|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-3: CSS (box model, grid, flex box) Mata Kuliah Desain dan Pemrograman Web** Pengampu: Tim Ajar Desain dan Pemrograman Web *September 2020* |

**Nama : Rizqi Rohmatul Huda**

**Kelas : TI – 2G**

**No.Absen : 26**

**NIM : 2141720264**

**Topik**

* CSS Box Model
* CSS *Flex Box*
* CSS *Grid*

**Tujuan**

Mahasiswa diharapkan dapat:

1. Mahasiswa mampu menerapkan konsep CSS *Display*
2. Mahasiswa mampu menerapkan konsep CSS Box Model
3. Mahasiswa mampu menerapkan konsep CSS *Flex Box*
4. Mahasiswa mampu menerapkan konsep CSS *Grid*

**Perhatian**

*Jobsheet* ini harus dikerjakan *step-by-step* sesuai langkah-langkah praktikum yang sudah diberikan. Soal dijawab langsung pada lembar *form* jawaban yang sudah disediakan

**Pendahuluan**

**Apa itu CSS *Layouting* ?**

CSS *layouting* adalah tehnik untuk mengatur tata letak sebuah halaman web dengan menggunakan kode CSS. Tujuan dari CSS Layouting adalah agar halaman yang dibuat terlihat rapih, menarik sesuai dengan yang diharapkan oleh si pembuat.

CSS *layouting* terdiri dari beberapa bagian yaitu; dimensi, *overflow*, Box model, float dan Position Sebelum melakukan praktikum ada beberapa tools yang harus disediakan diantaranya yaitu

* Code editor
* Web browser

**CSS *Display***

Tag pada HTML digunakan untuk memberikan ‘maksud’ / ‘arti’ pada sebuah konten (contohnya adalah p untuk paragraph, h1 untuk *heading* utama dan lain-lain)

Tag <div> dan tag <span> tidak memiliki arti apapun, keduanya digunakan untuk mengelompokkan tag-tag HTML dan memberikan informasi terhadap tag-tag tersebut.

**Praktikum Bagian 1. CSS *Layouting***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah satu file baru di dalam direktori dasarWeb, beri nama index.html. |
| 2 | Ketikkan ke dalam file index.html tersebut kode di bawah ini. |

|  |  |
| --- | --- |
| 3 |  |
| 4 | Simpan file tersebut, kemudian buka browser dan jalankan localhost/dasarWeb |
| 5 | Kemudian buatlah <div> pada halaman html yang sama seperti pada kode program di bawah ini |
| 6 |  |
| 7 | Amati hasil dari kedua program tersebut sama atau berbeda., jelaskan alasanya **(soal 1)**  **Jawab :**  Program pertama akan terlihat normal Ketika di run di browser, sedangkan pada kode program kedua pada bagian navigasi akan bewarna pink dan pada bagian warna background konten bewarna hijau. |
| 8 | Tambahkan *style* pada *class* **navigasi** dan *class* **main** seperti pada kode program di bawah ini |

|  |  |
| --- | --- |
| 9 |  |
| 10 | Jalankan program *capture* hasilnya dan jelaskan apa yang terjadi **(soal 2)**  **Jawab :**  Penambahan code program pada style css akan menghasilkan tampilan bagian navigasi akan bewarna pink dan pada bagian warna background konten bewarna hijau. |

# Hub antara tag <div> dan <span> pada *display* CSS

Mempunyai *property display*

|  |  |
| --- | --- |
| Tag | |
| <div></div> | <span></span> |
| Div {Display: *block*;} | Span {Display: *inline*;} |

**CSS *display* adalah**

Setiap tag pada HTML berada di salam sebuah kotak. *Property display* pada CSS mengatur perilaku dari kotak tersebut

(sumber; <https://css-tricks.com/almanac/properties/d/display>)

Setiap tag pada HTML memiliki nilai *default* untuk *property display*. Tetapi kita juga dapat mengubah perilaku dari tag tersebut dengan value-nya

(sumber; <https://developer.mozilla.org/en-US/docs/Web/CSS/display>)

**Value dari *display* ada empat**

***inline***

elemen HTML yang secara *default* tidak menambahkan baris baru Ketika dibuat

Karakteristik dari *dispay inline* adalah

1. lebar dan tinggi elemenya besarnya sesuai dengan konten yang ada di dalamnya
2. Tidak dapat mengatur tinggi dan lebar dari elemen *inline*
3. *Margin* dan *padding* hanya mempengaruhi elemen secara horizontal, tidak vertical

Adapun elemen-elemen inline adalah sebagai berikut;

# b, strong, i, em, a, span, sub, sub, button, input, label, select, textarea

(sumber; <https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_element>)

# Praktikum Bagian – 2. Display *Inline*

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Silakan hapus *style* pada soal no.2. kemudian tambahakan link 5 pada *class* navigasi seperti  kode program di bawah ini |
| 2 |  |

|  |  |
| --- | --- |
| 3 | Jalankan program *capture* hasil dan amati apa yang terjadi. **(Soal 3)**  **Jawab :**  Ketika style dihapus tampilan akan Kembali normal sama seperti percobaan nomor 1, bedanya hanya penambahan jumlah link, yaitu penambahan link 5. |

***inline-block*** Jika pada elemen *inline* kita tidak bisa mengatur tinggi dan lebar dari suatu elemen maka kita bisa mengaturnya dengan menggunaka *inline-block*. tidak ada elemen yang secara *default* memiliki *property***.**

Begitu juga dengan *property display* pada CSS.

*display: inline-block*;

kita harus ubah secara manual *property* tersebut Secara umum prilaku dasar dari dari elemen *inline-block*

sama dengan elemen *inline*

**Praktikum Bagian – 3. Display *inline-block***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 2 | silakan tambahkan *weight*, *hight* dan *display* pada *style* di elemen **a** seperti pada kode program dibawah ini |
| 2 |  |
| 3 | Jalankan program *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 4)**  **Jawab :**  Pada bagian link akan terbagi menjadi bagian-bagian berbentuk persegi dengan warna background pink yang masing-masing bagian terisi satu link. |

***Block*** *Block* adalah elemen HTML yang secara *default* menambahkan baris baru Ketika dibuat Jika tidak diatur lebar-nya, maka lebar *default* dari elemen *block* akan memenuhi lebar dari *browser* / *parent*-nya sehingga kita dapat mengatur tinggi dan lebar dari elemen *block.* Di dalam elemen *block*, kita dapat menyimpan tag dengan elemen *inline*, *inline-block*, atau bahkan elemen *block* lagi

Adapun contoh-contoh elemen *block* adalah ; h1-6, p, ol, ul, li, form, hr, div

(sumber; <https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements>)

# Praktikum Bagian – 4. Display *block*

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Beri *style* pada elemen h1, h2 dan p dengan *background-color* warna *lightgreen* seperti  pada kode program di bawah ini |
| 2 |  |
| 3 | Jalankan program dan coba resize halaman web tersebut. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 5)**  **Jawab :**  Pada bagian a, h1, h2, p warna background akan menyesuaikan dari *background-color* yang pada style. Pada bagian a bewarna pink dan h1, h2, p bewarna hijau. |
| 4 | Kemudian selanjutnya silakan tambahkan style pada class main seperti pada kode program dibawah ini |

|  |  |
| --- | --- |
| 5 |  |
| 6 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 6)**  **Jawab :**  Tampilan pada main akan berubah lebih kecil daripada tampilan normal dan setiap bagian seperti a, main, dan h2 pada main akan bewarna sesuai  *background-color* pada style. |

Untuk *value display* yang terakhir adalah *none*, none ini dapat digunakan untuk menghilangkan sebuah elemen

**Dimensi dan *Overflow* pada CSS**

Dimensi memiliki dua *property* di CSS yaitu *width* untuk lebar dan *height* untuk tinggi. Satuan dari dimensi ada macam macam diantaranya adalah **px, %, in, cm, mm pc pc**

*Overflow* adalah *property* CSS yang digunakan untuk mengatur perilaku elemen yang tidak cukup pada suatu *parent.*

Ada empat *value* dari *property overflow* diantaranya yaitu;

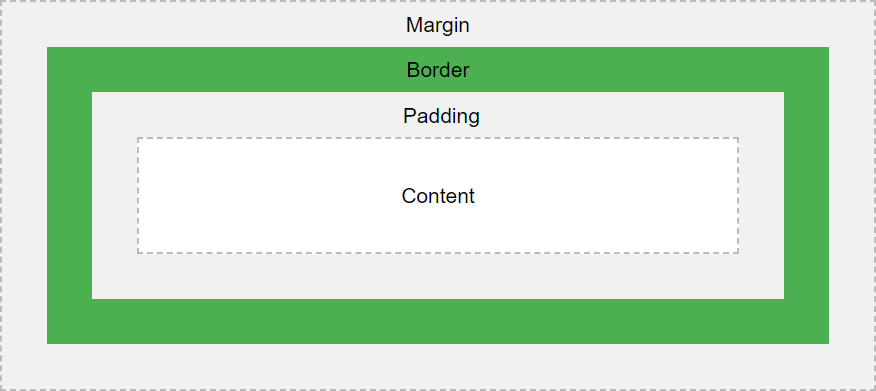
1. *Visible* : *Value default*
2. *Auto* : CSS akan secara otomatis akan menambahkan *scroll* jika konten tidak cukup
3. *Hidden* : konten akan disembunyikan atau tidak kelihatan
4. *Scroll* : seperti auto, akan memunculkan *scroll*, tetapi jika *content* cukup scroll akan tetap ada

***Box model* pada CSS**

Setiap elemen di halaman *website* berada di dalam sebuah *box* (kotak). Kita bisa mengatur ukuran dan posisi kotak tersebut. Kita bisa memberi warna / gambar sebagai *background* kotak tersebut (sumber; <http://css-tricks.com/the-css-box-model/>)

Box model pada CSS mendefinisikan ‘kotak’ yang dihasilkan oleh sebuah elemen, lalu menampilkannya sesuai dengan format visualnya. (sumber; <http://www.w3.org/TR/CSS2/box.html>)

CSS box model terdiri dari 4 komponen yaitu ; margin, border, padding dan *content* seperti pada gambar di bawah ini. (sumber; <http://www.w3schools.com/css/css_boxmodel.asp>)



*Gambar 1.* Komponen Box Model

1. *Margin*: area transparan di sekitar kotak (diluar *border*)
2. *Border*: batas disekeliling *conten* dan *padding*
3. *Padding*: area transparan di dalam kotak (antara *content* dan *border*)
4. *Content*: konten sebenarnya di dalam *box*, bisa berupa teks atau gambar Cara mengatur propertinya seperti pada tabel 1 dibawah ini

*Tabel 1. Cara mengatur property box model*

|  |  |  |
| --- | --- | --- |
| ***Margin*** | ***Padding*** | ***Border*** |
| *Margin-top .sMargin-right Margin-bottom Margin-left*  *margin* | *Padding -top Padding -right Padding -bottom Padding -left*  *Padding* | *Border -top Border -right Border -bottom Border -left*  *Border* |

## Box Model : Margin

**Praktikum Bagian – 5. *Box Model : Margin***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buatlah 2 file seperti pada gambar di bawah ini. File yang pertama dengan nama  index.html dan file yang kedua style.css |
| 2 |  |
| 3 | Jalankan program, amati hasilnya dan jelaskan **(Soal 7)**  **Jawab :**  Program akan menampilkan 3 box dengan warna dan ukuran yang berbeda |
| 4 | Selanjutnya adalah memberi *margin*, dimana *margin* adalah *area transparan* yang ada di sekitar kotak. Tambahkan *style* pada style.css **margin-top** sebesar 100px pada **class.satu** simpan dan kemudian jalankan pada web browser. Capture dan amati hasilnya  **(Soal 8)**  **Jawab :**  Posisi ketiga box menjadi lebih ke bawah sedikit dengan penambahan margin-top sebesar 100px. |
| 5 | Kemudian tambahkan ukuran ***margin*** yang lain seperti pada kode program berikut ini; |
| 6 |  |

|  |  |
| --- | --- |
| 7 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 9)**  **Jawab :**  Posisi box satu akan menjahui box dua dan tiga sesuai dengan posisi margin. |

*Ovelapping* margin

Terjadi Ketika kita menggabungkan dua buah margin. Kiri dan kanan atau atas dengan bawah Secara teori jika terjadi maka akan di ambil nilai yang paling besar.

*Negative* margin akan membuat kotak berlawanan arah. Bisa digunakan jika kita ingin menyembunyikan elemen.

*Auto* margin adalah nilai yang dapat kita kasih kedalam margin khusus untuk margin kiri dan margin kanan.. auto ini akan membuat elemenya berada di tengah-tengah halaman web browser

*Shorthand* margin adalah cara menyingkat penulisan margin

## Box Model : Padding, Border & Box Sizing

*Padding*

Cara pakai *padding* sama seperti margin

* *Padding* tidak bisa di pakai *negative*
* *Padding* tidak bisa di beri nilai *auto*
* *Padding* mempengaruhi ukuran dari *box* dari suatu elemen

*Border*

Cara menulisnya;

Border: width style color;

*Style* pada border; solid, dotted (titik titik), dashed(garis-garis), double Border merubah ukuran dari kotak atau elemen

box model secara **default** memiliki perhitungan untuk nilai width x height suatu elemen sebagai berikut: Width = width + left padding + right padding + left border + right border

Height = height + top padding + bottom padding + top border + bottom border

*Box sizing*

*Box sizing* adalah properti yang menerima nilai *padding* dan *border* pada suatu elemen termasuk sebagai nilai total dari *width* dan *height* suatu elemen.

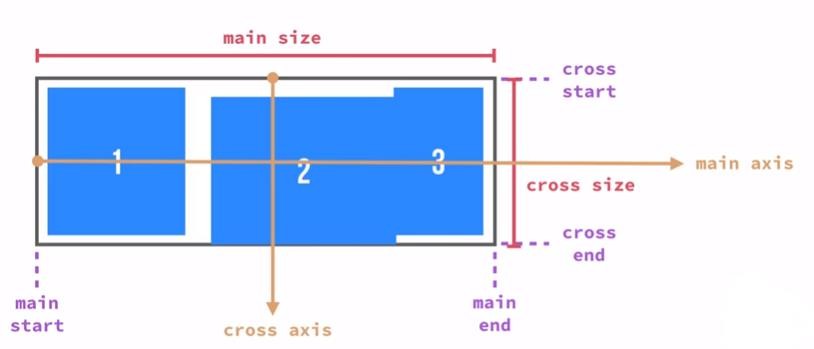
box-sizing: content-box (default)| border-box | unser | initial | inherit;

## Flex Box

Konsep dan definisi

Model layout 1 dimensi yang dapat mengatur jarak dan penjajaran antar item dalam sebuah *container* (sumber; <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout>) .Yang dimaksud dengan satu dimensi adalah hanya dapat mengatur satu dimensi pada saat tertentu, antara baris atau kolom, tidak bisa keduanya sekaligus.

*Flex Box* atau bisa di sebut juga *Flexbox Layout Module* adalah sebuah modul yang menawarkan cara yang efektif untuk Menyusun, mensejajarkan dan mendistribusikan jarak antar item di dalam sebuah *container*, meskipun ukuranya dinamis atau bahkan kita tidak tahu. (sumber; [https://css-](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) [tricks.com/snippets/css/a-guide-to-flexbox/](https://css-tricks.com/snippets/css/a-guide-to-flexbox/))



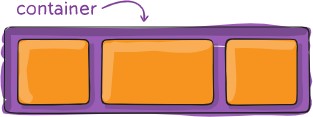
*Gambar 2 Istilah-istilah pada flex Box*

(sumber; <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>)

* **Main axis**; Sumbu utama dari sebuah *container* yang menentukan urutan dari penempatan item secara horizontal
* **Main start**/**main end**; Mulai dan berakhirnya items yang disimpan di dalam *container*
* **Main size**; ukuran (*width/height*) dari *container* yang akan membuat dimensi dari items nya relative terhadap size

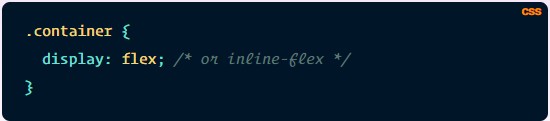
## Property pada container

*Container* adalah pembungkus dari elemen element



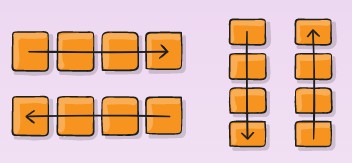
## Display

*Gambar 3 Container*

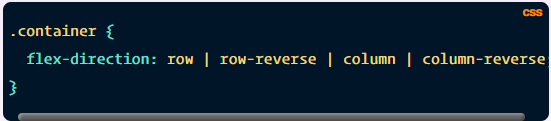


*Display* Membuat sebuah elemen *parent* menjadi flex box, dan membuat elemen elemen di dalamnya bisa berprilaku flex juga.

## Flex- direcrtion



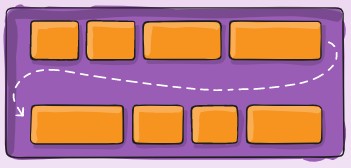
*Gambar 4 Flex- direcrtion*



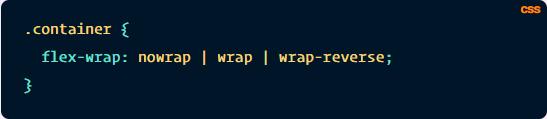
*Flex-direction* Mengatur arah / urutan dari items di dalam *container*

## Flex-wrap

Secara *default*, semua *items* di dalam *container* akan berada pada satu baris meskipun ukuranya sudah sudah tidak cukup, *wrap* memungkinkan untuk memindahkan items ke baris di bawahnya



*Gambar 5 Flex-Wrap*



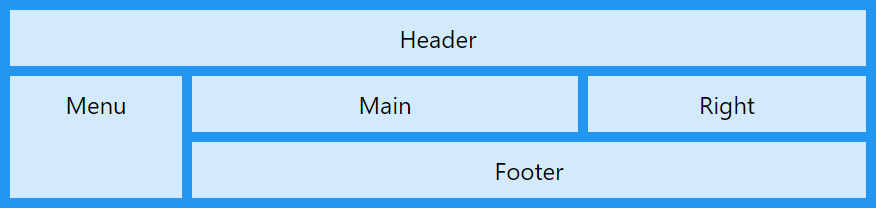
Untuk property yang lain bisa kunjungi website beriku [https://css-tricks.com/snippets/css/a-guide-to-](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) [flexbox/](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)

**Praktikum Bagian – 6. *Flex Box***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ketikkan kode program di bawah ini kemudian simpan file dengan nama flexbox.html |
| 2 |  |
| 3 | Dan untuk file style.css sebagai berikut |
| 4 |  |
| 5 | Jalankan program, amati hasilnya dan jelaskan **(Soal 10)**  **Jawab :**  Dengan menggunakan metode flex-box seperti diatas maka kita akan lebih mudah mengatur jarak dan penjajaran antar item dalam sebuah *container.* |
| 6 | Tambahkan *style* pada class **container-satu** pada file style1.css seperti pada kode program di bawah ini |

|  |  |
| --- | --- |
| 7 |  |
| 8 | Jalankan program. *capture* hasil, amati dan jelaskan apa yang terjadi. **(Soal 11)**  **Jawab :** artikel berada diatas halaman web browser sehingga terlihat lebih rapih. |
| 9 | Tambahkan property **display** pada selector container-satu dengan value **flex**. Capture dan jelaskan hasil nya **(Soal 12)**  **Jawab :** membagi konten yang semula ke bawah menjadi dua bagian menyamping |

**CSS *Grid***

*Grid layout module*

*Gambar 6 Grid Layout Module*

*CSS grid layout module* menawarkan *system layout* berbasis grid dengan baris dan kolom sehingga mempermudah melakukan desain halaman web tanpa menggunakan float dan position

Modul CSS baru untuk mendefinisikan system layout berbentuk grid dalam 2 dimensi (baris dan kolom)

*Grid element*

Grid layout terdiri dari satu elemen induk dengan satu atau lebih elemen anak Contoh

*CSS Grid Terminology*

* *Grid container*

Element pembungkus grid, didefinisikan dengan menuliskan:

display: grid;

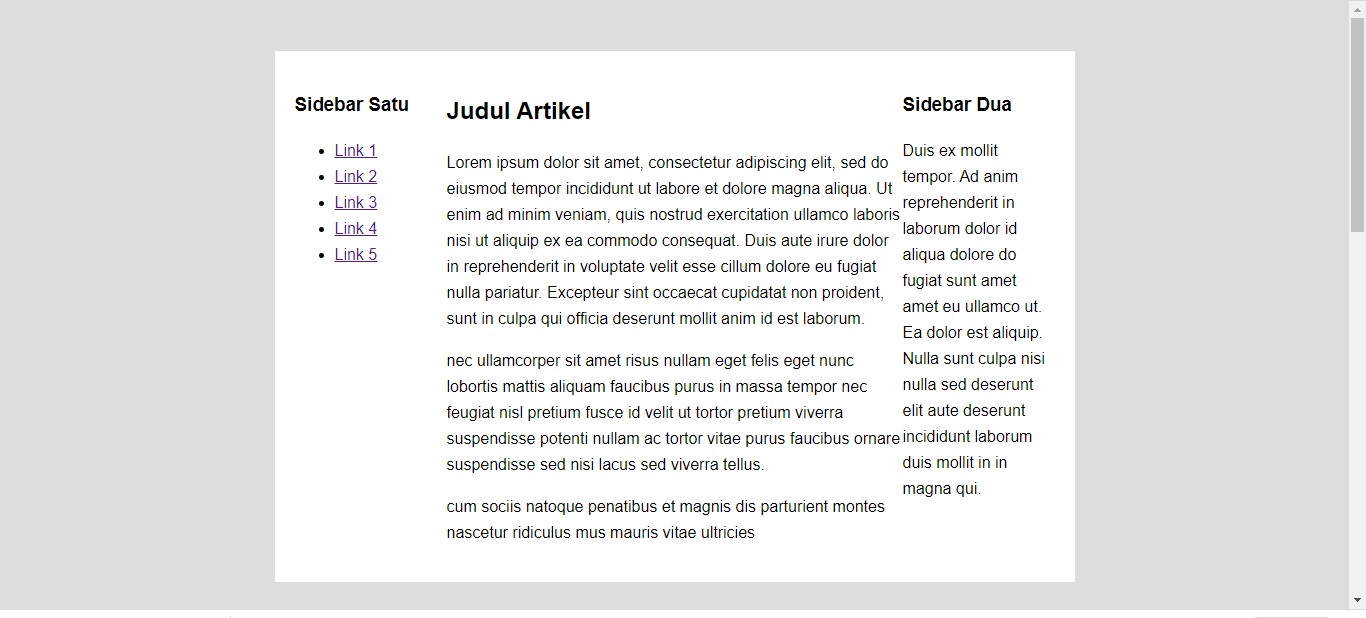
* *Grid item*; element element yang berada (1 level) di dalam grid container
* *Grid line*; Garis horizontal (kolom) atau vertical (baris) yang memisahkan grid menjadi beberapa bagian dan ditandai dengan angka
* *Grid cell*; Perpotongan/pertemuan antara baris dan kolom di dalam grid
* *Grid area*; Kumpulan lebih dari satu grid cell yang membentuk kotak
* *Grid track*; Ukuran/jarak antara 2 grid line, bisa horizontal (kolom) atau vertical (baris)
* *Grid gap*; jarak antar grid track/cell

Untuk lebih detail tentang grid bisa mempelajari di website berikut ini <https://www.w3schools.com/css/css_grid.asp>

**Praktikum Bagian – 7. *CSS Grid***

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Ketikkan kode program di bawah ini kemudian simpan file dengan nama grid.html |
| 2 |  |
| 5 | Jalankan program, amati hasilnya dan jelaskan **(Soal 13)**  **Jawab :** dengan menggunakan grid kita dapat memodifikasi konten menjadi beberapa bagian dengan mudah. |

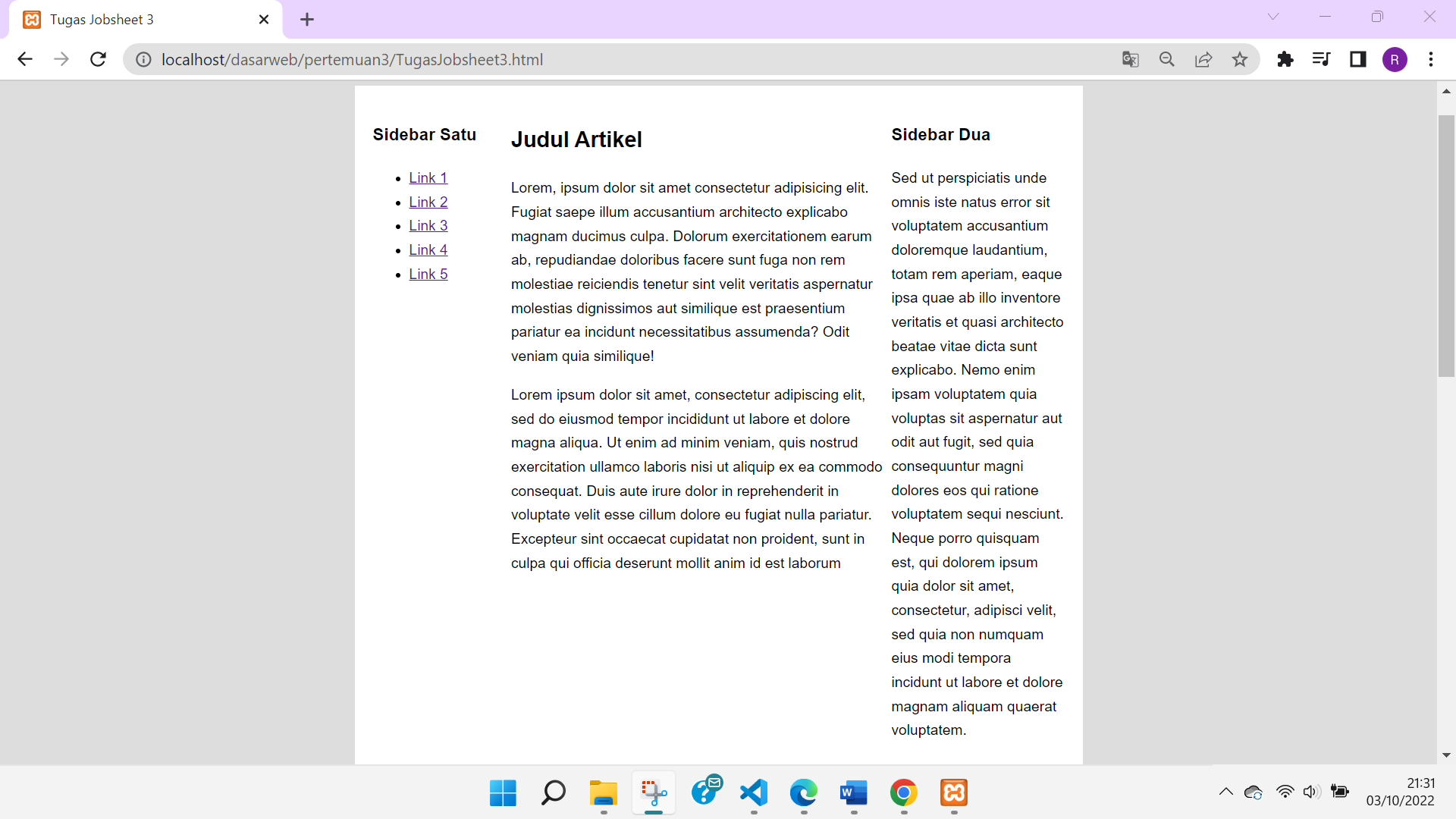
Tugas *Jobsheet* 3:

1. Buatlah tampilan dari hasil *flex-box* sebelumnya menjadi seperti ini

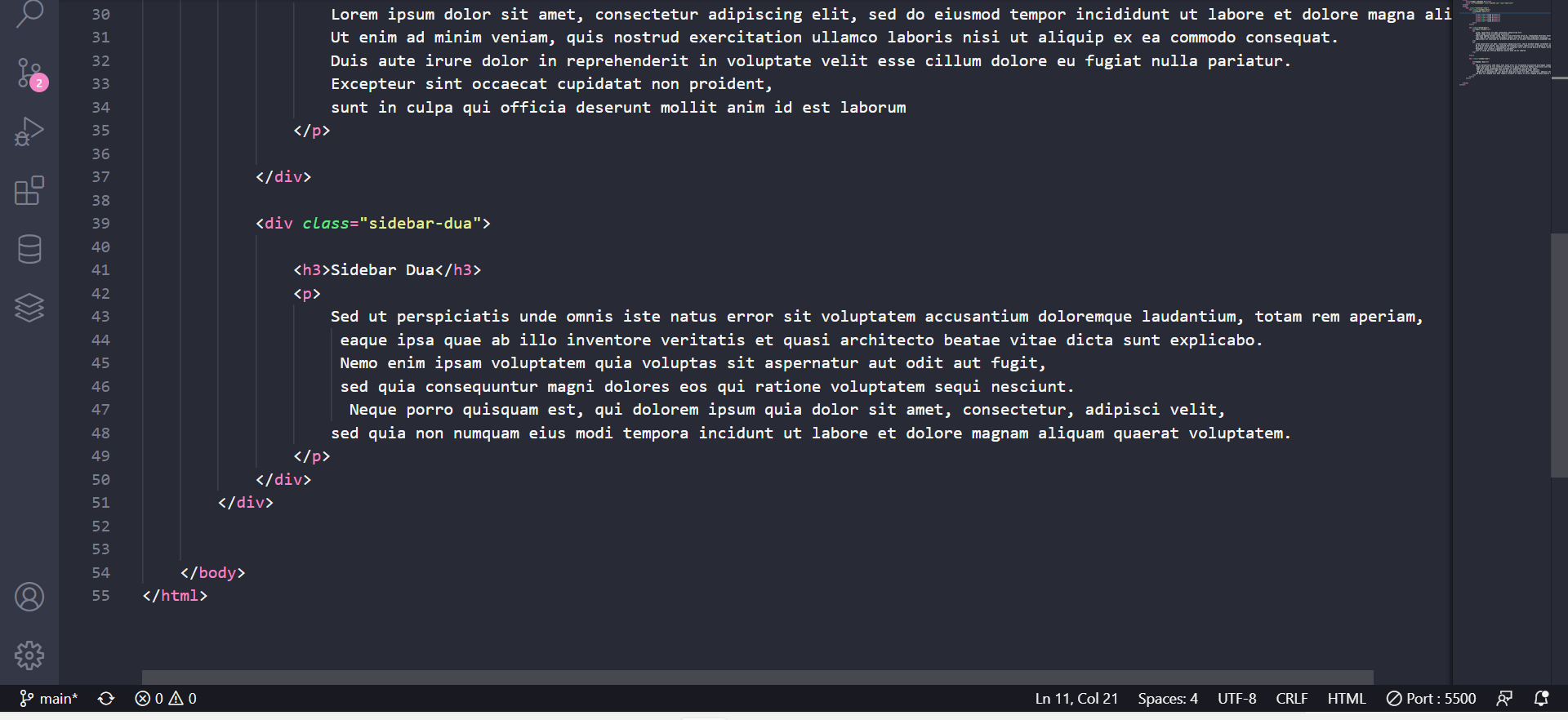
**Kata kunci**; gunakan *property flex* dan *order* pada masing-masing *selector* itemnya. Kemudian

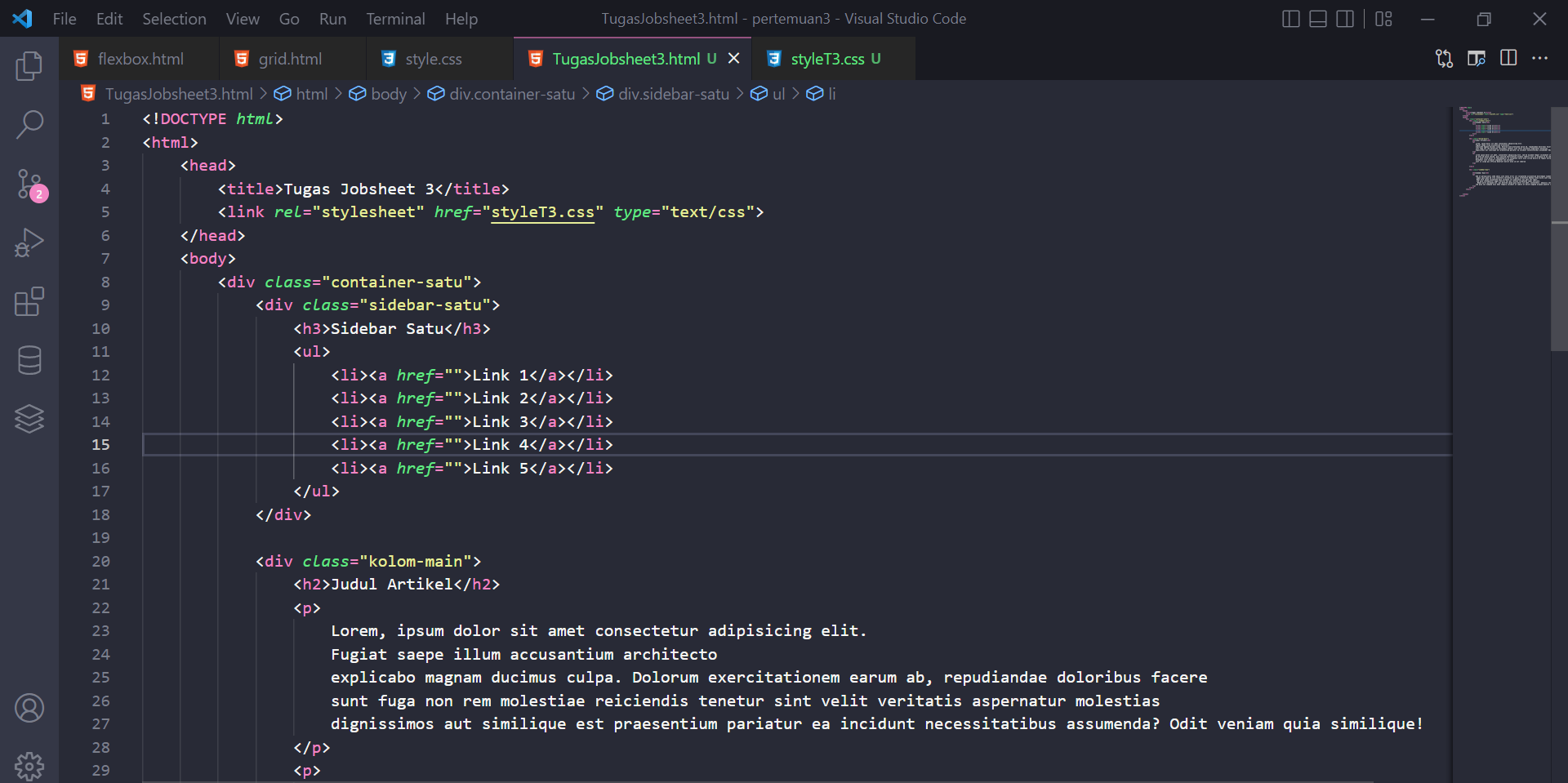
*capture* kode program dan hasilnya

**Jawaban Tugas Jobsheet 3**

****

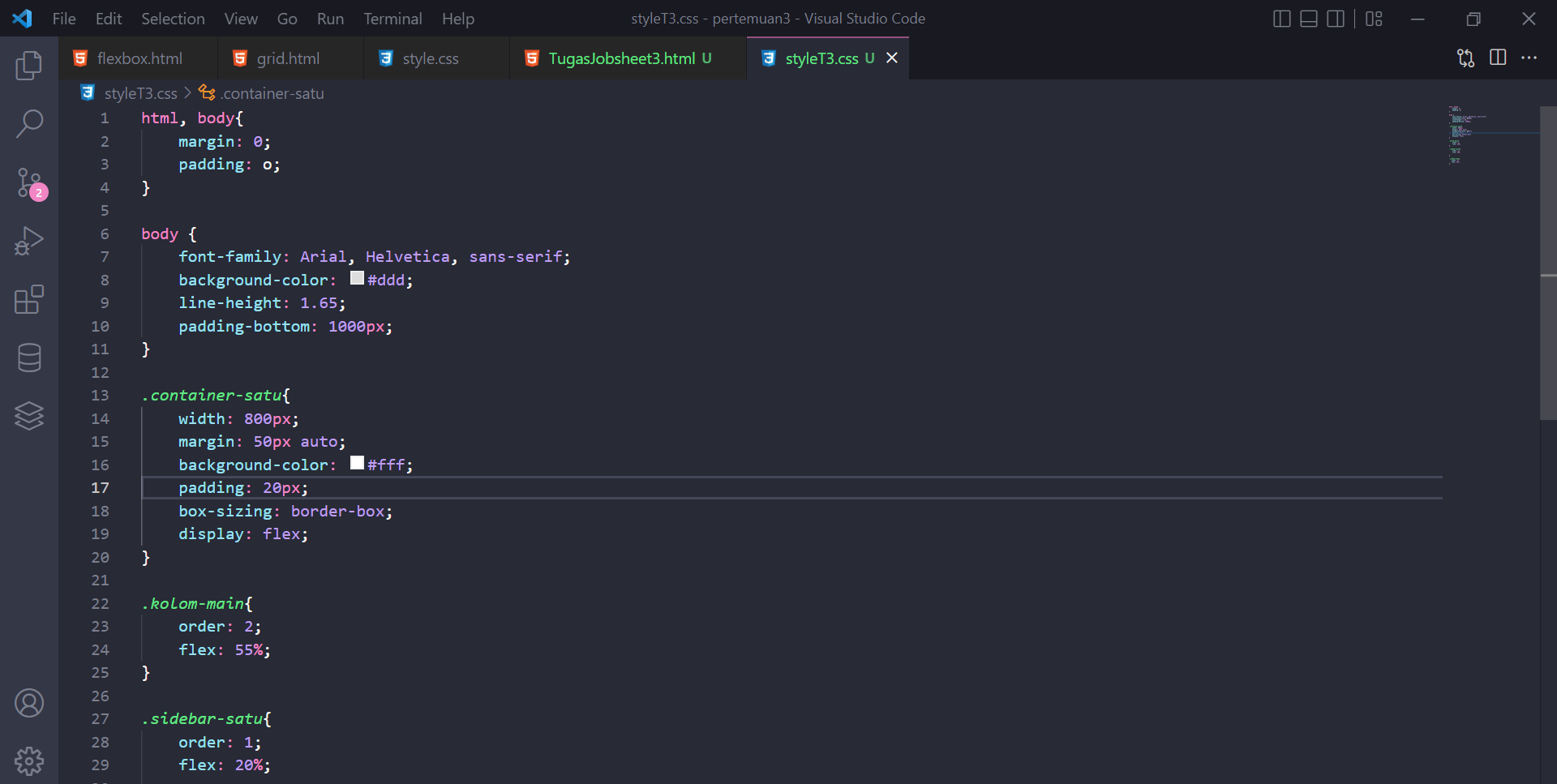
Code html :





Code css :





# Referensi:

1. Jason Beaird, The principles of Beautiful Web Design
2. Rian Ariona, Belajar HTML dan CSS (Tutorial Fundamental dalam mempelajari HTML dan CSS)
3. Adi Hadisaputra, HTML dan CSS Fundamental dari Akar menuju Daun John Duckett,HTML dan CSS design and build websites
4. <https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements>
5. <https://css-tricks.com/almanac/properties/d/display>
6. <http://www.w3.org/TR/CSS2/box.html>)
7. <http://www.w3schools.com/css/css_boxmodel.asp>)