Soal Praktikum 4

1. Apa perbedaan antara selector ID dan class dalam CSS, dan kapan Anda sebaiknya menggunakan masing-masing?
2. Bagaimana cara mengubah warna teks pada sebuah elemen HTML menggunakan CSS?
3. Mengapa penting untuk memahami konsep dasar CSS dalam pengembangan web?
4. Jelaskan sintaksis dasar CSS untuk mendefinisikan gaya pada sebuah elemen.
5. Apa yang dimaksud dengan "cascading" dalam Cascading Style Sheets (CSS)?

Bagaimana konsep ini mempengaruhi cara gaya diterapkan pada sebuah halaman web?

1. Bagaimana Anda dapat mengatur tata letak elemen-elemen HTML menggunakan CSS? 7. Apa yang dimaksud dengan desain responsif, dan mengapa penting untuk memperhatikannya dalam pengembangan web?
2. Bagaimana cara mengaplikasikan gaya CSS pada elemen-elemen tertentu menggunakan selector ID dan class?
3. Apa yang dimaksud dengan pseudo-class dalam CSS? Berikan contoh penggunaannya.
4. Bagaimana Anda bisa mengevaluasi efektivitas desain web menggunakan CSS, dan apa langkah-langkah yang dapat Anda ambil untuk meningkatkannya?

-------------------------------------------------------------------------------------------------------

1. Buatlah tampilan web seperti di bawah ini dengan ketentuan: untuk bagian navigasi ketika kursor di arahkan akan berubah warna dan untuk layout harus sama persis untuk pewarnaan silahkan tentukan sendiri.
2. Silahkan isi konten dari setiap artikel, tidak mesti cuman 2, lebih dari 2 lebih baik, terkati sidebar silahkan link ke artikel.



**ANSWER :**

1. Sintaks

Selector ID : Diawali dengan tanda pagar (#)

Selector Class : Diawali dengan tanda titik (.).

Keunikan

Selector ID : Harus unik dalam satu dokumen HTML.

Selector Class : Dapat digunakan pada banyak elemen HTML.

Spesifisitas

Selector ID : Memiliki spesifisitas yang lebih tinggi daripada class.

Selector Class : Memiliki spesifisitas yang lebih rendah daripada ID.

Penggunaan

Selector ID : Ideal untuk elemen tunggal dan unik dalam halaman.

Selector Class : Ideal untuk mengelompokkan elemen dengan gaya yang sama.

JavaScript

SelectorID:Lebih mudah dan cepat diakses melalui JavaScript (document.getElementById()).

Selector Class : Membutuhkan metode seperti document.getElementsByClassName() atau querySelectorAll() yang mengembalikan koleksi elemen.

**Kapan Sebaiknya Menggunakan Selector ID:**

* **Elemen Struktural Utama:** Gunakan ID untuk elemen-elemen utama dalam tata letak halaman Anda, seperti #header, #navigation, #main-content, #sidebar, #footer. Ini membantu dalam memberikan struktur yang jelas pada CSS Anda.
* **Target JavaScript:** Jika Anda perlu menargetkan elemen tertentu secara spesifik menggunakan JavaScript (misalnya, untuk memanipulasi konten atau perilaku elemen tersebut), ID adalah pilihan yang baik karena kemudahannya dalam pemilihan elemen menggunakan document.getElementById().
* **Link Internal (Anchor Links):** ID sering digunakan sebagai target untuk link internal (anchor links) di dalam halaman. Contoh: <a href="#bagian-tertentu">Menuju Bagian Tertentu</a>.
* **Keunikan yang Mutlak Diperlukan:** Jika Anda memiliki elemen yang benar-benar unik dan tidak akan pernah ada duplikatnya di halaman tersebut, ID adalah pilihan yang tepat.

1. **Menggunakan Properti color**

Ini adalah cara yang paling dasar dan umum untuk mengubah warna teks. Anda dapat menerapkan properti color langsung pada elemen HTML melalui atribut style (inline CSS) atau melalui selector CSS (internal atau eksternal stylesheet).

1. CSS memungkinkan kamu membuat situs yang **responsif**, artinya tampil bagus di berbagai ukuran layar (PC, tablet, HP) menggunakan media queries.

@media (max-width: 768px) {

.container {

flex-direction: column;

}

}

1. Sintaksis dasar CSS (Cascading Style Sheets) digunakan untuk menentukan gaya tampilan elemen HTML. Struktur umumnya terdiri dari selector, property, dan value. Berikut adalah bentuk dasar sintaksis CSS:

Sintaks Dasar CSS:

css

selector {

property: value;

}

Selector: Menunjukkan elemen HTML yang ingin diberi gaya (style).

Property: Merupakan atribut atau aspek dari elemen yang ingin diubah (misalnya warna, ukuran huruf, margin, dll).

Value: Nilai dari property tersebut.

Contoh:

css

p {

color: blue;

font-size: 16px;

}

Artinya: Semua elemen <p> (paragraf) akan memiliki teks berwarna biru dan ukuran

font 16 piksel.

Tambahan:

Setiap pasangan property dan value diakhiri dengan titik koma ;.

Dapat menambahkan beberapa property dalam satu selector.

CSS bisa ditulis secara:

Inline: langsung di atribut HTML (style="")

Internal: di dalam tag <style> di file HTML

Eksternal: di file .css terpisah, lalu dihubungkan ke HTML dengan <link>.

1. **Cascading** berarti **berjenjang atau bertingkat**, dan ini mengacu pada cara CSS memutuskan **gaya mana yang akan diterapkan** ketika ada beberapa aturan yang berlaku untuk elemen yang sama.

CSS akan memilih aturan berdasarkan urutan prioritas berikut:

**Inline styles** (langsung di tag HTML) → style="color:red"

**Internal stylesheet** (di <style> dalam <head>)

**External stylesheet** (di file .css)

**Spesifisitas** (ID > Class > Tag)

**Urutan kemunculan** (yang ditulis terakhir akan menimpa yang sebelumnya jika setara)

1. .container {

display: flex;

justify-content: space-between;

}

**Positioning** – seperti relative, absolute, fixed, sticky

**Float & Clear** – cara lama, masih digunakan di beberapa kasus

**Margin dan Padding** – untuk memberi jarak antar elemen

1. **Desain responsif** adalah pendekatan untuk membuat situs web yang bisa menyesuaikan tampilannya di berbagai ukuran layar (PC, tablet, HP).
2. Html : <div id="utama">Halo Dunia</div>

Css : #utama {

color: blue;

}

Dengan class html : <p class="highlight">Ini penting</p>

<p class="highlight">Ini juga</p>

Dengan class css : .highlight {

background-color: yellow;

}

1. **Pseudo-class** adalah "kelas semu" yang memungkinkan kamu menargetkan **elemen dalam kondisi tertentu**, tanpa menambahkan class di HTML.

**Css:**

a:hover {

color: red;

}

Artinya: saat mouse diarahkan ke link (hover), warnanya jadi merah.

Contoh lainnya:

:first-child → Elemen pertama dalam parent

:last-child → Elemen terakhir

:focus → Saat elemen input aktif

:nth-child(2) → Anak kedua

1. Jawab :
2. **Responsif?** – Coba di berbagai perangkat & ukuran layar
3. **Aksesibilitas?** – Kontras warna cukup? Bisa dibaca screen reader?
4. **Konsistensi?** – Warna, font, dan layout konsisten?
5. **Kecepatan loading?** – CSS terlalu berat?
6. **Keterbacaan & UX?** – Teks mudah dibaca? Navigasi jelas?

**Cara meningkatkannya:**

* Gunakan sistem grid atau framework seperti Bootstrap/Tailwind
* Optimalkan penggunaan class, hindari duplikat gaya
* Kompres file CSS (.min.css)
* Gunakan **custom properties** (--warna-utama: #333;) untuk konsistensi
* Manfaatkan **tools audit desain** seperti Chrome DevTools, Lighthouse, atau CSSLint