CSS

# Pseudo Class & Element

Pseudo Class adalah selector CSS yang digunakan untuk mengakses bagian tertentu dalam HTML yang tidak ‘terlihat’ (tidak tertulis di dalam HTML) atau bagian dari HTML yang tidak bisa diakses dengan selector sederhana lain.

**:** = biasanya digunakan dalam menerapkan gaya pada css.

**:link** = Berfungsi untuk menambahkan style pada link yang belum pernah di kunjungi atau klik.

**:hover** = Berfungsi untuk menambahkan style pada elemen pada saat mouse berada di atasnya.

**:active** = Berfungsi untuk menambahkan style pada elemen yang sedang aktif.

**:visited** = Berfungsi untuk menambahkan style pada link yang sudah pernah di kunjungi.

**:first-child** = Memilih element pertama dari beberapa element yang sama.

**:last-child** = Memilih elemen terakhir dari beberapa elemnt yang sama.

**:nth-child(n)** = Memilih elemen berdasarkan ekspresi aljabar sederhana (contoh "2n", atau "4n-h1").

**:root** = secara khusus merujuk pada elemen root, yang pada umumnya adalah elemen <html>

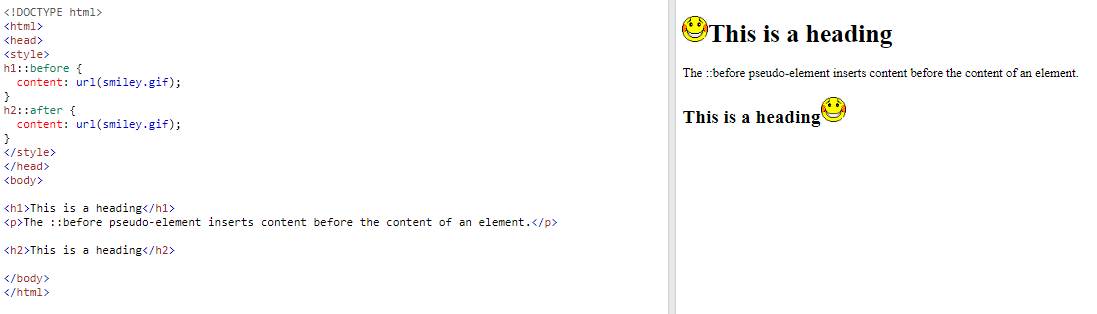
Variabel Global (:root): Jika Anda juga berencana untuk mendefinisikan variabel global (custom properties) yang akan digunakan di seluruh stylesheet, maka :root adalah pilihan yang lebih umum dan sering digunakan.

*\*jadi jika style di :root itu bisa digunakan di file html yang berbeda.*

Pseudo Element merupakan sebuah selector yang berfungsi untuk menyeleksi elemen atau tag secara otomatis, tapi bisa juga di artikan sebagai selector yang menciptakan elemen atau tag palsu tanpa menuliskan kode HTML.

**::** = biasanya digunakan dalam memanipulasi atau memilih bagian khusus dari element yang tidak dapat diwakili oleh element aslinya.

**::before & :: after** = Psudo elemnt yang memungkinkan kita untuk menambahkan konten baru (sebelum/ sesudah element) pada halaman HTML hanya melalui CSS saja, element ini tidak benar-benar ada dalam struktur html, namun tetap akan digenerate oleh browser sehingga akan tetap tampak.



**::first-letter** = Psudo element yang memungkinkan kita untuk merubah (menambahkan style) huruf pertama pada sebuah text.

**::first-line** = Psudo element yang memungkinkan kita untuk merubah (menambahkan style) baris pertama pada sebuah text.

*pseudo class source :* [*https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes*](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes)

# CSS Siblings

+ = Akan memilih element setelahnya

Misalkan : div + p { … } maka akan memilih ***satu element p setelah div*** saja.



~ = Akan memilih element setelahnya.

Misalkan : div ~ p { … } maka akan memilih ***semua element p setelah div*** saja.



# Spesificity

Specificity css adalah besar nilai atau bobot dari suatu element dalam css.

|  |  |
| --- | --- |
| Inline = 1000 | #id = 100 |
| .class = 10 | Element = 1 |

# Text Improvements / Font Face

***@font-face*** adalah aturan dalam CSS yang memungkinkan pengguna untuk menyematkan font ke dalam dokumen web dan menggunakan font tersebut untuk mengatur tampilan teks pada halaman tersebut. (Format font untuk web *woff*).

Ekstensi file pada font-face (.eot , .ttf / .otf , .svg , .woff).

Contoh : *@font-face {*

*font-family: “amble”;*

*src: url(amble/Amble.woff);*

*}*

*body { font-family: “amble”; }*

# Development Techniques / Vendor Prefix

Vendor-prefix, digunakan agar code css yang dibuat dapat support di semua web browser.

Contoh : *-webkit-<property-css3> (Chrome, Opera New, Safari)*

*-moz-<property-css3> (Firefox)*

*-ms-<property-css3> (Internet Explorer)*

*-o-<property-css3> (Opera old)*

# Layouting

## Display

None = Element tidak ditampilkan pada halaman.

Block = Element ditampilkan sebagai blok, mengambil seluruh lebar yang tersedia.

(Posisi Element berikutnya akan berada di bawahnya dan dapat memberikan jarak padding & margin ke seluruh areanya).

Inline = Element ditampilkan sejajar dengan text dan element lain, tanpa mengambil seluruh lebar.

(Posisi Element berikutnya akan berada di sampingnya / inline dan hanya memberikan jarak padding & margin ke sumbu x atau horizontalnya saja).

Inline-block = Element ditampilkan sejajar dengan element lain, namun dapat memiliki lebar dan tinggi sendiri.

(Posisi Element berikutnya akan berada di sampingnya / inline namun berupa blok sehingga dapat memberikan jarak padding & margin di seluruh areanya).

Flex = Element menjadi *container flex*, memungkinkan layout lebih kompleks dan responsive.

Grid = Element menjadi *container grid*, memungkinkan layout grid yang kompleks dan responsive.

Table = Element ditampilkan sebagai tabel.

## Dimension

**Width & height,** mengatur lebar dan tinggi pada elemen HTML dengan properti width dan height, width yang berarti lebar dan height yang berarti tinggi.

Untuk pixel itu absolute karena dianggap sebagai nilai yang tetap.

Sedangkan % dianggap relative karena menyesuaikan dengan ukuran device atau parentnya.

Nilai Property :

**Auto =** menghitung ukuran width dan height secara otomatis.

**Length =** Mengatur dengan menggunakan px, in, cm, mm, pt, pc.

**% =** Mengatur width dan height berdasarkan dengan persen.

**Initial =** Mengatur width dan height ke nilai default.

**Inherit =** Mengatur width dan height ke nilai warisan dari element induknya.

## Overflow

Overflow, alat untuk mengelola bagaimana konten yang melampaui batas elemen akan ditampilkan.

Visible = Akan menampilkan konten di dalam element meskipun telah melebihi batasnya.

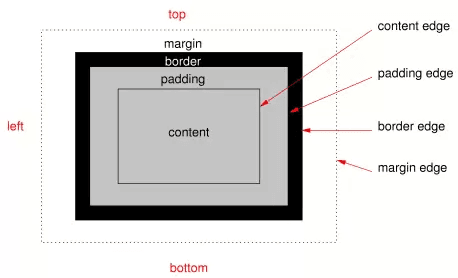
Hidden = Akan menyembunyikan konten di dalam element jika telah melebihi batasnya.

Scroll = Akan menampilkan scrolling atas bawah dan kanan kiri untuk kontennya jika melebihi batasnya.

Auto = Akan menampilkan scrolling atas bawah jika kontennya melebihi batasnya.

## Box Model

CSS Box Model adalah sebuah konsep dimana setiap element yang terdapat pada halaman web diproses sebagai kotak (box) yang di dalamnya terdiri dari 4 lapisan yakni content, padding, border, margin.



Margin = Margin dalam CSS adalah ruang di sekitar elemen yang digunakan untuk menentukan jarak antara elemen tersebut dan elemen lainnya.

Padding = Padding dalam CSS adalah ruang tambahan yang diberikan di sekitar isi elemen. Ini adalah jarak antara isi elemen dan tepi elemen itu sendiri.

Border = Merupakan garis tepi pembatas element. (dotted, solid, dash, double)

## Float

Float digunakan untuk mengatur layout dengan membuat sebuah elemen berada di kiri atau kanan sebuah Container.

**Left =** Elemen akan mengambang ke sisi kiri container-nya. Bisa juga menggunakan value inline-start.

**Right =** Elemen akan mengambang ke sisi kanan container-nya. Bisa juga menggunakan value inline-end.

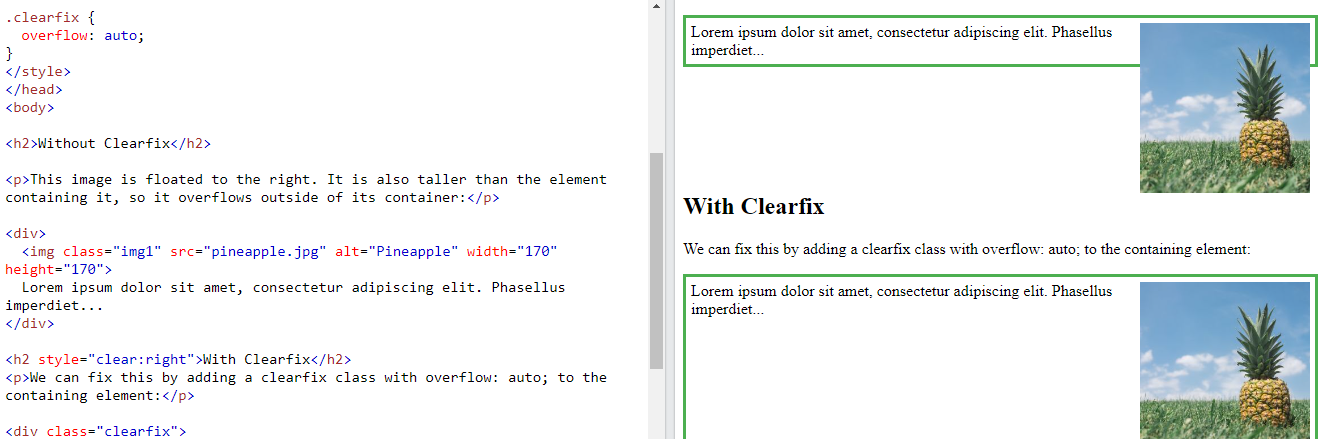
**None =** Elemen tidak akan mengambang. Ini adalah nilai default.

**Inherit =** Elemen akan mewarisi nilai properti float dari elemen induk.

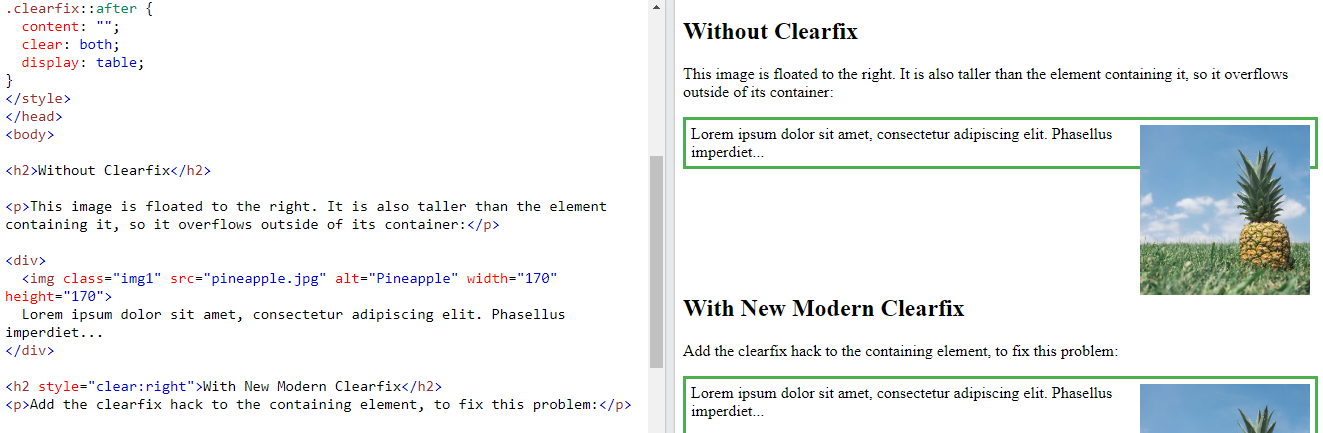
**Clear Float**

Membersihkan float adalah langkah-langkah yang diambil untuk memastikan bahwa elemen berikutnya di dalam layout tidak terpengaruh oleh elemen yang mendahuluinya yang di-float-kan. Float digunakan untuk menyusun elemen di sekitar elemen float, tetapi seringkali, float dapat menyebabkan elemen berikutnya naik dan menyatu dengan elemen yang di-float-kan.

Menggunakan overflow: auto



Menggunakan clearfix::after { … }



Atau bisa menggunakan Mikro Clearfix

## Css Reset

CSS reset adalah teknik yang digunakan dalam pengembangan web untuk menyamakan nilai default properti CSS di berbagai browser.

Cara 1 : *\* { padding: 0; margin: 0; }*

*\*body, h1, p { margin: 0; padding: 0; }*

Cara 2 : Css Reset yang dibuat oleh Eric Meyer.

## Position

Position dalam CSS adalah properti yang digunakan untuk mengontrol posisi elemen pada halaman web.

Static = Posisi element secara default. Properti CSS seperti z-index, left, top, right, bottom, tidak akan berpengaruh nilainya terhadap element yang posisinya static.

Relative = Posisi element relative sama dengan posisi static, hanya saja posisi relative elementnya dapat diubah ubah atau diatur posisinya seperti dengan menggunakan z-index, left, top, right, bottom.

*\*ibaratnya di posisi yang sama namun naik 1 dimensi*

Absolute = Posisi element absolute sama dengan posisi relative, namun jika relative posisinya akan seperti static tapi yang diberikan relative nya akan naik 1 dimensi, sedangkan absolute akan naik 1 dimensi tapi akan menempati posisi element sebelumnya, absolute juga dapat kita atur posisinya.

Fixed = Posisi Fixed membuat element mengambang diantara element lainnya, dan juga bisa diatur posisinya.

Sticky = Elemen akan selalu tetap pada posisi yang telah ditentukan saat melewati viewport, seperti elemen yang di-relatif-kan sampai batas tertentu, dan kemudian diubah menjadi elemen yang di-fixed-kan setelah itu.

## Z-Index

Properti z-index ini berguna untuk mengatur elemen mana yang ada di bagian depan dan belakang, biasa digunakan ketika ada elemen yang saling tumpang tindih. Namun untuk bisa menggunakan ini, perlu menggunakan properti position selain static .

# Display Improvements

## Border Radius

Border Radius merupakan properti untuk membuat suatu element memiliki tepian yang elips atau umumnya disebut sudut lengkungan.

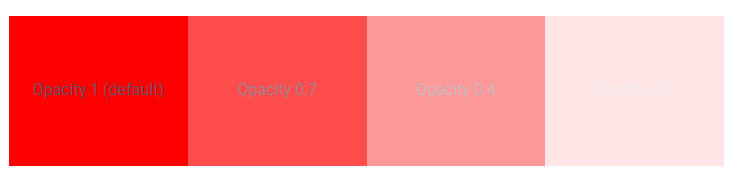
border-radius: 25px;

## Opacity

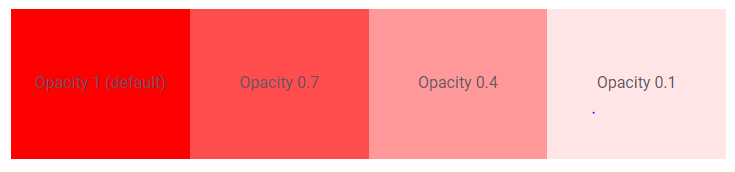
Opacity memiliki nilai default 1 (satu) dan opacity tidak diwarisi, tetapi karena induk memiliki opacity yang diterapkan sehingga semua yang ada di dalamnya ikut transparan. Bisa juga menerapkan opacity pada elemen induk agar tidak berpengaruh pada elemen didalamnya dengan menggunakan nilai color RGBA.

Opacity: 0.5;

Contoh Opacity



Contoh Opacity dengan menggunakan RGBA.

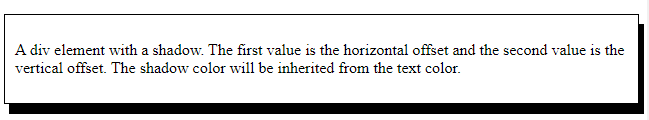


## Box Shadow

Dengan properti box-shadow bisa menggunakannya untuk membuat efek bayangan pada box seperti elemen <div> atau elemen yang lainnya. Dalam box-shadow juga bisa menentukan arah horizontal, vertikal, blur, spread dan warna.

\*Inset, x-offset, y-offset, blur, \*spread, color

#example1 { border: 1px solid; padding: 10px; box-shadow: 5px 10px black; }



Box-shadow akan menambahkan bayangan pada element dan element element lain yang diberikan box shadow di dalamnya.

## Text Shadow

Dengan properti text-shadow Anda bisa menggunakannya untuk membuat efek bayangan pada text atau tulisan. Dalam text-shadow Anda bisa menentukan arah horizontal, vertikal, blur dan warna.

x-offset, y-offset, blur, color

text-shadow: 2px 2px red;



## Linear Gradient

Dengan property linear-gradient kita bisa membuat gradasi warna secara linear / lurus. Kita bisa menggunakan property linear-gradient sebagai value dari property background.

Sudut(deg), arah, warna-awal, warna-akhir.

\*color-stop

background: linear-gradient(to bottom, red 20%, yellow 50%, purple 100%);

## Radial Gradient

Dengan property radial-gradient kita bisa membuat gradasi warna secara radial / putaran / lingkaran. Kita bisa menggunakan property linear-gradient sebagai value dari property background.

Eclips (default) / circle,

Jangkauan

closest-corner = jarak dari titik pusat gradient ke \*sudut terdekat\* element.

closest-side = jarak dari titik pusat gradient ke \*sisi terdekat\* element.

farthest-corner = jarak dari titik pusat gradient ke \*sudut terjauh\* gradient.

farthest-side = jarak dari titik pusat gradient ke \*sisi terjauh\* gradient.

\*sebenarnya tidak terbatas anda bisa menambahkan jarak secara eksplisit seperti circle at center

Posisi y (top, center, bottom, px)

Posisi x (left, center, right, px)

Warna awal

Warna akhir

**Contoh :**

*background: radial-gradient(circle at 30% 70%, /\* Bentuk lingkaran, posisi pusat x: 30%, y: 70% \*/*

*#ff0000, /\* Warna awal, misalnya merah \*/*

*#0000ff /\* Warna akhir, misalnya biru \*/*

*);*

## Filter

Filter adalah salah satu keunggulan CSS yang dapat digunakan penulis untuk memberi berbagai efek visual (seperti filter Photoshop tetapi untuk browser).

Blur() = Menerapkan blur ke gambar. Blur pada filter hanya bisa menggunakan px.

Ringan ( 0 – 2 px ) Moderat ( 3 – 5 px )

Signifikan ( 6 - 10 px ) Kuat ( > 10 px )

Brigthness() = Digunakan untuk mengatur kecerahan element. Nilainya dari 0 - > 1(default).

Contrast() = Digunakan untuk mengatur tingkat kontras element. Nilainya dari 0 - > 1 (default).

*\*perbedaan contrass dan brightness adalah jika contrass lebih ke mempengaruhi tingkat perbedaan antara warna sedangkan brightness lebih mempengaruhi ke tingkat kecerahan atau kegelapan.*

Drop-shadow() = Menerapkan efek drop shadow ke gambar input. Atau memberikan bayangan mirip seperti dengan box shadow.

Color, length x, length y, blur

***filter: drop-shadow(red 1rem 1rem 10px);***

*\*drop-shadow akan memberikan shadow yang mempengaruhi element itu sendiri tidak mempengaruhi element lainnya.*

Grayscale() = Digunakan untuk mengkonversi element ke skala abu abu. Nilainya dari 0 – 1.

Hue-rotate() = Digunakan untuk memutar roda warna pada element. Nilainya dapat menggunakan deg.

***filter: hue-rotate(45deg); /\* Memutar roda warna sebanyak 45 derajat \*/***

Invert() = Digunakan untuk ***membalikan warna element***, sehingga element yang semula berwarna dapat menjadi hitam jika diubah sepenuhnya menggunakan invert. Nilai pada invert bisa menggunakan 0 – 1 atau %.

***filter: invert(.6);***

Opacity() = sama seperti opacity biasa, digunakan untuk mengurangi opacity. Nilai yang bisa diberikan seperti 0 – 1 dan %.

***filter: opacity(.5);***

Saturate() = Digunakan untuk mengatur intensitas saturasi warna element. Nilai yang bisa diberikan seperti 0 - > 1 dan %.

***filter: saturate(4);***

Sepia() = Digunakan untuk mengubah warna menjadi tampilan sepia yang identik memberikan kesan tua dan hangat. Nilai yang bisa diberikan seperti 0 – 1 dan %.

***filter: sepia(.8);***

Backdrop-filter adalah properti CSS yang digunakan untuk memberikan efek filter pada elemen dan elemen-elemen di belakangnya (elemen di lapisan bawah).

**Contoh : *backdrop-filter: blur(5px);***



## Transform

CSS Transform atau transformasi memungkinkan kita untuk bisa membuat efek translate (memindahkan), rotate (memutarkan), scale (memperbesar / kecil), dan skew (memiringkan) pada element.

Translate (Memindahkan)

translateX, translateY, translate(X,Y)

**Contoh : transform: translate(-50px, 100px);**

Rotate (Memutarkan)

rotate(angle)

**Contoh : transform: rotate(30deg);**

Scale (Memperbesar / memperkecil)

scaleX, scaleY, scale(X, Y)

**Contoh: transform: scale(2.2, 1.2);**

Skew (Memiringkan)

skewX(angle), skewY(angle), skew(X, Y)

**Contoh: transform: skew(30deg, 20deg);**

Matriks (Menggabungkan semua transform)

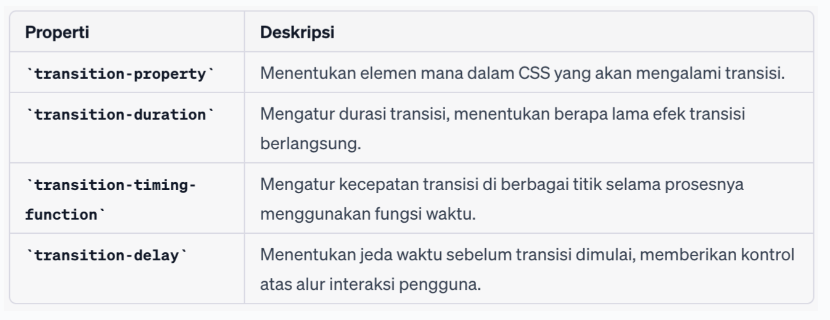
matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

**Contoh : transform: matrix(1, -0.3, 0, 1, 0, 0);**

# Animation

## Transition

CSS Transition atau Transisi adalah Efek Transis yang memungkinkan sebuah elemen secara bertahap berubah dari satu gaya ke yang lain. Hanya ada keadaan awal dan akhir saja.



Function ease, ease-in, ease-out, ease-in-out, linear, cubic-bezier(w,x,y,x)

*div {*

*transition: [transition-property] [transition-duration]*

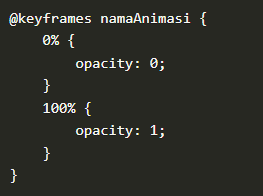
*[transition-timing-function] [transition-delay];*

*}*

*Contoh : transition: margin-left 3s ease-in;*

## Animation

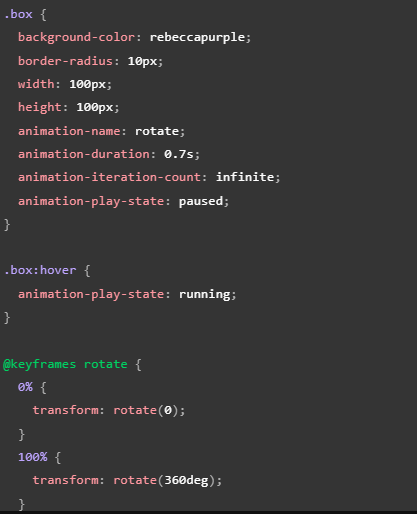
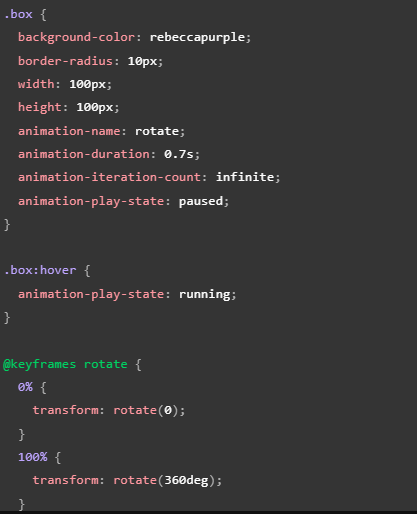
Memberikan animasi pada styling, pergerakan animasi bisa melebihi dari 2 keadaan dengan menggunakan @keyframe. Syarat dari animasi kita harus mendefinisikan keyframenya terlebih dahulu.



Property yang digunakan pada animation yaitu

|  |  |
| --- | --- |
| Property | Definisi |
| Animation-name | Menentukan nama dari animasi yang telah didefinisikan menggunakan @keyframe . |
| Animation-duration | Menentukan durasi dalam satuan **s** atau **ms**. |
| Animation-timing-function | Menentukan cara element berubah seperti **ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(w,x,y,x)**. |
| Animation-delay | Menentukan waktu tunggu sebelum animasi dimulai. |
| Animation-iteration-count | Menentukan berapa kali animasi akan diulang, nilai yang dapat juga diberikan seperti **infinite**. |
| Animation-direction | Menentukan arah animasi, nilai yang dapat diberikan seperti **normal, reverse,** atau **alternate** (Bulak Balik) |
| Animation-fill-mode | Menentukan keadaan akhir element setelah animasi selesai, ada **none, forwards**(di akhir posisi)**, backward**(kembali ke awal)**, both** dll. |
| Animation-play-state | Menentukan apakah animasi sedang berjalan atau dijeda. **Paused / running** |

Contoh penggunaan animation :



# CSS Unit

Dalam CSS, "value" merujuk pada nilai yang diberikan kepada properti CSS, sedangkan "unit" merujuk pada satuan yang diterapkan pada nilai tersebut. Misalnya, jika Anda memberikan nilai lebar elemen dengan width: 200px, "200" adalah nilai dan "px" adalah satuan.

Terdapat beberapa unit satuan yang dapat kita berikan seperti :

Integer = Bilangan Bulat atau bilangan yang tanpa titik atau desimal seperti 1, -5, 100

Number = Bilangan Pecahan, bilangan yang mencakup seluruh jenis angka, 1, -5, 3.14, -19.02

Percentage = Bilangan Bulat dengan diakhiri % (relative).

Dimension = merujuk pada aspek ukuran atau dimensi dari elemen HTML, seperti lebar, tinggi, atau kedalaman.

|  |  |  |
| --- | --- | --- |
| Length | Absolute Length | **Px, pt, pc, cm, mm, In** |
| Relative Length | percentage (**%**), relative terhadap ukuran parent nya |
| **em** relative terhadap ukuran font parent (16px)  *\*jika em akan bertambah dan akan mengalami*  *penumpukan.* |
| **rem** relative terhadap ukuran font ‘root’ (16px) |
| **ch** relative terhadap karakter ‘0’ dari font digunakan |
| **vh** (%) relative terhadap tinggi viewport.  Contoh: Jika tinggi viewport adalah 600 piksel, maka 50vh akan setara dengan 300 piksel. |
| **vw** (%) relative terhadap lebar viewport  Contoh: Jika lebar viewport adalah 600 piksel, maka 50vw akan setara dengan 300 piksel. |
| Elemen **vh** sama dengan 1/100 dari tinggi viewport. Contohnya, jika tinggi perambah 900px, 1vh akan menjadi 9px. Demikian pula, jika lebar viewport 750px, 1vw akan menjadi 7.5px. |
| **Vmin** (%), relatif terhadap dimensi terkecil viewport |
| **Vmax** (%), relatif terhadap dimensi terbesar viewport |
| Jika tinggi viewport adalah 600 piksel dan lebar viewport adalah 800 piksel, maka: 10vmin = 60px 10vmax = 80px  \*vmin akan mengambil ukuran viewport yang lebih kecilnya  misalkan di hp vmin itu lebar nya jadi ukurannya akan menyesuaikan dengan lebar ukuran handphone  \*vmax akan mengambil ukuran viewport yang lebih besarnya  misalkan di hp vmax itu tinggi nya jadi ukuran lebarnya akan menyesuaikan dengann tinggi dari handphonenya  \*karena jika viewport hp itu biasnya tinggi layarnya (vmax) dan lebarnya pendek (vmin) |
| Angle | **deg** degree / derajat |
| **grad** gradiant / 1 putaran penuh = 400 grad |
| **rad** radiant / berhubungan dengan busur lingkaran |
| **turn** putaran penuh |
| Time | **ms** (milisecond) |
| **s** (second) |
| Resolution, biasanya digunakan saat mencetak sesuatu.  \*penggunaannya juga biasanya digunakan dalam media query.  *@media print and (min-resolution: 300dpi) { ... }* | **dpi** (dot per inch) |
| **dpcm** (dot per centimeter) |
| **dppx** (dot per pixel) |

*\*viewport = merupakan lebar atau ukuran dari layar yang digunakan*

# Function CSS

Dalam konteks CSS (Cascading Style Sheets), "function" sering kali merujuk pada berbagai fungsi atau metode yang dapat digunakan untuk memanipulasi nilai atau melakukan operasi tertentu dalam properti CSS.

Di dalam css terdapat function seperti **calc(), rgb(), rgba(), hsl(), hsla(), url(), linear-gradient(), radial-gradient(), var(), translate().**

Calc() = sebuah function pada css yang memungkinkan kita untuk melakukan operasi matematika pada nilai dari sebuah properti.

\*Nilai yang bisa diberikan calc()

Length (px, pt, cm, …), angle, time, percentage, angka (opacity & z-index).

Contoh : *width: calc(20px + 30px);*

*Width: calc(calc(100%/3) – calc(1rem \* 2));*

Var() = Variabel dalam CSS diperkenalkan melalui fitur yang disebut **"Custom Properties" / :root** atau lebih umum dikenal sebagai CSS Variables. Variabel CSS memungkinkan Anda menyimpan nilai-nilai yang dapat digunakan ulang dalam seluruh stylesheet, memberikan cara yang lebih terorganisir dan efisien untuk mengelola gaya.

Cara membuat variabel di css cukup dengan **--nama-variabel: value;**

Contoh:

*:root { --warna-utama: blue; --ukuran-teks: 16px; }*

*.header { color: var(--warna-utama); font-size: var(--ukuran-teks); }*

# Flexbox