Muhammad Gus Nadir

202110370311481

**HTML (Hypertext Markup Language)**

Pada intinya, HTML adalah serangkaian kode singkat yang diketik ke dalam sebuah file teks. Ini adalah tag-tag yang memberi daya pada kemampuan HTML. Teks disimpan sebagai file HTML dan dilihat melalui peramban web. Peramban tersebut membaca file dan menerjemahkan teks ke dalam bentuk yang terlihat, sesuai dengan kode-kode yang digunakan oleh penulis untuk membuat apa yang menjadi hasil tampilan yang terlihat. Menulis HTML memerlukan penggunaan tag secara benar untuk menciptakan visi dari penulis.

Tag-tag adalah yang memisahkan teks normal dari kode HTML. Tag-tag adalah kata-kata yang berada di antara apa yang dikenal sebagai tanda kurung sudut, yang memungkinkan grafik, gambar, dan tabel muncul di halaman web. Tag-tag yang berbeda memiliki fungsi yang berbeda. Tag-tag paling dasar mengaplikasikan pemformatan pada teks. Saat antarmuka web perlu menjadi lebih dinamis, Cascading Style Sheets (CSS) dan aplikasi JavaScript dapat digunakan. CSS membuat halaman web lebih dapat diakses dan JavaScript menambahkan kekuatan pada HTML dasar.

* Gunakan elemen HTML semantik seperti <header>, <nav>, <main>, <article>, dan lainnya untuk memberikan makna dan struktur yang jelas pada halaman.
* Desain halaman web agar responsif, artinya dapat menyesuaikan tampilan dengan baik di berbagai perangkat dan ukuran layar.
* Gunakan Cascading Style Sheets (CSS) untuk mengatur tampilan dan gaya halaman web secara terpisah dari struktur HTML.
* Ukuran file gambar harus dioptimalkan agar halaman web dapat memuat lebih cepat. Gunakan format gambar yang tepat (JPEG untuk foto, PNG untuk grafik dengan latar belakang transparan, dll.).
* Usahakan halaman web dapat diakses oleh semua orang, termasuk mereka yang memiliki disabilitas. Gunakan atribut alt pada gambar, dan perhatikan prinsip aksesibilitas lainnya.
* Pastikan halaman web dapat berfungsi dengan baik di berbagai browser, seperti Chrome, Firefox, Safari, dan Edge.
* Selalu komentari kode Anda dengan baik. Ini akan membantu kita dan orang lain dalam memahami kode di masa depan.
* Pisahkan konten, gaya, dan perilaku (HTML, CSS, dan JavaScript) dalam tiga file terpisah untuk memudahkan pemeliharaan.
* Selalu berhati-hati dengan keamanan web. Validasi input dari pengguna dan hindari penyusupan SQL (SQL injection) atau serangan sumber lintas situs (Cross-Site Scripting, XSS).

Contoh Sederhana HTML

1. <!**DOCTYPE** html> //mendefinisikan bahwa dokumen ini adalah dokumen HTML.
2. <**html**> //elemen akar (root) dari halaman HTML yang mengelilingi semua elemen lain dalam halaman HTML.
3. <**head**> //berisi informasi tentang halaman, seperti judul dan metadata.
4. <**title**>programiz</**title**>  //menentukan judul untuk halaman HTML yang ditampilkan di bilah judul peramban.
5. </**head**>
6. <**body**> //mendefinisikan konten utama halaman dan merupakan wadah untuk semua konten yang terlihat, seperti judul, paragraf, daftar, dll.
7. <**h1**>HTML Tutorial</**h1**>  //mendefinisikan sebuah judul.
8. <**p**>You'll learn about HTML.</**p**> //mendefinisikan sebuah paragraf dalam dokumen HTML.
9. </**body**>
10. </**html**>

**CSS (Cascading Style Sheet)**

CSS menjelaskan bagaimana elemen-elemen HTML akan ditampilkan di layar, kertas, atau media lainnya. CSS digunakan untuk mendefinisikan gaya-gaya pada halaman web , termasuk desain, tata letak, dan variasi tampilan untuk perangkat yang berbeda dan ukuran layar yang berbeda.Stylesheet eksternal disimpan dalam file CSS.

* Kelompokkan properti-properti yang sama bersama-sama untuk memudahkan pemeliharaan. Misalnya, kelompokkan semua properti font bersama-sama.

Contoh:

1. **font-family**: Arial, sans-serif;
2. **font-size**: 16px;
3. **font-weight**: bold;

* Usahakan untuk tidak menggunakan gaya inline (misalnya, menggunakan atribut style dalam elemen HTML), kecuali diperlukan.
* Letakkan CSS dalam file eksternal terpisah. Ini memisahkan gaya dari struktur halaman web dan membuatnya lebih mudah untuk dikelola dan di-maintain.
* Pastikan desain web responsif. Gunakan media queries untuk mengatur tampilan halaman web pada berbagai perangkat dan resolusi layar.

Contoh :

1. @media (**max-width**: 768px) {
2. /\* Aturan CSS untuk perangkat dengan lebar maksimum 768px \*/
3. }

* Ketika memungkinkan, pertimbangkan menggunakan Flexbox atau Grid Layout untuk mengatur tata letak elemen-elemen dalam halaman web. Ini memberikan kontrol yang lebih baik dan meningkatkan responsivitas.

Aturan struktur CSS cukup sederhana. Selektor menunjuk pada elemen HTML yang ingin digayakan atau distyling, blok deklarasi berisi satu atau lebih deklarasi yang dipisahkan oleh titik koma. Setiap deklarasi mencakup nama properti CSS dan nilai, dipisahkan oleh titik dua. Deklarasi CSS selalu diakhiri dengan titik koma, dan blok deklarasi dikelilingi oleh kurung kurawal.

Contoh:

Semua elemen <p> akan berwarna biru dan diberi tebal.

1. <style>
2. p {
3. **color**: blue;
4. **font-weight**: bold;
5. }
6. </style>

**JavaScript**

JavaScript adalah bahasa pemrograman komputer yang dinamis. Bahasa ini ringan dan paling umum digunakan sebagai bagian dari halaman web, di mana implementasinya memungkinkan skrip sisi klien berinteraksi dengan pengguna dan membuat halaman yang dinamis.

Client-side JavaScript adalah bentuk paling umum dari bahasa ini. Skrip harus disertakan dalam atau dirujuk oleh dokumen HTML agar kode tersebut dapat diinterpretasikan oleh peramban web. Ini berarti bahwa sebuah halaman web tidak perlu menjadi HTML statis, tetapi dapat mencakup program-program yang berinteraksi dengan pengguna, mengendalikan peramban, dan secara dinamis membuat konten HTML.

Keuntungan JavaScript

Beberapa keunggulan penggunaan JavaScript adalah sebagai berikut:

* Interaksi dengan server yang lebih sedikit – Kita dapat memvalidasi input pengguna sebelum mengirimkan halaman ke server. Hal ini menghemat lalu lintas server, yang berarti beban yang lebih ringan pada server.
* Umpan balik langsung kepada pengunjung - Pengunjung tidak perlu menunggu halaman untuk dimuat ulang untuk melihat apakah mereka telah melewatkan sesuatu.
* Interaktivitas yang lebih tinggi – kita dapat membuat antarmuka yang bereaksi ketika pengguna mengarahkan kursor mouse pada elemen atau mengaktifkannya melalui papan ketik.
* Antarmuka yang lebih kaya - kita dapat menggunakan JavaScript untuk menyertakan elemen-elemen seperti komponen seret-dan-lepas (drag-and-drop) dan slider untuk memberikan Antarmuka yang Kaya kepada pengunjung situs.

Keterbatasan JavaScript

* Tidak dapat membaca atau menulis file secara langsung pada sisi klien. Ini dilakukan karena pertimbangan keamanan.
* Tidak dapat digunakan untuk aplikasi jaringan karena tidak ada dukungan yang tersedia.
* Tidak memiliki kemampuan multi-threading atau multi-prosesor.

JavaScript dapat disisipkan langsung ke dalam halaman web atau dimasukkan dalam file dengan ekstensi .js. Biasanya, JavaScript lebih baik ditempatkan setelah HTML dan CSS telah dimuat di halaman web, sehingga biasanya diletakkan di bagian penutup tag <body> dalam HTML. Berikut adalah contoh template dasar HTML yang mencakup penempatan JavaScript di bagian akhir <body>:

1. <html>
2. <head>
3. <title>Judul Halaman Anda</title>
4. <!-- Masukkan stylesheet CSS Anda di sini -->
5. </head>
6. <body>
7. <!-- Konten HTML Anda berada di sini -->
9. <script type="text/javascript">
10. //I am Javascript. I make website funny place.
11. </script>
12. <!-- OR With External .js File-->
13. <script type="text/javascript" src="./myFile/myScript.js">
14. </script>
15. </body>
16. </html>

Reference

<https://www.investopedia.com/terms/h/html.asp#toc-hypertext-markup-language-basics>

<https://www.programiz.com/html/what-is-html#:~:text=HTML%20is%20a%20standard%20language,pages%20more%20engaging%20and%20interactive>.

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

<https://www.tutorialspoint.com/javascript/javascript_overview.htm>

<https://medium.com/analytics-vidhya/what-is-javascript-a-really-brief-explanation-3cd111dd87d9>