Paduan Dasar Untuk HTML, CSS dan JS

**Hastho Rahtomo**

**12 Desember 2023**

**Contents**

[Pengenalan 2](#_Toc152445080)

[CSS 3](#_Toc152445081)

[**Display** 3](#_Toc152445082)

[**1) block:** 4](#_Toc152445083)

[**2) inline:** 4](#_Toc152445084)

[**3) inline-block:** 4](#_Toc152445085)

[**4) none:** 4](#_Toc152445086)

[**5) flex:** 5](#_Toc152445087)

[**6) grid:** 5](#_Toc152445088)

[**Position** 5](#_Toc152445089)

[**2) relative:** 5](#_Toc152445090)

[**3) absolute:** 6](#_Toc152445091)

[**4) fixed:** 6](#_Toc152445092)

[**5) sticky:** 6](#_Toc152445093)

# **Pengenalan**

Pada Paduan sebelumnya kamu telah belajar mengenai basic HTML dan CSS dan di paduan kali ini, kamu akan lebih berfokus pada CSS yang ditunjukan untuk “menghias” web-mu agar lebih menarik.

Kode yang diberikan pada paduan ini akan ditulis menggunakan eksternal CSS yang telah kamu pelajari pada paduan sebelumnya. Pastikan untuk mempersiapkan beberapa file ini:

***index.html***

***style.css***

# **CSS**

## **Display**

Display digunakan untuk mengatur tampilan sebuah elemen ketika file dijalankan. Dengan menggunakan display, kamu dapat mengatur elemen tersebut mulai dari posisi elemen terhadap elemen lain, posisi elemen ketika web-mu di scroll, bahkan menentukan apakah elemen tersebut akan terlihat user atau tidak.

Berikut masing-masing display yang dapat kamu gunakan:

### **1) block:**

Elemen ini akan memulai di baris baru dan memenuhi lebar penuh dari container yang mengandungnya. Setiap elemen block akan menduduki baris baru, sehingga elemen-elemen berikutnya akan muncul di bawahnya.

Contoh:

div {

    display: block;

  }

**2) inline:**

Elemen ini akan tampil dalam konteks baris yang sama dengan elemen-elemen sebelumnya dan sesudahnya. Ini berarti elemen tersebut hanya akan memakan lebar sejauh yang diperlukan dan tidak akan memaksa baris baru.

Contoh:

span {

    display: inline;

  }

### **3) inline-block:**

Mirip dengan inline, tetapi elemen ini masih memiliki sifat blok, sehingga Anda dapat mengatur lebar dan tinggi untuk elemen tersebut.

Contoh:

img {

    display: inline-block;

  }

### **4) none:**

Elemen tersebut tidak akan ditampilkan di halaman web. Ini berguna untuk menyembunyikan elemen-elemen secara dinamis melalui JavaScript atau CSS.

Contoh:

.hidden {

    display: none;

  }

### **5) flex:**

Elemen ini akan menjadi container fleksibel yang memungkinkan pengaturan tata letak fleksibel menggunakan properti lain seperti justify-content dan align-items.

Contoh:

.flex-container {

    display: flex;

  }

### **6) grid:**

Elemen ini akan menjadi container grid yang memungkinkan pengaturan tata letak menggunakan properti grid CSS.

Contoh:

.grid-container {

    display: grid;

  }

## **Position**

Sesuai namanya, position digunakan untuk mengatur posisi dari sebuah elemen. Kamu dapat mengatur agar posisi dari sebuah elemen dapat berubah/bergerak mau pun tetap pada ukurannya apapun yang terjadi.

Berikut masing-masing position yang dapat kamu gunakan:

**1) static:**

Ini adalah nilai default. Elemen dengan posisi statis ditempatkan dalam alur dokumen biasa. Artinya, elemen tersebut akan tampil sesuai dengan urutan alami di halaman.

Contoh:

.element {

    position: static;

  }

**2) relative:**

Elemen dengan posisi relatif akan diposisikan relatif terhadap posisinya sendiri yang normal. Dengan kata lain, Anda dapat menggunakan properti top, right, bottom, dan left untuk memindahkan elemen tersebut dari posisi normalnya.

Contoh:

.element {

    position: relative;

    top: 10px;

    left: 20px;

  }

### **3) absolute:**

Elemen dengan posisi absolut dihapus dari alur dokumen normal dan ditempatkan relatif terhadap elemen terdekat yang memiliki posisi bukan statis. Jika tidak ada elemen tersebut, elemen absolut akan ditempatkan relatif terhadap elemen <html>.

.element {

    position: absolute;

    top: 50px;

    left: 100px;

  }

### **4) fixed:**

Elemen dengan posisi tetap ditempatkan relatif terhadap viewport (area tampilan pengguna) dan tidak akan bergerak ketika halaman digulir. Ini cocok untuk elemen yang ingin tetap terlihat pada layar, misalnya, menu navigasi.

.element {

    position: fixed;

    top: 0;

    right: 0;

  }

### **5) sticky:**

Elemen dengan posisi "sticky" tetap relatif terhadap viewport sampai elemen mencapai posisi tertentu saat digulir. Setelah mencapai batas tersebut, elemen menjadi "fixed" dan tetap pada posisi itu.

.element {

    position: sticky;

    top: 20px;

  }