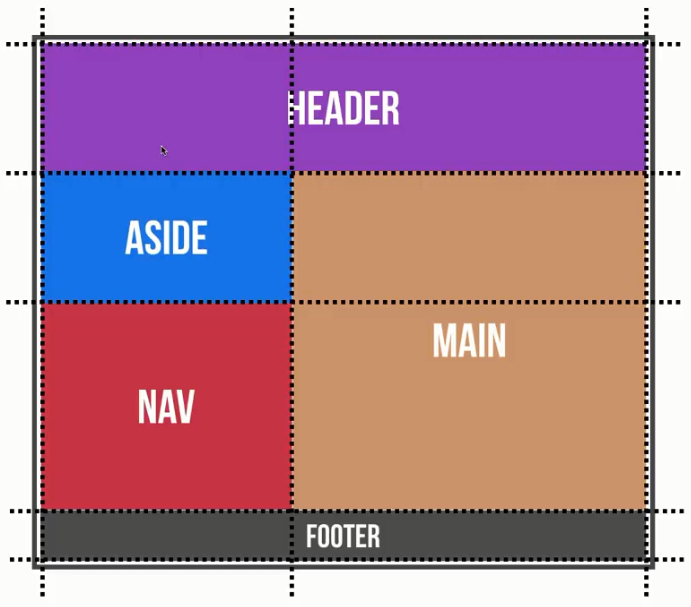
**CSS GRID SYSTEM**

**APA DAN KENAPA?**

CSS Grid adalah modul CSS baru untuk mendefinisikan sistem layout berbentuk grid dalam 2 dimensi (baris & kolom). Karena CSS Grid isa 2 dimensi (baris sekaligus kolom) maka lebih unggul dari CSS Flex Box, karena flex box Cuma bisa 1 dimensi (baris saja atau kolom saja).

CSS Grid bukannya pengganti Flex Box, nanti bisa kita gabungkan penggunaannya. CSS Grid System membuat HTML lebih sederhana, untuk membuat web responsive akan menjadi lebih mudah (tinggal tukar menukar posisi grid). Contoh berikut untuk cssgridunpas1.html, cssgridunpas1.css



**TERMINOLOGI / ISTILAH**

Disarankan install Mozilla Firefox karena ia memiliki grid inspector. Grid inspector ada di saat melakukan inspect di halaman browser, akan memperlihatkan grid secara visual (lebih mudah dan enak dilihat).

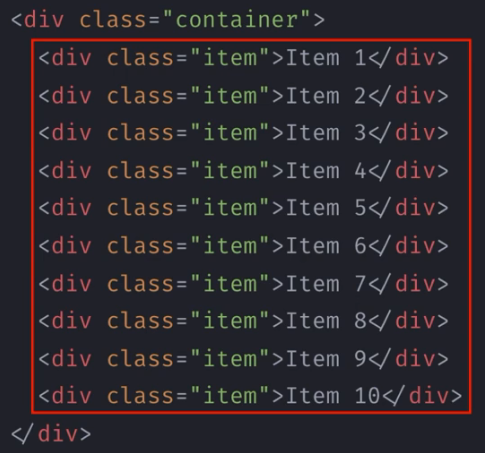
**Grid Container**

Adalah element pembungkus grid, didefinisikan dengan menuliskan = display: grid;



**Grid Item**

Merupakan element yang berada persis 1 level di dalam grid container



**Grid Line**

Adalah garis horizontal (kolom) atau vertikal (baris) yang memisahkan grid menjadi beberapa bagian dan ditandai dengan angka. Grid line akan punya nomer, jumlahnya pasti lebih 1 dari baris maupun kolomnya. Positif itu dari kiri ke kanan dan dari atas ke bawah, negatif itu dari kanan ke kiri dan dari bawah ke atas.



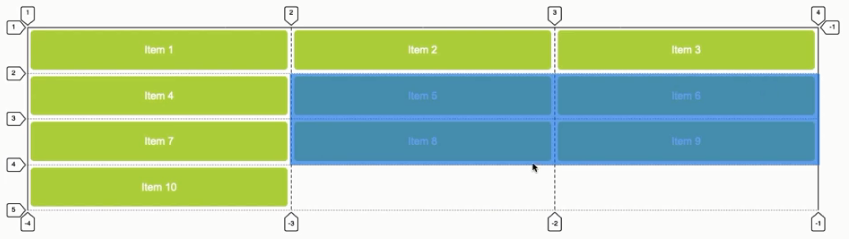
**Grid Cell**

Adalah perpotongan atau pertemuan antara baris dan kolom di dalam grid. Misal contoh 1 cell :



**Grid Area**

Adalah kumpulan lebih dari satu cell yang membentuk kotak, jadi cell gaboleh acak. Gaboleh longkap satu, bentuknya harus bergabung.





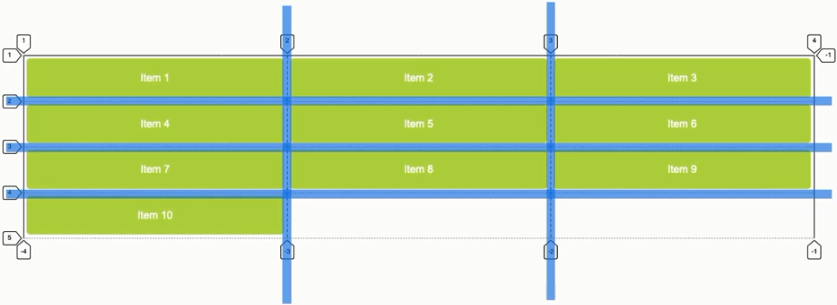
**Grid Track**

Adalah ukuran atau jarak antara 2 grid line, bisa horizontal (kolom) atau vertikal (baris), bisa juga disebut sebagai ukuran cell. Bisa atur ukuran tinggi tiap baris atau ukuran lebar tiap kolom.

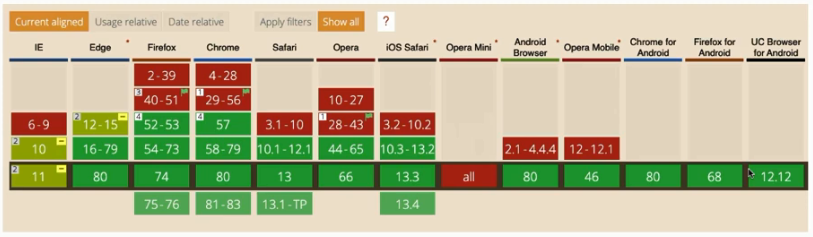


**Grid Gab**

Merupakan jarak antara grid track atau jarak antara cell. Erat kaitannya dengan cell spacing.



**BROWSER SUPPORT (UNTUK CSS GRID)**



Hijau artinya sudah disupport secara penuh, kuning artinya belum penuh, merah itu belu support.

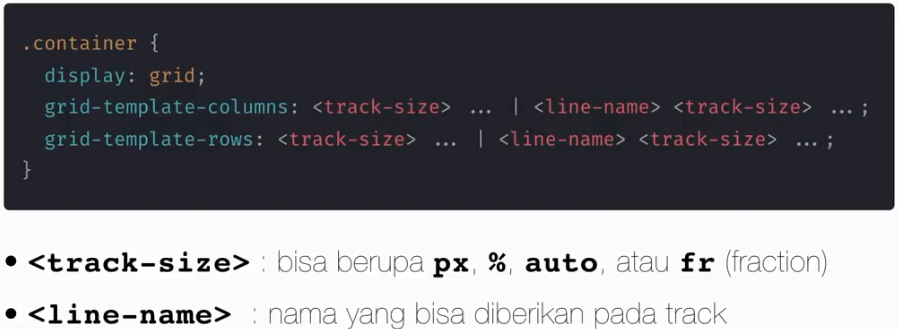
**CSS GRID PROPERTIES**



**PROPERTY MENGATUR KOLOM DAN BARIS (CONTAINER)**

Property grid-template-columns dan grid-template-rows

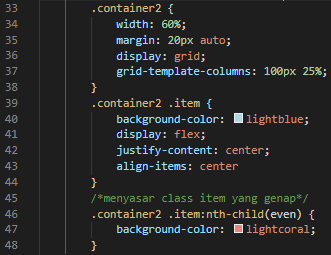
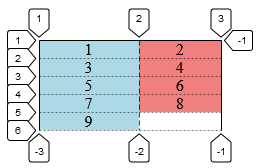
Mendefinisikan kolom / baris dengan cara menuliskan nilai dipisahkan oleh spasi. Nilai merepsentasikan ukuran grid track dan spasi merepresentasikan grid line.

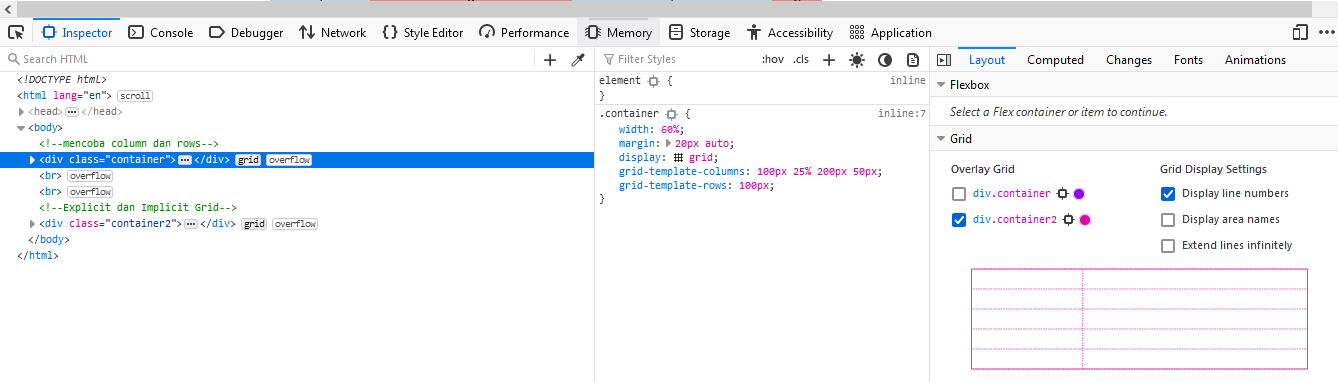




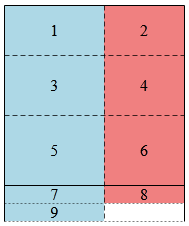
**Explicit Dan Implicit Grid**

Explicit grid adalah ketika kita secara jelas menuliskan ukuran dari grid (kolom dan baris). Implicit grid adalah ketika grid menentukan secara otomatis sisa-nya dari kolom dan baris yang tidak ditentukan secara explicit. Misal isi item ada 9 tapi cuma dikasih grid 2 buah untuk column :



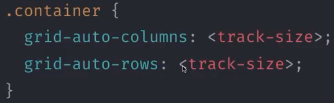
Cobalah lakukan inspect di Mozilla Firefox seperti di atas. Garis solid menandakan explicit grid dan garis putus - putus menandakan explicit items / explicit tracks, sementara titik - titik adalah implicit items / implicit tracks. Misal item ada 9 dan dibikin grid 2 untuk column dan 3 untuk baris, no 7, 8, 9 termasuk implicit.

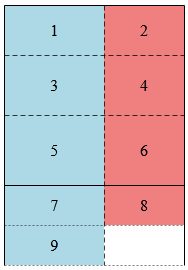
Manfaat dari keberadaan explicit dan implicit grid karena konten nantinya akan dinamis dan jumlahnya bisa jadi tidak bisa ditentukan. Untuk atur ukuran items yang implicit dapat ditangani dengan property berikut.

Property grid-auto-columns dan grid-auto-rows

Untuk mengatur grid track yang tidak dituliskan pada grid-template (grid track)



Misal item ada 9 dan dibikin grid 2 untuk column dan 3 untuk baris, no 7, 8, 9 termasuk implicit dan ditambahkan pengaturan untuk implicit items-nya.



**Auto Flow**

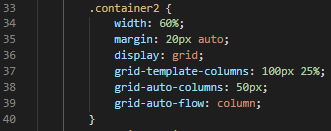
Biasanya item yang ga cukup di kolom default-nya akan membuat baris barus (baris yang implicit), gimana kalo supaya kolom yang implicit?

Property grid-auto-flow

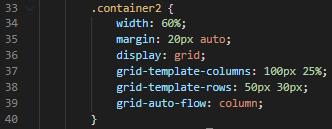
Untuk mengatur penempatan item / cell pada grid track, termasuk yang tertulis secara implicit. Bisa memilih untuk yang bablas akan menjadi baris atau column, defaultnya row (baris).



Misal contoh dibawah untuk nyoba implicit items ditaruh menjadi kolom :





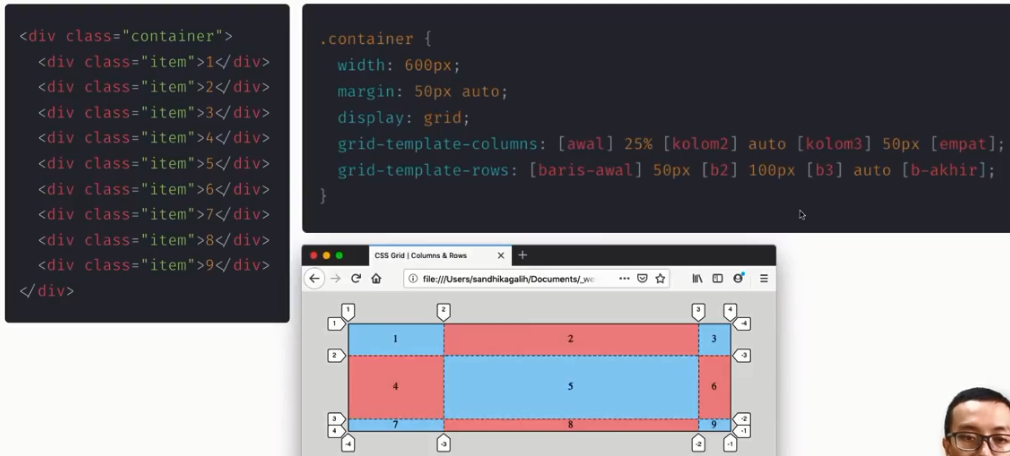




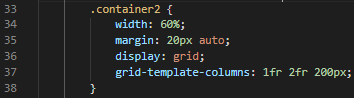
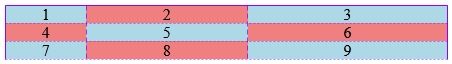
**Memberi Nama Dan Fraction**

Nomor baris akan bisa dipanggil sesuai dengan nama untuk garis yang diberikan. Untuk nilai fraction (fr) akan responsive juga kalo misal container (pembungkusnya) responsive. Proses fraction adalah dijumlahkan dulu semua nilai fraction barulah lebar total akan dibagi dan dikalkulasi. Dalam fraction kalo ada nilai solid, makan lebar total akan dikurangi nilai solid terlebih dahulu.

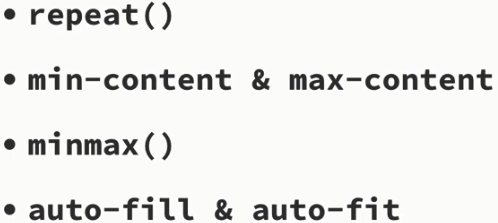
contoh kasih nama grid-line :



contoh fraction :

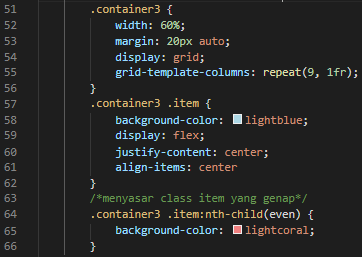
 

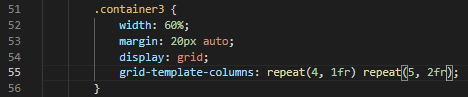
**SPECIAL FUNCTION & KEYWORDS**



Value Special Function repeat()

Untuk menentukan ukuran grid track secara berulang.



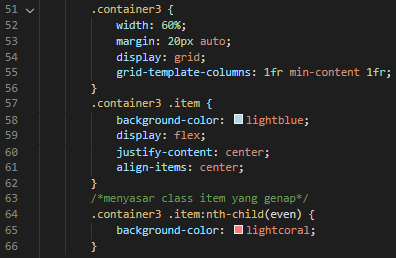
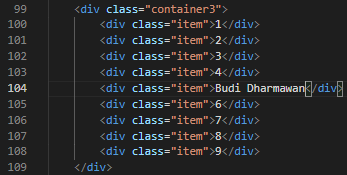


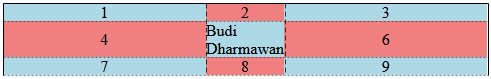
Value min-content & max-content

Menentukan seberapa besar ukuran grid track (horizontal maupun vertikal) berdasarkan content dari sebuah item. Mau maksimal atau mau minimal ?

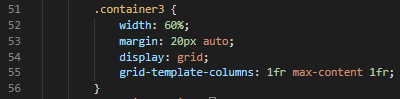


Untuk min-content akan membuat content dipisah menjadi masing - masing kata, kemudian lebar ditentukan dari kata yang terpanjang





Untuk max-content tidak akan memecah konten, dan tetap dijadikan satu baris. Ukurannya akan sesuai konten yang ada dalam 1 baris.

Value auto-fill & auto-fit

Menentukan jumlah item untuk berada pada grid-track, biasanya digunakan di dalam repeat()



Misal container-nya diberikan border solid supaya kelihatan, coba juga tampilkan grid saat inspect agar terlihat perbedaan antara auto-fill dan auto-fit. Untuk auto-fill akan menentukan akhir dari grid bisa misalnya item bisa muat lagi sesuai container, sedangkan auto-fit akan menentukan batas akhir grid sesuai dengan item yang terakhir. Misal dibawah ada 3 item di dalam sebuah container (pembungkus). Sifat grid dan item ini akan responsif sesuai dengan viewport.

 auto-fit

auto-fill

Value Special Function minmax()

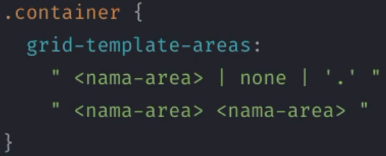
Menentukan ukuran minimal dan makasimal grid track. Kalo bisa ukuran maksimal, maka pakai ukuran maksimal, kalo sudah tidak muat (kecil) maka akan dikurangi hingga batas minimal tetap.



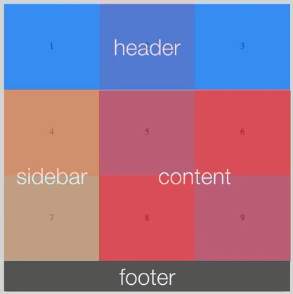
**Grid & Gab**

Property grid-template-areas

Mendifinisikan grid template menggunakan nama dari area yang ditulis pada property grid-area pada item. Item disini yaitu item yang parent-nya diberi display grid dan ada grid-template-areas.



Misal coba buat bentuk berikut di cssunpas4.html dan cssunpas4.css



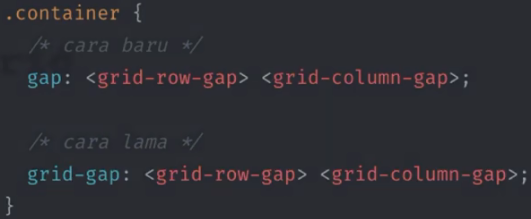
Shorthand grid-template

Merupakan shorthand untuk mendefinisikan grid-template-rows, grid-template-columns dan grid-template-areas dalam satu deklarasi saja.



Shorthand grid-gap (Cara Baru Pakai gap)

Merupakan cara memberi gab sekaligus gabungan dari cara lama (grid-column-gap & grid-row-gap) sudah tidak dipakai, atau cara baru (column-gap & row-gap) dalam satu deklarasi.



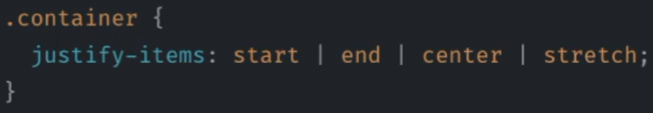
Shorthand grid

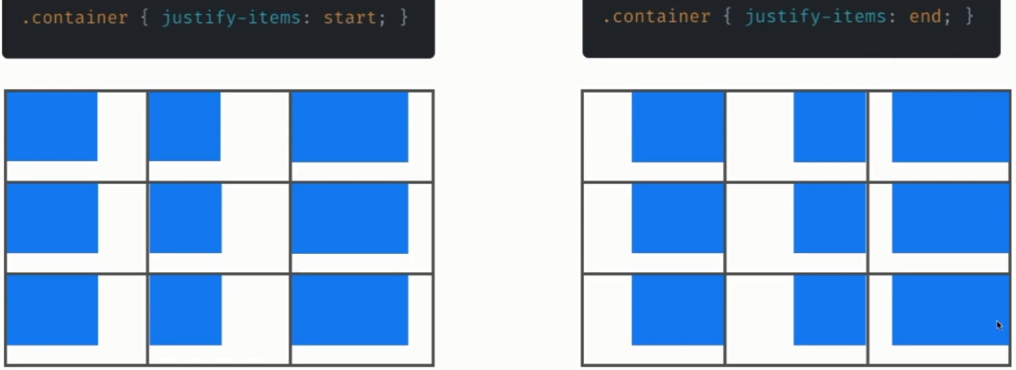
Merupakan shorthand paling lengkap, yaitu untuk mendefinisikan grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns dan grid-auto-flow dalam satu deklarasi. Umumnya jarang dipakai karena penulisan akan terlalu panjang.

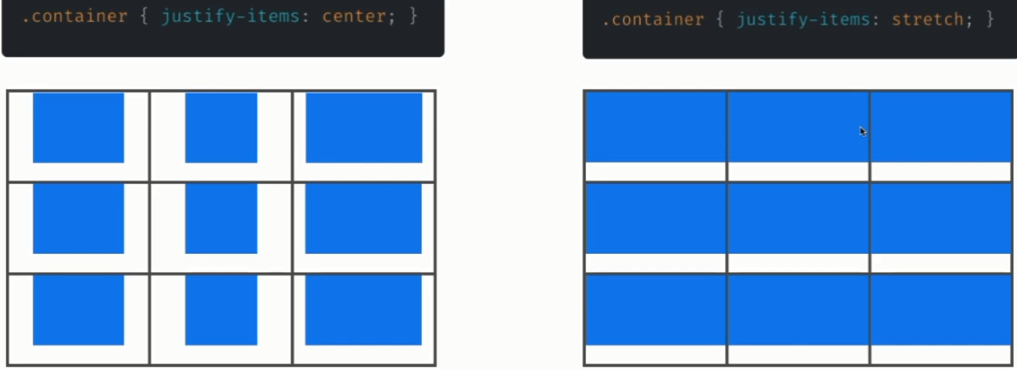
**Grid Alignment**

Property justify-items

Mensejajarkan grid-items pada sumbu horizontal. Defaultnya stretch, tapi stretch tidak akan berpengaruh kalo item - itemnya dikasih lebar (width). Value start akan membuat lebar sesuai isi konten saja (kalo tidak memiliki width).

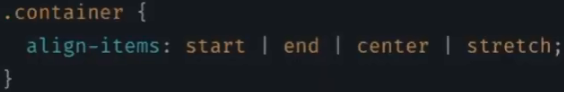


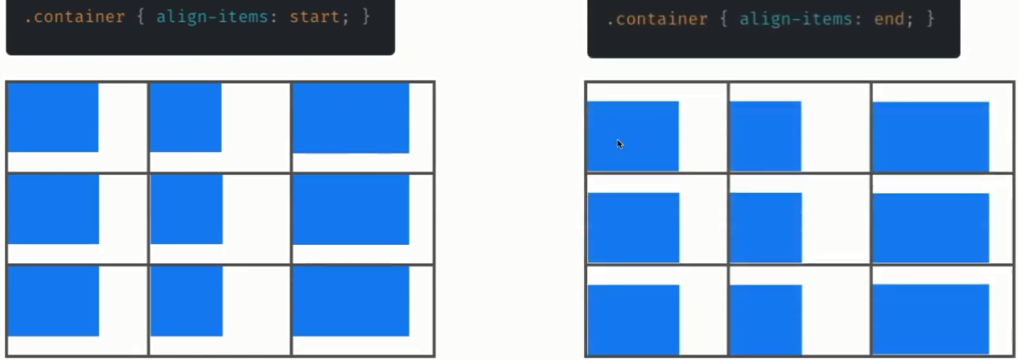


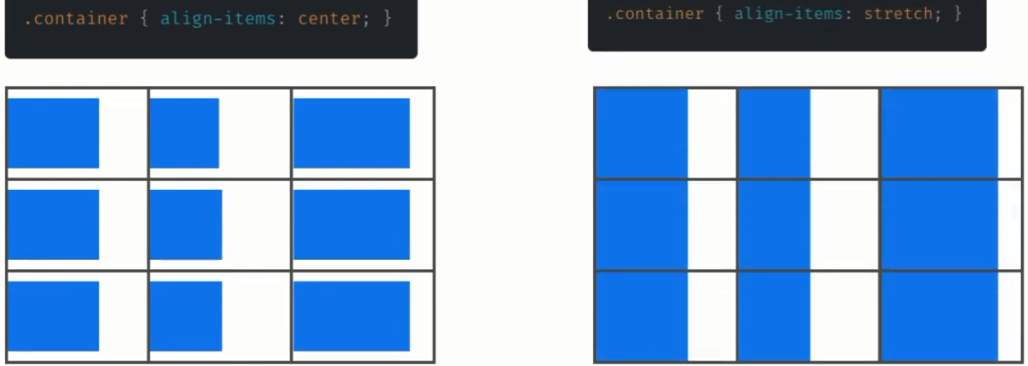


Property align-items

Mensejajarkan grid-items pada sumbu vertikal. Defaultnya stretch, tapi stretch tidak akan berpengaruh kalo item - itemnya dikasih tinggi (height).

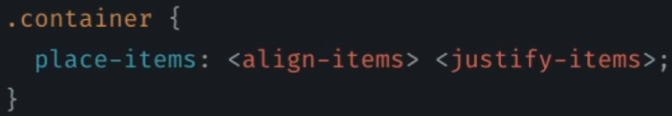






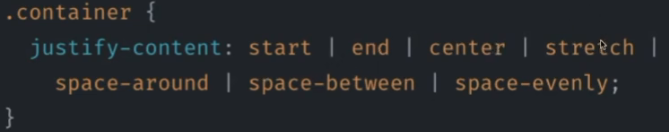
Shorthand place-items

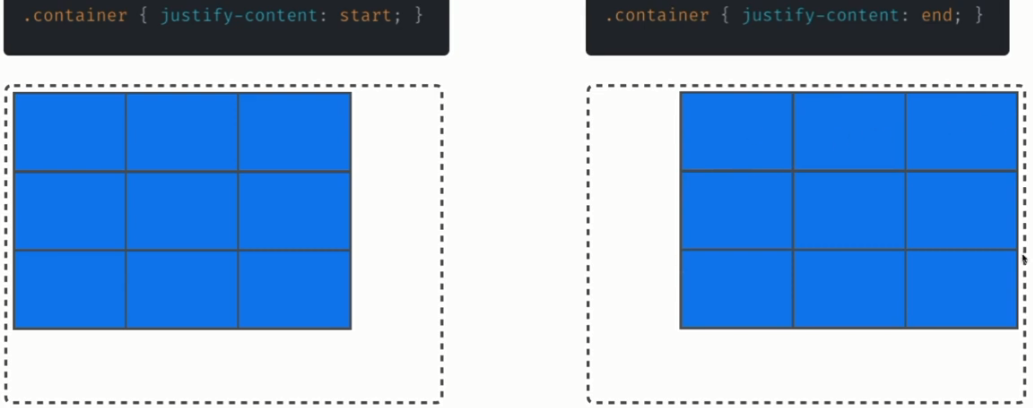
Merupakan shorthand untuk menulis justify-items dan align-items dalam satu deklarasi.

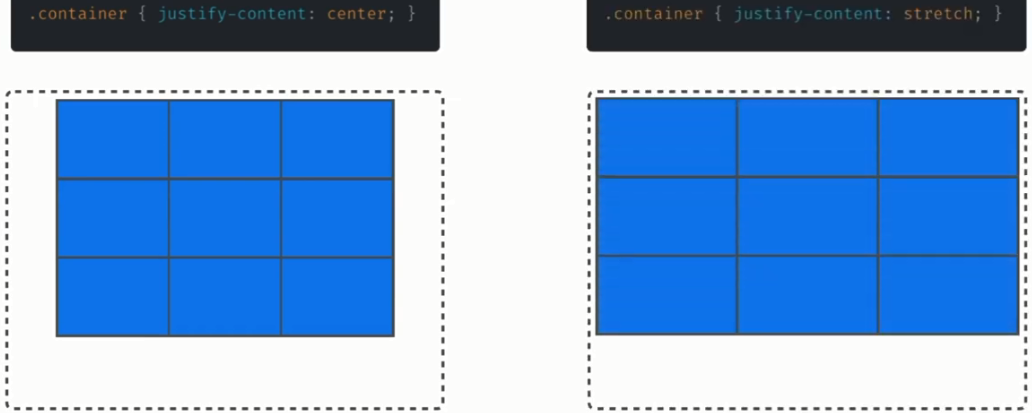


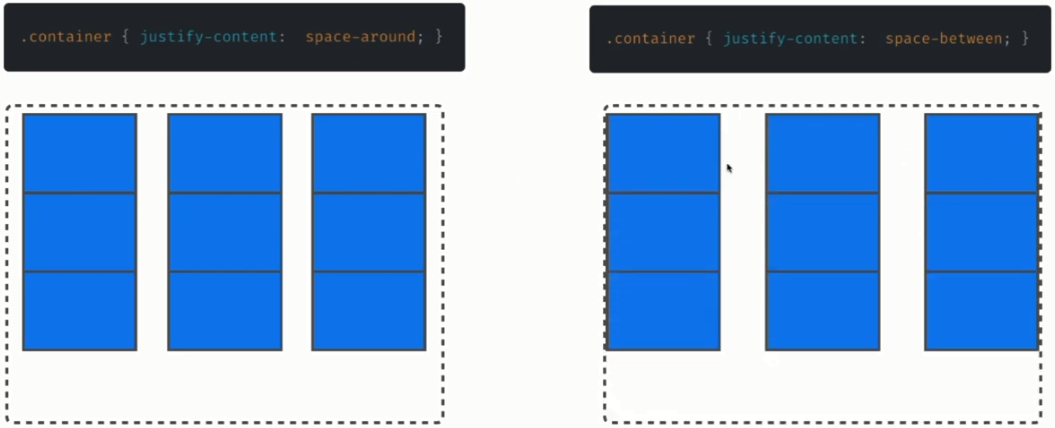
Property justify-content

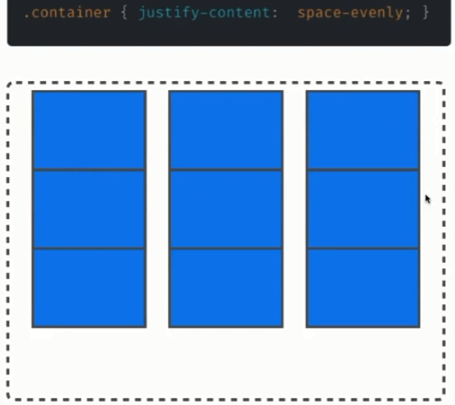
Mengatur posisi seluruh grid container pada sumbu horizontal. Ini bisa dilakukan ketika ukuran total grid lebih kecil dari ukuran containernya, biasanya ketika grid items-nya menggunakan ukuran yang fixed (px).





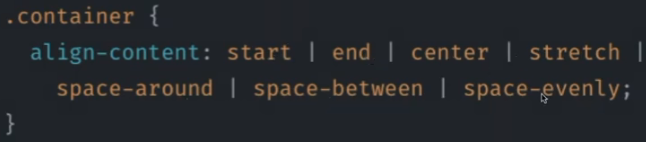


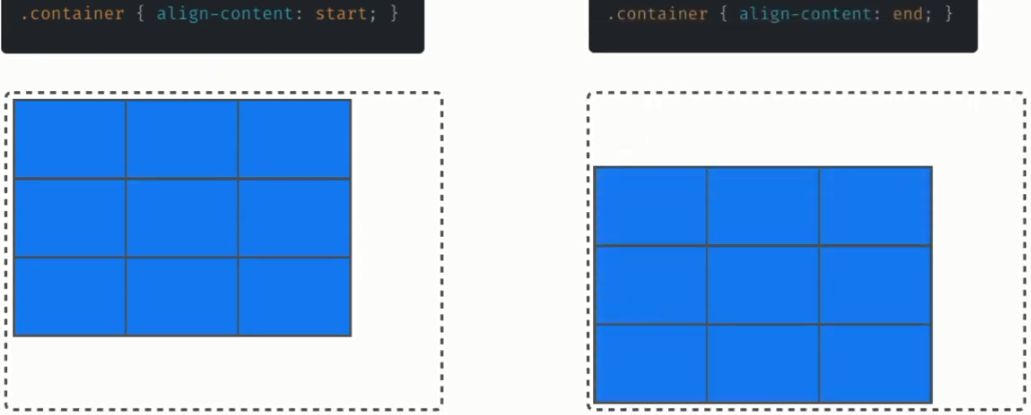


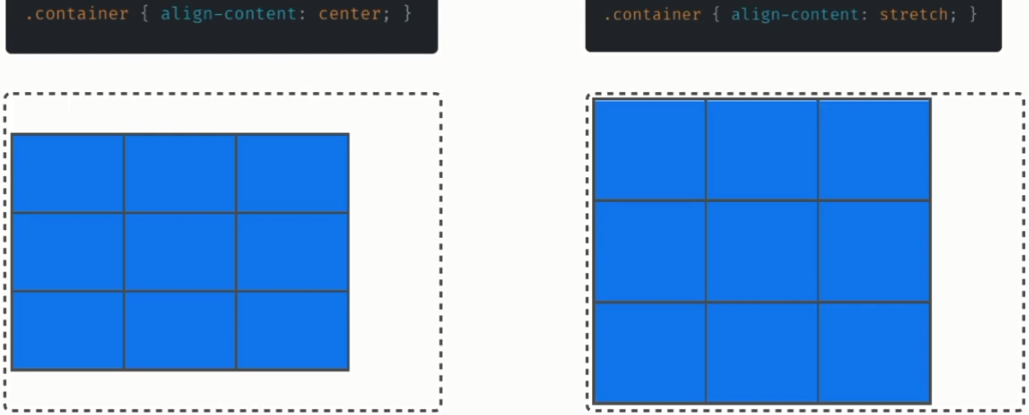


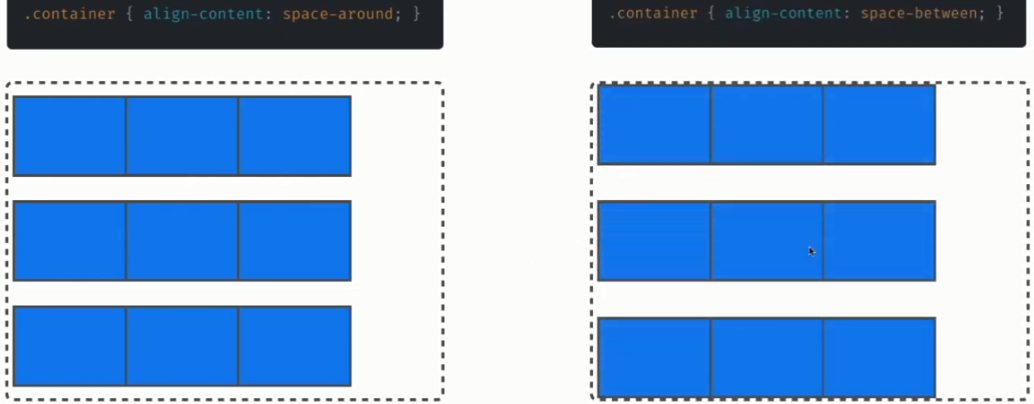
Property align-content

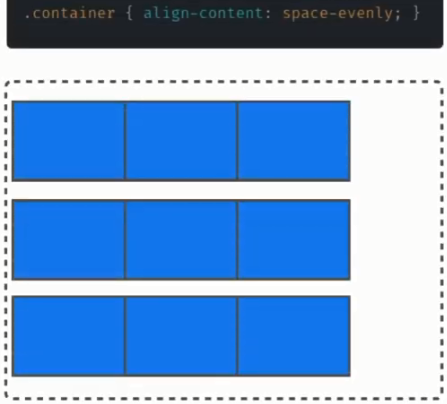
Mengatur posisi seluruh grid container pada sumbu vertikal. Ini bisa dilakukan ketika ukuran total grid lebih kecil dari ukuran containernya, biasanya ketika grid items-nya menggunakan ukuran yang fixed (px).











Shorthand place-content

Merupakan shorthand untuk menulis justify-content dan align-content dalam satu deklarasi.

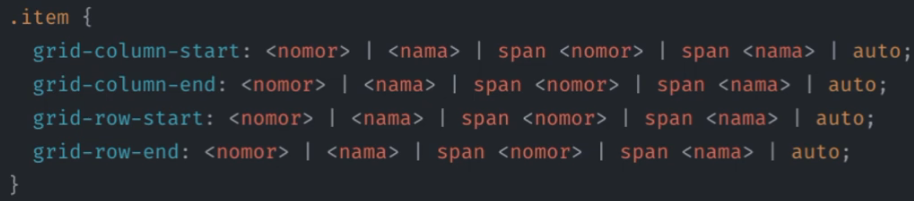


**GRID ITEM PROPERTIES**

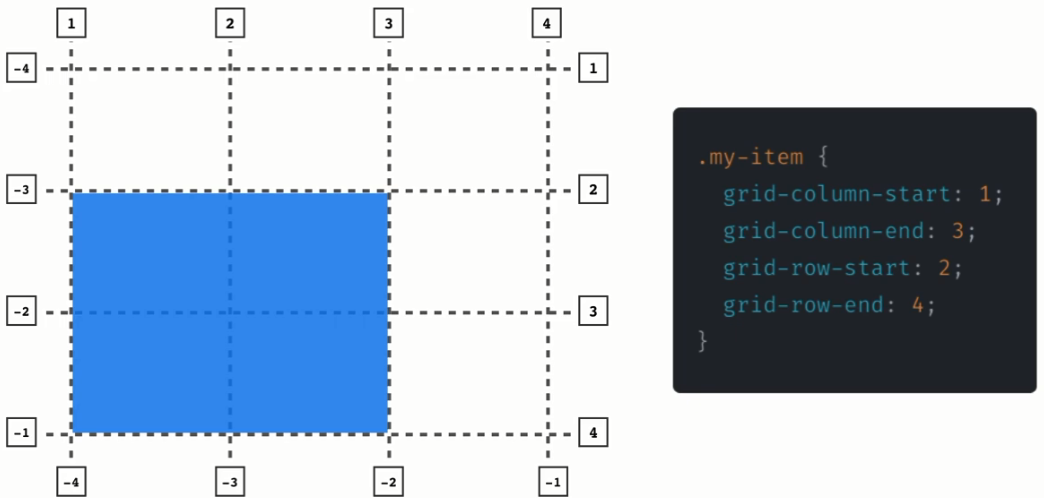
Merupakai property dari grid sytem yang bisa diberikan pada item-nya.

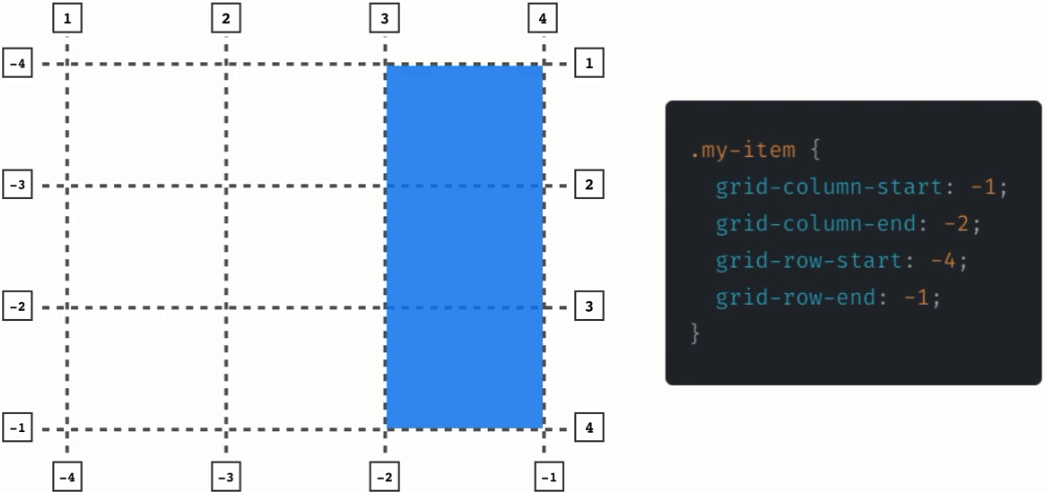
Property grid-column-start & grid-column-end Bersama Property grid-row-start & grid-row-end

Menentukan tempat dari grid item dengan mangacu pada nomor atau nama grid line yang spesifik.



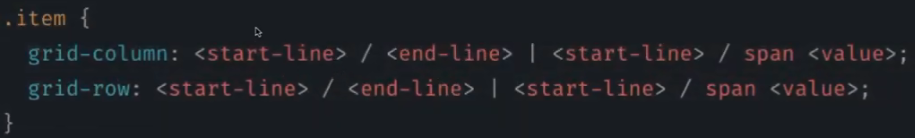






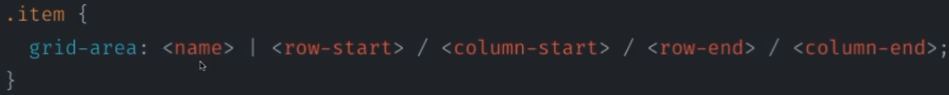
Shorthand grid-column & grid-row

Merupakan shorthand untuk grid-column-start & grid-column-end dan grid-row-start & grid-row-end. Untuk contohnya silahkan lihat di cssgridunpas5.html



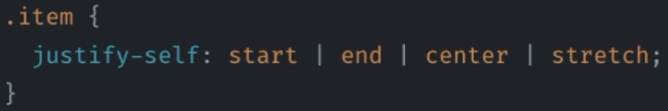
Property grid-area

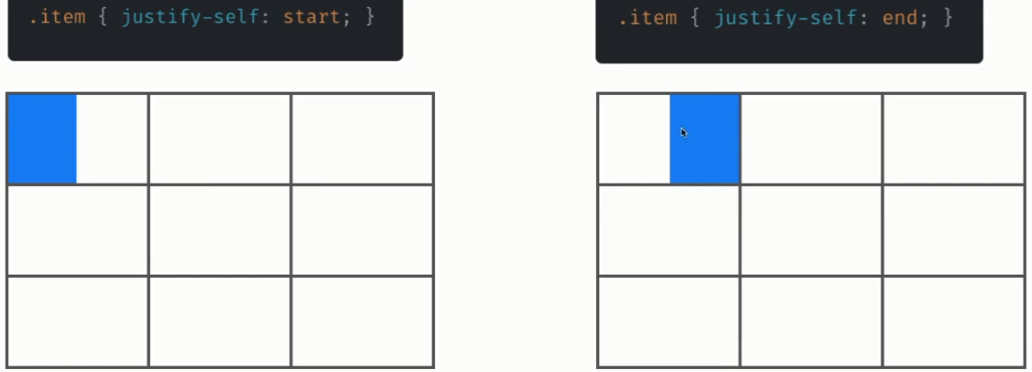
Ada 2 fungsi. Pertama, menantukan nama area pada item sesuai dengan template yang sudah dibuat sebelumnya melalui property grid-template-areas pada container. Kedua, bisa digunakan sebagai shorthand untuk grid-column-start & grid-column-end dan grid-row-start & grid-row-end. Untuk contohnya silahkan lihat di cssgridunpas5.html dan cssgridunpas5.css

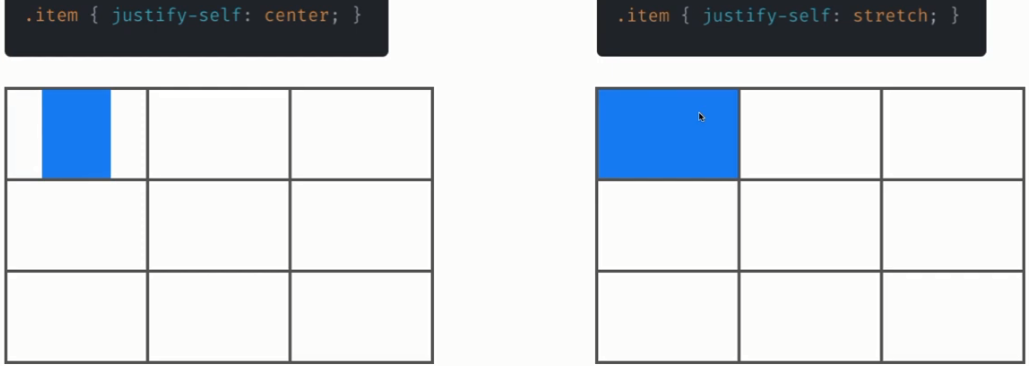


Property justify-self

Mengatur posisi item pada sebuah cell terhadap sumbu horizontal. Nilai default-nya stretch.

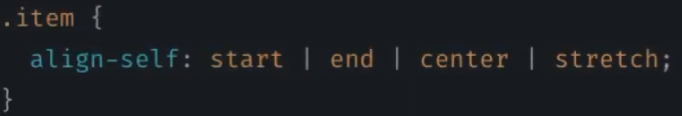


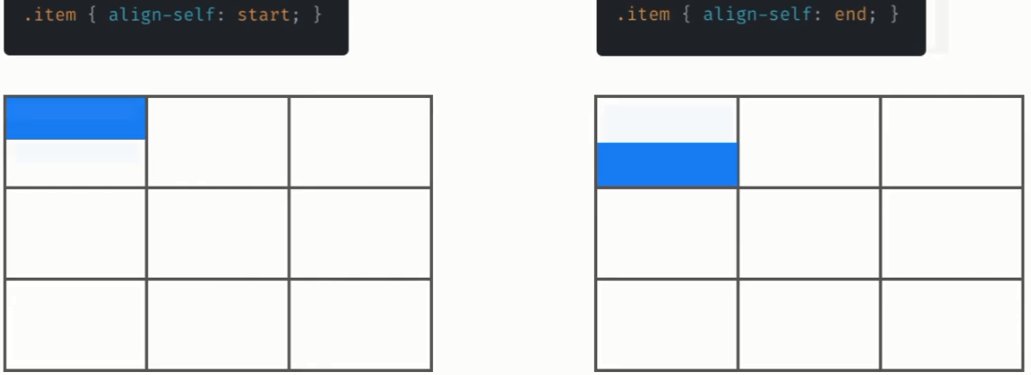


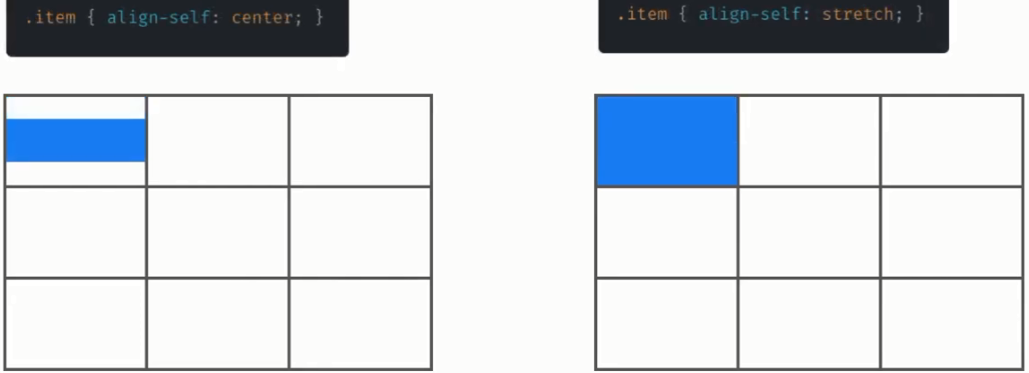


Property align-self

Mengatur posisi item pada sebuah cell terhadap sumbu vertikal. Nilai default-nya stretch.







Shorthand place-self

Merupakan shorthand posisi item pada sebuah cell terhadap sumbu vertikal dan sumbu horizontal (shorthand justify-self & align-self) dalam satu deklarasi.



NOTE : Kalo ada <div> di dalam container yang display-nya grid maka kalo grid-template-areas sudah ditentukan, tiap <div> otomatis akan memasukin masing - masing cell yang sudah dibuat dari container (sudah ditentukan baris dan kolom-nya).

NOTE : Kalo container sudah display-nya grid dan ukuran, template maupun area tidak ditentukan, maka banyaknya baris grid akan menjadi sesuai banyaknya element HTML di container tersebut, dan ukurannya hanya sesuai dengan isi konten element tersebut.