

Nama : Rezky Juliansyah Putra

Nim : 202110370311429

Kelas : pemrograman web (i)

Html

Elemen: Elemen adalah komponen dasar dalam HTML. Mereka biasanya terdiri dari tag pembuka, konten, dan tag penutup. Tag pembuka menandakan awal dari sebuah elemen, tag penutup menandakan akhirnya, dan konten adalah isi dari elemen tersebut. Contoh sederhana adalah elemen paragraf `<p>`, yang memiliki tag pembuka `<p>` dan tag penutup `</p>` di sekitar teks paragraf.

Struktur Dasar: Setiap dokumen HTML harus memiliki struktur dasar yang terdiri dari elemen-elemen seperti `<html>`, `<head>`, dan `<body>`. Elemen `<html>` adalah elemen terluar dan mengelilingi seluruh halaman web. Elemen `<head>` berisi informasi tentang halaman, seperti judul, meta tag, dan tautan ke file eksternal. Elemen `<body>` berisi konten yang akan ditampilkan kepada pengguna, seperti teks, gambar, dan tautan.

Tag: Tag adalah perintah yang digunakan dalam HTML untuk menentukan jenis elemen dan cara elemen tersebut akan ditampilkan. Tag umumnya ditulis dalam bentuk pasangan, yaitu tag pembuka dan tag penutup. Beberapa tag, seperti `` untuk gambar, tidak memiliki tag penutup dan dianggap lengkap dalam satu tag.

Atribut: Atribut adalah informasi tambahan yang dapat ditambahkan ke tag untuk memberikan detail atau pengaturan lebih lanjut. Misalnya, atribut `src` pada tag `` digunakan untuk menentukan sumber (URL) gambar yang akan ditampilkan.

Hyperlink: HTML digunakan untuk membuat tautan atau hyperlink antara halaman web. Ini dilakukan dengan menggunakan tag `<a>` (anchor) yang memiliki atribut `href` untuk menunjukkan URL halaman atau sumber yang akan dihubungkan.

List dan Tabel: HTML mendukung pembuatan list (daftar) dan tabel. Tag `` dan `` digunakan untuk membuat daftar tak berurutan (unordered list) dan daftar berurutan (ordered list), sementara tag `<table>`, `<tr>`, `<td>`, dan `<th>` digunakan untuk membuat tabel.

Formulir: HTML juga digunakan untuk membuat formulir interaktif. Tag seperti `<form>`, `<input>`, `<textarea>`, dan `<button>` digunakan untuk mengumpulkan input pengguna, seperti teks, pilihan, atau tombol yang dapat diklik.

Komentar: Komentar dalam HTML digunakan untuk menambahkan catatan atau dokumentasi ke dalam kode HTML. Komentar tidak akan ditampilkan di halaman web dan dimulai dengan `<!--` dan diakhiri dengan `-->`.

css

Selektor: Selektor adalah bagian dari aturan CSS yang digunakan untuk menentukan elemen mana yang akan diubah gaya tampilannya. Contoh selektor sederhana adalah nama tag HTML seperti `<p>`, `<h1>`, atau `<div>`. Ada juga selektor yang lebih spesifik seperti kelas (`.kelas-saya`) dan ID (`#id-saya`).

Deklarasi: Deklarasi adalah bagian aturan CSS yang berisi gaya atau properti yang akan diterapkan pada elemen yang dipilih. Setiap deklarasi terdiri dari dua bagian, yaitu properti dan nilainya, dipisahkan oleh tanda titik dua (`:`). Misalnya:

```
p {  
    color: blue;  
    font-size: 16px;  
}
```

Dalam contoh di atas, dua properti yang dideklarasikan adalah `color` (warna teks) dan `font-size` (ukuran font) untuk elemen `<p>`.

Kaskade: CSS memiliki konsep kaskade, yang mengatur bagaimana aturan CSS diterapkan pada elemen. Jika ada beberapa aturan yang berlaku untuk elemen yang sama, kaskade akan menentukan urutan prioritas dan bagaimana aturan-aturan itu digabungkan. Prioritas dapat ditentukan oleh jenis selektor, spesifikitas, dan urutan penulisan.

Warisan: Beberapa properti CSS dapat diwariskan dari elemen yang lebih tinggi ke elemen yang lebih rendah dalam struktur dokumen HTML. Ini berarti, jika Anda mengatur gaya di elemen induk, anak-anaknya akan mewarisi sebagian atau seluruh gaya tersebut, kecuali jika ada aturan yang menyimpannya.

CSS Eksternal dan Internal: CSS dapat ditempatkan di dalam elemen HTML menggunakan tag `<style>`, atau sebagai file terpisah dengan ekstensi `.css` yang dihubungkan dengan elemen HTML menggunakan tag `<link>`. CSS eksternal lebih umum digunakan untuk memisahkan tampilan dari struktur konten HTML.

Pseudo-class dan Pseudo-element: Pseudo-class dan pseudo-element adalah cara untuk memilih elemen berdasarkan keadaan atau posisinya dalam dokumen. Contoh pseudo-class adalah `:hover` yang digunakan untuk mengubah tampilan elemen saat kursor berada di atasnya. Pseudo-element seperti `::before` dan `::after` digunakan untuk menambahkan konten tambahan ke elemen.

Media Queries: Media queries memungkinkan Anda mengubah tampilan elemen berdasarkan ukuran layar atau perangkat yang digunakan pengguna. Ini penting untuk membuat desain responsif yang dapat menyesuaikan diri dengan berbagai perangkat dan layar.

JAVA SCRIPT

`console.log` untuk print

`("")` untuk string

`(angka)` untuk int

`(""+ "")` untuk menggabungkan String

`#` untuk mengupdate variable bisa menggunakan

```
let nama = "ujang"
```

```
console.log(nama);
```

```
nama="saipul"
```

```
console.log(nam);
```

maka input (ujang) dan (saipul)

Struktur Kontrol:

- Percabangan (if-else): Dengan menggunakan pernyataan if dan else, Anda dapat menjalankan kode berdasarkan kondisi tertentu.

javascriptCopy code

```
var usia = 18; if (usia >= 18) { console.log("Anda dewasa"); } else { console.log("Anda masih anak-anak"); }
```

- Perulangan (for, while): Anda dapat menggunakan perulangan for atau while untuk menjalankan kode secara berulang.

javascriptCopy code

```
for (var i = 0; i < 5; i++) { console.log("Iterasi ke-" + i); }
```

3. Fungsi:

- Mendefinisikan Fungsi: Anda dapat membuat fungsi untuk menjalankan kode tertentu.

javascriptCopy code

```
function sapa(nama) { console.log("Halo, " + nama + "!"); }
```

Memanggil Fungsi: Panggil fungsi dengan memberikan argumen.

javascriptCopy code

```
sapa("John"); // Output: "Halo, John!"
```

4. Objek:

- Objek Literal: Anda dapat membuat objek literal untuk mengelompokkan data.

javascriptCopy code

```
var orang = { nama: "John", usia: 30 };
```

- Mengakses Properti Objek: Akses properti objek dengan menggunakan . atau [].

javascriptCopy code

```
console.log(orang.nama); // Output: "John"
```

5. Perintah Lainnya:

Array: JavaScript memiliki banyak metode untuk bekerja dengan array, seperti push, pop, shift, dan lainnya.

- Event Handling: Anda dapat menangani peristiwa (event) di halaman web dengan JavaScript.
- Ajax: JavaScript dapat digunakan untuk mengambil dan mengirim data tanpa perlu merefresh halaman.