



CSS

Silver - Chapter 1 - Topic 3



Pada topik pertama, kita udah belajar tentang **web development** dan peran **fullstack web** di dalamnya.

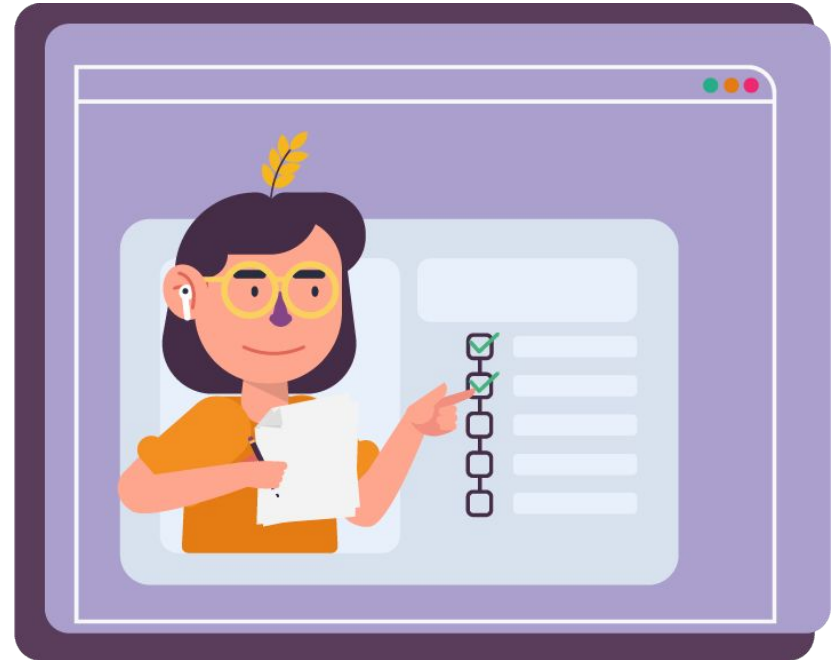
Pada topik kedua ini, kita bakal bahas tentang **HTML** dalam pengembangan suatu web. Cus langsung aja~





Detailnya, kita bakal bahas hal-hal berikut ini:

- Dasar CSS
- Struktur CSS
- Debugging HTML & CSS



Kita udah bahas HTML ditopik sebelumnya.
Tapi, sebenarnya HTML aja nggak cukup, lho!

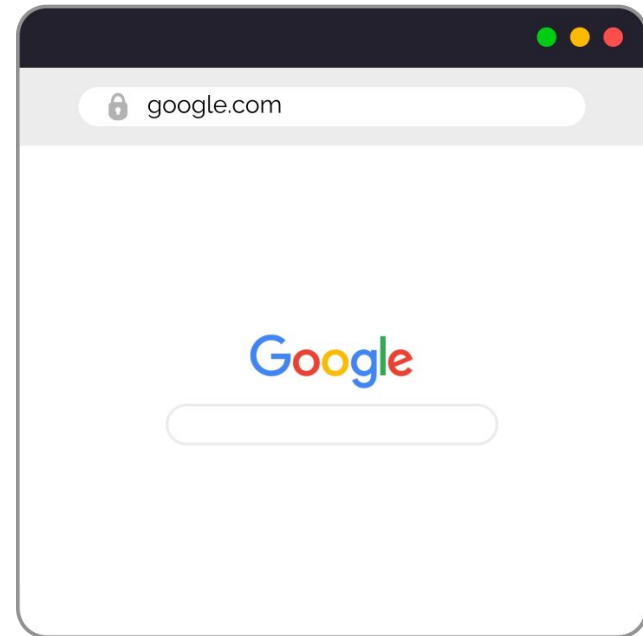
Ada yang namanya **CSS** biar tampilan web lebih ciamik~



Kayak tampilan Google di samping~

Pada materi di atas soal HTML, kamu udah tau kalo tampilan google kayak di samping dibuat menggunakan HTML.

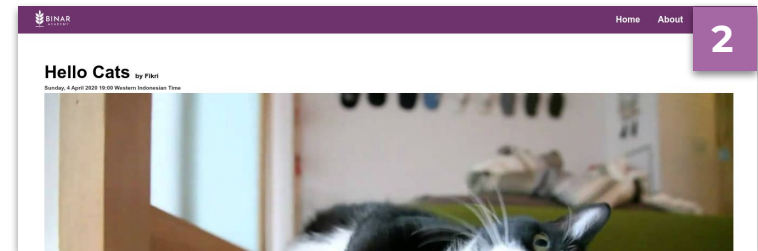
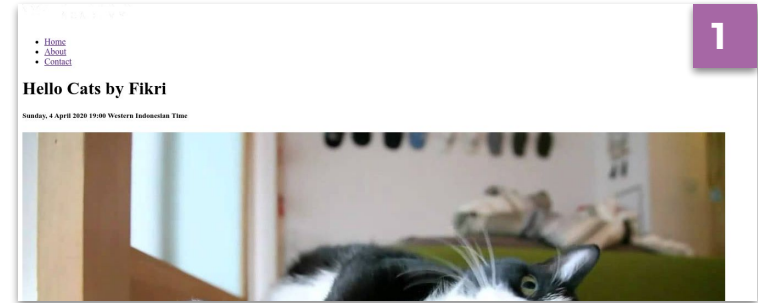
Tapi, HTML nggak bekerja sendiri, lho! Ada kontribusi CSS sehingga tampilan Google bisa ciamik dan berwarna-warni.



Biar lebih jelas, coba perhatikan perbedaan dua tampilan web di samping, ya!

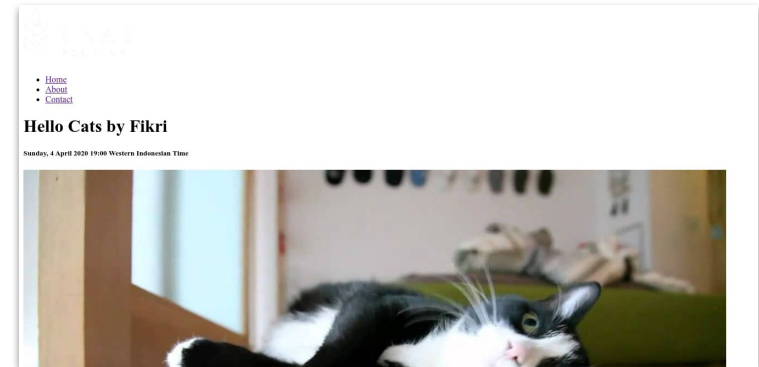
Website kedua terlihat lebih ciamik, kan? Yap~ itu karena udah diolah menggunakan CSS. Kalo web pertama, baru pake HTML aja, gaes!

CSS atau **Cascading Style Sheet** merupakan mekanisme yang mengatur gaya atau style pada halaman web agar memiliki visual yang menarik.



Supaya tampilan web ciamik, HTML dan CSS harus collab!

Soalnya, kalo web hanya dibuat menggunakan HTML tanpa dipoles CSS, hasilnya akan seperti gambar disamping! Nggak menarik kan?



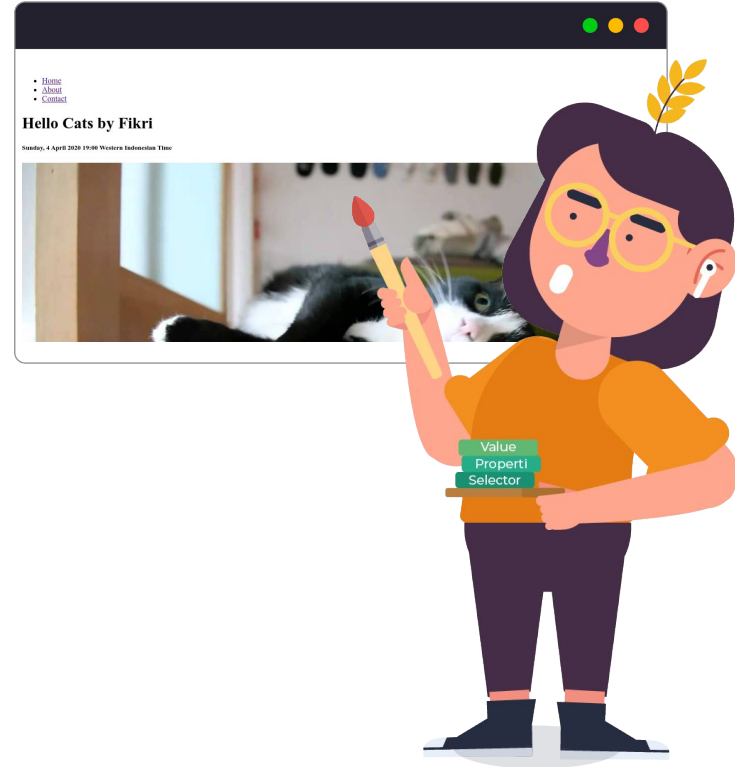
Terus, gimana sih cara HTML dan CSS bekerjasama?



Caranya kayak gini, gaes!

Saat dokumen HTML dibuat, aturan CSS diterapkan di dalamnya untuk menentukan cara dokumen itu ditampilkan.

Aturan CSS itu ada tiga, yaitu **selector**, **properti**, dan **value**. Tiga aturan ini juga disebut **struktur** atau **deklarasi**.





Selector itu elemen yang mau kita pilih untuk diterapkan properti tertentu. Misal, kita memilih paragraf pertama dari dokumen kita. Paragraf pertama itu adalah selector.

Properti adalah hal apa yang mau kita terapkan dalam suatu selector. Misal, paragraf pertama tadi mau kita coloring pake warna dan font size tertentu. Warna dan font size ini adalah properti.

Sementara **value** adalah bentuk dari properti. Misal, kita mau menggunakan warna hijau dengan font size 20. Hijau dan ukuran 20 adalah value.

Biar nggak bingung, coba liat gambar di samping yaa!

Fyi, kalo properti nya nggak jelas atau value nya nggak valid, maka browser engine CSS nggak akan bisa ngebaca dan hasilnya akan error. Misal, kamu memasukkan value “warna muntah kucing” dalam properti color. Engine CSS ya bakal bingung :’(

```
h1 {                               /* selector */
  color: green;
  font-size: 20px;
}
```

Ini Selector

```
h1 {
  color: green; /* property: value; */
  font-size: 20px;
}
```

ini namanya Properti

```
h1 {
  color: green; /* property: value; */
  font-size: 20px;
}
```

Ini namanya value



Nah, sekarang kita lihat gimana cara kerja HTML dan CSS.

Cara kerjanya ada tiga nih, gengs!

1. **Stylesheet eksternal**
2. **Stylesheet internal**
3. **Inline style**

Kita bahas satu persatu-satu, skoey!

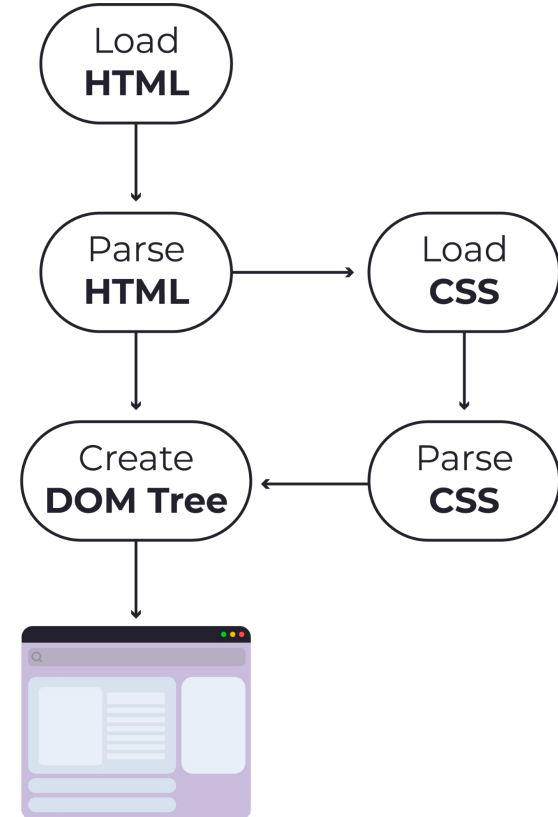


Eitss, sebelum kita bahas satu persatu-satu, kamu perlu tahu dulu cara kerja HTML - CSS secara umum.

Caranya cukup simpel kok, gengs! Ketika **browser menampilkan dokumen, dokumen itu harus di gabung dengan informasi stylenya**. Proses ini terdiri dari dua tahap :

1. Browser mengubah HTML dan CSS menjadi DOM (*Document Object Model*). DOM mewakili dokumen dalam memori komputer.
2. Browser akan menampilkan konten DOM.

Sekarang, kita bahas cara kerja HTML - CSS satu persatu, yuk! Cek di halaman selanjutnya yaa~





Stylesheet eksternal

Stylesheet eksternal adalah cara kerja HTML dan CSS dengan masing-masing file terpisah. Contohnya kayak gambar di samping ini!

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>My CSS experiment</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <h1>Hello World!</h1>
10    <p>This is my first CSS example</p>
11  </body>
12 </html>
```

index.html

```
1 h1 {
2   color: blue;
3   background-color: yellow;
4   border: 1px solid black;
5 }
6
7 p {
8   color: red;
9 }
```

style.css

Stylesheet internal

Stylesheet internal adalah cara kerja HTML dan CSS dengan masing-masing file yang digabung. Dengan kata lain, CSS tidak memiliki file terpisah. Jadi, CSS ditempatkan di dalam elemen `<style>` yang terdapat dalam kepala HTML. Contohnya kayak gambar di samping!

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <style>
      h1 {
        color: blue;
        background-color: yellow;
        border: 1px solid black;
      }

      p {
        color: red;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```



Inline style

Inline styles adalah cara kerja HTML dan CSS dimana elemen CSS hanya mempengaruhi satu elemen HTML yang terkandung dalam atribut **style**. Contohnya kayak gambar di samping!

Inline style ini sangat tidak dianjurkan karena tidak praktis dan merepotkan.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>My CSS experiment</title>
</head>

<body>
  <h1 style="color: blue;background-color: yellow;border: 1px solid black;">
    Hello World!
  </h1>
  <p style="color:red;">This is my first CSS example</p>
</body>

</html>
```



Kalau coding HTML dan CSS di ketiga cara diatas dijalankan, maka hasilnya akan seperti ini !





Nah, gimana sih kok bisa jadi kayak gitu? Yuk kita bahas detailnya~

Dalam file CSS tadi, kita menerapkan struktur untuk elemen-elemen yang ada pada file HTML.

Struktur pertama adalah selector `h1`, artinya kita menerapkan properti untuk selector `h1`. Nah, **properti yang diterapkan ada tiga**, yaitu `color`, `background-color`, dan `border`. Masing-masing properti juga memiliki **value seperti berikut ini**:

1. Warna biru merupakan value dari properti `color`.
2. Warna kuning merupakan value dari properti `background-color`. Properti pertama adalah `color`. Value nya warna biru.
3. Garis hitam berukuran 1 pixel merupakan value dari properti `border`.

Kombinasi struktur HTML dan CSS akan menghasilkan tampilan web lebih ciamik~

```
1 h1 {  
2   color: blue;  
3   background-color: yellow;  
4   border: 1px solid black;  
5 }  
6  
7 p {  
8   color: red;  
9 }
```

style.css



Sejauh ini, kamu udah paham kan apa itu HTML, CSS, dan gimana cara kerja antar keduanya? **Udah siap dong buat bikin website hehe~**





Pas proses bikin website, ada kemungkinan kamu mengalami berbagai permasalahan.

Misal, pas udah bikin HTML, eh hasilnya malah nggak sesuai ekspektasi. Padahal, udah sesuai sama tutorial~

Eitss, jangan galawww! Kamu bisa memecahkan permasalahan tersebut dengan **debugging**.





Apa sih debugging itu?

Debugging adalah **proses mencari permasalahan dalam website yang kamu buat**. Dengan kata lain, debugging itu proses membedah ulang apa yang udah dikerjakan dalam suatu website.



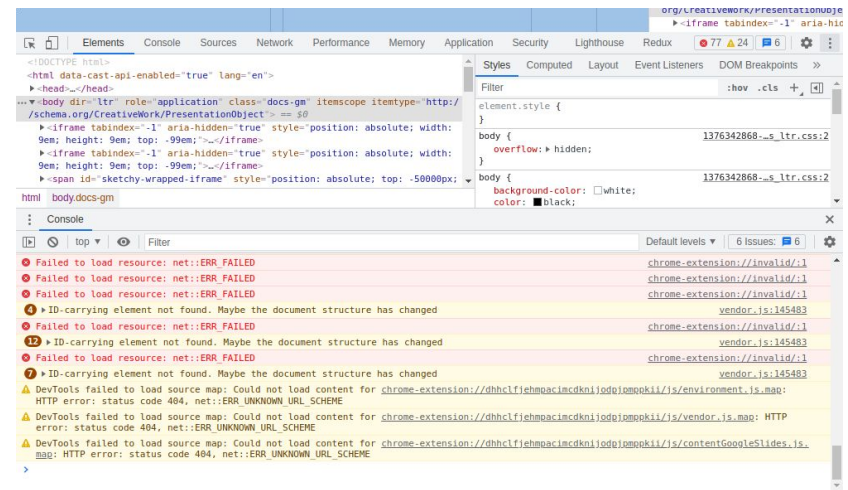


Gimana sih cara men-debugging?

Baik HTML maupun CSS, caranya sama aja. Kamu hanya perlu berbekal browser. Dalam browser, ada yang namanya **developer tools (devtools)**. Nah, devtools ini punya beberapa kegunaan, diantaranya:

- Untuk membedah struktur HTML yang sudah di-render di browser.
- Untuk membedah deklarasi CSS yang ada di browser.
- Untuk membedah hasil komputasi CSS yang akhirnya ditampilkan di browser.
- Melihat log dari eksekusi kode Javascript.

Devtools dapat diakses dengan menekan **F12** pada browser mu.





Sebagai contoh, coba perhatikan halaman web berikut ini!

Ada yang salah denganku?

Kayaknya sih gaada Aku Yakin

- **Unclosed elements:** If an element is **not closed properly**, then its effect can spread to areas you didn't intend
- **Badly nested elements:** Nesting elements properly is also very important for code behaving correctly. *strong strong emphasized? what is this?*
- **Unclosed attributes:** Another common source of HTML problems. Let's look at an example:



Source code nya seperti ini!

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="description" content="" />
    <meta name="author" content="" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, width=device-width" />
    <title>Adakah yang salah denganku?</title>
  </head>
  <body>
    <h1>Ada yang salah denganku?</h1>

    <p>Kayaknya sih gaada

    <a>Aku Yakin

  <ul>
    <li>Unclosed elements: If an element is <strong>not closed properly,
      then its effect can spread to areas you didn't intend

    <li>Badly nested elements: Nesting elements properly is also very important
      for code behaving correctly. <strong>strong <em>strong emphasized?</strong>
      what is this?</em>

    <li>Unclosed attributes: Another common source of HTML problems. Let's
      look at an example: <a href="https://www.mozilla.org/>link to Mozilla
      homepage</a>

  </ul>

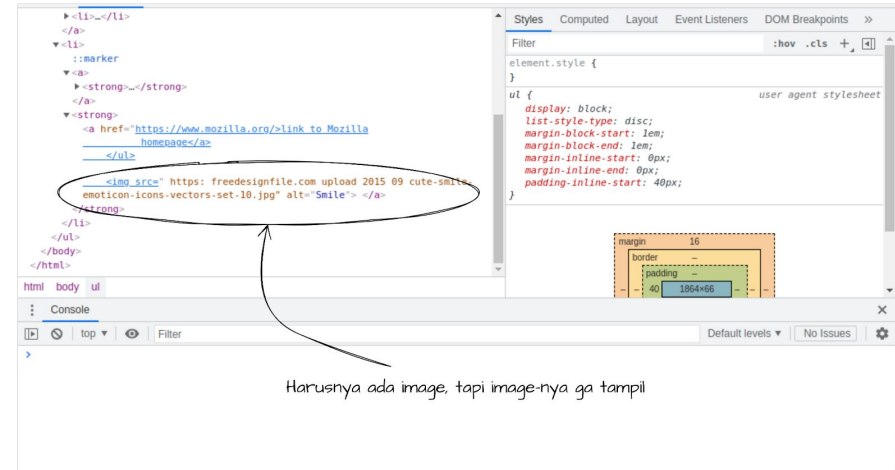
  
</body>
</html>
```



Kalo dilihat dari tampilannya sih nggak ada yang salah, tapi kalo kita melihat devtools nya, kita akan melihat permasalahan.

Seperti yang terlihat pada gambar di samping, seharusnya halaman tadi menampilkan image, tapi malah tidak tampil. Nah, setelah kita lihat dalam **devtools**, kita jadi tau ternyata ada sintaks pada source code yang belum tepat. Masih ada elemen-elemen yang belum ditutup.

Nah, karna udah tau dimana letak kesalahan, maka kita dapat membenahi kesalahan tersebut.



Saatnya kita
Quiz!





1. Bagaimana cara memilih elemen yang ingin kita beri aturan di CSS berdasarkan kelasnya?

- A. Menggunakan tanda pagar (#), sebagai contoh #nama_kelas
- B. Menggunakan tanda titik (.), sebagai contoh .nama_kelas
- C. Langsung kita sebut nama kelasnya seperti nama_kelas
- D. Menggunakan pseudo element seperti nama_kelas::after



1. Bagaimana cara memilih elemen yang ingin kita beri aturan di CSS berdasarkan kelasnya?

- A. Menggunakan tanda pagar (#), sebagai contoh #nama_kelas
- B. Menggunakan tanda titik (.), sebagai contoh .nama_kelas
- C. Langsung kita sebut nama kelasnya seperti nama_kelas
- D. Menggunakan pseudo element seperti nama_kelas::after

A.

Untuk menerapkan suatu aturan berdasarkan kelasnya di dalam stylesheet, kita memakai syntax `.nama_kelas`



2. Untuk membuat internal stylesheet, elemen apa yang kita butuhkan?

- A. `<link>`
- B. `<a>`
- C. `<button>`
- D. `<style>`



2. Untuk membuat internal stylesheet, elemen apa yang kita butuhkan?

- A. `<link>`
- B. `<a>`
- C. `<button>`
- D. `<style>`

Untuk melakukan internal stylesheet di HTML kita membutuhkan elemen `<style>`



3. Untuk membuat eksternal stylesheet, elemen apa yang kita butuhkan?

- A. `<link>`
- B. `<a>`
- C. `<button>`
- D. `<style>`



3. Untuk membuat eksternal stylesheet, elemen apa yang kita butuhkan?

- A. `<link>`
- B. `<a>`
- C. `<button>`
- D. `<style>`

Sebenarnya link bisa dipakai untuk banyak hal, seperti import file lain, mengganti favicon dan lain, Tapi untuk mengimport eksternal stylesheet, kita harus menggunakan link.



Referensi dan bacaan lebih lanjut~

1. https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Values_and_units
2. https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Box_model
3. https://www.w3schools.com/cssref/css_selectors.asp
4. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors
5. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started





Nah, selesai sudah pembahasan kita di Chapter 1 Topic 3 ini.

Selanjutnya, kita bakal bahas tentang **CSS Framework**.

Penasaran kayak gimana? Cus langsung ke topik selanjutnya~



Terima Kasih!



Next Topic

loading...