

# Review 07-01-2023

---

## HTML

Sebelum kita mulai mengetik **HTML**, **HTML** merupakan singkatan dari Hyper Text Markup Language, bahasa ini digunakan untuk membuat struktur suatu website, ekstensi file ialah **.html**.

Didalam file **HTML** tadi, kita harus mengetik `<!DOCTYPE html>` untuk menandai file tersebut memang **HTML**, lalu dilanjutkan dengan element `<html>...</html>` dan kita memulai mengetik kode **HTML** disitu.

Element merupakan tag pembuka dan tag penutup tetapi beberapa element bisa tidak dikasih tag penutup sendiri.

Struktur di element `<html>...</html>` biasanya terdapat element `<body>` yang digunakan sebagai konten utama, dan element `<head>` yang ber-isi dengan metadata seperti **title**, link **css** ataupun **js**

berikut macam macam element HTML dan kegunaan umumnya :  
untuk element `<head>`

1. `<title>` : Digunakan untuk memberi judul di tab browser.
2. `<meta>` : Metadata suatu website (lebih in-depth disini <https://htmlhead.dev/>).
3. `<link>` : Untuk menambahkan css external.
4. `<script>` : Untuk menambahkan javascript external (bisa ditaruh di `<body>`).
5. `<style>` : Digunakan untuk membuat script css di **HTML**.

untuk element `<body>`

1. `<p>` : Digunakan untuk membuat suatu paragraf.
2. `<a>` : Digunakan untuk membuat link ke halaman lain.
3. `<h1>` (h1-h6) : Digunakan untuk membuat judul/header.
4. `<img>` : Digunakan untuk menambah gambar ke website.
5. `<table>` : Digunakan untuk membuat table biasanya anak didalam `<table>` berisi `<tr>` lalu didalam `<td>` terkadang di `<tr>`.
6. `<form>` : Untuk mengambil data data user isi.
7. `<input>` : Untuk memberi textbox, cocok digunakan di `<form>`.
8. `<button>` : Digunakan untuk membuat tombol.
9. `<ul>` / `<ol>` : Digunakan untuk membuat list dan didalamnya terdapat `<li>` untuk membuat poin poin list.
10. `<div>` : sebuah container.

ada beberapa element yang menerima attribute, attribute ini digunakan untuk menyimpan informasi tambahan, misalnya di element `<a>` harus memiliki attribute **href** untuk bisa berfungsi dengan sempurna, penulisannya sangat mudah tinggal ketik :

```
<a href="https://www.w3schools.com/html/html_attributes.asp">W3</a>
```

Note :

-> Untuk element `<img>` membutuhkan attribute **src** agar bisa menampilkan gambar dan dapat ditambah **alt** untuk menampilkan teks ketika gambar gagal ditampilkan.

- > Untuk element `<script>` membutuhkan attribute `src` adalah link yang mengarah javascript external.
- > Untuk element `<link>` agar bisa memakai css external dengan benar harus diketik `<link rel="stylesheet" href="nama_file_css.css">`

Kita juga dapat memberikan komentar didalam `HTML` kita agar kita atau orang lain bisa memahami apa maksud dari struktur `HTML` yang dimaksud, untuk penulisannya :

```
<!-- Komentar taruh sini -->
```

Untuk lebih lanjut bisa ditanyakan atau browsing di internet khususnya di <https://www.w3schools.com/html/>.

## CSS

CSS Digunakan untuk memperindah halaman website yang kita buat, untuk pemakaiannya ada tiga cara dengan menggunakan attribute `style` atau element `<style>` atau dengan file external dengan menggunakan element `<link>`, untuk urusan penulisannya hampir mirip dengan ketiga cara tadi, untuk extensi file css `.css`.

Karena CSS ini ada banyak sekali bisa dibaca di :

<https://www.w3schools.com/css/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

Sebelum kita memberikan format/style kepada element `HTML`, kita harus men-select dengan `Selector`, `Selector` ini ada 3 jenis.

- > menggunakan `*` atau dengan kata lain global : ini akan men-select keseluruhan element `HTML` contoh :

```
* {  
  color: red; /*ini akan membuat semua text ber-warna merah*/  
}
```

- > menggunakan nama element : ini akan men-select di element spesifik contoh :

```
p {  
  color: red; /*ini akan membuat semua text yang berada di element <p> ber-warna merah*/  
}
```

- > menggunakan class : ini akan men-select element yang memiliki attribute class yang sama, dan dipisah dengan spasi kalau lebih dari 1, contoh :

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

Note :

Untuk CSS yang menggunakan attribute langsung ditulis perintah style CSS langsung tanpa harus menuliskan

selectornya

Berikut beberapa kode style selain color :

- > `padding: {angka}px` => digunakan untuk memberi jarak antara konten dengan border.
- > `margin: {angka}px` => digunakan untuk memberi jarak antara border dengan element yang lain.
- > `font-family: {nama-font}, {nama-font}` => digunakan untuk mengganti font.
- > `font-size: {angka}px` => digunakan untuk mengganti ukuran font.
- > `height: {angka}px` => digunakan untuk memberi tinggi yang spesifik kepada element.
- > `width: {angka}px` => digunakan untuk memberi lebar yang spesifik kepada element.
- > `box-shadow: {offset-x}px {offset-y} {blur-radius}px {spread-radius}px {warna}` => untuk memberi efek bayangan di element.
- > `text-shadow: {offset-x}px {offset-y} {blur-radius}px {spread-radius}px {warna}` => untuk memberi efek bayangan teks di suatu element.
- > `border: {angka}px {solid, dotted} {warna}` => digunakan untuk memberi tebal border/pembatas dan warnanya.
- > `opacity: {angka}%` => digunakan untuk transparansi suatu element.
- > `background-color: {warna}` => digunakan untuk memberi warna background di element.
- > `border-radius: {ukuran/persen}` => digunakan untuk mengatur ketumpulan border.
- > `overflow: {auto, hidden}` => digunakan untuk mengatur cara untuk menampilkan konten jika melebihi batas yang sudah ditentukan.
- > `text-decoration: {none, dotted, underline, line-through, overline}` => digunakan untuk mengatur dekorasi font/text.
- > `background-image: url({url})` => digunakan untuk memberi background gambar disuatu element.

dan masih banyak lagi

Note :

- > Tidak hanya ada `px` saja, bisa pakai `em` atau `rem`, kedua tersebut relative dengan ukuran font yang digunakan browser, atau menggunakan `vw` dan `vh` yang relative dengan ukuran tampilan browser.
- > Untuk `padding`, `margin`, `border` memiliki beberapa tambahan seperti `margin-left`, `right`, `top`, `bottom`.
- > Untuk `{}` berarti dapat di isi dengan apa yang di inginkan, tetapi harus sesuai dengan penamaan didalam `{}`

Dan ada hal yang paling penting ketika menyusun layout website, paling mudah menggunakan `flexbox`, ini dapat digunakan untuk membuat berbagai macam desain dan layout.

Pertama-tama harus menyalakannya dengan `display: flex;` lalu dikasih arah flex `flex-direction: {arah}` mulai dari `coloumn`, `row`, `reverse-row`, dll, kita juga bisa memberikan `gap: {angka}px` untuk memberi jarak dengan element didalam `flexbox`.

## Javascript

Javascript merupakan bahasa pemrograman yang High Level dan memiliki integrasi dengan web tetapi juga dapat digunakan untuk bagian server juga, bahasa ini merupakan interpreted, cara pendefinisian Javascript di web ada dua :

1. Dengan `<script>` dan nanti kode Javascript diselipkan didalam element tersebut.
2. Dengan file `.js` external dengan menggunakan element `<script type="text/javascript" src="nama-file.js"></script>`.

Mengingat kembali tipe data yang ada di Javascript

- > Char/Character : 1 Huruf.
- > String : Banyak huruf.
- > Boolean/Bool : true/false (iya atau tidak).
- > Number/Integer : merepresentasikan angka.
- > Undefined/Null : tipe data kosong tetapi undefined dan null beda (javascript begitu).
- > Function : merupakan lambda function / Anonymous function.
- > Array : merupakan kumpulan data.
- > JavaScriptObject : merupakan tipe data yang berbentuk key-value pair (key: value)

Paling simpel untuk menampilkan data kita bisa menggunakan `console.log` tetapi itu hanya akan muncul dengan console (browser, atau console yang lain) untuk membuka console di browser tergantung di browsernya (kalau di firefox tinggal tekan F12).

Pendeklarasian variabel ada 3 cara

- > `var nama_variabel = data_pertama;` => Variabel ini bersifat global jadi bisa diakses dimana saja, jika tidak ditaruh didalam function.
- > `let nama_variabel = data_pertama;` => Variabel ini hanya hidup di scope yang didefinisikan.
- > `const NAMA_VARIABEL = data_pertama;` => Variabel ini hanya hidup di scope yang di definisikan tetapi tidak bisa dimodifikasi.

Contoh :

```
let namaSiswa = "Zagar Dangure";
const namaGuru = "Pak Antoni";
var namaSiswa2 = "Alpin";
```

Apa itu **Scope** : Merupakan block kode yang mendefinisikan panjang hidup suatu variabel, di Javascript block tersebut dimulai dengan `{` dan ditutup dengan `}`.

Pendeklarasian Array sangat simpel sekali kita

```
let namaSiswa = ["Zagar", "Alpin", "Syahril"];
```

Kita bisa mengatur alur program dengan menggunakan `if`, `if` ini akan melakukan perbandingan di kondisi apakah layak untuk menjalankan block kodenya.

```
const kondisi = true;

if (kondisi) {
  console.log("Dijamin jalan, karena kondisi berisi true");
} else {
  console.log("Hanya jalan ketika kondisi false");
}
```

Jika kalau kita memiliki banyak kondisi yang harus kita bandingkan, kita bisa menggunakan `&&` kalau kedua/lebih kondisi harus benar atau `||` untuk salah satu kondisi saja yang harus benar.

```
let a = 1;
let kondisi = true;

if (kondisi && a == 1) {
  console.log("Dijamin jalan, karena kondisi berisi true");
} else {
  console.log("Hanya jalan ketika kondisi false");
}
```

Kalau kita ingin banyak sekali alur program, kita bisa memakai `else if`.

```
let nilai = 70;

if (nilai > 75) {
  console.log("Dijamin Tidak jalan");
} else if (nilai > 55) {
  console.log("Dijamin jalan");
} else if (nilai < 55) {
  console.log("Jelek amat nilainya");
}
```

Kalau semisal jelek melihat banyak sekali if else if, kita bisa menggunakan `switch`.

```
let hari = 'Rabu';
switch (hari) {
  case "Senin":
    // Kode
    break;
  case "Selasa":
    // Kode
    break;
  case "Rabu":
    // Kode
    break;
  case "Kamis":
    // Kode
    break;
  case "Jum'at":
    // Kode
    break;
  case "Sabtu":
    // Kode
    break;
  case "Minggu":
    // Kode
    break;
}
```

```

        break;
    default:
        // Kalau di-atas tidak ada yang cocok ini jalan
        break
    }

```

Kadang kita butuh pengulangan didalam kode kita, kita bisa menggunakan **while** dan **for** untuk melakukan perulangan kita, kedua perulangan tersebut berbeda jauh.

```

// variabel inisialisasi buat loop; pengecekan kondisi apakah layak loop
dilanjutkan; apa yang harus dijalankan setelah block kode selesai dijalankan
for(let i = 0; i < 5; i++) {
    // Melakukan kegiatan
}

let i = 0;
while(i < 5) {
    // Melakukan kegiatan
    i++;
}

```

Array Javascript mempunyai **Iterator**, fitur ini dapat mempermudah kita dalam melakukan loop/perulangan.

```

const myArray = ["Zagar", "Alpin", "Alifah", "Ghaza"];

// Bisa const atau let tergantung dengan mau diapakan
// value merupakan representasikan isi
// dan indek merupakan representasi posisi di dalam array
for (const [index, value] of myArray.entries()) {
    console.log(`${index} : ${value}`);
}
for (const value of myArray) {
    console.log(value)
}

```

Terakhir, Function di Javascript ini dapat membantu kita mengurangi duplikasi kode, dan cara pendeklarasiannya sangat sederhana.

```

function printNama(nama) {
    console.log("Hai " + nama);
}

printNama("Agus");

```

Note :

-> Karena web itu sangat luas dan berkembang cepat, jadi sulit untuk dibahas dengan waktu yang terbatas,

jadi aku rekomendasikan untuk membaca dokumen ataupun tutorial.

-> Kalau semisal bingung mau latihan apa, bisa tanya aja rekomendasi buat latihan (terakhir membuat halaman login memakai HTML, CSS, Javascript)