo-classes style elements in a specific STATE (hover, focus, position, etc.)

:hover

```
<!DOCTYPE html>
<html>
<head>
  <title>Pseudo Class - Hover</title>
  <style>
    a:hover {
      color: red;
      text-decoration: underline;
    }
  </style>
</head>
<body>

  <a href="#">Arahkan mouse ke sini</a>

</body>
</html>
```

:active

```
<!DOCTYPE html>
<html>
<head>
  <style>
    button:active {
      background-color: orange;
    }
  </style>
</head>
<body>

  <button>Klik saya</button>

</body>
</html>
```

Pseudo-Classes: Styling Berdasarkan Status Elemen

Pseudo-classes style elements in a specific STATE (hover, focus, position, etc.)

:focus

```
<!DOCTYPE html>
<html>
<head>
  <style>
    input:focus {
      border: 2px solid blue;
      background-color: #f0f8ff;
    }
  </style>
</head>
<body>

  <input type="text" placeholder="Klik di sini">

</body>
</html>
```

:visited

```
<!DOCTYPE html>
<html>
<head>
  <style>
    a:visited {
      color: purple;
    }
  </style>
</head>
<body>

  <a href="https://google.com">Kunjungi Google</a>

</body>
</html>
```

Pseudo-Elements: Styling Bagian dari Elemen

Pseudo-element membuat/men-styling BAGIAN dari elemen (pakai :: titik dua ganda)

```
/* ::before and ::after - add content */
.quote::before { content: open-quote; font-size: 2em; }
.quote::after { content: close-quote; font-size: 2em; }

/* Required field indicator */
.required::after {
  content: " *";
  color: red;
  font-weight: bold;
}
<!-- HTML -->
<label class="required">Email</label>
<!-- Displays: Email * (red asterisk) -->

/* Drop cap - large first letter */
p.article::first-letter {
  font-size: 3em;
  float: left;
  margin-right: 8px;
  color: navy;
  font-weight: bold;
}
/* Style first line only */
p.article::first-line {
  font-weight: bold;
  color: #333;
}
```

```
/* Custom text selection color */
::selection {
  background-color: #FFEB3B;
  color: black;
}
/* When user highlights text, it
shows yellow bg instead of blue */

/* Placeholder text styling */
input::placeholder {
  color: #999;
  font-style: italic;
}

<!-- HTML -->
<input type="text"
  placeholder="Enter your name...">
<!-- "Enter your name..." appears
in italic gray -->

/* Pseudo-CLASS vs Pseudo-ELEMENT */
: hover      /* STATE (single :) */
::before    /* PART (double ::) */
:first-child /* STATE (single :) */
::first-line /* PART (double ::) */
```

Pseudo-Elements: Styling Bagian dari Elemen

::before dan ::after (Quote)

```
<!DOCTYPE html>
<html>
<head>
  <title>::before dan ::after</title>
  <style>
    .quote::before {
      content: open-quote;
      font-size: 2em;
      color: gray;
    }

    .quote::after {
      content: close-quote;
      font-size: 2em;
      color: gray;
    }
  </style>
</head>
<body>

  <h2>Contoh Quote</h2>
  <p class="quote">
    CSS bisa menambahkan konten tanpa mengubah HTML.
  </p>

</body>
</html>
```

Drop Cap (::first-letter)

```
<!DOCTYPE html>
<html>
<head>
  <title>Drop Cap</title>
  <style>
    p.article::first-letter {
      font-size: 3em;
      float: left;
      margin-right: 8px;
      color: navy;
      font-weight: bold;
    }
  </style>
</head>
<body>

  <p class="article">
    CSS adalah bahasa untuk mengatur tampilan halaman web
    agar lebih menarik dan profesional.
  </p>

</body>
</html>
```

Specificity: Spesifisitas: Design Mana yang Menang?

Saat beberapa aturan CSS menargetkan elemen yang sama, spesifisitas menentukan pemenangnya!

Priority	Selector Type	Specificity Score	Example
Lowest	Element (tag name)	0-0-1 = 1 point	p { color: blue; }
Low	Class / Pseudo-class	0-1-0 = 10 points	.intro { color: green; }
Medium	ID	1-0-0 = 100 points	#header { color: red; }
High	Inline style	1000 points	style="color: pink"
Highest	!important	Overrides all	color: black !important;

Spesifisitas: Design Mana yang Menang?

```
/* Which color wins? */
p      { color: blue; } /* 1 point */
.intro { color: green; } /* 10 points */
#special { color: red; } /* 100 points */

<p id="special" class="intro">
  This is RED! (ID wins with 100 pts)
</p>

<p class="intro">
  This is GREEN! (class wins with 10 pts)
</p>

<p>This is BLUE! (only element applies)</p>
```

```
/* Combining selectors adds scores */
p.intro      { } /* 0-1-1 = 11 points */
#nav .link   { } /* 1-1-0 = 110 points */
#nav ul li a { } /* 1-0-3 = 103 points */

/* EQUAL specificity? Last rule wins! */
p { color: blue; }
p { color: red; } /* RED wins (last) */

/* Avoid !important - it's hard to override */
p { color: red !important; } /* BAD practice */
```

Complete HTML + CSS Example

index.html:

```
<header id="main-header">
  <h1>My Website</h1>
  <nav>
    <a href="#" class="nav-link">Home</a>
    <a href="#" class="nav-link active">About</a>
    <a href="https://google.com"
      class="nav-link">Google</a>
  </nav>
</header>

<main class="container">
  <article class="post">
    <h2>First Post</h2>
    <p class="intro">Introduction text.</p>
    <p>Regular paragraph.</p>
  </article>
  <article class="post featured">
    <h2>Featured Post</h2>
    <p class="intro">Special post!</p>
  </article>
</main>

<footer id="main-footer">
  <p>Copyright 2024</p>
</footer>
```

styles.css:

```
/* Element selectors */
body { font-family: Arial; line-height: 1.6; }
h1, h2 { color: #333; }

/* ID selectors */
#main-header { background: #333; color: white;
                padding: 20px; }
#main-footer { background: #666; color: white;
                padding: 10px; text-align: center; }

/* Class selectors */
.container { max-width: 1200px; margin: 0 auto; }
.post { padding: 20px; margin: 15px 0;
        border: 1px solid #ddd; border-radius: 8px; }
.intro { font-size: 18px; color: #555; }
.featured { border-left: 5px solid #4CAF50;
            background: #f9fff9; }

/* Descendant selectors */
#main-header nav a { color: white;
                    text-decoration: none; margin: 0 15px; }

/* Pseudo-classes */
.nav-link:hover { text-decoration: underline; }
.nav-link.active { border-bottom: 2px solid white; }

/* Pseudo-elements */
.featured h2::before { content: "* "; color: #4CAF50; }

/* Attribute selectors */
a[href^="https"]::after { content: " (ext)";
                        font-size: 0.8em; }
```

The Box Model

Understanding content, padding, border, and margin

Every Element is a Box

Di CSS, setiap elemen HTML adalah kotak persegi panjang dengan 4 lapisan:

MARGIN (outside - transparent)



Dari dalam ke luar:

1. CONTENT - Teks/gambar sebenarnya (lebar x tinggi)
2. PADDING - Jarak antara konten dan border
3. BORDER - Garis di sekitar elemen
4. MARGIN - Jarak antara elemen ini dan elemen lain

Layer	What It Does	Visible?
Content	Teks, gambar	Ya
Padding	Jarak dalam	Background terlihat
Border	Garis tepi	Ya (jika diatur)
Margin	Jarak luar	Tidak (transparan)

Area Konten: Lebar & Tinggi

```
/* Fixed Size */
.box {
    width: 300px;    /* exactly 300px wide */
    height: 200px;   /* exactly 200px tall */
}

/* Percentage (relative to parent) */
.half-width {
    width: 50%;      /* half of parent width */
}

/* Auto (default behavior) */
.auto-box {
    width: auto;      /* fills available space */
    height: auto;     /* grows with content */
}

/* Min/Max - RESPONSIVE design! */
.responsive-container {
    width: 100%;      /* full width */
    max-width: 1200px; /* but max 1200px */
    min-width: 320px;  /* but min 320px */
    margin: 0 auto;    /* center it! */
}
```

HTML + CSS Example:

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Container & Card Layout</title>

    <style>
        body {
            background: #f4f4f4;
            font-family: Arial, sans-serif;
        }
        .container {
            width: 100%;
            max-width: 800px;
            margin: 0 auto; /* centered */
            padding: 20px;
            background: #eaeaea;
        }
        .card {
            width: 100%; /* fills container */
            min-height: 100px;
            background: white;
            margin-bottom: 15px;
            padding: 15px;
            box-shadow: 0 2px 5px rgba(0,0,0,0.1);
            border-radius: 6px;
        }
    </style>
</head>

<body>

    <div class="container">
        <div class="card">Card 1</div>
        <div class="card">Card 2</div>
    </div>

</body>
</html>
```

Padding: Space Inside the Border

Padding membuat jarak antara konten dan border. Warna background mengisi area padding.

```
/* Padding Syntax */
.box {
  /* All 4 sides equal */
  padding: 20px;

  /* Vertical | Horizontal */
  padding: 10px 30px;
  /* top/bottom=10px, left/right=30px */

  /* Top | Right | Bottom | Left */
  padding: 10px 20px 15px 25px;

  /* Individual sides */
  padding-top: 10px;
  padding-right: 20px;
  padding-bottom: 15px;
  padding-left: 25px;
}
```

HTML + CSS Example:

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Padding Example</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 20px;
    }
    .no-padding {
      background: lightblue;
      border: 2px solid blue;
      margin-bottom: 15px;
      /* Text touches the border */
    }
    .with-padding {
      background: lightgreen;
      border: 2px solid green;
      padding: 20px;
      margin-bottom: 15px;
      /* 20px space between text and border */
    }

    /* Common use: buttons need padding */
    .btn {
      padding: 12px 24px; /* 12px top/bottom, 24px left/right */
      background: #4CAF50;
      color: white;
      border: none;
      cursor: pointer;
      border-radius: 5px;
    }
  </style>
</head>

<body>
  <h2>Perbedaan Padding</h2>
  <div class="no-padding">No padding</div>
  <div class="with-padding">With padding</div>
  <h2>Contoh Button dengan Padding</h2>
  <button class="btn">Klik Saya</button>
</body>
</html>
```

Border: Tepi Elemen

```
/* Border Shorthand: width style color */
.box { border: 2px solid black; }

/* Border Styles */
.solid { border: 2px solid red; } /* _____ */
.dashed { border: 2px dashed blue; } /* - - - */
.dotted { border: 2px dotted green; } /* . . . */
.double { border: 4px double black; } /* ===== */
.none { border: none; } /* removed */

/* Individual Sides */
.card {
  border: none;
  border-bottom: 3px solid #4CAF50;
  /* Only bottom border! */
}

/* Border Radius (rounded corners) */
.rounded { border-radius: 10px; } /* rounded */
.pill { border-radius: 50px; } /* pill */
.circle { border-radius: 50%; } /* circle */
.custom {
  border-radius: 20px 0 20px 0;
  /* TL:20 TR:0 BR:20 BL:0 */
}
```

HTML + CSS Example:

```
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<title>Border Example</title>
<style>
body{font-family:Arial;background:#f4f4f4;padding:20px}

.card-basic{border:1px solid
#ddd;border-radius:8px;padding:20px;background:#fff;margin:10px 0}

.card-accent{border-left:5px solid
#2196F3;padding:20px;background:#f5f9ff;margin:10px 0}

.avatar{width:100px;height:100px;border-radius:50%;border:3px solid
#fff}

.pill-btn{border:none;border-radius:50px;padding:10px
30px;background:#4CAF50;color:#fff}
</style>
</head>
<body>

<h3>Card Basic</h3>
<div class="card-basic">Basic card</div>
<h3>Card Accent</h3>
<div class="card-accent">Accent card</div>
<h3>Avatar</h3>

<h3>Pill Button</h3>
<button class="pill-btn">Click Me</button>
</body>
</html>
```

Margin: Jarak di Luar Elemen

Margin membuat jarak ANTAR elemen. Selalu transparan (tidak punya warna background).

HTML + CSS Example:

```
/* Same syntax as padding */
.box {
    margin: 20px;           /* all sides */
    margin: 10px 30px;      /* vert horiz */
    margin: 10px 20px 15px 25px; /* T R B L */
}

/* CENTER a block element */
.centered {
    width: 800px;          /* MUST have width! */
    margin: 0 auto;        /* auto = equal sides */
    /* Result: element centered in parent */
}

/* MARGIN COLLAPSE (important!) */
.box1 { margin-bottom: 30px; }
.box2 { margin-top: 20px; }
/*
    Expected gap: 30 + 20 = 50px
    ACTUAL gap:   30px (larger one wins!)

    Vertical margins COLLAPSE.
    Horizontal margins do NOT collapse.
*/
```

```
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<title>Margin Example</title>
<style>
body{background:#f4f4f4;font-family:Arial}

.page-container{
    max-width:960px;
    margin:0 auto;
    padding:20px;
}

.section{
    background:#fff;
    padding:30px;
    margin-bottom:20px; /* space ke bawah saja */
}
</style>
</head>
<body>

<div class="page-container">
    <section class="section">Section 1</section>
    <section class="section">Section 2</section>
    <section class="section">Section 3</section>
</div>

</body>
</html>
```

Box-Sizing: Aturan CSS Terpenting

box-sizing mengubah CARA perhitungan lebar dan tinggi. Ini sangat penting dipahami!

content-box (default – confusing!):

```
/* DEFAULT: width = content ONLY */
.box {
  width: 300px;
  padding: 20px; /* adds to width! */
  border: 5px solid; /* adds to width! */
}
```

```
/* Actual rendered width:
content: 300px
+ padding-left: 20px
+ padding-right: 20px
+ border-left: 5px
+ border-right: 5px
= TOTAL: 350px !!!
```

You said 300px but got 350px!
This breaks layouts constantly. */

border-box (recommended!):

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Border Box Example</title>
<style>

/* WAJIB di paling atas */
*, ::before, ::after{
  box-sizing: border-box;
}

body{
  font-family: Arial;
  padding: 20px;
  background: #f4f4f4;
}

.box{
  width: 300px;
  padding: 20px;
  border: 5px solid black;
  background: #fff;
}

</style>
</head>
<body>

<div class="box">
  Lebar tetap 300px walaupun ada padding & border.
</div>

</body>
</html>
```

Properti Display: Perilaku Elemen

Display	Takes Full Width?	Starts New Line?	Can Set Width/Height?	Default Elements
block	Yes	Yes	Yes	div, p, h1-h6, section, header
inline	No	No	No (ignored!)	span, a, strong, em, img
inline-block	No	No	Yes	None (must set manually)
none	–	–	–	Element is completely hidden

Properti Display: Perilaku Elemen

```
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <title>Display Inline Example</title>
</head>
<body style="font-family: Arial; padding:20px;">

  <h2>Contoh Display: Inline</h2>

  <span style="background:yellow;">Inline 1</span>
  <span style="background:orange;">Inline 2</span>

  <p>
    Kedua span di atas tampil <strong>sejajar dalam satu
    baris</strong>
    karena secara default <code>span</code> adalah elemen
    inline.
  </p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Display Inline-Block</title>
  <style>
    .box{
      display:inline-block;
      width:120px;
      height:60px;
      background:lightgreen;
      margin:5px;
    }
  </style>
</head>
<body>

  <div class="box">Box 1</div>
  <div class="box">Box 2</div>

</body>
</html>
```

Properti Display: Perilaku Elemen

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Display None</title>
</head>
<body>

<p>Paragraf ini terlihat</p>
<p style="display:none;">Paragraf ini disembunyikan</p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Display Block</title>
<style>
.box{
    background: lightblue;
    margin:10px 0;
}
</style>
</head>
<body>

<div class="box">Block 1</div>
<div class="box">Block 2</div>

</body>
</html>
```

Complete Example: Card Component

HTML:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="card.css">
</head>
<body>
  <div class="card-container">

    <div class="card">
      <div class="card-image"></div>
      <div class="card-body">
        <h2 class="card-title">Card Title</h2>
        <p class="card-text">
          This card demonstrates the box model
          with padding, margin, and border.
        </p>
        <button class="card-btn">Learn More</button>
      </div>
    </div>

    <div class="card">
      <div class="card-image"></div>
      <div class="card-body">
        <h2 class="card-title">Another Card</h2>
        <p class="card-text">Same CSS classes!</p>
        <button class="card-btn">Read More</button>
      </div>
    </div>

  </div>
</body>
</html>
```

card.css:

```
/* Reset - ALWAYS start with this! */
*, ::before, ::after {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body { font-family: Arial; padding: 20px;
  background: #f0f0f0; }

/* Card container */
.card-container {
  max-width: 900px;
  margin: 0 auto; /* center page */
  display: flex; gap: 20px;
}

/* Card component */
.card {
  width: 300px; /* content width */
  background: white;
  border: 1px solid #ddd;
  border-radius: 12px; /* rounded corners */
  overflow: hidden; /* clip image */
  margin-bottom: 20px; /* space between */
}

.card-image {
  width: 100%; height: 180px;
  background: #ddd;
}

.card-body { padding: 20px; }
.card-title { margin-bottom: 10px; color: #333; }
.card-text { color: #666; margin-bottom: 15px;
  line-height: 1.6; }

.card-btn {
  display: inline-block;
  padding: 10px 24px; /* inner space */
  background: #4CAF50; color: white;
  border: none; border-radius: 6px;
  cursor: pointer;
}

.card-btn:hover { background: #388E3C; }
```

Tugas: Membuat Website Profile Company

Terapkan semua yang kamu pelajari dengan membangun halaman lengkap dari nol:

Proyek : Portofolio Pribadi Profile Company

- Menerapkan:
 - Selector (element, class, id)
 - Box model (margin, padding, border)
 - Display
 - Border & border-radius
 - Box-sizing
 - Pseudo-class (:hover)
 - Layout sederhana

- **Struktur Wajib**
 - Header
 - Logo (boleh text)
 - Navigation menu (Home, About, Services, Contact)
 - Hover effect pada menu
- **Hero Section**
 - Judul besar (h1)
 - Paragraf deskripsi perusahaan
 - Tombol "Contact Us"
 - Gunakan padding dan background
- **About Us**
 - Paragraf tentang perusahaan
 - Gunakan margin yang rapi
 - Tambahkan gambar berbentuk kotak atau rounded
 -
- **Contact Section**
 - Form sederhana:
 - Nama
 - Email
 - Pesan
 - Tombol submit
 - Gunakan styling input (padding & border-radius)