



Layout - Flexbox

Flexbox itu sebenarnya cuma *mode layout* di CSS yang diaktifkan dengan

```
display: flex
```

atau

```
display: inline-flex
```

Begitu dipasang, elemen itu jadi **flex container**, dan semua elemen langsung di dalamnya jadi

flex items

1 Apa itu Flex Container?

Flex container adalah elemen pembungkus yang mengatur posisi dan ukuran anak-anaknya (*flex items*) berdasarkan aturan *flexbox*.

Contoh:

```
.container {  
  display: flex; /* atau inline-flex */  
}
```

Semua **child langsung** dari `.container` akan ikut aturan flex.

2 Properti Utama di Flex Container

Ini properti yang dipakai untuk mengatur perilaku anak-anaknya:

A. **flex-direction**

Menentukan arah susunan flex items.

- `row` → default, susun dari kiri ke kanan
- `row-reverse` → susun dari kanan ke kiri
- `column` → susun dari atas ke bawah
- `column-reverse` → susun dari bawah ke atas

```
.container {  
  flex-direction: row; /* default */  
}
```

B. **justify-content**

Mengatur *posisi horizontal* (kalau `flex-direction: row`) atau *vertikal* (kalau `flex-direction: column`).

- `flex-start` → mulai dari awal
- `flex-end` → mulai dari akhir
- `center` → di tengah
- `space-between` → jarak sama, ujung nempel ke tepi
- `space-around` → jarak sama tapi ada padding kiri-kanan
- `space-evenly` → jarak sama rata semua sisi

C. **align-items**

Mengatur *posisi vertikal* (kalau `flex-direction: row`) atau *horizontal* (kalau `flex-direction: column`) untuk satu baris.

- `stretch` → default, item mengisi tinggi penuh
- `flex-start` → di atas
- `flex-end` → di bawah
- `center` → di tengah
- `baseline` → sejajar di garis teks

D. **flex-wrap**

Default-nya flexbox memaksa semua item berada di satu baris.

Kalau mau dibungkus ke baris berikutnya:

- `nowrap` → default
- `wrap` → pindah ke baris baru kalau penuh
- `wrap-reverse` → sama seperti `wrap` tapi urutan baris dibalik

E. align-content

Dipakai kalau **ada banyak baris/kolom** (pakai `flex-wrap`).

Mengatur jarak antar baris:

- `stretch` → isi penuh
- `flex-start` → rapat di awal
- `flex-end` → rapat di akhir
- `center` → di tengah
- `space-between` → jarak antar baris sama, ujung nempel
- `space-around` → jarak antar baris sama dengan padding kiri-kanan

3 Hubungan dengan Flex Items

Selain properti di atas untuk **container**, setiap **item** juga bisa diatur individual:

- `flex-grow` → seberapa besar dia bisa membesar
- `flex-shrink` → seberapa kecil dia bisa menyusut
- `flex-basis` → ukuran awal sebelum pembesaran/penyusutan
- `align-self` → posisi khusus item itu saja

NOTE:

Di **flex item**, ada properti `flex` yang memang bisa ditulis seperti `flex: 3` dan itu adalah **shorthand** dari tiga properti berbeda:

```
flex: flex-grow flex-shrink flex-basis;
```

1 Bagian-bagiannya

1. `flex-grow`

Angka yang menentukan *seberapa besar* item bisa **membesar** dibanding item lain jika ada ruang kosong.

- `0` → (default) tidak membesar otomatis

- `1` → membesar, bagi rata ruang kosong dengan item lain yang juga `1`
- `3` → membesar 3x lebih cepat dibanding item yang `1`

2. `flex-shrink`

Angka yang menentukan *seberapa kecil* item boleh **menyusut** jika ruang tidak cukup.

- `0` → tidak menyusut
- `1` → menyusut proporsional (default)
- `2` → menyusut 2x lebih cepat

3. `flex-basis`

Ukuran awal item sebelum dihitung grow/shrink (bisa `px`, `%`, atau `auto`).

- `auto` → ukurannya pakai width/height normal
- `0` → abaikan ukuran awal, langsung hitung grow

2 Kalau `flex: 3` artinya apa?

Kalau cuma tulis `flex: 3`, itu **shorthand** yang sama dengan:

```
flex-grow: 3;  
flex-shrink: 1;  
flex-basis: 0;
```

Artinya:

- Ukuran awal dianggap `0` (jadi semua ruang dibagi berdasarkan grow)
- Item ini tumbuh **3x lebih cepat** dari item yang `flex: 1`
- Masih boleh menyusut jika ruang kurang