# Layer Order

Layer order adalah sesuatu yang menentukan urutan suatu element diantara element yang lain. Apabila kalian pernah menggunakan aplikasi desgin grafis seperti Adobe Photoshop, kalian pasti familiar dengan permainan urutan layers.

Layer order pada html dan css tidak lah sesimple pada photoshop, pada kenyataannya layer order adalah salah satu pelajaran css yang cukup sulit dan membutuhkan imajinasi dan perencanaan yang bagus. Pada umumnya seluruh web designer dan front-end developer akan merancang dan membuat sitenya sebisa mungkin untuk menghindari rumitnya permainan layer order. Dengan begitu tidak akan ada kesulitan pada webnya.

Tetapi website tertentu contohnya website dengan design parallax mungkin akan membutuhkan perencanaan layer order yang rumit. Atau apabila kalian mencampur 2 atau lebih template aplikasi untuk digunakan di satu aplikasi, terkadang terjadi kekacauan z-index. (Bahkan di bootsrap 1).

Layer order akan terpengaruh oleh beberapa hal, misalnya seperti:

1. Urutan ditulisnya html node pada DOM di html document.
2. Hierarchy node tersebut pada rancangan DOM, hubungan parent dan childnya
3. Property position pada node tersebut
4. Nilai property z-index pada element tersebut

## 1. Urutan HTML dan DOM

Html tag yang ditulis paling awal secara default akan berada di posisi yang lebih dalam pada tampilan web, dan element yang ditulis atau di load belakangan akan berada paling luar. Ini akan cukup sulit dilihat tapi bisa dibayangkan, karena element html yang ditampilkan di layar browser secara default tidak akan bertumpukan dan overlay satu sama lain, melainkan akan terus tersusun ke bawah.

## 2. Hierarchy node

Apabila hierarchy node pada DOM adalah tiga tumpuk, seperti di bawah ini:

<div class=”wrapper”>

<div class=”container”>

<div class=”child”></div>

</div>

<div>

Maka child berada di paling depan, container di tengah, dan wrapper di urutan paling belakang. Tetapi apabila terdapat beberapa susunan seperti di bawah ini:

<div class=”wrapper”>

<div class=”container”>

<div class=”child”></div>

</div>

<div>

<div class=”wrapper-2”>

<div class=”container-2”>

<div class=”child-2”></div>

</div>

<div>

Walaupun child adalah urutan teratas, tetapi wrapper-2 tetap 1 tahap lebih tinggi dari child, sehingga dengan DOM seperti ini, css akan membuat urutan seperti berikut ini:

0: wrapper

1: container

2: child

3: wrapper-2

4: container-2

5: child-2

(0 terdalam – 5 tertinggi)

## 3. Property Position

Dengan memberikan property position (diluar position static) pada element, urutan element akan berubah. Element yang diberi position akan berada di lapisan teratas, tetapi konsep teratas pada position lain dengan teratas pada element yang tidak memiliki position. Semua element yang memiliki position, practically berada pada satu lapisan dimana tidak mungkin berada setara dengan element biasa. Bisa di simpulkan seluruh kedalaman element yang memiliki position adalah = lapisan terakhir non-position + urutan pada position.

contohnya seperti ini:

<div class=”wrapper”>

<div class=”container”>

<div class=”child”></div>

</div>

<div class=”other-container”>

<div class=”other-child”>

<div>

<div>

<div class=”wrapper-2”>

<div class=”container-2”>

<div class=”child-2”></div>

</div>

<div>

Andai kata element yang di highlight kuning memiliki position sedangkan yang tidak di highlight tidak diberi position.

Maka urutan dari element adalah kurang lebih seperti ini

Non-position:

0: wrapper

1: container

2: other-container

3: other-child

4: wrapper-2

(berarti 4 adalah angka terluar dari non-position, yaitu wrapper-2)

Position:

5 (4 + 1): child

6 (4 + 2): container-2

7 (4 + 3): child-2

## 4. Z-index Property

Z-index property adalah property yang bisa dipakai pada saat satu element yang memiliki position. Tanpa position atau apabila position static, maka z-index tidak akan berfungsi. Z-index memiliki kemampuan untuk menciptakan satu GROUP layer baru dimana mungkin berada di dalam atau di luar element berposisi atau tidak.

Masih menggunakan contoh yang sama:

<div class=”wrapper”>

<div class=”container”>

<div class=”child”></div>

</div>

<div class=”other-container”>

<div class=”other-child”>

<div>

<div>

<div class=”wrapper-2”>

<div class=”container-2”>

<div class=”child-2”></div>

</div>

<div>

Apabila saya memberikan z-index: 1 pada child, maka yang terjadi:

Group Non-position:

0: wrapper

1: container

2: other-container

3: other-child

4: wrapper-2

Group Position (z-index 0):

-

6 (4 + 2): container-2

7 (4 + 3): child-2

Group Z-index +1:

8 (7 + 1): child

Apabila saya memberikan z-index: -1 pada child, maka yang terjadi:

Group Z-index -1:

-1 (0 - 1): child

Group Non-position:

0: wrapper

1: container

2: other-container

3: other-child

4: wrapper-2

Group Position (z-index 0):

-

6 (4 + 2): container-2

7 (4 + 3): child-2

## 5. Kombinasi dari keseluruhannya

Pemakaian kombinasi dari urutan node, hierarchy, position dan z-index bisa sangat membingungkan apabila tanpa perencanaan yang rapih dan tersusun. Karena pada saat kita membawa satu element ke dalam group lapisan yang baru, anak-anak dan keturunan dari element tersebut juga akan terbawa. Ini yang akan sangat membingungkan.

Misalnya dengan contoh yang sama:

<div class=”wrapper”>

<div class=”container”>

<div class=”child”></div>

</div>

<div class=”other-container”>

<div class=”other-child”>

<div>

<div>

<div class=”wrapper-2”>

<div class=”container-2”>

<div class=”child-2”></div>

</div>

<div>

Saya akan memindahkan container-2 ke lapisan group z-index +1

Non-position:

0: wrapper

1: container

2: other-container

3: other-child

4: wrapper-2

(berarti 4 adalah angka terluar dari non-position, yaitu wrapper-2)

Position (z-index 0)::

5 (4 + 1): child

-

-

Group Z-index +1:

8 (7 + 1): container-2

9 (7 + 2): child-2