

HTML & CSS For **EDUCATOR**

Femi Dena Juang
femijuang@gmail.com

2020

KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadirat Allah swt yang telah memberikan segala rahmat, taufik, hidayah, nikmat dan karunia-Nya, sehingga penulis dapat menyelesaikan karya yang berjudul "HTML & CSS For Educator". Shalawat beserta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW beserta keluarganya, para sahabatnya, dan para pengikutnya hingga akhir zaman.

Buku sederhana ini hadir untuk mempermudah proses PBM pada mata kuliah Web Programming. Dalam penyusunan buku ini, penulis menyadari sepenuhnya bahwa masih terdapat banyak kekurangan dan keterbatasan ilmu pengetahuan yang penulis miliki, Namun berkat adanya dorongan dan bantuan dari berbagai pihak, akhirnya karya sederhana ini dapat terselesaikan

Semoga buku sederhana ini dapat bermanfaat bagi penulis khususnya dan bagi para pembaca pada umumnya. Apabila terdapat kekurangan dan kesalahan adalah semata-mata keterbatasan ilmu yang penulis miliki. Apabila terdapat kesempurnaan itu berasal dari Allah SWT.

Bandung, 29 Mei 2020

PENULIS

DAFTAR ISI

KATA PENGANTAR.....	1
DAFTAR ISI.....	2
PROLOG.....	1
PERTEMUAN 1 Introduction.....	2
PERTEMUAN 2 Text Align.....	6
PERTEMUAN 3 Link Elemen.....	15
PERTEMUAN 4 Table.....	19
Rowspan dan Colspan.....	27
PERTEMUAN 5 Kontainerisasi.....	33
PERTEMUAN 6 Kontainerisasi II.....	36
PERTEMUAN 7 Form Elements.....	41
PERTEMUAN 8 Ordered & Unordered List.....	46
PERTEMUAN 9 Border Radius & Box Shadow.....	48
PERTEMUAN 10 Layouting.....	54
PERTEMUAN 11 Hover Effect.....	59
PERTEMUAN 12 Transition Effect.....	63
PERTEMUAN 13 Simple Persentation Page.....	66
PERTEMUAN 14 Box Sizing.....	73
PERTEMUAN 15 Flexbox.....	80
PERTEMUAN 16 Responsive Layout.....	88
PERTEMUAN 17 Animation.....	96
PERTEMUAN 18 Transform Concept.....	103
Transform 3D.....	108
PERTEMUAN 19 Button Slider.....	112
PERTEMUAN 20 Compile The HTML & CSS ?.....	117
EPILOG.....	128
REFERENSI.....	129

PROLOG

Perjalanan mendesain halaman web penuh dengan lika-liku sekaligus menyenangkan. Banyak metode yang dapat ditempuh dengan peralatan serupa atau bahkan berbeda dari satu metode dengan metode lain yang tersedia. Bahasa yang digunakan untuk untuk saling menampilkan data pada browser adalah HTML dengan protokol http.

Tujuan penggunaan pendekatan html dan http salah satunya agar data dapat dikirim per-bongkah, sehingga baik proses pengiriman maupun penerimaan dapat dilakukan lebih ringan terhadap sumber daya serta untuk menjaga kompatibilitas pada semua jenis platform yang saling tersambung.

Buku sederhana ini tidak membahas satu persatu akan elemen-elemen html maupun propertis-propertis css secara keseluruhan. Disini, pembaca disuguhkan materi ala “hot plate”, dalam artian setelah membaca mesti praktekan perlatihan yang terdapat pada setiap pertemuan (bab).

Setiap latihan melibatkan dua dokumen yakni dokumen html dan dokumen css. Adapun teknik penggunaan yang digunakan pada buku ini ialah teknik css secara eksternal (dokumen css terpisah dari html). Latihan seperti ini sangat sesuai bagi para pemula dan para edukator html-css agar peserta didik memiliki suatu kebiasaan dan kedisiplinan baik ketika memulai latihan html dan css.

PERTEMUAN 1 Introduction

HTML atau hypertext markup language adalah bahasa markup penstruktur konten website dimana semua konten website distrukturkan oleh html, sehingga setiap elemen (bagian) memiliki status node (titik hirarki) tertentu.

CSS atau cascading style sheet adalah porsi yang dikhkususkan untuk pengaturan serta desain konten-konten html. Css dapat dilakukan secara inline bersama code html atau secara internal terpisah secara tak langsung dari code html maupun secara eksternal (dokumen berbeda).

HTML & CSS dikelola oleh organisasi dunia seperti W3C dan WHATWG. Dua organisasi ini dapat dibilang “rumah” utama bagi html dan css. Meskipun demikian, banyak organisasi ataupun instansi yang memberi kontribusi untuk kelangsungan dan kemajuan html & css.

Organisasi diatas bertugas menggulirkan standar universal dan komprehensif atas penerapan html dan css. Instansi dan para developer yang nanti “menterjemahkan” standar tersebut ke dalam proyek mereka. Hal ini sangat terlihat pada dunia peramban web (browser).

Saat buku saku sederhana ini ditulis, html berada pada versi 5.2 sedangkan css pada versi 3.0. Html dan css versi selanjutnya adalah html 6.0 dan css 4.0. Pada setiap versi rilis terdapat perubahan dan penambahan fasilitas-infrastruktur baru seperti pada html 5.2 terdapat elemen canvas yang tidak tersedia pada versi sebelum 5.X.

Perlu juga dicamkan pada setiap rilis biasanya terdapat beberapa elemen html yang dianggap obsolete (kadarluarsa). Artinya sudah tidak dianjurkan untuk digunakan para developer web, bahkan ada yang mesti ditinggalkan karena sudah tidak berfungsi.

HTML memiliki struktur utama yang menjadi kerangka atas setiap konten yang dibubuhkan. Setiap bagian memiliki maksud tersendiri dan menjadi sebuah standar umum pengembangan web. Walaupun demikian, ada beberapa bagian yang “boleh” tak diikutsertakan pada code html dan tidak mengakibatkan fatal error alias halaman dan konten website tetap ter-render dengan baik pada browser.

Namun, hal ini tidak dianjurkan untuk tujuan yang sifatnya jangka panjang dan universal, gunakanlah standar html dan css dengan disiplin serta baik. Berikut contoh kerangka utama html:

```
<!DOCTYPE html>
<html>
<head>
    <title>latihan</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat.css">
</head>
<body>

</body>
</html>
```

Bagian-bagian pembangun html disebut dengan ‘tag’. Bagian yang mencakup tag buka dan tag tutup disebut sebagai ‘element’ html. Dengan kata lain elemen html adalah setiap bagian ber-tag buka dan tutup. Tag buka seperti : < > , tag tutup seperti : </ >.

Tag tutup sedikit berbeda dengan tag buka, ia memiliki blackslash ‘/’, ini menandakan akhir suatu elemen tersebut. Standar tag buka dan tutup

mesti diterapkan secara disiplin guna menghindari kesalahan yang berakibat merugikan render konten website pada browser.

Penjelasan struktur html:

<!DOCTYPE html>

Semacam ‘deklarasi’ bahwa ini adalah dokumen html, sehingga browser mengenali dokumen tersebut.

<html>

Tag buka html.

<head>

Tag buka bagian head. Bagian ini mirip dengan konsep kop surat pada kegiatan tulis-menulis. Head memiliki beberapa sub konten berupa link menuju dokumen lain, inisiasi device dengan ukuran layar berbeda, deklarasi unicode yang digunakan dan bagian lainnya.

<title>latihan</title>

element title bersungsi sebagai carier judul dokumen. Judul dokumen akan ditampilkan pada tab browser.

<meta charset="utf-8">

Tag meta dapat disebut informasi tambahan atas dokumen html. Salah satu tujuannya adalah menjaga kompatibilitas secara luas pada platform berbeda.

Utf adalah jenis encoding karakter yang digunakan. Utf-8 dianggap mewakili hampir seluruh kebutuhan karakter yang sering digunakan.

<link rel="stylesheet" href="lat.css">

tag <link> sesuai namanya untuk menyambungkan dokumen html dengan dokumen lain secara eksternal (dokumen terpisah). ‘rel’ disini adalah

relation artinya jenis hubungan pada dokumen target, 'href' adalah referensi lokasi dokumen target.

</head>
tag tutup dari head.

<body>

Tag buka elemen body, dimana disinilah setiap konten website dibubuhkan dengan memanfaatkan infrastruktur html tentunya. Ada banyak jenis elemen yang dapat dimanfaatkan. Buku ini tidak membahas semua element html. Silahkan merujuk pada sumber eksternal atas elemen-elemen html!

</body>
Tag tutup elemen body.

</html>
Tag tutup elemen html. Tanda bahwa dokumen html berakhir disini.

Persyaratan untuk melakukan latihan dan pengetesan html-css adalah sebagai berikut:

- ✓ Sistem operasi
- ✓ Browser (gunakanlah browser mainstream)
- ✓ Editor (kate, notepad++, dan lain-lain)
- ✓ Secangkir Kopi

Buku saku ini menggunakan pendekatan "hot on the plate" dalam artian ketimbang menjelaskan secara teoritikal setiap bagian html dan css, metode mengenali lalu menghafal setiap materi dalam praktik pada setiap pertemuan.

PERTEMUAN 2 Text Align

Seiring pertemuan per pertemuan, elemen-elemen HTML secara tak langsung diperkenalkan pada pembaca. Buku ini menggunakan pendekatan latihan sambil menghafal elemen HTML tersebut.

Text align ialah teknik perataan text dalam HTML melalui CSS. Mirip seperti pada aplikasi text processor, perataan text yang tersedia adalah rata kiri, rata kanan, rata tengah dan rata kiri-kanan. Tentu, setiap istilah atau kata kunci yang digunakan baik pada HTML maupun CSS menggunakan bahasa Inggris.

Langkah-langkah pertemuan 2:

1. nyalakan komputer
2. install browser dan editor pilihan anda
3. buka browser dan editor tersebut, biarkan standby
4. siapkan struktur utama HTML, ketik pada editor lalu save dengan nama lat1.html. Beri title "Latihan 1", isi href pada tag <link> dengan "lat1.css".

struktur HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 1</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat1.css">
</head>
<body>

</body>
</html>
```

Langkah-langkah diatas selalu berlaku sampai halaman terakhir buku saku ini ! Seiring waktu, ada beberapa point yang mesti dibiasakan pada diri pembelajar, mereka adalah :

- ✓ biasakan untuk selalu menggunakan lowercase/huruf kecil dalam pengetikan nama elemen html
- ✓ biasakan memberi title sederhana, namun mewakili projek
- ✓ biasakan melakukan penjorokan/indentasi pada sub-sub elemen html
- ✓ biasakan mengikutsertakan ekstensi file “.html” dan “.css” saat melakukan save/save as. Setiap latihan selalu melibatkan dua dokumen tersebut.
- ✓ Biasakan membuat folder khusus per pertemuan/projek

langkah ke-5 adalah mengisi bagian <body> dengan konten berupa text menggunakan elemen <h> dan <p>, p = paragraf, h = header. Elemen <h> digunakan untuk Judul konten dan auto bold, dia memiliki enam size, dimulai dari paling besar <h1> sampai terkecil (ukuran standar text 16px) <h6>. Elemen <p> sesuai dengan semantika atas p = paragraf adalah untuk keperluan konten paragraf.

Ketik ini pada bagian <body> :

```
<h1>Pertemuan 2</h1>
<h3>Latihan 1</h3>

<p>Latihan pertama html-css Latihan pertama html-cssLatihan pertama
html-cssLatihan pertama html-cssLatihan pertama html-cssLatihan
pertama html-cssLatihan pertama html-cssLatihan pertama html-
cssLatihan pertama html-cssLatihan pertama html-cssLatihan pertama
html-cssLatihan pertama html-cssLatihan pertama html-cssLatihan
pertama html-cssLatihan pertama html-cssLatihan pertama html-
cssLatihan pertama html-css</p>
```

Pada titik ini, kita telah memiliki struktur html sebagai berikut:

Notes:

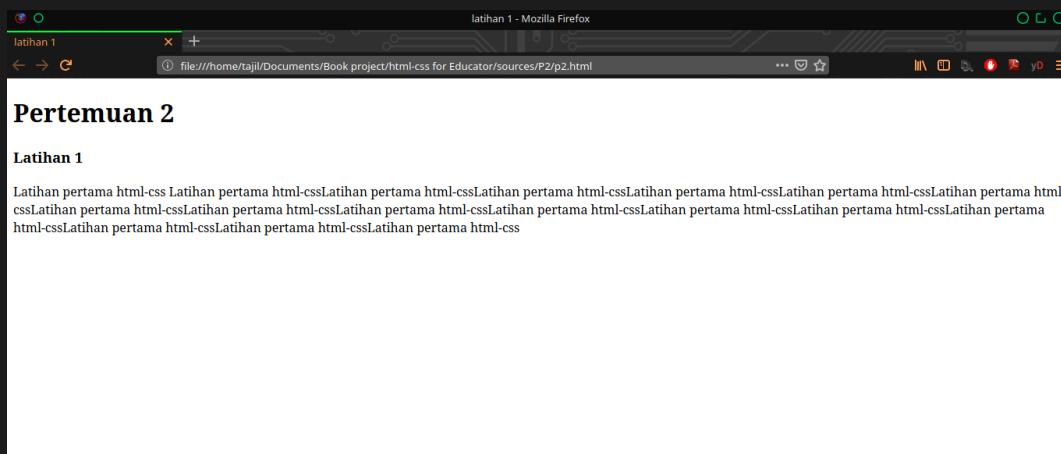
Konten atas elemen <h>, <p> dan elemen lain berada diantara tag bukan dan tag tutup elemen tersebut !

Indentasi sub elemen harus sedikit kedalam, agar lebih mudah untuk pengenalan dan pengecekan keperluan kedepan.

Langkah ke-6, menampilkan pada browser. Terdapat beberapa teknik dalam membuka file html pada browser:

- buka browser, ctrl+O, arahkan pada dokumen lat1.html
 - klik dua kali pada lat1.html, akan terbuka langsung di browser

Figure lat1.html pada browser:



Notes:

- Secara default setiap elemen ditempatkan mulai dari pojok kiri atas. Konten pada elemen <h> atau <p> secara otomatis membuat baris baru ketika mencapai akhir sisi jendela kanan.
 - Mayoritas elemen html memberi garis baru otomatis atas elemen lain.
 - Pemberian spasi kosong pertama akan di-render browser, spasi selanjutnya berapapun banyaknya akan diabaikan. Begitu pula dengan garis baru secara manual akan diabaikan. Ada beberapa elemen yang tidak memberi garis baru secara default, seperti elemen link: <a>. Elemen ini memiliki properti (sifat) inline, sedangkan elemen yg memberi garis baru otomatis disebut block elemen.
 - Elemen yang di-render browser adalah elemen yang ditulis lebih awal, sehingga mana yang ditulis paling awal pada editor, dialah yang akan dirender terlebih dahulu mirip seperti konsep antrian.

Langkah ke-7 adalah menambahkan file tipe gambar (jpeg, png atau gif boleh mana saja). Siapkan sebuah file gambar, beri nama sesimpel mungkin, contoh xyz.png. Dalam hal embed atau attachment file gambar dan dokumen eksternal lainnya, hal krusial pertama adalah menuliskan dengan benar nama dan lokasi file-file tersebut. Oleh karena itu,

sebaiknya buat simpel nama dan lokasi file tersebut agar memudahkan dalam coding dan pengecekan kedepan apabila diperlukan. Dari sisi organisasi file juga terlihat lebih rapi dan profesional-like.

Berikut ini ialah step-step yang mesti dilalui :

1. Buat folder dilokasi file lat1.html berada, beri nama “image”
2. Masukan file gambar kita pada folder tersebut.
3. Tambahkan code berikut pada konten lat1.html kita :
``
4. save perubahan pada editor, lalu refresh browser
refresh browser dapat dilakukan dengan click tombol refresh pada browser atau menekan F5

struktur code html saat ini:

```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 1</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat1.css">
</head>
<body>

    <h1>Pertemuan 2</h1>
    <h3>Latihan 1</h3>

    <p>Latihan pertama html-css Latihan pertama html-cssLatihan pertama
    html-cssLatihan pertama html-cssLatihan pertama html-css</p>

</body>
</html>
```

Notes:

- “src” adalah source atau sumber file target, buuhkan alamat lengkap dan nama file lengkap dengan ekstensinya.
- “alt” adalah alternate alias alternatif yang akan muncul apabila terjadi problem saat menampilkan file gambar pada “src”.
- Ukuran gambar pada browser sesuai dengan ukuran native (asli) file tersebut. Untuk dapat mengontrol ukuran gambar, kita harus mengaturnya dengan CSS seperti pada langkah dibawah.

Mari kita atur ukuran gambar yang telah kita sertakan pada elemen .

1. Pertama, pada editor, buat dokumen baru, file-new atau cara lain silahkan.
2. Save dengan nama “lat1.css”, simpan tepat disamping dokumen lat1.html.
3. Sertakan nama “lat1.css” dibagian <head> pada elemen <link> seperti berikut:

```
<link rel="stylesheet" href="lat1.css">
```

Setiap ada perubahan sekecil apapun pada editor, jangan lupa untuk melakukan “save” !! lalu refresh browser untuk melihat perubahan yang dilakukan.

Dengan begini, kita telah menghubungkan dokumen lat1.html dengan dokumen lat1.css. Metode yang digunakan disini adalah CSS secara eksternal.

4. Untuk dapat mengatur propertis suatu elemen html, kita mesti “menargetkan-nya” terlebih dahulu. Hal ini dapat dilakukan dengan beberapa cara. Dalam buku ini pembiasaan yang dilakukan adalah metode pemberian “class” dengan css secara eksternal.

Target kita adalah elemen , maka beri class elemen ini dengan nama “gambar” seperti berikut:

```

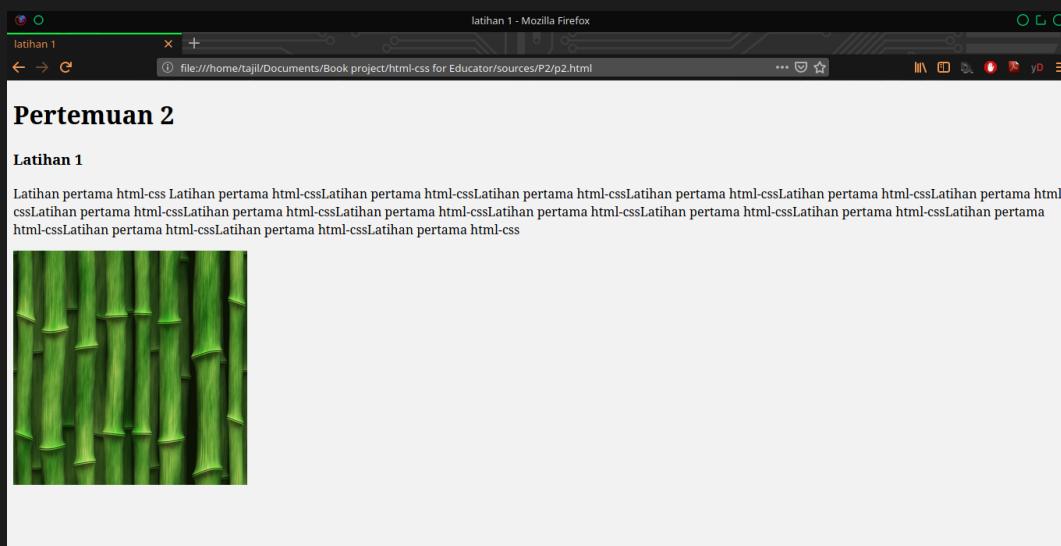
```

5. Sekarang kita beralih pada dokumen lat1.css. Panggil class yang menjadi target, sesuai dengan nama class pada html dan penulisan pada dokumen css dimulai dengan titik (.). Berikut contoh code css:

```
.gambar {  
    width : 300px;  
    height : 300px;  
}
```

save dokumen css, lalu refresh browser!

Tampilan html pada browser :

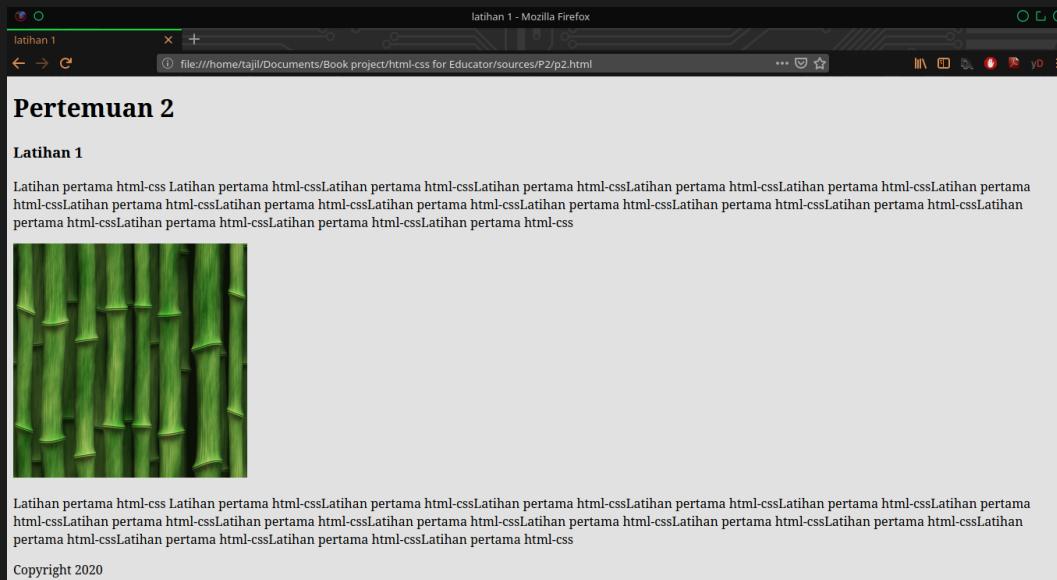


Notes:

- Hati-hati dengan salah ketik alias typo, dapat mengakibatkan code tidak berfungsi atau bahkan membuat elemen lain terganggu. Selalu cek and recheck segala kesalahan apabila tampilan pada browser belum sesuai dengan perubahan pada editor.
- Perhatikan jenis file gambar kita, cek dahulu ekstensinya apakah jpg, png atau lain-lain.

- Struktur css dimulai dengan titik (.) dalam penggunaan class, kemudian propertis dan value (nilai) atas propertis berada dalam kurang kurawal.
- Akhir dari setiap propertis dibatasi dengan titik koma (;
- class yang sama dapat digunakan berulang-ulang serta dapat dibubuhkan pada lebih dari satu elemen html selama memiliki pengaturan propertis elemen yang sama. Apabila berbeda pengaturan maka membuat class dengan nama baru.

Langkah ke-8 adalah menambahkan satu paragraf persis dibawah gambar diatas, lalu sedikit text untuk bagian footer, sehingga hasilnya seperti berikut:



Langkah ke-9 text align

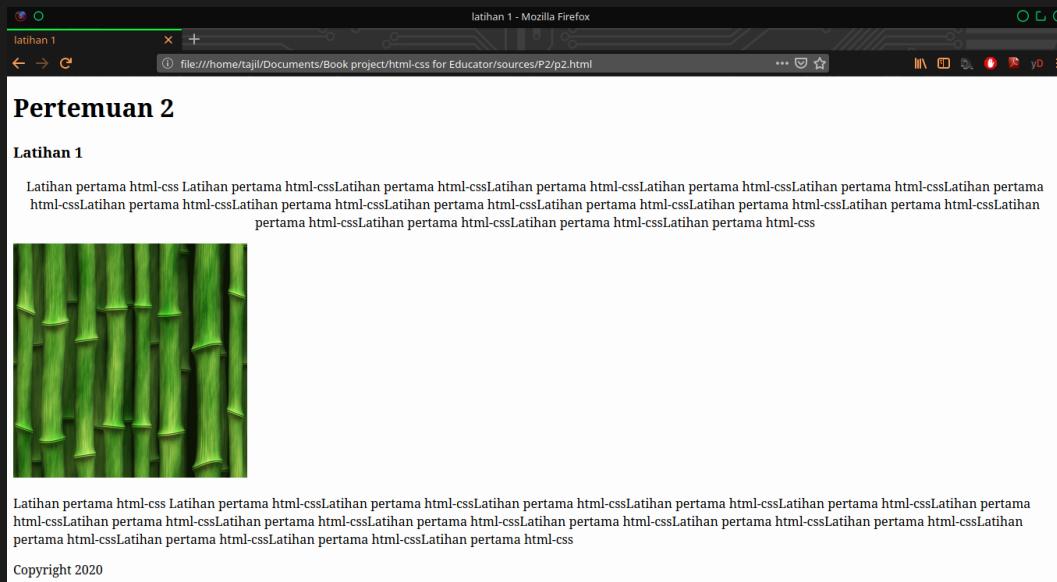
Akhirnya kita sampai disni, step-step diatas memang mesti dilalui sebagai langkah awal. Text align alias perataan text terdiri atas empat macam: left, right, center dan justify. Mari terapkan hal tersebut sebagai berikut:

Pertama, tentukan target elemen mana yang akan diberi text align ? Tentu target kita adalah elemen <p>, saat ini terdapat tiga elemen <p> pada html tergantung pada yang mana mau diberi perataan arah mana. Misal kita

beri class dengan nama “paragraf” pada elemen `<p>` pertama, kemudian buat css untuk `<p>` tersebut pada dokumen lat1.css tepat dibawah class gambar seperti berikut:

```
.gambar {  
    width   : 300px;  
    height  : 300px;  
}  
  
.paragraf {  
    text-align : center;  
}
```

save editor dan refresh browser. Tampilan saat ini:



Notes:

- ✓ Pelajari dan pahami setiap langkah pada pertemuan dua ini, karena hal tersebut berlaku sampai pertemuan terakhir.
 - ✓ Pelajari elemen-elemen html pada sumber-sumber lain, lalu coba praktikan!
 - ✓ Ssalah satu sumber gratis dan komprehensif terdapat pada www.w3cshcool.com

PERTEMUAN 3 Link Elemen

Halaman website seperti halnya buku dapat terdiri atas lebih dari satu halaman. Pada html elemen link atau [alias anchor berperan penting dalam menghubungkan berbagai halaman html.](#)

Penggunaan elemen [sangat mudah, persis seperti elemen dimana kita harus mempersiapkan dokumen lain lalu membubuhkan nama dan alamat dokumen lain tersebut pada elemen bersangkutan.](#)

Contoh :

```
<a href="lat2b.html">Halaman 2</a>
```

Notes:

- href adalah sumber dokumen lain berupa nama+ekstensi file beserta alamat apabila berbeda folder atau bahkan berbeda direktori.
- “Halaman 2” adalah label elemen [tersebut, yang mana akan ditampilkan pada browser berupa teks “Halaman 2” yang dapat diklik.](#)
- Ketika label tersebut diklik user, user diarahkan pada dokumen yang tertera pada bagian “href”.

Langkah-langkah pertemuan 3 :

1. Pertama, buka editor
2. Kedua, buat tiga dokumen (lat2.html, lat2b.html dan lat2.css), simpan pada folder khusus misal folder dengan nama “Latihan 2”.
3. Ketiga, pada dokumen lat2.html dan lat2b.html buatlah struktur utama html terlebih dahulu. Jangan lupa sering-sering melakukan “save” walau perubahan sekecil apapun!.

4. Keempat, setiap kali kita membuat dokumen html dan css baru, maka disaat yang sama kita harus menyesuaikan bagian <title> dan href dalam <link> pada bagian <head>.
 5. Kelima, isi bagian body dengan beberapa konten, hindari copy-paste atas code html dan css bagi kalian yang baru belajar. Agar lekas paham dan lekas mahir dalam materi basic pada buku sederhana ini !!! berlelah-lelah dahulu, kemudian barulah mengecap manisnya “output”. Bedakan sedikit isi konten dokumen lat2.html dengan lat2b.html.
 6. Keenam, pada dokumen lat2.html dibagian paling bawah konten tambahkan elemen <a> seperti berikut:

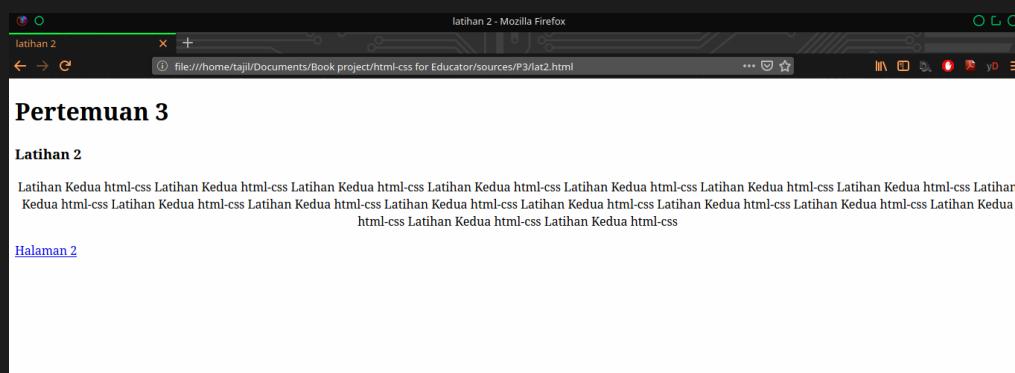
[Halaman 2](lat2b.html)

Contoh konten dokumen lat2.html :

Notes:

Biasakan penjorokan/indentasi atas sub-sub elemen agar terlihat rapi dan dapat lebih mudah dibedakan mana parent mana child (sub elemen). Indentasi dapat dilakukan dengan spasi atau tab.

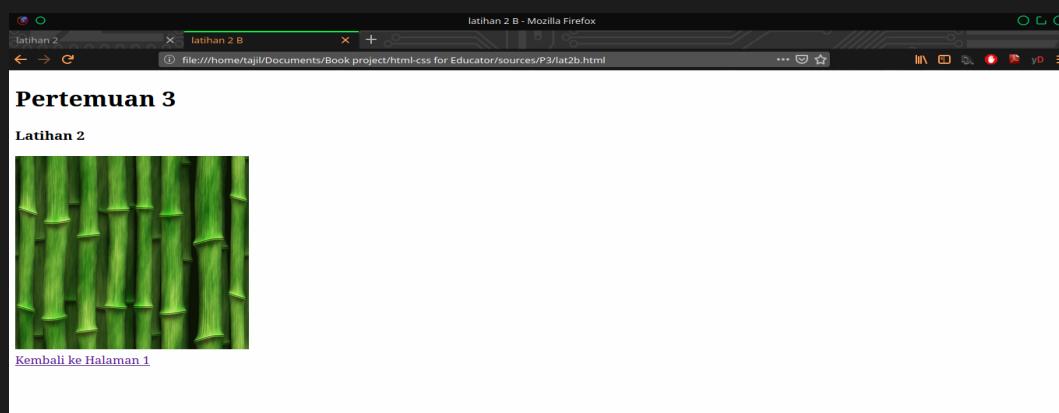
Tampilan pada browser:



Adapun konten dokumen lat2b.html adalah sebagai berikut:

```
lat2b.html — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat2b.html lat2.css lat2.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 2 B</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat2.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 3</h1>
11  <h3>Latihan 2</h3>
12
13
14  
15
16
17  <br/>
18  <a href="lat2.html">Kembali ke Halaman 1</a>
19
20 </body>
21 </html>
22
23
```

Tampilan pada browser:



Notes:

Pada dokumen lat2.html kita melabeli elemen <a> disini dengan “Halaman 2” dan mengarahkan href pada lat2b.html. Sedangkan pada dokumen lat2b.html dilabeli “Kembali ke Halaman 1” serta menset href menuju lat2.html. Ini merupakan konsep dasar dalam meracik tombol next dan back.

Ketujuh, buka dokumen lat2.html di browser, lalu test apakah link (<a>) sudah terkonfigurasi dengan benar dengan cara klik link “Halaman 2”, kemudian apabila sudah benar tinggal klik link kembali ke halaman 1. Apabila belum berjalan sesuai dengan rencana, cek n ricek apabila ada kesalah ketik (typo), kurang tag tutup suatu elemen, ketiadaan titik koma pada css, kurang kurung kurawal pada css, perubahan pada dokumen yang ada belum di “save” atau bahkan browser belum di re-refresh setelah terjadi perubahan pada dokumen.

PERTEMUAN 4 Table

Html memiliki fasilitas tabel untuk tujuan tabulasi data. Dahulu, sebelum CSS3 rilis, para developer dan web disainer banyak yang memanfaatkan tabel html untuk kepentingan layouting website. Tabel tujuannya adalah untuk tabulasi data, bukan untuk layouting atau dekorasi halaman website.

Cara tersebut sah-sah saja walaupun kurang sejalan dengan semantikan elemen tabel pada html. Kini dengan CSS3 telah hadir, keperluan layouting dan dekorasi telah dipenuhi oleh css. CSS3 bahkan menyuguhkan serangkaian protokol animasi sehingga menyuguhkan tampilan muka lebih menarik dan hidup.

Untuk mempelajari html dan css secara mendalam hendaklah pembaca mencari dan mempelajari pada sumber lain. Buku ini tidak membahas kedua hal tersebut secara menyeluruh dan mendalam.

Tabel terdiri atas kolom dan baris. Kombinasi kolom dan baris inilah yang membentuk tabel. Tabel (`<table>`) merupakan salah satu elemen pada html yang juga terdiri atas baris (`<tr>`) dan kolom (`<row>`).

Langsung kita ke TKP dengan contoh dibawah:

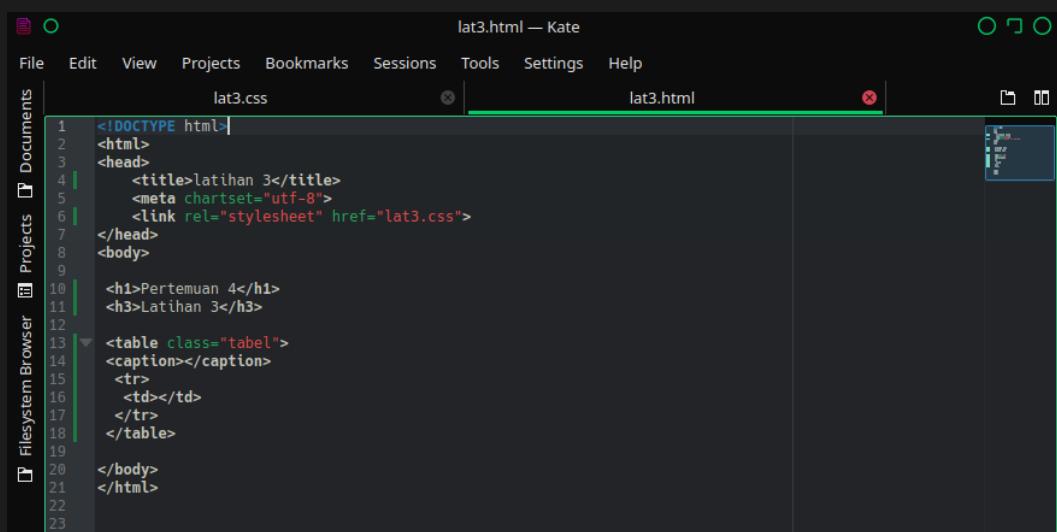
```
<table>
<caption></caption>
<tr>
  <td></td>
</tr>
</table>
```

Elemen tabel terdiri atas tag table buka dan tag tabel tutup, didalamnya terdapat tag `<tr>` sebagai baris dan `<td>` sebagai kolom. Adapun `<caption>` berfungsi sebagai label tabel tersebut. Caption dapat ditempatkan diatas (default) atau dipaling bawah sebelum tag penutup tabel.

Step by step pertemuan 4 ini adalah sebagai berikut:

1. Pertama, seperti biasa siapkan browser dan editor
2. Kedua, buat dua dokumen baru; lat3.html dan lat3.css
3. Ketiga, buat struktur utama html pada lat3.html
4. Keempat, sesuaikan konten pada <title> dan href pada <link> dibagian <head>.
5. Kelima, bubuhkan elemen tabel seperti pada contoh diatas dibagian <body>
6. save, lalu buka di browser

code html:



The screenshot shows the Kate text editor interface. On the left, there's a sidebar with tabs for 'Documents', 'Projects', and 'Filesystem Browser'. The 'Documents' tab is active, showing two files: 'lat3.css' and 'lat3.html'. The 'lat3.css' file contains the following CSS code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 3</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat3.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 4</h1>
11  <h3>Latihan 3</h3>
12
13  <table class="tabel">
14    <caption></caption>
15    <tr>
16      <td></td>
17    </tr>
18  </table>
19
20 </body>
21 </html>
22
23
```

The 'lat3.html' file is currently selected and shows the following HTML code:

```
<h1>Pertemuan 4</h1>
<h3>Latihan 3</h3>

<table class="tabel">
<caption></caption>
<tr>
<td></td>
</tr>
</table>
```

Bagaimana hasilnya ?

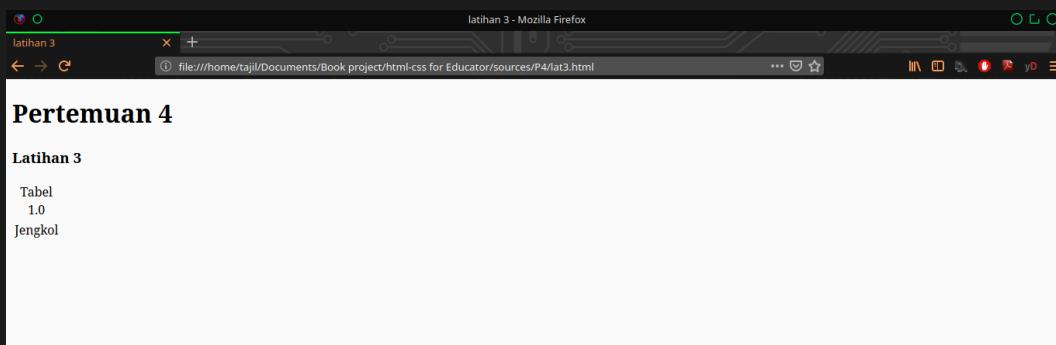
Pasti tidak tampil sama sekali tabelnya bukan?

Ada dua faktor, pertama tabel tidak akan dirender apabila belum memiliki konten, walaupun sudah kita ketik pada bagian <body>. Untuk itu, kita mesti memberi konten tabel yakni pada bagian caption dan td seperti berikut :

The screenshot shows the Kate text editor interface. On the left, there's a sidebar with 'Documents', 'Projects', and 'Filesystem Browser'. The main area has two tabs: 'lat3.css' and 'lat3.html'. The 'lat3.css' tab contains the following CSS code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 3</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat3.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 4</h1>
11  <h3>Latihan 3</h3>
12
13  <table class="tabel">
14    <caption>Tabel 1.0</caption>
15    <tr>
16      <td>Jengkol</td>
17    </tr>
18  </table>
19
20 </body>
21 </html>
22
23
```

Nah, sekarang coba save lalu refresh browser. Tampilan pd browser :



Faktor kedua adalah tentang border. Border aritnya garis pembatas. Semua elemen html memiliki border masing-masing, akan tetapi secara default tidak ditampilkan. Dalam hal ini tabelpun tidak terkecuali. Oleh karena itu dibutuhkan CSS untuk mengatur properti tabel kita.

Berikut langkah yang mesti dilakukan:

Beri elemen table class dengan nama "tabel". Hindari penamaan yang rumit atau kata kunci bawaan dari html dan css!!!

Buat css untuk tabel pada dokumen lat3.css sebagai berikut:

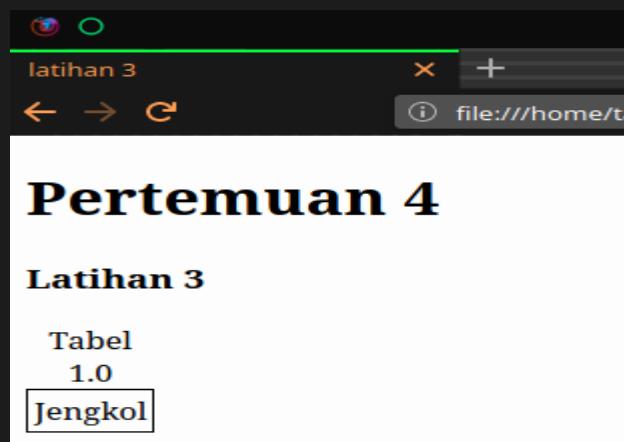
The screenshot shows the Kate text editor interface. On the left, there's a sidebar with 'Documents' and 'Projects'. The main area has two tabs: 'lat3.css' and 'lat3.html'. The 'lat3.css' tab contains the following CSS code:

```
1 .table {  
2     border : 1px solid;  
3 }  
4  
5
```

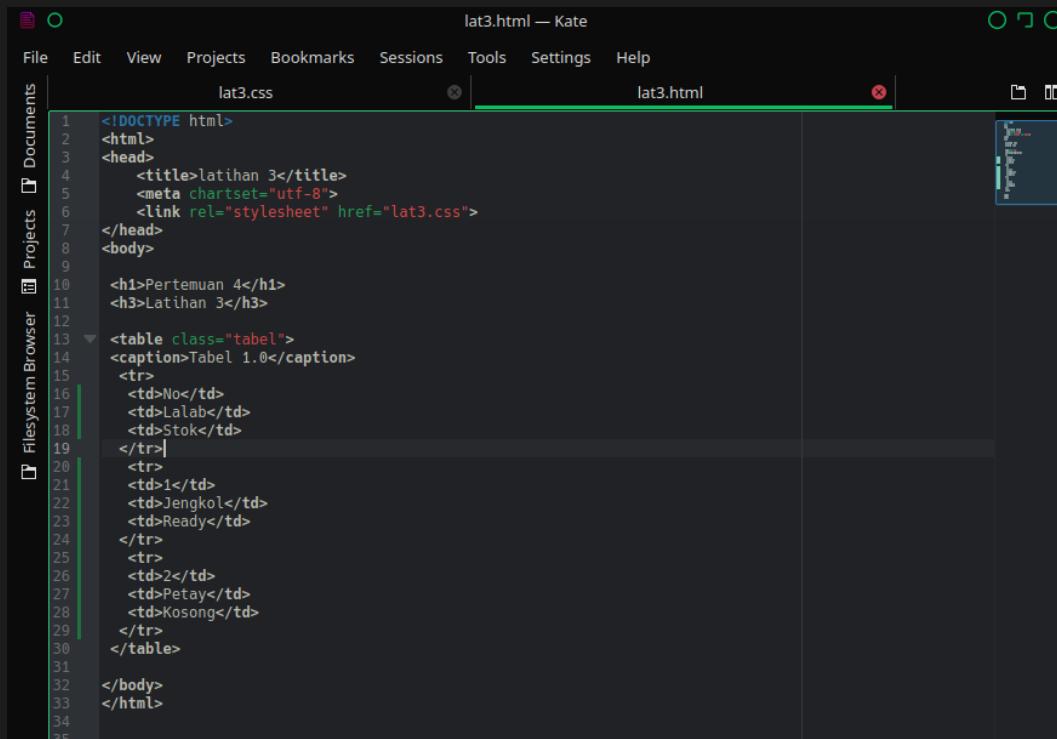
Notes:

- Properti border adalah untuk mengatur sifat-sifat border elemen html. Dalam hal ini kita menentukan khusus untuk tabel dalam class “tabel”. Properti “border” memiliki tiga syntax, masing-masing dipisah dengan spasi (wajib memakai spasi antara).
- border : 1px solid color;
- “1px” artinya ketebalan garis setebal 1 pixel, semakin besar angka semakin tebal garis.
- “solid” adalah style border dan wajib dicantumkan. Solid adalah tipe garis tidak terputus. Silahkan pelajari lebih lanjut soal style border secara mandiri.
- “color” adalah warna dari garis. Disini kita bebas menggunakan warna apaja dan penerapannya dengan bahasa Inggris, contoh: red,blue,green dan lainnya. Bagian ini tidak wajib dicantumkan (seperti pada contoh tabel yang diberikan diatas), karena secara otomatis akan dipilihkan warna default, yakni hitam (black).

Hasil:



Berikutnya, mari kita tambahkan jumlah kolom dan baris agar benar-benar terlihat seperti tabel pada lazimnya juga melakukan penyesuaian konten kolom pada baris pertama dengan memberi konten “No”, “Lalab” dan “Stok” masing-masing pada kolom pertama, kedua dan ketiga tersebut sebagai berikut:



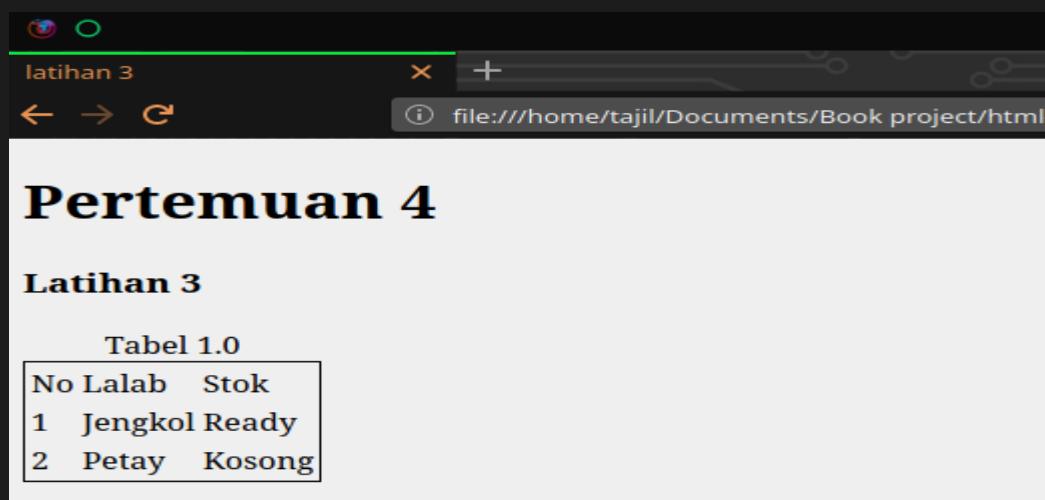
The screenshot shows the Kate text editor interface. The menu bar includes File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, and Help. The title bar says "lat3.html — Kate". There are two tabs open: "lat3.css" and "lat3.html". The "lat3.html" tab contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 3</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat3.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 4</h1>
11  <h3>Latihan 3</h3>
12
13  <table class="tabel">
14    <caption>Tabel 1.0</caption>
15    <tr>
16      <td>No</td>
17      <td>Lalab</td>
18      <td>Stok</td>
19    </tr>
20    <tr>
21      <td>1</td>
22      <td>Jengkol</td>
23      <td>Ready</td>
24    </tr>
25    <tr>
26      <td>2</td>
27      <td>Petay</td>
28      <td>Kosong</td>
29    </tr>
30  </table>
31
32  </body>
33 </html>
34

```

Tampilan pada browser:

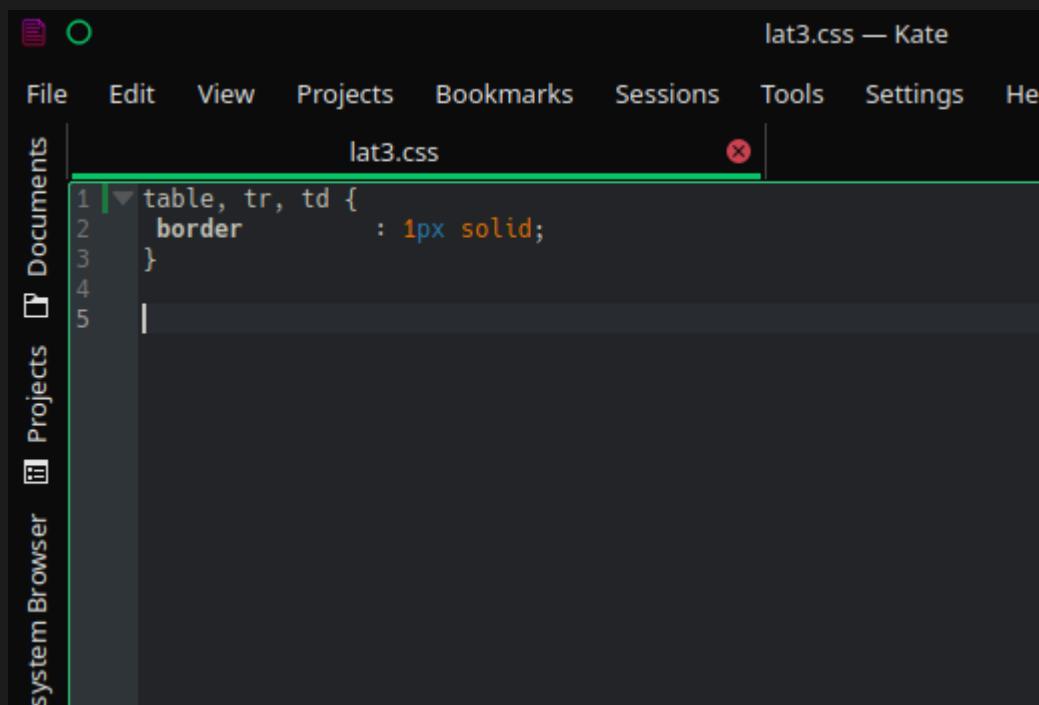


Perhatikan tampilan pada browser diatas, apa yang dirasa kurang ?

Ketidaaan sekat antar kolom dan baris yang belum tersedia. Oleh karena itu, kita mesti mengatur css tabel pada dokumen css untuk menampilkan border setiap <tr> dan <td>.

Sebelumnya kita telah membuat class “table” dan menampilkan border setebal 1 pixel bertipe solid, kini kita dapat menghapus class tersebut. Penerapan css tidak melulu melalui class, css dapat diterapkan dengan memanggil langsung “nama” elemen target.

Perhatikan contoh css dibawah:



The screenshot shows the Kate text editor interface. The title bar says "lat3.css — Kate". The menu bar includes File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, and Help. On the left, there's a vertical sidebar with icons for Documents, Projects, and System Browser. The main area shows the CSS code:

```
1 ▼ table, tr, td {  
2   border      : 1px solid;  
3 }  
4  
5 |
```

Memanggil nama elemen secara langsung tidak perlu dimulai dengan titik, karena ini bukan class, tapi elemen html. Tampilan pada browser:

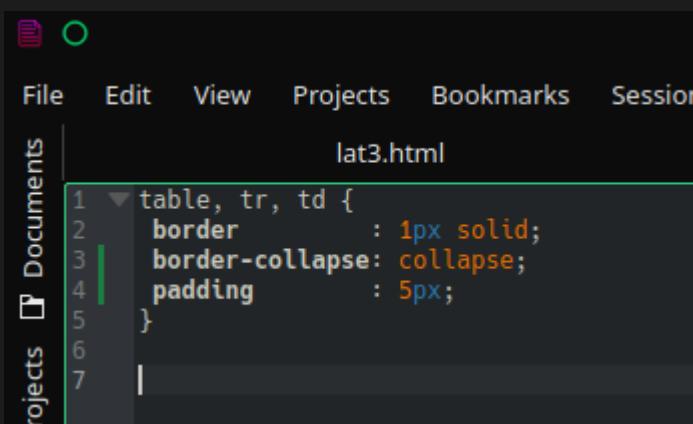


Tabel 1.0 diatas terlihat jelas setiap bagian kolom memiliki border tersendiri sehingga terkesan tidak menyatu sebagai suatu tabel. Untuk itu diperlukan finishing touch pada css tabel tersebut dengan menambahkan beberapa properti tambahan seperti padding dan border collapse.

Notes:

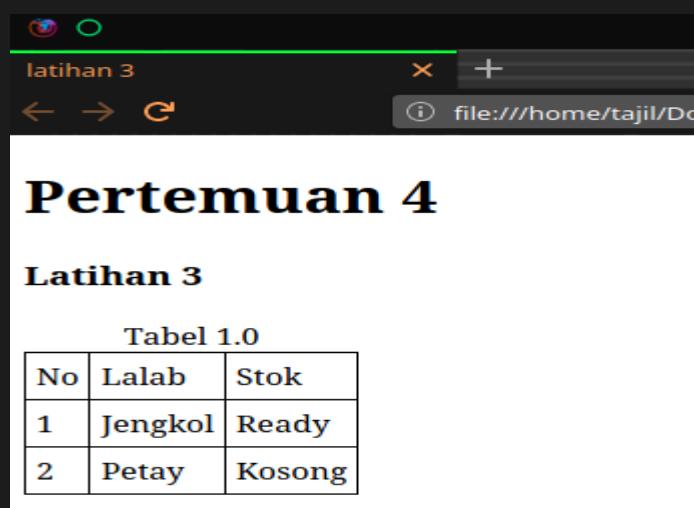
- Padding adalah batas konten terhadap bordernya masing-masing.
- border-collapse adalah properti bagi tabel apakah border yang ditampilkan collapse (disatukan) atau tidak collapse seperti tampilan tabel 1.0 diatas, dimana setiap kolom terpisah dan memiliki border tersendiri, ini merupakan properti default pada tabel.

Code CSS tabel :



```
lat3.html
File Edit View Projects Bookmarks Sessions
Documents
1 <table, tr, td {
2   border          : 1px solid;
3   border-collapse: collapse;
4   padding         : 5px;
5 }
6
7 |
```

Hasil pada browser :

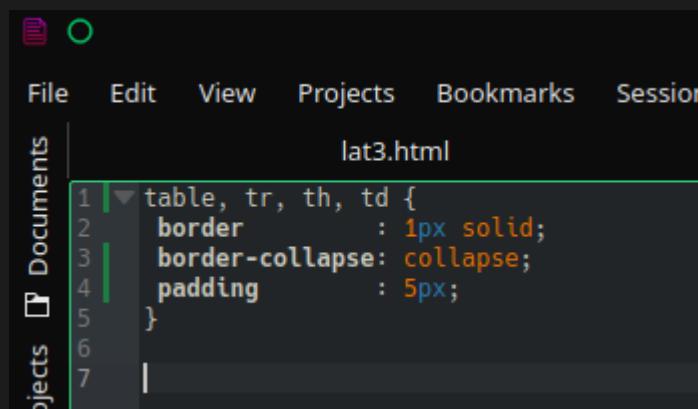


The browser window shows the title "Latihan 3" and the URL "file:///home/tajil/Do". The main content area displays the following table:

No	Lalab	Stok
1	Jengkol	Ready
2	Petay	Kosong

Notes:

- ✓ Penggunaan nama class css atau secara langsung nama elemen seperti diatas dapat dibubuhkan lebih dari satu pada propertis yang sama. Pada css tabel diatas kita membubuhkan tiga bagian berbeda (table, tr, td) namun dengan isi css (propertis) yang sama. Perlu dicamkan, menggunakan metode seperti ini harus menggunakan tanda koma sebelum membubuhkan elemen atau bagian elemen lain.
- ✓ Tabel pada html memiliki bagian khusus untuk digunakan sebagai header tabel. Apabila kolom disimbolkan dengan tag <td>, maka untuk bagian header gunakan <th> sebagai unit pembangun kolom baris header tabel pada dokumen html. Jangan lupa sertakan juga 'th' pada css tabel agar memiliki propertis yang sama dengan 'tr' dan 'td' Perhatikan contoh dibawah:



A screenshot of a code editor window titled "lat3.html". The left sidebar shows a tree view with "Documents" expanded, showing "lat3.html" and "objects". The main area contains the following CSS code:

```
1 table, tr, th, td {  
2     border          : 1px solid;  
3     border-collapse: collapse;  
4     padding         : 5px;  
5 }  
6  
7 |
```

Hasil pada browser:



The browser displays the following content:

Latihan 3

Tabel 1.0

No	Lalab	Stok
1	Jengkol	Ready
2	Petay	Kosong

Rowspan dan Colspan

Keperluan tabulasi memang bermacam-ragam, sehingga dibutuhkan tabel dengan beberapa kolom dimerger atau beberapa baris dimerger atau bahkan terdapat kedua hal tersebut dalam satu tabel. Dalam hal inilah rowspan dan colspan unjuk gigi, rowspan menyatukan dua atau lebih baris <tr> sedangkan colspan menyatukan dua atau lebih kolom <td>.

Mari kita praktekan kedua hal tersebut dengan persiapan sebagai berikut:

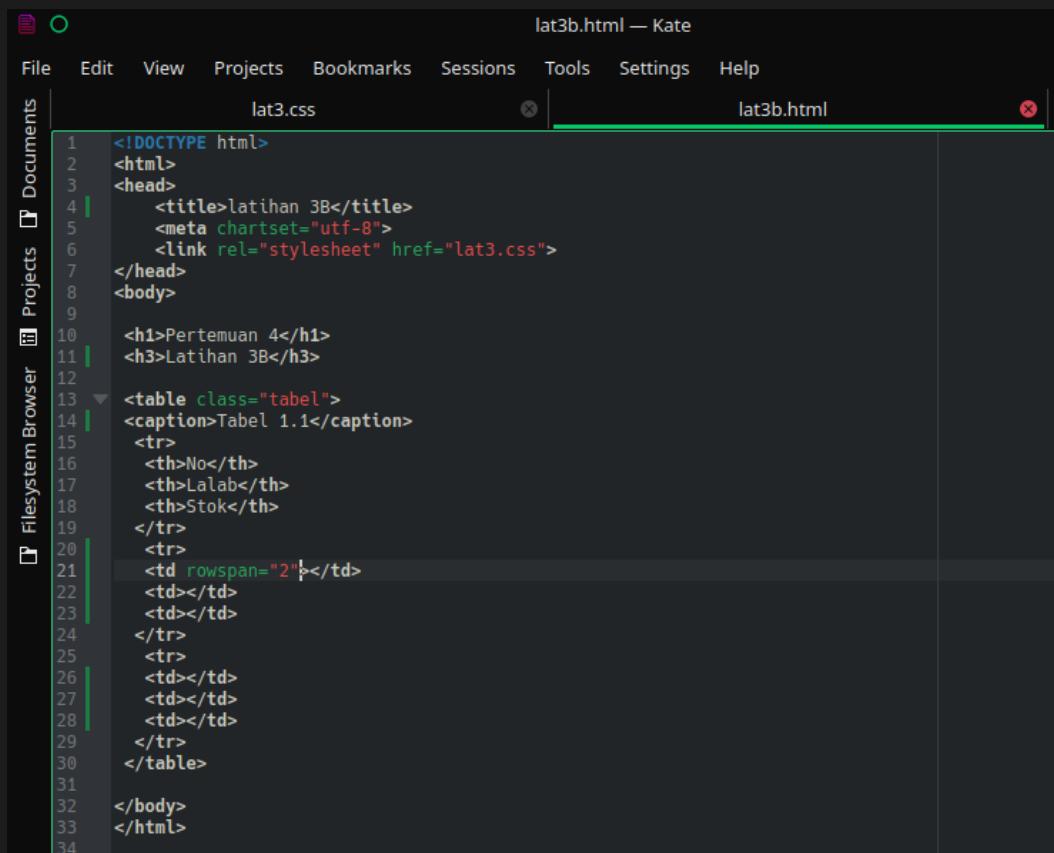
1. File new pada editor, save sebagai “lat3b.html”
2. Buat struktur utama html
3. Buat tabel dengan spesifikasi 3x3 (tiga baris dan tiga kolom tiap baris)
4. Ingat! Ketika isi kolom pada baris pertama belum diberi konten, tabel tidak akan di render memanjang-luas, namun berdempet tipis-tipis. Untuk itu, beri konten apa saja tiga kolom pada baris pertama agar tabel terlihat. Baris kedua dan ketiga tidak perlu diberi konten.
5. Link-kan lat3b.html dengan dokumen css sebelumnya (lat3.css).
6. Tampilkan dibrowser, apabila belum tampil, cek n ricek segala yang mungkin seperti terdapat typo, salah membuka dokumen html yang dimaksud, belum di save pada editor, belum di refresh pada browser dan lain sebagainya. Contoh tampilan pada browser:

No	Lalab	Stok

Tabel diatas terdiri atas 3 baris dan setiap baris terdiri atas 3 kolom, dengan singkat disebut dengan tabel 3x3. Target kita untuk latihan rowspan dan colspan ada pada baris kedua dan ketiga.

Khusus rowspan dan colspan akan kita lakukan secara inline. Css inline artinya konfigurasi css berada bersama pada elemen html.

Pertama, kita lakukan rowspan, diterapkan pada <td> di baris kedua (<tr> kedua pada tabel), sehingga code html saat ini adalah sebagai berikut:



The screenshot shows the Kate text editor interface. On the left, there's a sidebar with tabs for 'Documents', 'Projects', and 'Filesystem Browser'. The main area has two tabs: 'lat3.css' and 'lat3b.html'. The 'lat3b.html' tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>latihan 3B</title>
<meta charset="utf-8">
<link rel="stylesheet" href="lat3.css">
</head>
<body>

<h1>Pertemuan 4</h1>
<h3>Latihan 3B</h3>

<table class="tabel">
<caption>Tabel 1.1</caption>
<tr>
<th>No</th>
<th>Lalab</th>
<th>Stok</th>
</tr>
<tr>
<td rowspan="2">1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</table>

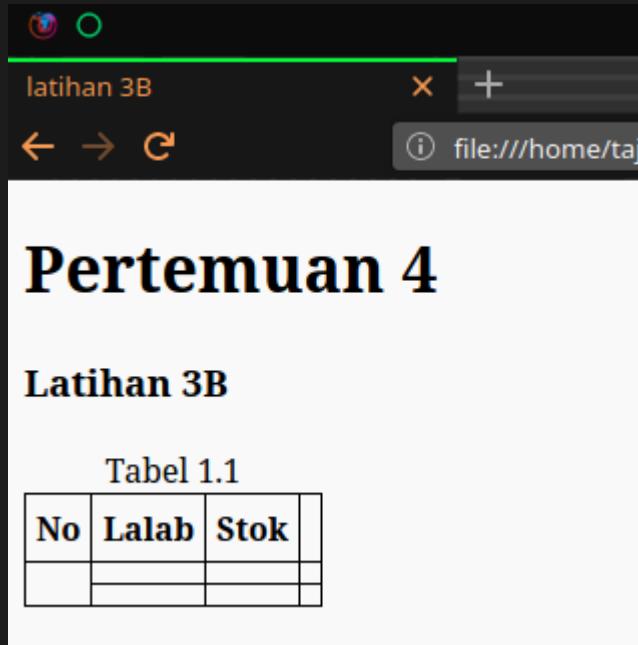
</body>
</html>
```

Notes:

rowspan = 2 berarti merger dua baris kebawah. Pada kasus struktur table diatas <td> pada baris kedua dan ketiga akan dimerger.

Proses merger baik baris atau kolom berarti memakan ruang lain. Aritnya merger baris berarti menggunakan baris lain untuk digabung, merger kolom berarti menggunakan kolom lain untuk digabung. Dengan kata lain, setelah menerapkan rowspan dan colspan, kita harus menyesuaikan struktur tabel pada html.

Berikut tampilan setelah diberi rowspan = 2 pada baris kedua:



Perhatikan! Rowspan =2 membuat <td> baris kedua juga menggunakan <td> dibawahnya. Hal tersebut membuat struktur baris ketiga (<tr> ketiga) bergeser ke samping menjadi kolom keempat. Maka, apabila yang dibutuhkan adalah tabel 3x3 dengan kolom pada baris kedua dan ketiga merger, hapuslah kolom (<td>) yang tidak diperlukan.

Tampilan code html setelah disesuaikan :

A screenshot of a code editor titled "lat3b.html — Kate". The left sidebar shows "Projects" and "Documents". The main pane displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 3B</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat3.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 4</h1>
11  <h3>Latihan 3B</h3>
12
13  <table class="tabel">
14    <caption>Tabel 1.1</caption>
15    <tr>
16      <th>No</th>
17      <th>Lalab</th>
18      <th>Stok</th>
19    </tr>
20    <tr>
21      <td rowspan="2"></td>
22      <td></td>
23      <td></td>
24    </tr>
25    <tr>
26      <td></td>
27      <td></td>
28    </tr>
29  </table>
30
31  </body>
32 </html>
```

The code editor interface includes tabs for "lat3.css" and "lat3b.html", and a status bar at the bottom.

Tampilan pada browser:



Berikutnya, kita merger kolom kedua dan ketiga pada baris kedua dengan membubuhkan “colspan = 2” pada <td> kedua baris kedua.

```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 3B</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat3.css">
</head>
<body>

    <h1>Pertemuan 4</h1>
    <h3>Latihan 3B</h3>

    <table class="tabel">
        <caption>Tabel 1.1</caption>
        <tr>
            <th>No</th>
            <th>Lalab</th>
            <th>Stok</th>
        </tr>
        <tr>
            <td rowspan="2"></td>
            <td colspan="2"></td>
        </tr>
        <tr>
            <td></td>
            <td></td>
        </tr>
    </table>

</body>
</html>
```

Hasil pada browser:



Perhatikan! Seperti pada kasus rowspan, colspan membuat kolom kedua pada baris kedua menggunakan space kolom ketiga baris kedua, sehingga membuat struktur kolom yang ada bergeser menjadi kolom keempat.

Hal yang mesti dilakukan pada kasus colspan adalah membuang kolom (<td>) yang tak diperlukan, sehingga code html kita menjadi seperti berikut:

```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 3B</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat3.css">
</head>
<body>
    <h1>Pertemuan 4</h1>
    <h3>Latihan 3B</h3>
    <table class="tabel">
        <caption>Tabel 1.1</caption>
        <thead>
            <tr>
                <th>No</th>
                <th>Lalab</th>
                <th>Stok</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td rowspan="2"></td>
                <td colspan="2"></td>
            </tr>
            <tr>
                <td></td>
                <td></td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

Tampilan pada browser:

A screenshot of a web browser window titled "latihan 3B". The slide contains the following text and table:

Pertemuan 4

Latihan 3B

Tabel 1.1

No	Lalab	Stok

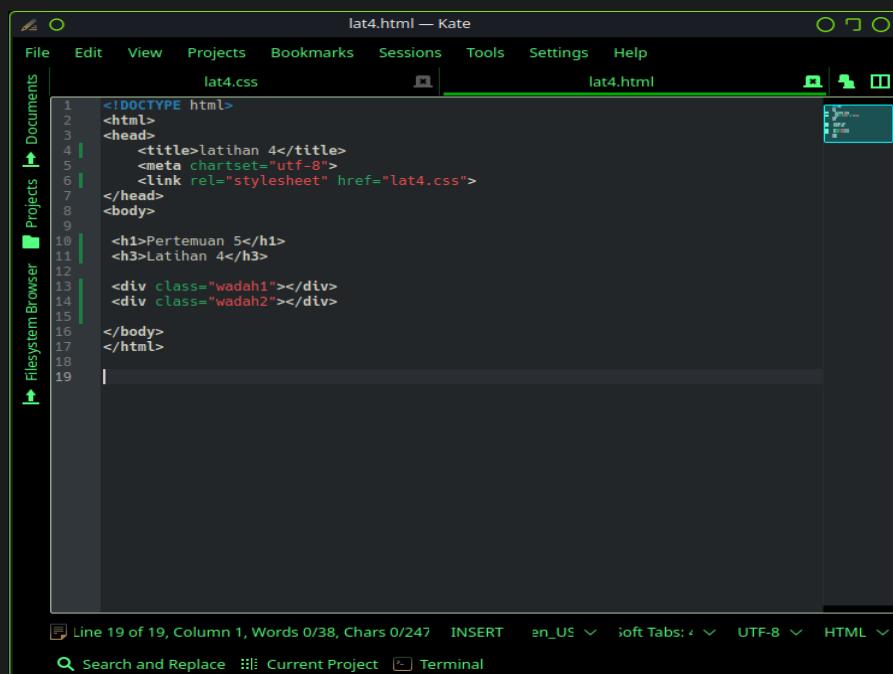
PERTEMUAN 5 Kontainerisasi

Kontainer secara sederhana diartikan sekaligus digunakan sebagai “wadah”. Wadah atau tempat digunakan untuk menyimpan sesuatu, kemudian diposisikan bahkan didekorasi dengan warna latar atau aksen tertentu.

Penggunaan elemen html sebagai kontainer sangat berguna dan memudahkan pengaturan struktur html untuk dijadikan dasar utama layout halaman web. Elemen html yang sering digunakan sebagai kontainer adalah <div>. Div sendiri memiliki makna semantika “division”, artinya pembagian/kavling alias kontainer itu sendiri.

Adapun persiapan untuk melakukan latihan kontainer adalah sebagai berikut:

1. Buat dua dokumen; lat4.html dan lat4.css
2. Sesuaikan konten <title> dan <link> dibagian <head>
3. Buat struktur utama html pada dokumen lat4.html
4. Isi konten body dengan dua elemen <div>, lalu beri class dengan nama “wadah1” dan “wadah2”.

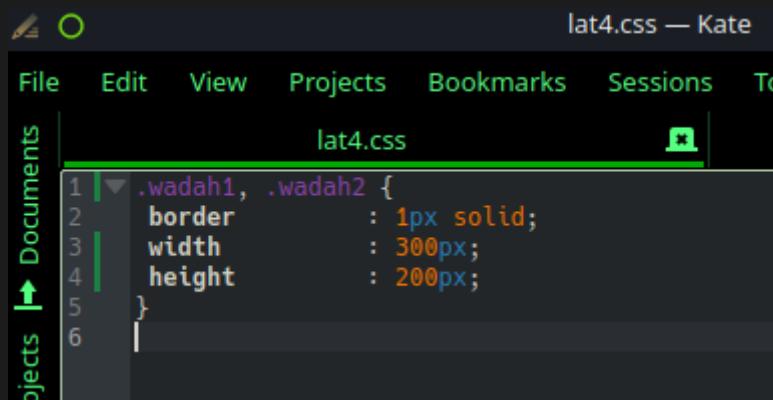


```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 4</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat4.css">
</head>
<body>
    <h1>Pertemuan 5</h1>
    <h3>Latihan 4</h3>
    <div class="wadah1"></div>
    <div class="wadah2"></div>
</body>
</html>
```

6. Save dan buka di browser.

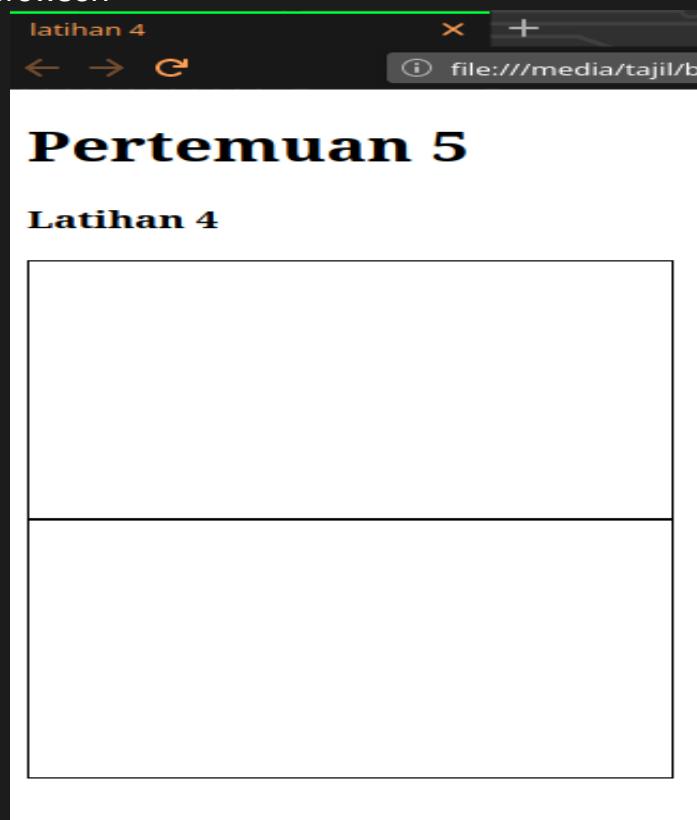
Pasti ketika dibuka dibrowser belum menunjukan apapun. Hal ini karena secara default setiap elemen html tidak menampilkan bordernya. Perlu diaktifkan terlebih dahulu via css melalui propertis ‘border’ serta penentuan ukuran lebar dan tinggi elemen.

Mari kita tambah propertis css tersebut pada kedua div :



```
lat4.css — Kate
File Edit View Projects Bookmarks Sessions To
lat4.css
1 .wadah1, .wadah2 {
2   border      : 1px solid;
3   width       : 300px;
4   height      : 200px;
5 }
6
```

Hasil dibrowser:



Sekarang terlihat nampak border berwarna hitam dengan ukuran 300x200 pixel. Kedua kontainer tersebut dapat diisi berbagai konten dengan menggunakan berbagai elemen hmtl. Kontainerisasi juga membantu mempermudah mendesain layout website dan membantu dalam kategorisasi konten halaman web seperti kategori header, section, footer dan lainnya.

Elemen <div> merupakan salah satu elemen yang memberi garis baru otomatis. Pada contoh diatas <div> kedua langsung berada tepat dibawah <div> pertama. Hal ini patut dipahami supaya keperluan mendesain layout website menjadi lebih mudah.

Adapun border yang ditampilkan dapat dihilangkan setelah layouting selesai atau bahkan tetap menggunakan style seperti itu (sesuai selera). Silahkan diuji-cobakan mengganti ketebalan dan warna border. Apabila menginginkan konfigurasi berbeda antara <div> pertama dengan <div> kedua, maka harus diberi class berbeda dengan nama class berbeda pula.

Kemudian, coba isi kedua div tersebut dengan konten seperti paragraf atau tipe image (jpg, png, ...). Kemudahan utama dalam menggunakan kontainer sebagai dasar layouting halaman web adalah kita tidak perlu mengatur setiap elemen satu-persatu untuk diposisikan pada lokasi tertentu, cukup pindahkan kontainer/wadah pada lokasi tujuan, dengan serta-merta seluruh isi kontainer mengikuti.

PERTEMUAN 6 Kontainerisasi II

Halaman website pada umumnya dapat dikategorikan menjadi beberapa golongan umum layout, terdiri atas satu halaman flat, terdiri atas dua bagian utama kiri dan kanan (salah satu bagian adalah panel navigasi), terdiri atas tiga bagian vertikal kiri-tengah-kanan, terdiri atas empat bagian atas-kiri-tengah-kanan atau empat bagian atas-kiri-tengah-kanan dan bawah.

Penentuan jumlah dan posisi layout tentu berdasarkan kebutuhan dan selera, baru kemudian disesuaikan dengan tren terkini. Diluarsana, sudah banyak tersedia cara lebih instan dalam membangun halaman website dengan menggunakan framework atau dalam bahasa spesifik adalah content management system (CMS) seperti wordpress, drupal, blog dan lain-lain.

Untuk para pemula disarankan bersabar dahulu untuk tidak menggunakan hal tersebut sebagai langkah awal belajar html-css. Hal ini karena walaupun tersedia berbagai framework gratis, kemampuan manual dalam programing html-css sangat besar manfaatnya. Pada level tertentu bahkan dapat membuat framework sendiri dengan cita rasa khas.

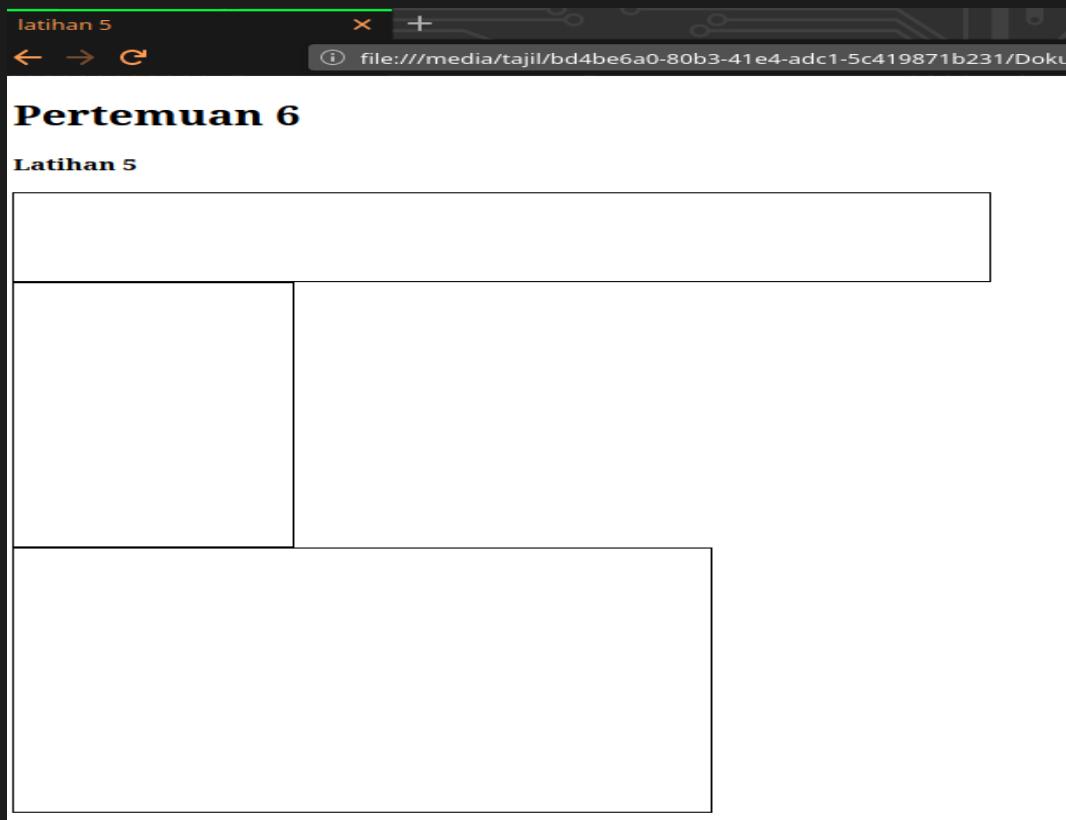
Adapun persiapan latihan pertemuan 6 ialah sebagai berikut:

1. Buat dua dokumen; lat5.html dan lat5.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link pada bagian head
4. Isi konten html dengan tiga elemen <div>
5. Beri ketiga div adi class berbeda, tentu nama class berbeda pula
6. Isi css untuk ketiga div seperti berikut:

The screenshot shows the Kate text editor interface. The title bar reads "lat5.css — Kate". The menu bar includes File, Edit, View, Projects, Bookmarks, Sessions, and Tools. A toolbar with icons for file operations is visible above the main editor area. The left sidebar has sections for Documents, Projects, and Filesystem Browser, with "Documents" currently selected. The main editor area contains the following CSS code:

```
1 .wadah1 {  
2   border : 1px solid;  
3   width  : 700px;  
4   height : 100px;  
5 }  
6  
7 .wadah2 {  
8   border : 1px solid;  
9   width  : 200px;  
10  height : 300px;  
11 }  
12  
13 .wadah3 {  
14   border : 1px solid;  
15   width  : 500px;  
16   height : 300px;  
17 }  
18
```

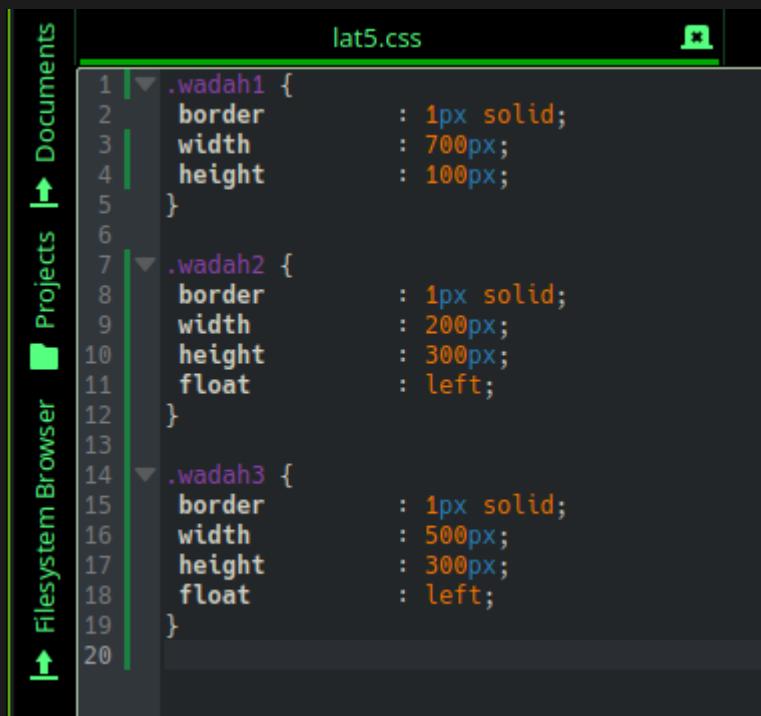
Lalu, tampilkan di browser !



Pada tahap ini anggap div-div tersebut sebagai blok pembangun layout website kita. Pada pertemuan ke-5 telah dijelaskan bahwa, div adalah elemen yang memberi garis baru secara otomatis, walaupun masih tersedia space kosong dibelah kanan layar.

Secara teknis, <div> adalah jenis elemen kategori block. Kategori block akan memberi garis baru otomatis setelah dia di render browser. Untuk dapat merubah propertis elemen html sesuai keperluan, css adalah jawabannya.

Tambahkan propertis css “float” dengan value “left” pada div kedua dan ketiga seperti berikut:

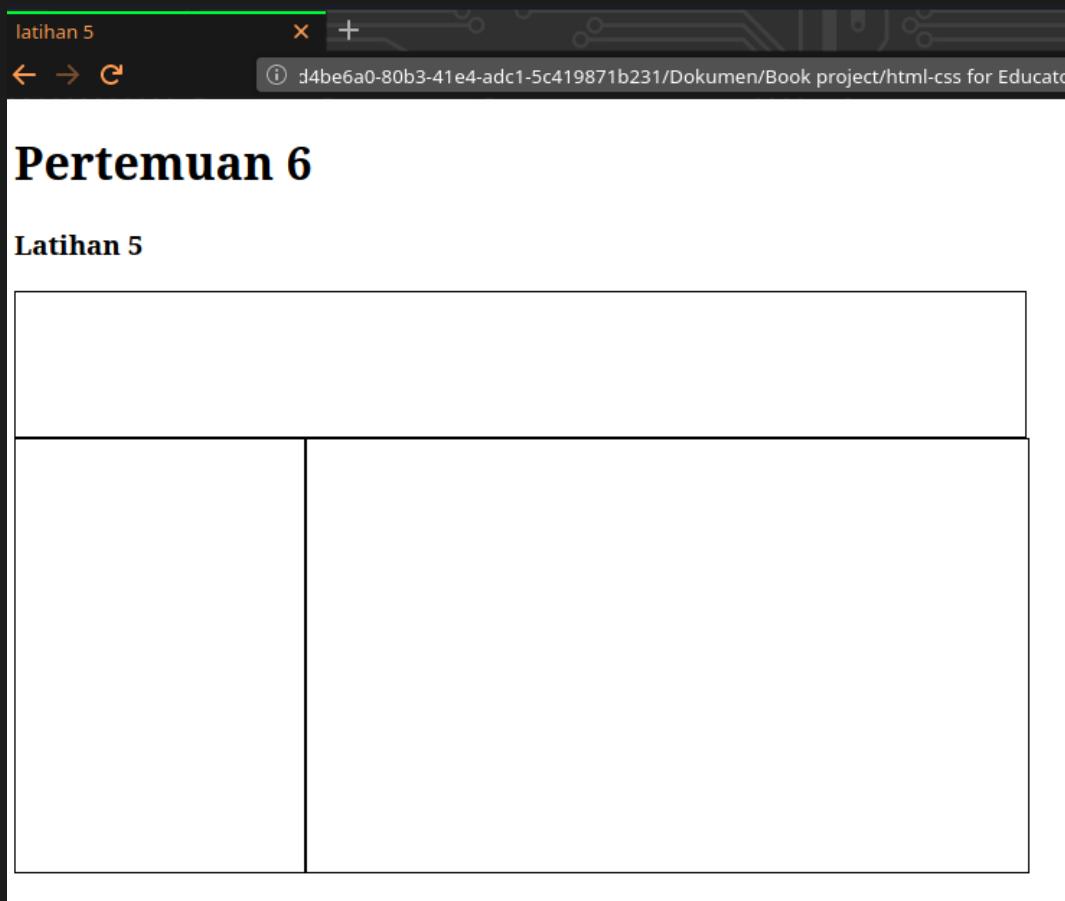


```
lat5.css
Documents
Projects
Filesystem Browser
1 .wadah1 {
2   border      : 1px solid;
3   width       : 700px;
4   height      : 100px;
5 }
6
7 .wadah2 {
8   border      : 1px solid;
9   width       : 200px;
10  height      : 300px;
11  float       : left;
12 }
13
14 .wadah3 {
15   border      : 1px solid;
16   width       : 500px;
17   height      : 300px;
18   float       : left;
19 }
20
```

Notes:

Propertis ‘float’ membuat suatu elemen memiliki sifat float (melayang, mengapung....) sehingga dapat pula ditumpuk diatas elemen lain. Value atau nilai dari propertis float hanya ada dua opsi; left atau right.

Save lalu refresh browser. Tampilan pada browser:



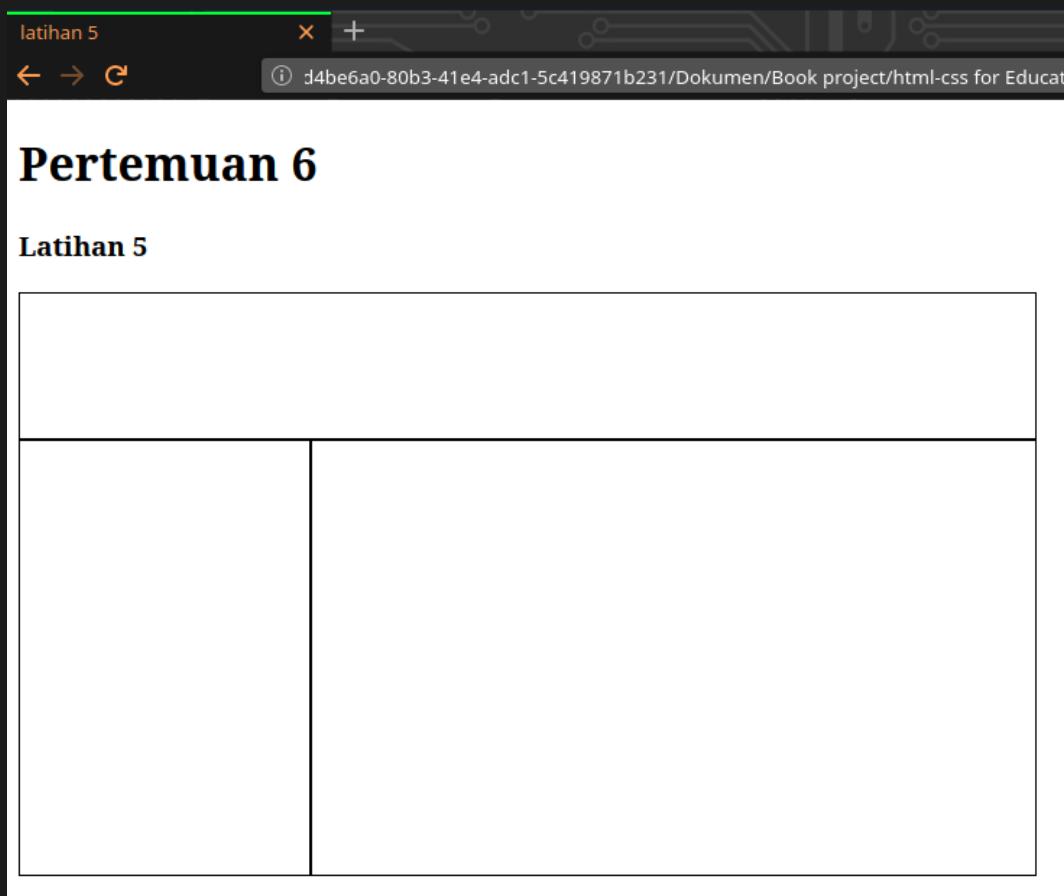
Kini para tersebut mulai terlihat seperti sebuah desain layout website yang siap diisi oleh berbagai konten!

Namun Perhatikan ! Bagian kanan dari div ketiga agak sedikit lebih lebar ketimbang div bagian atas. Padahal sedari awal kita telah memberi ukuran width dan height dengan angka tertentu yakni div pertama lebar 700px, div kedua lebar 200px dan div ketiga lebar 500px. Hasil penjumlahan lebar div kedua dan ketiga adalah $200 + 500 = 700$. Seharusnya sudah sama dengan lebar div pertama 700px. Lantas mengapa demikian ?

Ketika mendesain layout website, yang menjadi perhitungan lebar dan tinggi (width dan height) tidak hanya kalkulasi angka pada properti 'width' dan 'height' saja, melainkan terdapat bagian yang juga mesti turut dikalkulasikan seperti ketebalan border setiap elemen apabila border ditampilkan seperti latihan diatas. Margin dan padding juga termasuk yang mesti dimasukan dalam kalkulasi ukuran layout website.

Terdapat berbagai macam cara untuk mengatasi hal seperti ini. Penambahan atau pengurangan ukuran width atau height suatu elemen misalnya atau penyesuaian besaran margin dan padding.

Kita akan mengatur kembali ukuran width div pertama agar sejajar dengan width div dibawahnya. Tambah width div pertama menjadi 702px, save lalu refresh browser. Hasil setelah ditambah ukuran width:



Sekarang, layout benar-benar siap diisi konten. Cobalah isi dengan konten judul pada bagian atas (header), konten tipe link [<a>](#) pada bagian kiri dan konten tipe text (kombinasi

<h1>

 dan

<p>

) pada bagian kanan sebagai homepage.

PERTEMUAN 7 Form Elements

Form artinya formulir. Formulir kita ketahui adalah kumpulan inputan yang terorganisir untuk penjemputan/pengumpulan suatu data. Data tersebut kemudian digolongkan untuk dianalisis.

Persis seperti itu, form html juga untuk tujuan “inputasi” data lalu digolongkan untuk keperluan analisa atau sekedar untuk ditampilkan kembali melalui browser.

Elemen-elemen form html umumnya dimanfaatkan oleh bahasa pemrograman kategori side scripting language seperti PHP, Javascript dan lain sebagainya.

Elemen form html terdiri atas beberapa macam yakni:

- ✓ input type text
- ✓ input type number
- ✓ input type radio
- ✓ input type checkbox
- ✓ input type button
- ✓ input type reset
- ✓ select

Sebelum kita praktekan, ada hal yang patut dicamkan, yakni perihal pelabelan elemen form. Pelabelan diberikan tepat diantara tag buka dan tag tutup elemen tersebut, namun ada yang berada di dalam tag buka elemen, seperti input type text, number, button dan reset.

Berikut ini persiapan melakukan praktek di pertemuan ini:

1. Buat dua dokumen; lat6.html dan lat6.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link pada head
4. Isi bagian body seperti contoh dibawah:

The screenshot shows the Kate text editor interface. The title bar reads "lat6.html — Kate". The menu bar includes File, Edit, View, Projects, Bookmarks, Sessions, Tools, Settings, and Help. A toolbar on the left contains icons for Documents, Projects, Filesystem Browser, and other functions. Two tabs are open: "lat6.css" (closed) and "lat6.html". The "lat6.html" tab contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 6</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat6.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 7</h1>
11  <h3>Latihan 6</h3>
12
13  <input type="text" placeholder="Nama" />
14  <input type="text" placeholder="Password" />
15  <input type="number" placeholder="Telp." />
16  <br/>
17
18  <input type="radio">WNI</input>
19  <br/>
20  <input type="radio">WNA</input>
21  <br/>
22
23  <input type="checkbox">Pria</input>
24  <br/>
25  <input type="checkbox">Wanita</input>
26  <br/>
27
28  <input type="button" value="Submit" />
29  <input type="reset" />
30
31
32 </body>
33 </html>
34
```

Notes:

- Konten html diatas dipisahkan ke dalam empat golongan, golongan pertama dari baris ke-13–16, golongan kedua dari baris ke-18-21, golongan ketiga dari baris ke-23-26 dan golongan keempat dari baris ke-28-29.
- Apabila nomor baris belum muncul pada editor, maka harus diaktipan secara manual pada editor yang anda gunakan. Baca panduan yang disediakan developer.
- Golongan pertama adalah tipe elemen input yang pelabelannya dilakukan didalam tag buka elemen tersebut via placeholder. Beberapa elemen dapat ditutup langsung dengan membubuhkan "/"

sebelum kurung sikukanan ">". Kebiasaan menjaga tag buka dan tutup elemen html adalah suatu yang sangat baik untuk kepentingan jangka panjang dan menapaki tingkat yang lebih tinggi.

- Golongan kedua dan ketiga adalah elemen form html yang pelabelannya dilakukan tepat diantara tag buka dan tutup elemen tersebut.
- Adapun golongan terakhir, pelabelan dilakukan di dalam tag open elemen via "value". Untuk tipe "reset" tidak perlu dilabeli, dia otomatis mengandung label "reset".
- Semua tipe elemen input diatas adalah kategori inline alias tidak akan memberi garis baru pada elemen setelah dia. Maka untuk memberi garis baru sebagai sekat atau jeda antara, gunakan elemen break:
. Elemen break adalah salah satu elemen yang tidak memiliki tag penutup, sehingga ditutup dengan cara menambahkan "/" sebelum siku kanan ">".

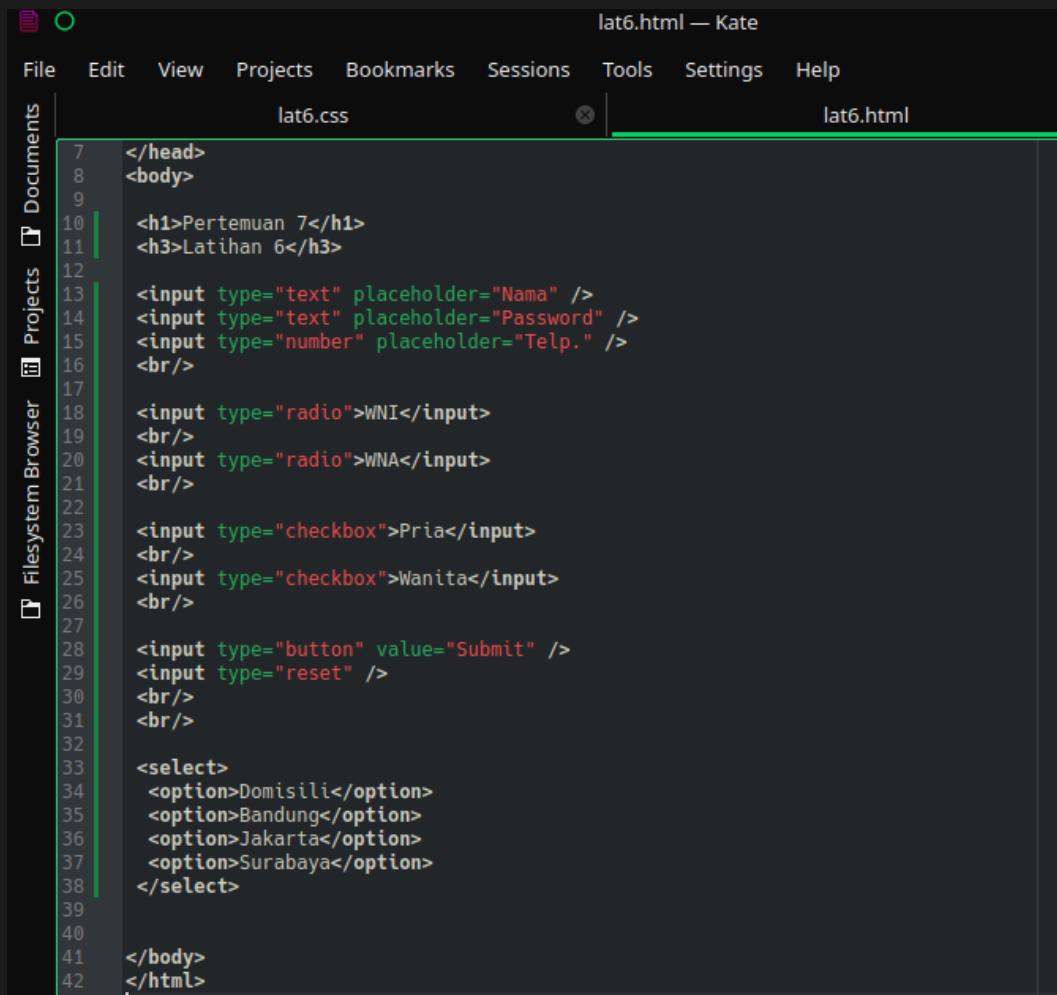
Tampilan pada browser:

The screenshot shows a web browser window with the title bar "Latihan 6 - M". The address bar shows the URL "file:///media/tajil/Storage/UINK/EBOOK/Book project/html-css for Edu". The main content area displays a form titled "Latihan 6". The form includes three input fields: "Nama", "Password", and "Telp.". Below these fields is a group of radio buttons for selecting nationality: "WNI" and "WNA". Underneath the nationality buttons are two checkboxes for gender: "Pria" and "Wanita". At the bottom of the form are two buttons: "Submit" and "Reset".

NB:

Untuk inputan tipe password, html dewasa ini menyediakan elemen input khusus password : <input type="password" placeholder="password"/>. Input ini langsung merubah hasil ketikan menjadi asterik (*) dan juga menyediakan tombol untuk melihat atau show password.

Selanjutnya mari kita praktikan elemen <select> sebagai berikut:



```
lat6.html — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat6.css lat6.html
7 </head>
8 <body>
9
10 <h1>Pertemuan 7</h1>
11 <h3>Latihan 6</h3>
12
13 <input type="text" placeholder="Nama" />
14 <input type="text" placeholder="Password" />
15 <input type="number" placeholder="Telp." />
16 <br/>
17
18 <input type="radio">WNI</input>
19 <br/>
20 <input type="radio">WNA</input>
21 <br/>
22
23 <input type="checkbox">Pria</input>
24 <br/>
25 <input type="checkbox">Wanita</input>
26 <br/>
27
28 <input type="button" value="Submit" />
29 <input type="reset" />
30 <br/>
31 <br/>
32
33 <select>
34 <option>Domisili</option>
35 <option>Bandung</option>
36 <option>Jakarta</option>
37 <option>Surabaya</option>
38 </select>
39
40
41 </body>
42 </html>
```

Elemen ini terdiri atas tag buka dan tutup regular. Didalamnya terdapat <option> sebagai pilihan menu ketika diklik.

Gunakan <option> pertama sebagai pengantar atau sebagai label elemen ini.

Tampilan pada browser:

latihan 6 - M

latihan 6

← → ⌂ file:///media/tajil/Storage/UINK/EBOOK/Book project/html-css for Educa

Pertemuan 7

Latihan 6

WNI
 WNA
 Pria
 Wanita

Domisili

Tampilan ketika “Domisili” di klik :

latihan 6 - M

latihan 6

← → ⌂ file:///media/tajil/Storage/UINK/EBOOK/Book project/html-css for Educa

Pertemuan 7

Latihan 6

WNI
 WNA
 Pria
 Wanita

Domisili

Domisili
Bandung
Jakarta
Surabaya

PERTEMUAN 8 Ordered & Unordered List

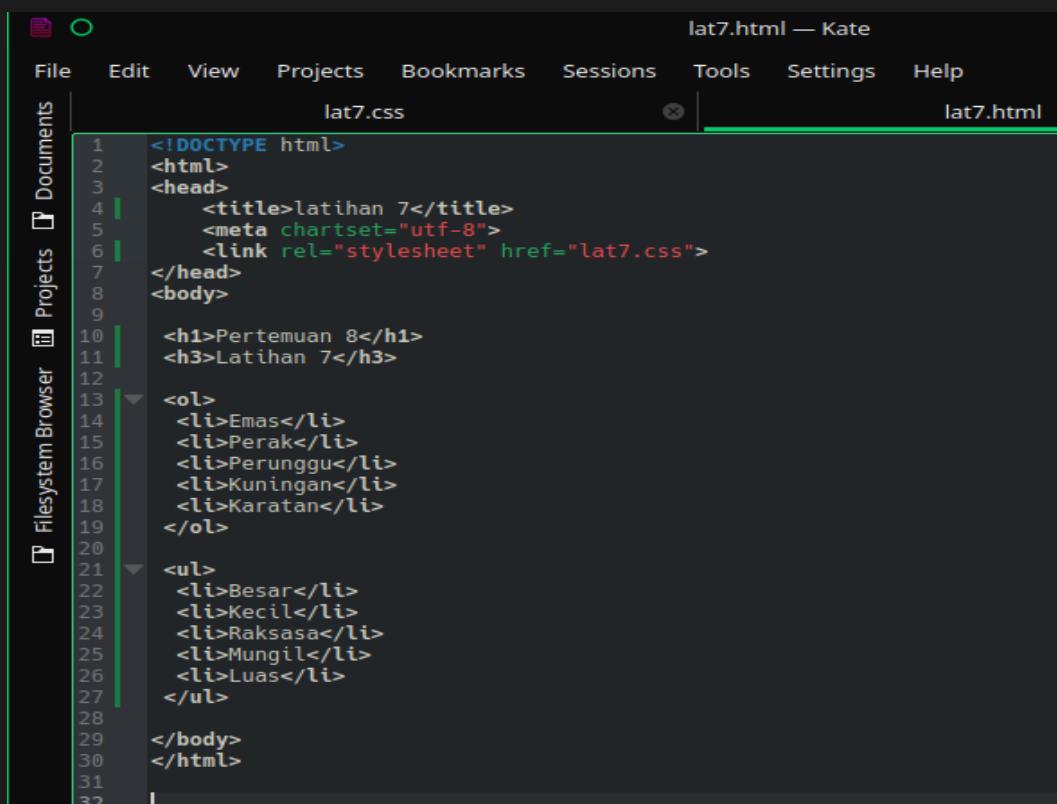
Html menyediakan fasilitas untuk membuat list baik ber-style tersusun atau tidak tersusun. Pada html istilah tersusun sama dengan ordered dan tak tersusun sama dengan unordered.

Ordered list dengan menggunakan elemen ``, `ol` sebagai “ordered”. Unordered list dengan menggunakan elemen ``, `ul` sebagai “unordered”.

Baik `` maupun `` memiliki sub konten yakni sub elemen ``, `li` disini sebagai “list” item.

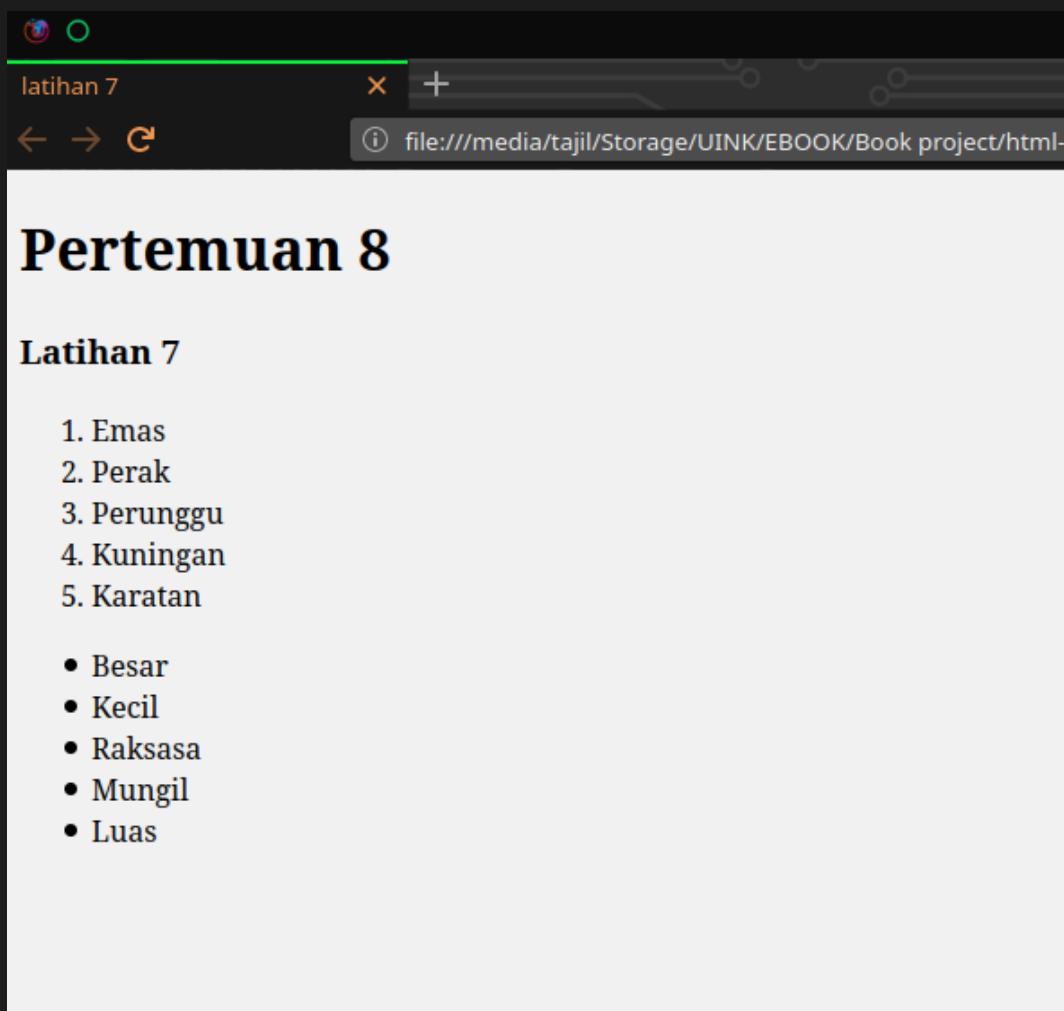
Berikut adalah persiapan praktik pertemuan 8:

1. Buat dua dokumen; lat7.html dan lat7.css
2. Buat struktur utama html
3. Sesuai konten title dan link pada head
4. isi body seperti contoh dibawah ini:



```
lat7.html — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat7.css | lat7.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 7</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat7.css">
7 </head>
8 <body>
9
10 <h1>Pertemuan 8</h1>
11 <h3>Latihan 7</h3>
12
13 <ol>
14   <li>Emas</li>
15   <li>Perak</li>
16   <li>Perunggu</li>
17   <li>Kuningan</li>
18   <li>Karatan</li>
19 </ol>
20
21 <ul>
22   <li>Besar</li>
23   <li>Kecil</li>
24   <li>Raksasa</li>
25   <li>Mungil</li>
26   <li>Luas</li>
27 </ul>
28
29 </body>
30 </html>
31
32
```

Tampilan pada browser:



Notes:

- ul dan ol akan memberi garis baru otomatis.
- ul dan ol memiliki sub elemen sama yakni
- Untuk kepentingan advance, ul dan ol dapat diganti style. Cek dokumentasi elemen html pada sumber lain.

PERTEMUAN 9 Border Radius & Box Shadow

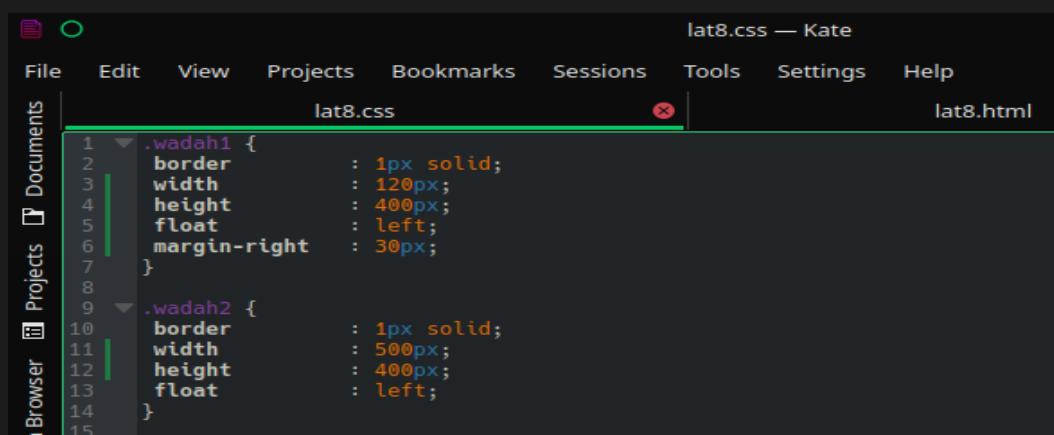
Meracik layout website secara sederhana bukan berarti tidak mencapai layout elegan dan looking profesional alike. Sedikit taburan kreatifitas membuat sesuatu dapat terlihat kompleks bahkan tidak jarang membuat decak kagum yang melihat.

Border radius dan Box shadow adalah bumbu pelengkap setelah menguasai materi border dan kontainerisasi. Dengan amunisi seperti ini saja kita dapat meracik suatu layout lumayan ciamik. Untuk itu kuasai soal border radius dan box shadow ini.

Border radius sesuai nama dan istilah tersebut untuk mengatur roundness atau kelengkungan border setiap elemen. Sama seperti ukuran ketebalan border, border radius serta box shadow dapat menggunakan satuan pixel (px). Ada sejumlah satuan yang dapat diterapkan pada value css seperti em, pt, % dan lainnya.

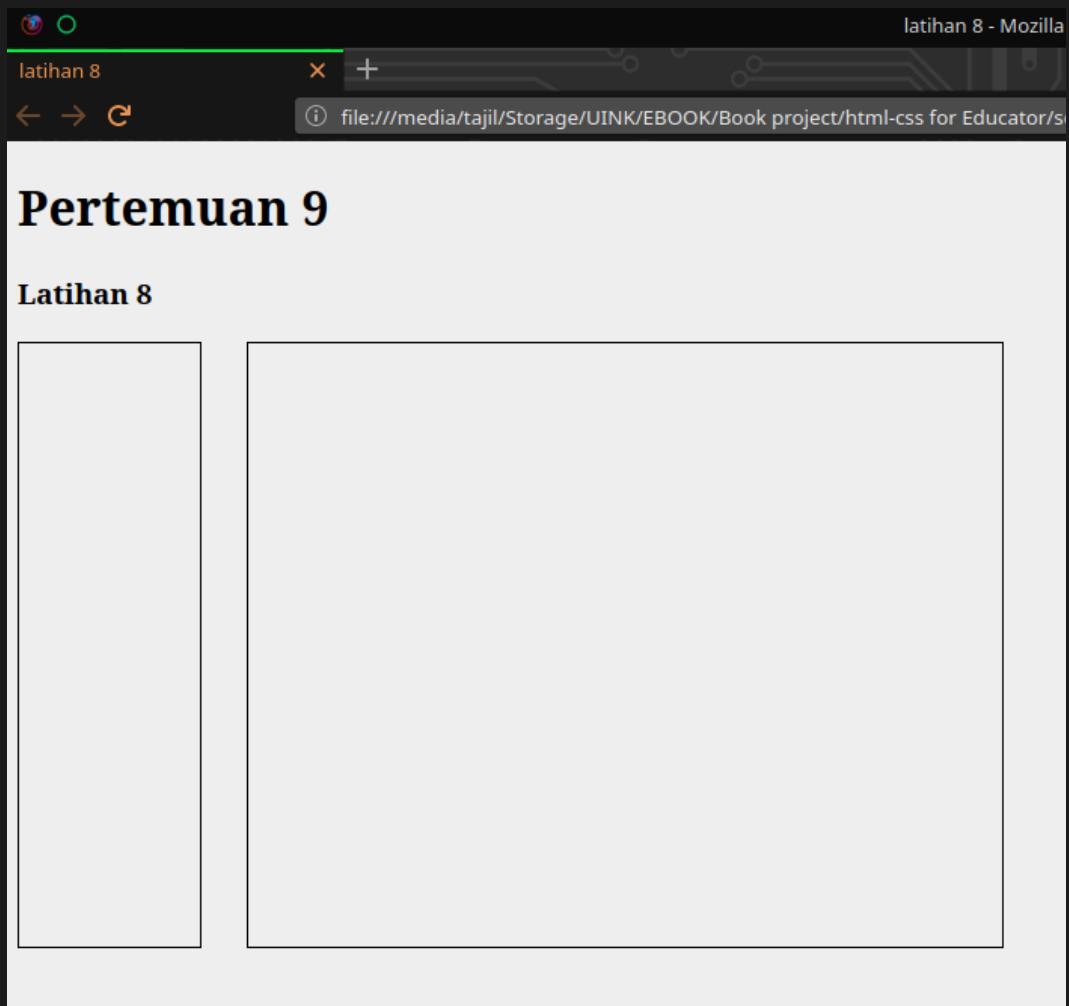
Persiapan latihan pertemuan 9 :

1. Buat dua dokumen; lat8.html dan lat8.css
2. buat struktur utama html
3. sesuaikan konten title dan link pada head
4. buat dua buah kontainer dengan menggunakan elemen <div>
5. beri kedua div tersebut class dengan nama berbeda
6. terapkan css berikut :



```
lat8.css — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat8.css x lat8.html
Documents Projects Browser
1 .wadah1 {
2   border      : 1px solid;
3   width       : 120px;
4   height      : 400px;
5   float       : left;
6   margin-right: 30px;
7 }
8
9 .wadah2 {
10  border      : 1px solid;
11  width       : 500px;
12  height      : 400px;
13  float       : left;
14 }
15
```

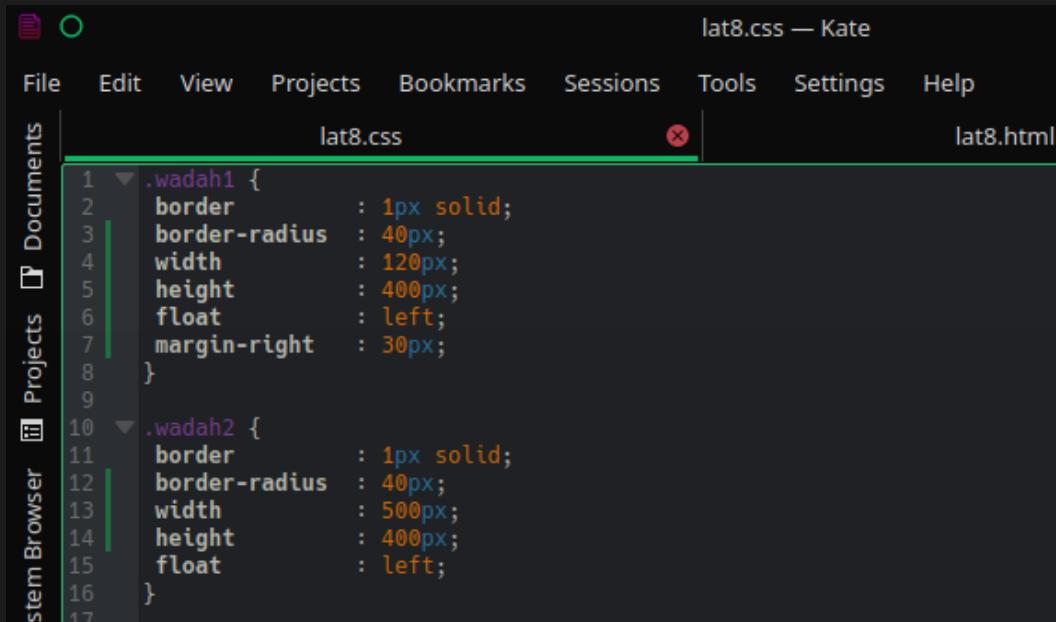
Hasil pada browser:



Notes:

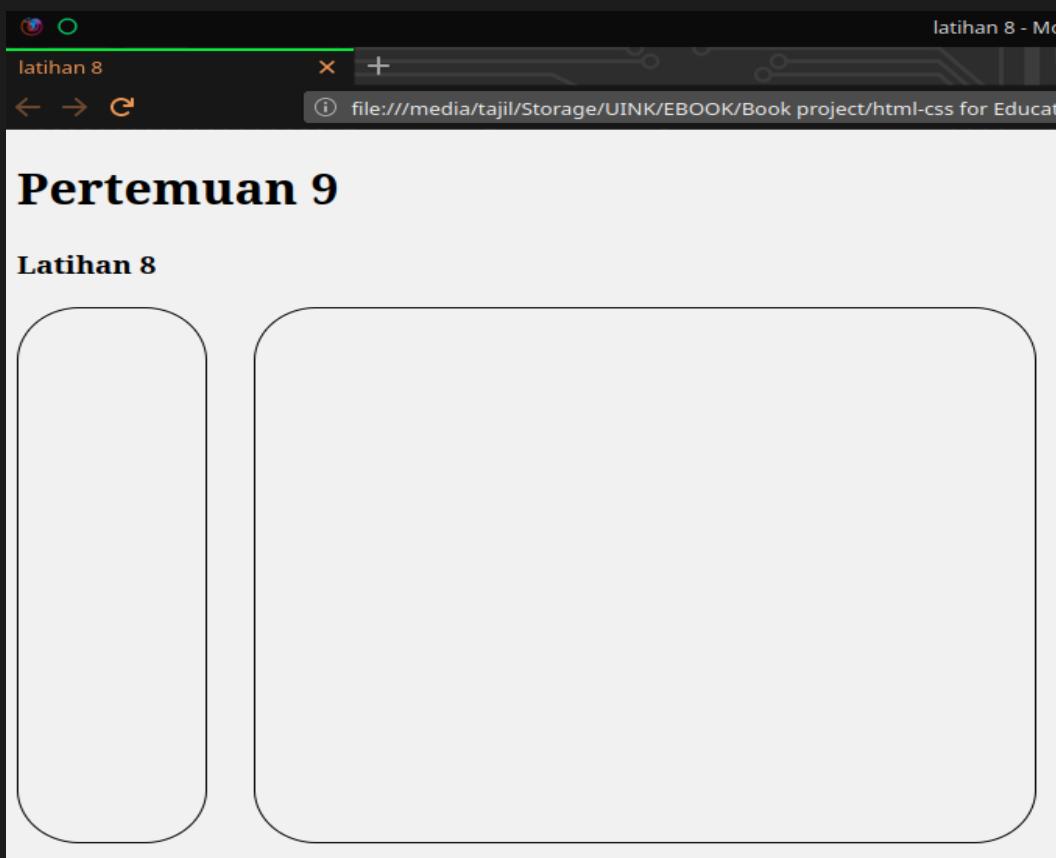
- Untuk dapat melakukan side by side menggunakan propertis 'float', setiap elemen target mesti diberi propertis ini.
- <div> pertama diberi margin-right:30px, artinya batas luar sebelah kanan terdapat jarak 30 pixel.
- Margin adalah batas luar suatu elemen, kita dapat mengatur margin sekaligus empat arah (kanan, kiri, atas dan bawah) dengan sintax "margin:auto;" atau mengatur secara individual salah satu batas luar tersebut seperti contoh diatas "margin-right".
- Ujicobalah dengan warna border sesuai selera, contoh "border: 1px solid red;".

Selanjutnya mari kita terapkan border radius dengan menambahkan properti dan value border radius pada elemen div diatas.

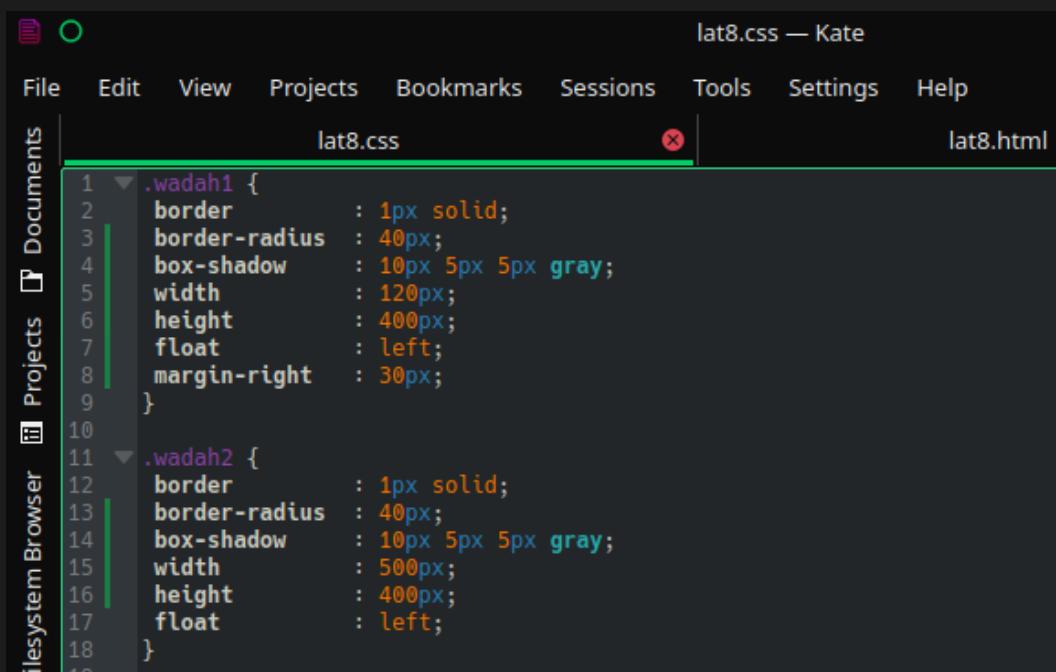


```
lat8.css — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat8.css lat8.html
1 .wadah1 {
2   border          : 1px solid;
3   border-radius   : 40px;
4   width           : 120px;
5   height          : 400px;
6   float           : left;
7   margin-right    : 30px;
8 }
9
10 .wadah2 {
11   border          : 1px solid;
12   border-radius   : 40px;
13   width           : 500px;
14   height          : 400px;
15   float           : left;
16 }
17
```

Hasil pada browser:

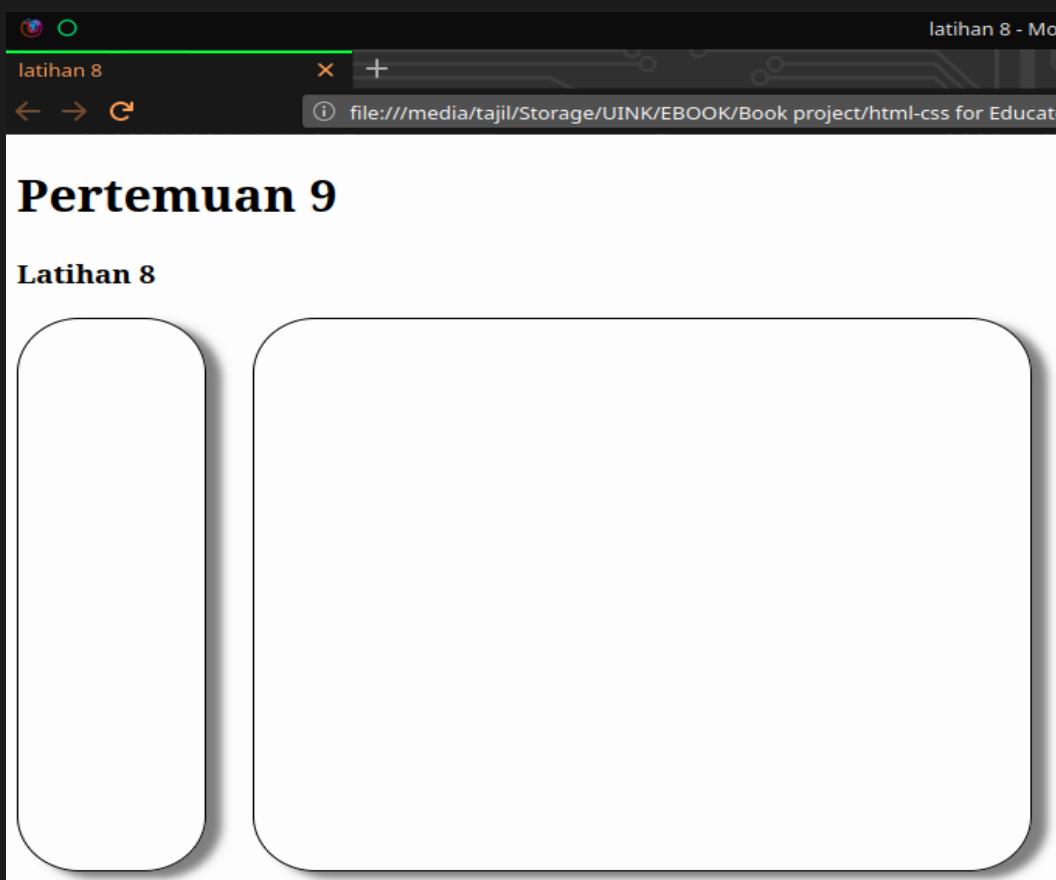


Sekarang saatnya menerapkan box shadow pada kedua elemen div kita.



```
lat8.css — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat8.css lat8.html
1 .wadah1 {
2   border : 1px solid;
3   border-radius : 40px;
4   box-shadow : 10px 5px 5px gray;
5   width : 120px;
6   height : 400px;
7   float : left;
8   margin-right : 30px;
9 }
10
11 .wadah2 {
12   border : 1px solid;
13   border-radius : 40px;
14   box-shadow : 10px 5px 5px gray;
15   width : 500px;
16   height : 400px;
17   float : left;
18 }
```

Hasil pada browser:

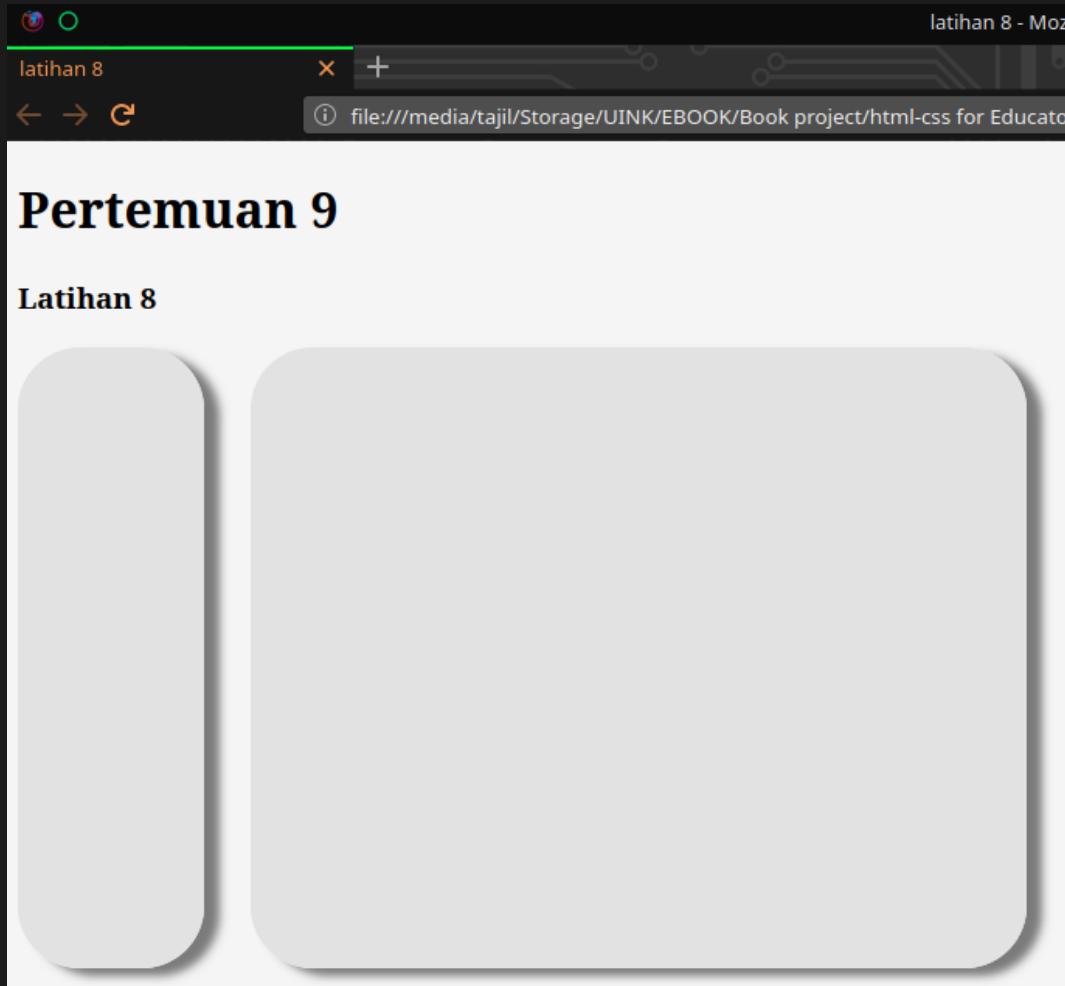


Notes:

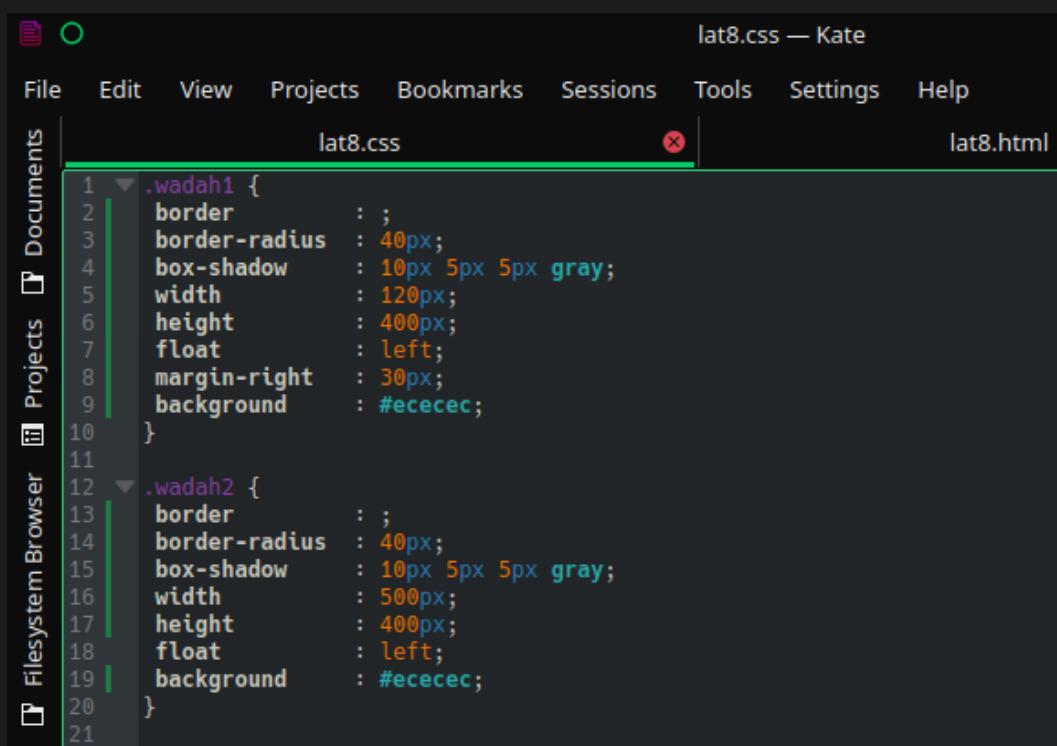
box-shadow: 10px 5px 5px gray;

- Digit pertama (10px) adalah ketebalan shadow arah belah kanan
- Digit kedua (5px) adalah posisi vertikal shadow dari elemen, semakin besar nilai, semakin dia turun ke arah bawah.
- Digit ketiga (5px) adalah tingkat blur shadow, rata-rata cukup menggunakan 3-5px.
- 'gray' adalah warna shadow

Pada tahap ini apabila border elemen sudah tidak diperlukan, kita bisa meniadakannya. Sehingga menghasilkan tampilan seperti berikut:



Code css :



```
lat8.css — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat8.css lat8.html
1 .wadah1 {
2   border : ;
3   border-radius : 40px;
4   box-shadow : 10px 5px 5px gray;
5   width : 120px;
6   height : 400px;
7   float : left;
8   margin-right : 30px;
9   background : #ececec;
10 }
11
12 .wadah2 {
13   border : ;
14   border-radius : 40px;
15   box-shadow : 10px 5px 5px gray;
16   width : 500px;
17   height : 400px;
18   float : left;
19   background : #ececec;
20 }
21
```

Notes:

Tahap ini kita meniadakan border, alias mengembalikan sifat border elemen ke kondisi default state tak terlihat.

Menambah warna latar (background) dan mengisi value background dengan "#ececec". Value warna atas border atau background dapat menggunakan kode warna dalam bahasa inggris atau dengan menggunakan kode hexadesimal seperti contoh diatas serta dapat menggunakan RGB color. Silahkan pelajari cara penggunaan warna melalui format hexadesimal dan RGB disumber lain.

PERTEMUAN 10 Layouting

Pertemuan 10 dikususkan untuk melakukan praktik layouting dengan bekal materi pada pertemuan yang telah lalu. Diawali dengan langkah perencanaan desain dalam bentuk sederhana dapat dilakukan sketsa diatas kertas atau bahkan hanya menggunakan imajinasi.

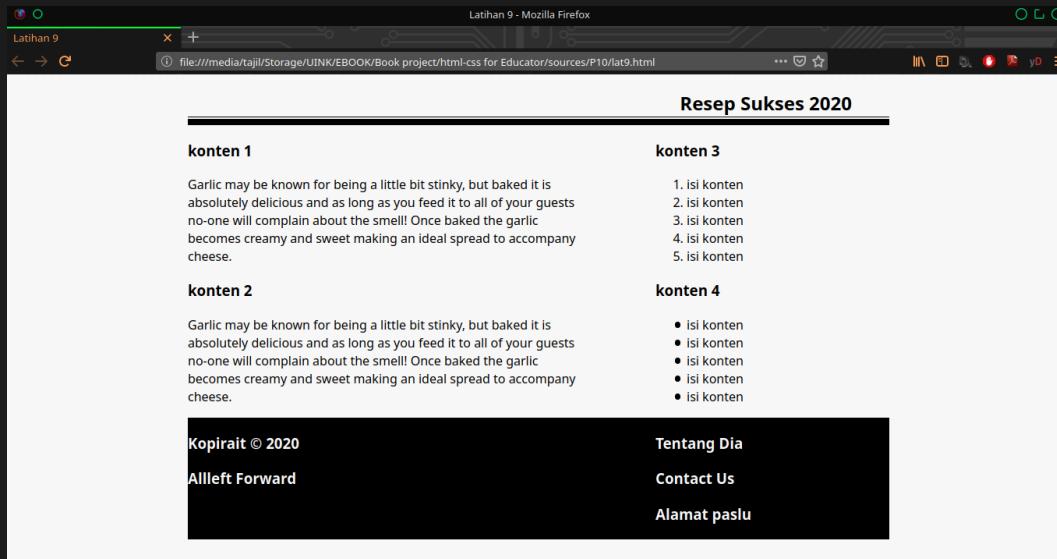
Kita akan mendesain satu halaman web yang terdiri atas tiga bagian utama; bagian header (atas), bagian section (tengah) dan bagian footer (bawah). Setiap bagian tersebut mengandung beberapa sub bagian menggunakan teknik kontainerisasi via elemen <div>. Hal ini dimaksudkan agar melakukan penempatan elemen menjadi lebih mudah.

Persiapan praktik pertemuan 10:

1. Buat dua dokumen; lat9.html dan lat9.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link pada head
4. Perhatikan figure dibawah, praktikan tanpa melihat sumber code html dan css yang disertakan dibawah. Teruslah latihan dan bereksperimen agar dapat mendesain semirip mungkin figure ini atau bahkan sampai tingkat sama. Gunakan berbagai cara dalam mendisain, hasil boleh jadi serupa, namun metode yang digunakan dapat sangat berbeda. Ada banyak jalan menuju Bandung!

Barulah boleh melihat contoh code html dan css yang disertakan ketika benar-benar buntu !

Contoh tampilan pada browser:



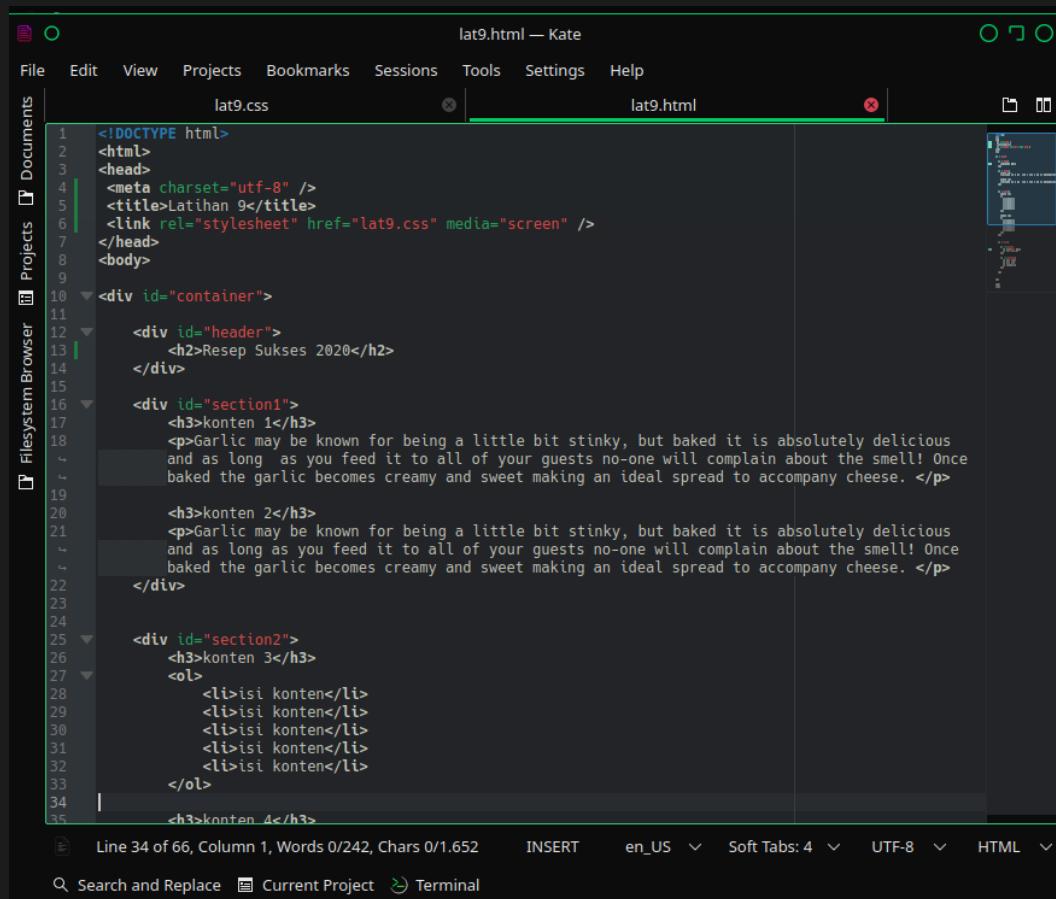
Notes:

- ✓ Perhatikan dengan seksama layout diatas. Analisa dengan cermat perbagian agar mudah me-redesain ulang persis seperti figure tersebut.
- ✓ Agar membantu dalam mendesain layout website, dapat dimulai dengan menentukan dasar layout via elemen <div>. Tampilkan border div tersebut untuk menentukan width dan height yang presisi bagi setiap bagian.
- ✓ Buat tiga elemen <div> dengan class yang berbeda, atur width dan height ketiga div mirip seperti figure diatas.
- ✓ Baru setelah itu mengisi tiap div dengan konten seperti pada yang dicontohkan diatas.
- ✓ Bisa saja tiap bagian div memiliki sub div seperti pada <div> kedua (section) dan ketiga (footer).
- ✓ Bagian footer diberi latar hitam
- ✓ bagian header memiliki garis bawah. Hal ini dapat dilakukan dengan kombinasi elemen garis horizontal (<hr>) dengan border elemen div bagian header. Mirip seperti margin yang dapat digunakan sekaligus empat arah atau manual perarah, borderpun demikian, dapat ditampilkan sekaligus empat sisi seperti latihan

yang telah kita lalui atau per satu sisi yang diinginkan, contoh:
border-left: 1px solid red;

- ✓ Pada div kedua baian berlabel konten 3 dan konten 4 memanfaatkan ordered dan unordered list (dan).
- ✓ Div bagian header hanya mengandung konten tipe teks. Karena terlihat besar dan bold, maka dapat dipastikan menggunakan elemen heading (<h1>). Jangan lupa posisikan disebelah kanan.

Source code html part 1:



The screenshot shows the Kate text editor interface with two tabs open: 'lat9.css' and 'lat9.html'. The 'lat9.css' tab contains the following CSS code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Latihan 9</title>
<link rel="stylesheet" href="lat9.css" media="screen" />
</head>
<body>
```

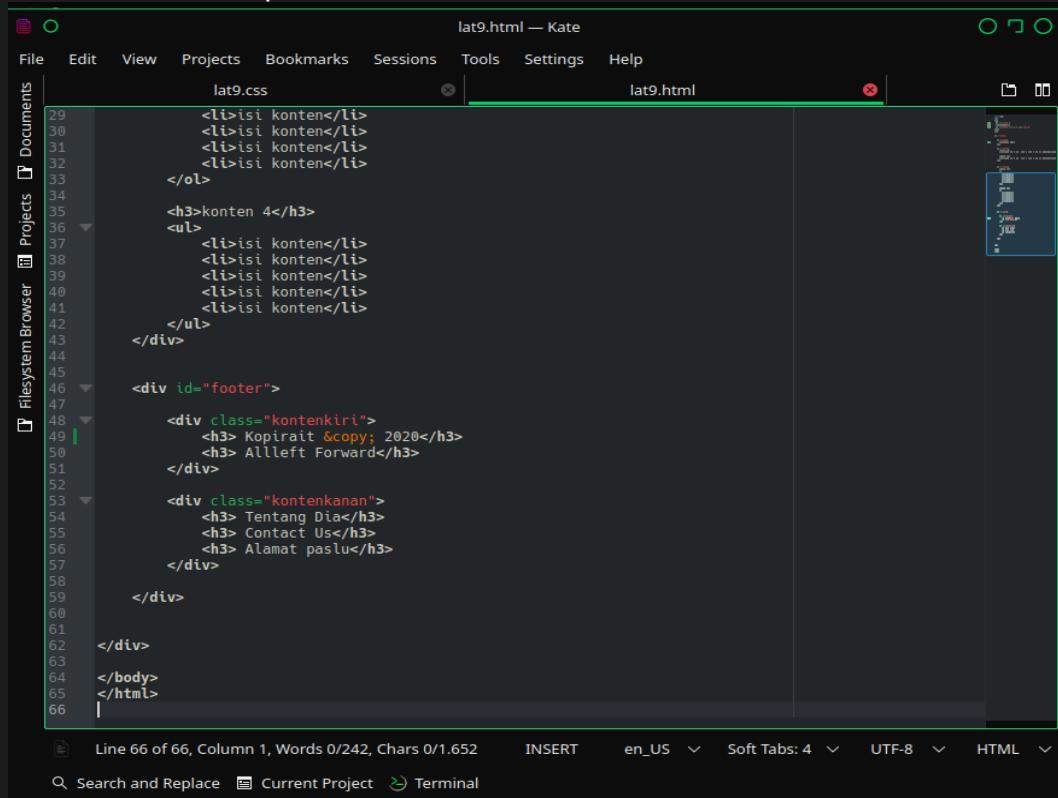
The 'lat9.html' tab contains the following HTML code:

```
<div id="container">
<div id="header">
<h2>Resep Sukses 2020</h2>
</div>
<div id="section1">
<h3>konten 1</h3>
<p>Garlic may be known for being a little bit stinky, but baked it is absolutely delicious and as long as you feed it to all of your guests no-one will complain about the smell! Once baked the garlic becomes creamy and sweet making an ideal spread to accompany cheese. </p>
<h3>konten 2</h3>
<p>Garlic may be known for being a little bit stinky, but baked it is absolutely delicious and as long as you feed it to all of your guests no-one will complain about the smell! Once baked the garlic becomes creamy and sweet making an ideal spread to accompany cheese. </p>
</div>
<div id="section2">
<h3>konten 3</h3>
<ol>
<li>isi konten</li>
<li>isi konten</li>
<li>isi konten</li>
<li>isi konten</li>
<li>isi konten</li>
</ol>
<h3>konten 4</h3>

```

The editor's status bar at the bottom shows: Line 34 of 66, Column 1, Words 0/242, Chars 0/1.652. It also includes settings for INSERT mode, en_US locale, Soft Tabs: 4, UTF-8 encoding, and HTML mode.

Source code html part 2:



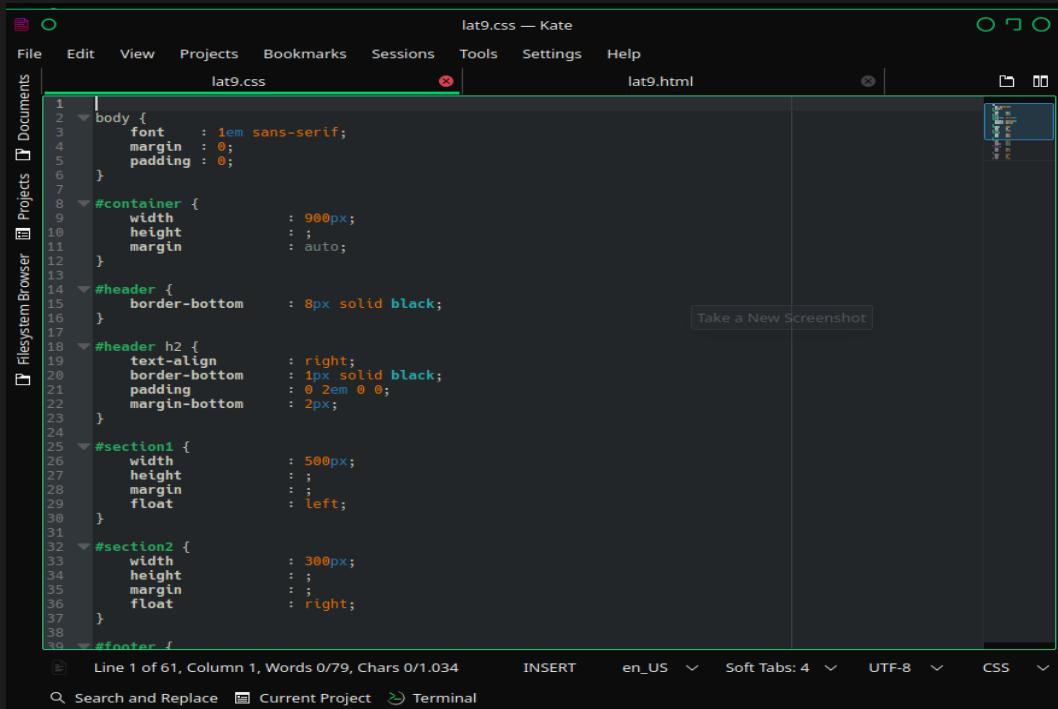
The screenshot shows the Kate text editor interface with two tabs open: 'lat9.css' and 'lat9.html'. The 'lat9.html' tab is active, displaying the following HTML code:

```
<ol>
    <li>isi konten</li>
    <li>isi konten</li>
    <li>isi konten</li>
    <li>isi konten</li>
</ol>
<h3>konten 4</h3>
<ul>
    <li>isi konten</li>
    <li>isi konten</li>
    <li>isi konten</li>
    <li>isi konten</li>
    <li>isi konten</li>
</ul>
</div>

<div id="footer">
    <div class="kontenkiri">
        <h3> Kopirait © 2020</h3>
        <h3> Alleft Forward</h3>
    </div>
    <div class="kontenkanan">
        <h3> Tentang Dia</h3>
        <h3> Contact Us</h3>
        <h3> Alamat paslu</h3>
    </div>
</div>
</body>
</html>
```

The status bar at the bottom indicates: Line 66 of 66, Column 1, Words 0/242, Chars 0/1.652. The bottom navigation bar includes: Search and Replace, Current Project, Terminal.

Source code css part 1:



The screenshot shows the Kate text editor interface with two tabs open: 'lat9.css' and 'lat9.html'. The 'lat9.css' tab is active, displaying the following CSS code:

```
body {
    font : 1em sans-serif;
    margin : 0;
    padding : 0;
}

#container {
    width : 900px;
    height : ;
    margin : auto;
}

#header {
    border-bottom : 8px solid black;
}

#header h2 {
    text-align : right;
    border-bottom : 1px solid black;
    padding : 0 2em 0 0;
    margin-bottom : 2px;
}

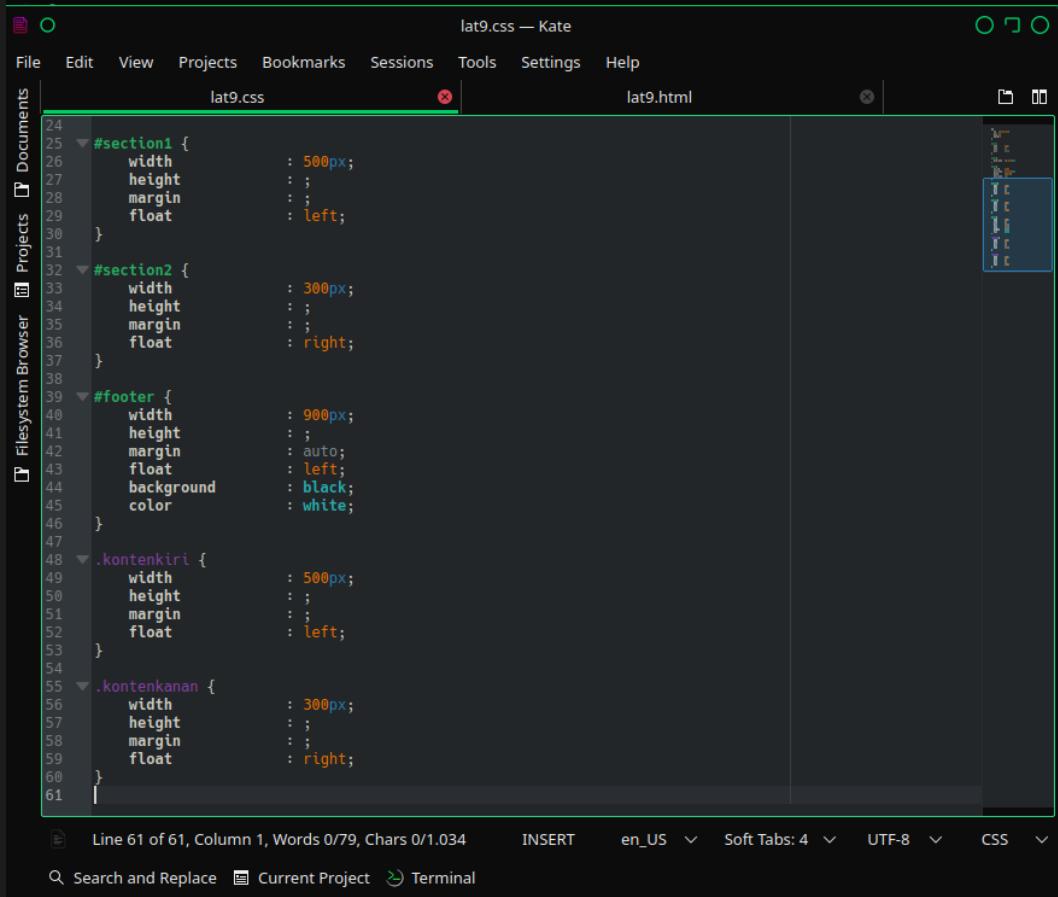
#section1 {
    width : 500px;
    height : ;
    margin : ;
    float : left;
}

#section2 {
    width : 300px;
    height : ;
    margin : ;
    float : right;
}

#footer {
```

The status bar at the bottom indicates: Line 1 of 61, Column 1, Words 0/79, Chars 0/1.034. The bottom navigation bar includes: Search and Replace, Current Project, Terminal.

Source code css part 2:



```
lat9.css — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
lat9.css lat9.html
Documents Projects FilesystemBrowser
24
25  #section1 {
26      width      : 500px;
27      height     : ;
28      margin     : ;
29      float      : left;
30  }
31
32  #section2 {
33      width      : 300px;
34      height     : ;
35      margin     : ;
36      float      : right;
37  }
38
39  #footer {
40      width      : 900px;
41      height     : ;
42      margin     : auto;
43      float      : left;
44      background : black;
45      color      : white;
46  }
47
48  .kontenkiri {
49      width      : 500px;
50      height     : ;
51      margin     : ;
52      float      : left;
53  }
54
55  .kontenkanan {
56      width      : 300px;
57      height     : ;
58      margin     : ;
59      float      : right;
60  }
61
```

Line 61 of 61, Column 1, Words 0/79, Chars 0/1.034 INSERT en_US Soft Tabs: 4 UTF-8 CSS Search and Replace Current Project Terminal

PERTEMUAN 11 Hover Effect

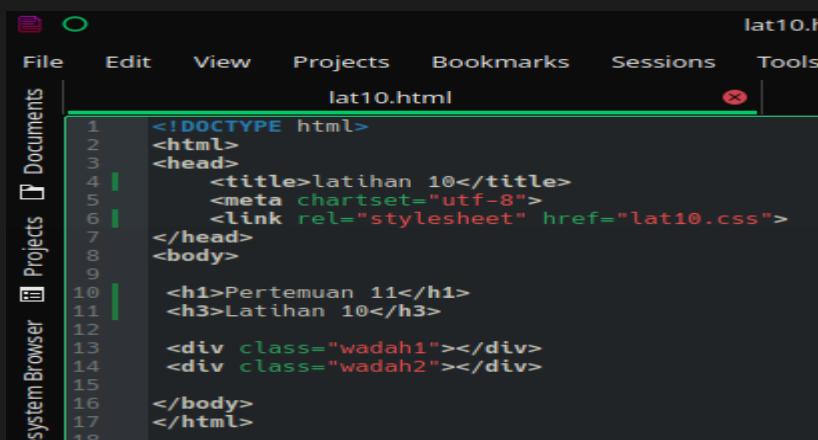
Hover dapat diartikan sebagai menunggu dekat atau melayang-layang. Makna kedua yang tepat untuk diterapkan dalam dunia html walaupun makna pertama juga dapat digunakan, yang jelas ini tidak terlalu krusial. Hal krusialnya adalah penguasaan terhadap hover ini.

Konsep ‘hover’ dalam css disebut juga sebagai “kelas palsu” atau “pseudo class”. Pseudo artinya palsu/lacung/gadungan. Secara singkat mengapa demikian karena hover bertingkah seperti propertis kedua bagi elemen bersangkutan. Dia hanya muncul ketika terpicu oleh event (kejadian) hover by mouse atau ketika disorot pointer (cursor). Setelah pointer tidak menunjuk elemen tersebut, efek hover pun menghilang dan berlaku propertis class asli si elemen.

Persiapan melakukan praktik pertemuan 11:

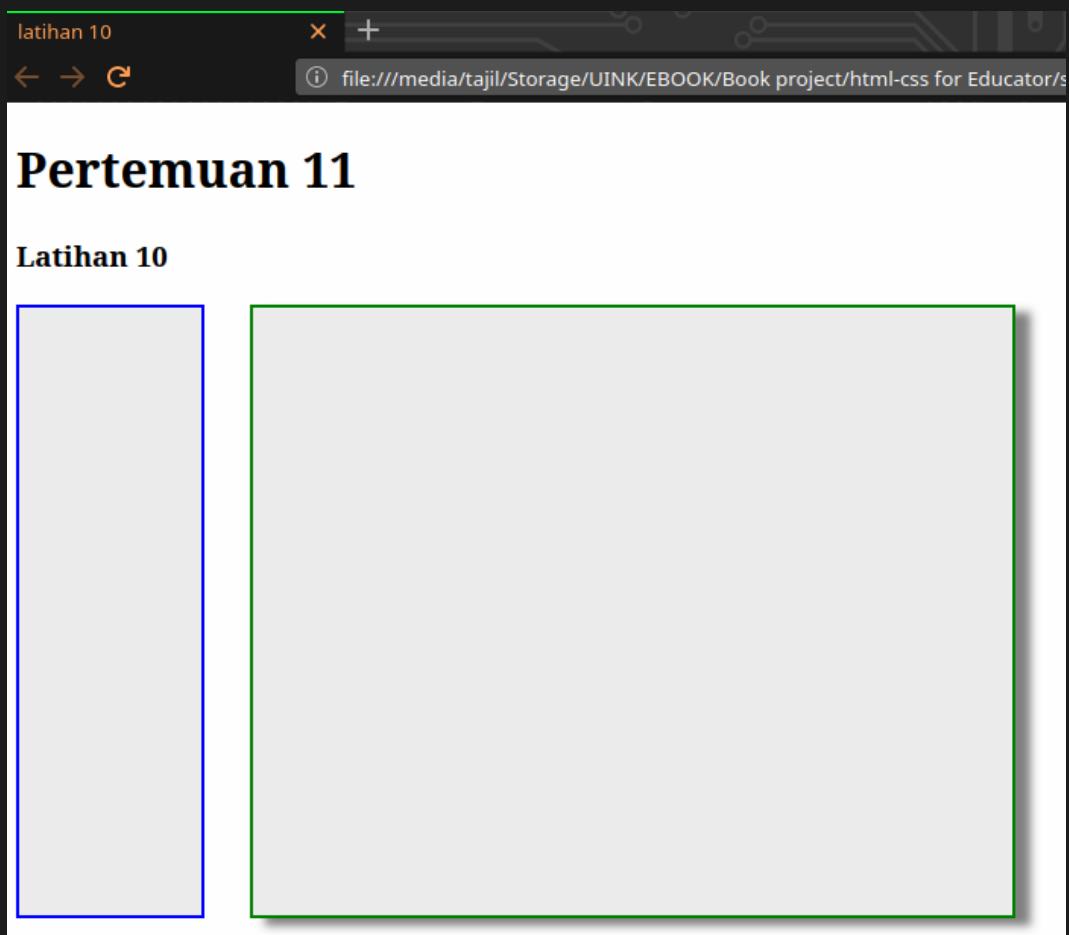
1. Buat dua dokumen; lat10.html dan lat10.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link pada head
4. Bubuhkan dua elemen <div> dan beri class
5. Tampilkan border div, lalu atur width dan height secara proporsional

Contoh code html:



```
<!DOCTYPE html>
<html>
<head>
    <title>latihan 10</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="lat10.css">
</head>
<body>
    <h1>Pertemuan 11</h1>
    <h3>Latihan 10</h3>
    <div class="wadah1"></div>
    <div class="wadah2"></div>
</body>
</html>
```

Tampilan pada browser:



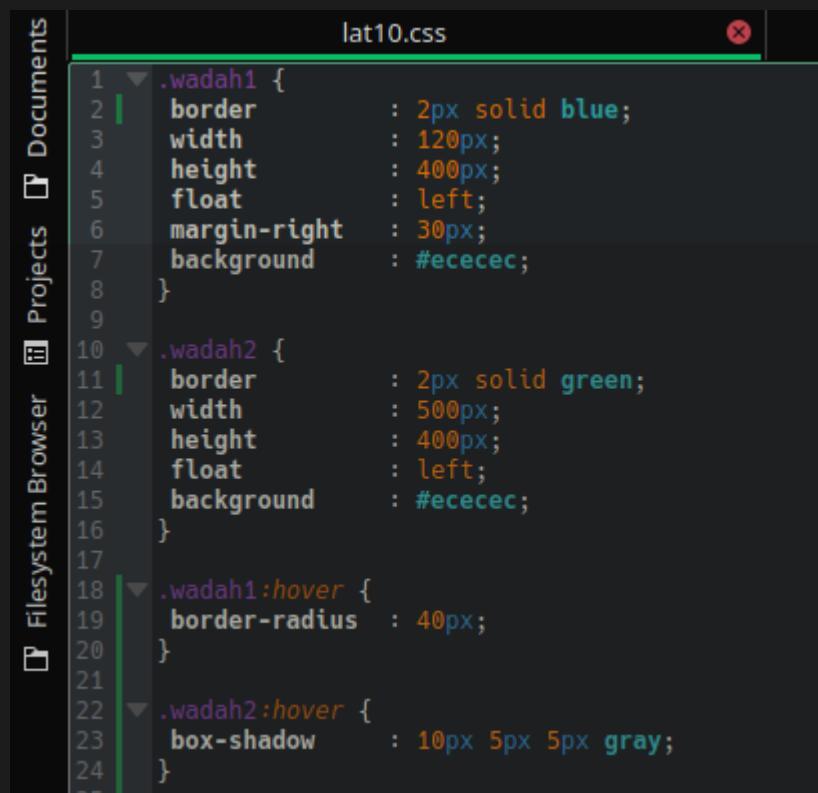
Cara menerapkan hover adalah dengan membuat ulang class css dengan nama yang sama, kemudian "hover" disematkan/disandingkan pada class yang menjadi target, antara nama class dan hover dipisah dengan titik dua (:). Contoh:

```
.wadahA {  
    .....  
}  
  
.wadahA:hover {  
    .....  
}
```

Notes:

- ✓ Konten class hover dapat berupa propertis yang berbeda dengan konten class original.
- ✓ Konten class hover dapat berupa propertis yang sama, akan tetapi dengan value yang berbeda seperti perbedaan ukuran, posisi, warna dan nilai berbeda lainnya.
- ✓ Penerapan hover cukup simpel, namun tergantung kreativitas dan imajinasi yang dimiliki, hover yang diatur sedemikian rupa dapat terlihat seperti efek profesional dan rumit.

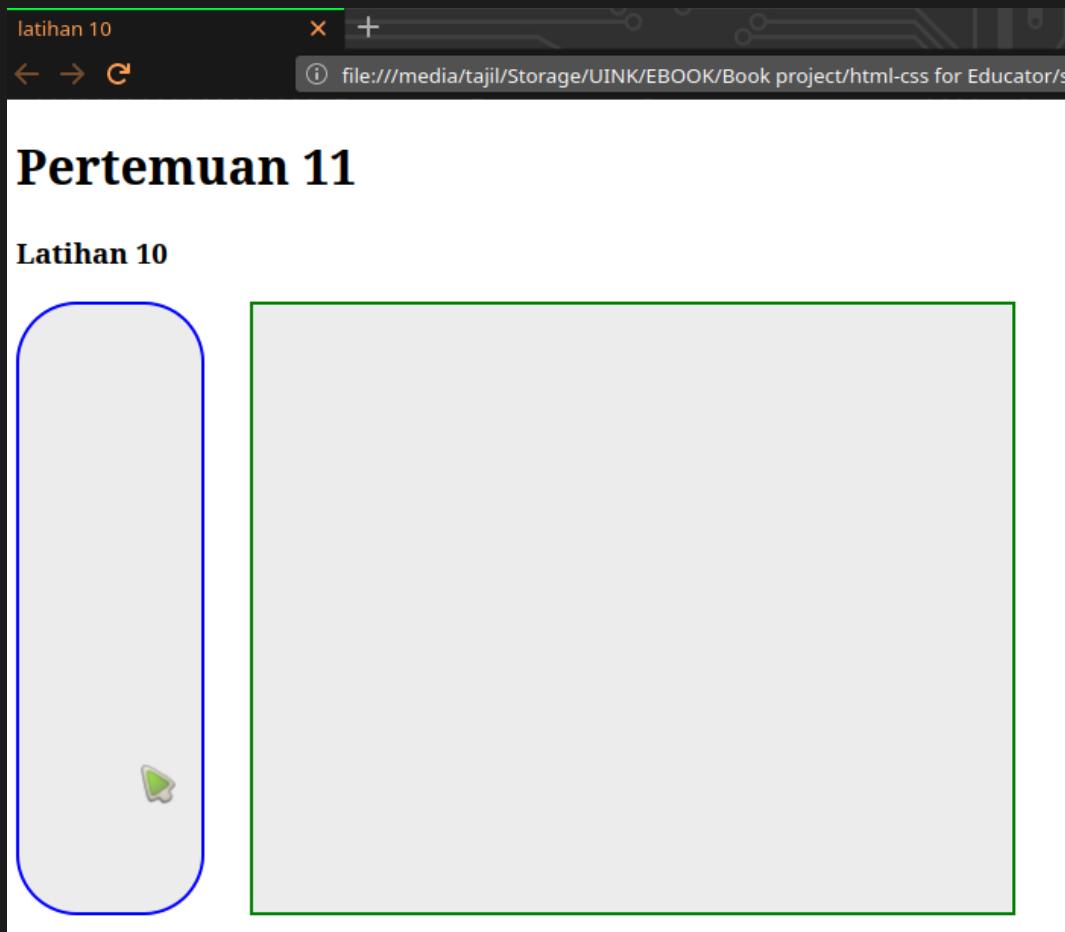
Contoh code css dengan hover:



```
lat10.css
1 .wadah1 {
2   border      : 2px solid blue;
3   width       : 120px;
4   height      : 400px;
5   float       : left;
6   margin-right: 30px;
7   background   : #ececec;
8 }
9
10 .wadah2 {
11   border      : 2px solid green;
12   width       : 500px;
13   height      : 400px;
14   float       : left;
15   background   : #ececec;
16 }
17
18 .wadah1:hover {
19   border-radius: 40px;
20 }
21
22 .wadah2:hover {
23   box-shadow   : 10px 5px 5px gray;
24 }
```

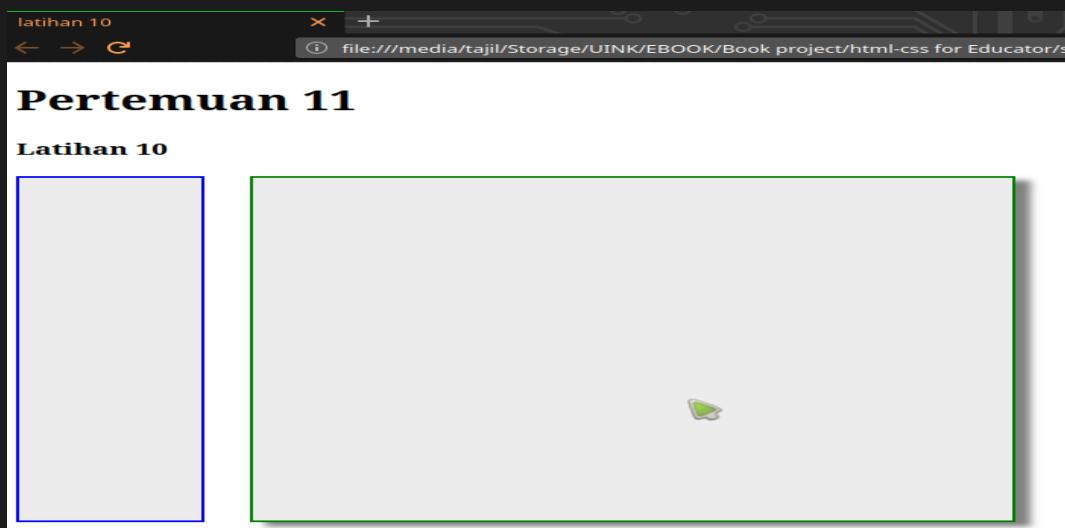
Dengan struktur css diatas, ketika pointer menunjuk div pertama dengan class “wadah1” akan terjadi penambahan sementara propertis “border-radius” dengan value “40px”. Adapun “wadah2” terjadi penambahan sementara propertis “box-shadow” dengan nilai yang tertera.

Hasil ketika div dengan wadah1 di hover:



Terlihat div tersebut memiliki propertiis border-radius ketika kondisi hovered, namun kembali seperti semula ketika unhovered.

Berikut ketika div dengan class "wadah2" di hover:



PERTEMUAN 12 Transition Effect

Transition dalam bahasa Indonesia diartikan transisi atau peralihan. Peralihan dari satu kondisi ke kondisi lain dalam satuan waktu. Pada dunia html menggunakan second atau detik dalam menentukan ukuran transisi.

Efek transisi akan kita praktekan bersama efek hover. Tak jarang, kombinasi kedua ini membuat kesan ciamik ketika berinteraksi dengan halaman web.

Pertama, siapkan dua dokumen; lat11.html dan lat11.css

Kedua, buat struktur utama html

Ketiga, sesuaikan konten title dan link pada bagian head

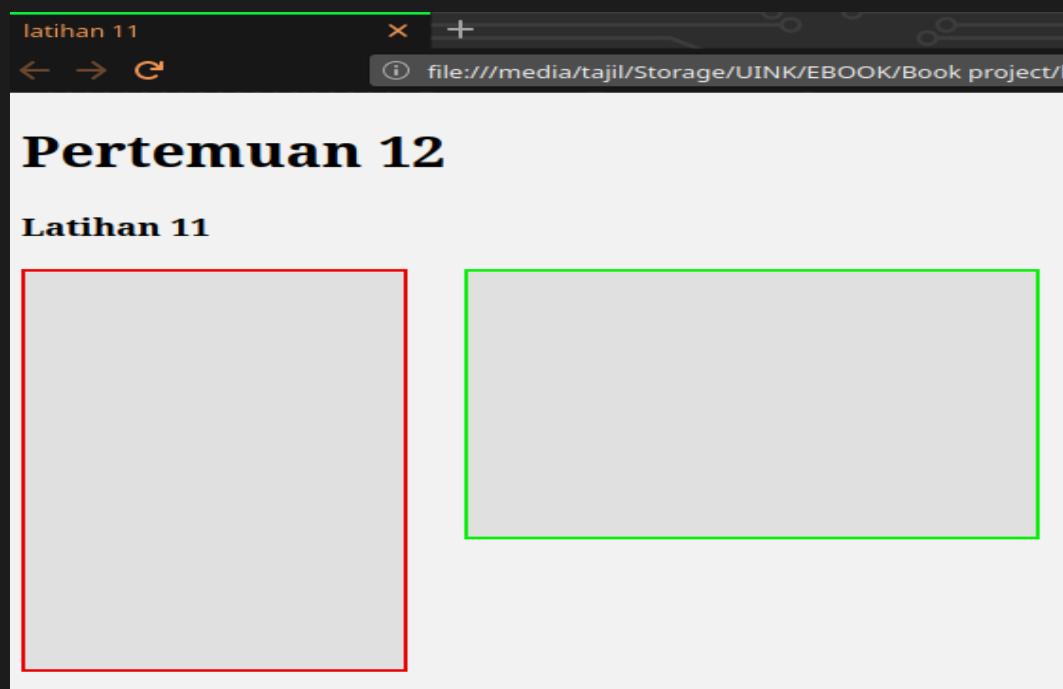
Keempat, buat dua buah objek menggunakan elemen <div>

Kelima, beri kedua div tadi class dengan nama berbeda

Keenam, atur width dan height keduanya dengan proporsional

Ketujuh, beri warna latar dan warna border

Contoh tampilan pada browser:



Code css:

```
File Edit View Projects Bookmarks Sessions
lat11.css
1 .wadah1 {
2   border      : 2px solid red;
3   width       : 200px;
4   height      : 300px;
5   float       : left;
6   margin-right: 30px;
7   background   : #ececfc;
8 }
9
10 .wadah2 {
11   border      : 2px solid lime;
12   width       : 300px;
13   height      : 200px;
14   float       : left;
15   background   : #ececfc;
16 }
17
18 .wadah1:hover {
19   border-radius: 40px;
20 }
21
22 .wadah2:hover {
23   box-shadow   : 10px 5px 5px gray;
24 }
```

Notes:

Perhatikan dengan seksama, kedua class diatas telah diberikan efek hover dengan propertis border radius untuk wadah1 dan box-shadow untuk wadah 2.

Efek hover akan seketika muncul dan dieksekusi ketika pointer mengarah ke kedua <div> tersebut.

Untuk melihat proses transisi dari class normal menuju class hover, transition jawabannya. Mari kita tambahkan propertis "transition" pada css diatas.

Apabila 'transition' hanya terdapat pada class original, maka efek transisi hanya terdapat ketika pointer menjauh dari elemen <div>, tapi tidak ketika kondisi hover.

Apabila 'transition' hanya terdapat pada class hover, maka efek transisi hanya tersedia ketika kondisi hover.

Agar efek transisi tersedia sebelum dan sesudah event hover, bubuhkan propertis ‘transition’ baik pada class original maupun class dengan hover.

```
File Edit View Projects Bookmarks Sessions
lat11.css
1 .wadah1 {
2   border          : 2px solid red;
3   width           : 200px;
4   height          : 300px;
5   float           : left;
6   margin-right    : 30px;
7   background      : #ececec;
8 }
9
10 .wadah2 {
11   border          : 2px solid lime;
12   width           : 300px;
13   height          : 200px;
14   float           : left;
15   background      : #ececec;
16 }
17
18 .wadah1:hover {
19   border-radius   : 40px;
20   transition      : 1s;
21 }
22
23 .wadah2:hover {
24   box-shadow      : 10px 5px 5px gray;
25   transition      : 1s;
26 }
27
```

Nah, sekarang coba lakukan hover pada kedua elemen <div>, lalu perhatikan dalam rentang satu secon/detik terjadi perubahan propertis elemen sebagaimana dibubuhkan pada class hover.

Transition diisi dengan value angka+second, contoh diatas kita menggunakan “1s” sebagai value atas transition. Semakin besar angka, semakin transisi berjalan lambat.

Standar efek transisi untuk penggunaan profesional namun masih memperhatikan kenyamanan user adalah 0.7s atau 0,7 detik, contoh:

“ transition : 0.7s; ”.

PERTEMUAN 13 Simple Persentation Page

Pembaca yang budiman pasti sudah tidak asing dengan persentasi dan bagaimana membuat dokumen khusus untuk digunakan dalam sesi persentasi. Baik untuk tujuan akademik, sosial atau bentuk aktivitas lain, persentasi adalah salah satu metode dalam menyampaikan ide dan pemikiran kepada khalayak ramai.

Membuat dokumen persentasi cukup mudah dengan banyaknya produk aplikasi persentasi baik jenis komersial atau berbayar maupun jenis gratis (free or opensource software). Namun, bagaimana dengan html dan css, apakah mampu menyediakan infrastruktur untuk “sekedar” meracik data untuk tujuan ditampilkan dalam suatu persentasi ?

Setelah mempelajari materi dan praktek tentang anchor <a>, border, border-radius, hover serta transition efek, maka selanjutnya kita sudah cukup layak untuk mendesain projek sederhana seperti membuat halaman persentasi dengan html dan css only.

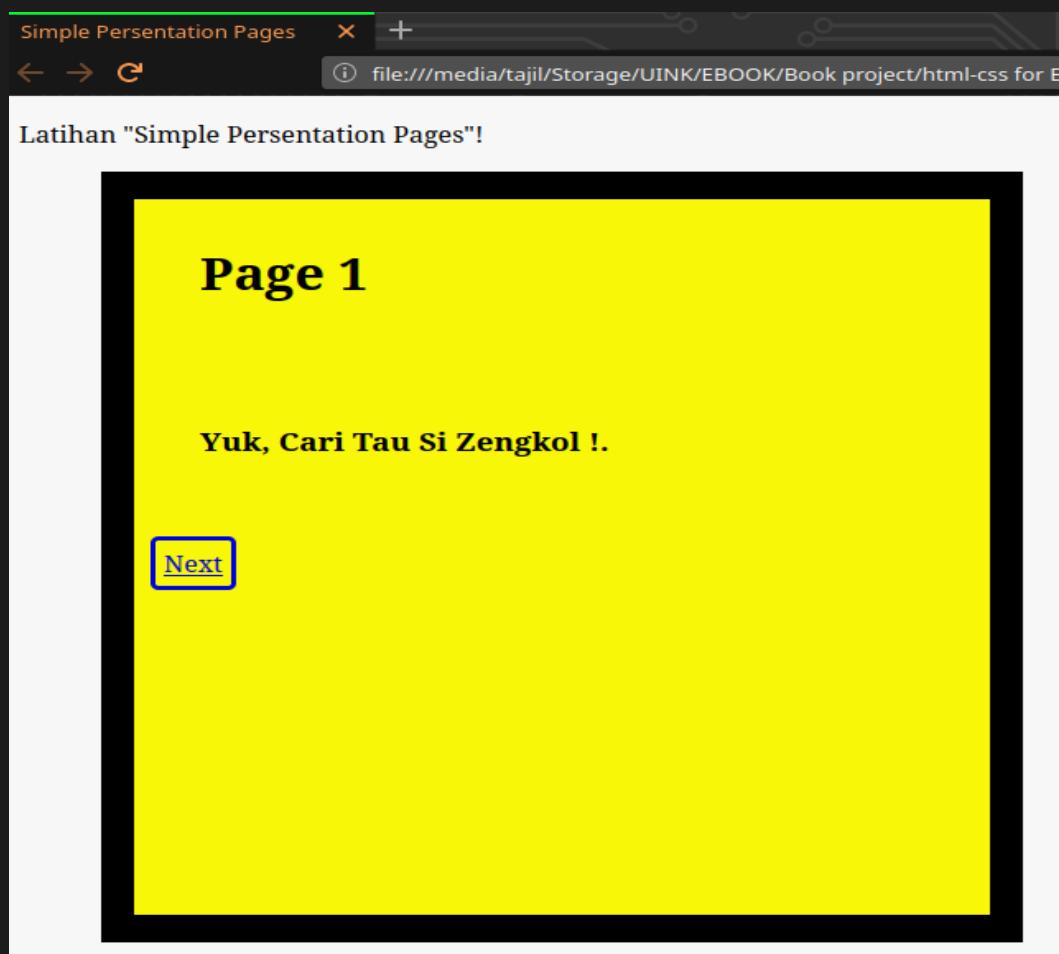
Hemat penulis, di tingkat sekolah maupun universitas, persentasi via web browser masih sangat jarang dilakukan. Memang sangat dimaklumi, boleh jadi belum sepraktis menggunakan produk aplikasi membuat dokumen persentasi dan memang perlu memiliki pengetahuan dasar soal html dan css. Setidaknya kedua hal tersebut yang menjadi penghalang utama menurut penulis.

Adapun persiapan praktek pada pertemuan ini adalah :

1. Buat tiga empat dokumen; lat12html, lat12.css, page2.html dan page3.html.
2. Buat struktur utama pada tiga dokumen html diatas.
3. Sesuaikan konten title dan href link dibagian head.
4. Tiga dokumen html dihubungkan dengan satu dokumen css (lat12.css). Boleh juga apabila ingin berimprovisasi menggunakan lebih dari satu dokumen css.

5. Isi konten <body> setiap dokumen html dengan konten yang agak berbeda, karena lat12.html adalah halaman pertama, page2.html sebagai halaman kedua dan page3.html sebagai halaman terakhir persentasi kita.
6. Beri tautan dari dan ke halaman selanjutnya serta halaman sebelumnya tentunya dengan menggunakan elemen anchor (<a>).
7. Beri dekorasi sesuai selera pada setiap halaman via lat12.css.
8. Buka di browser, tes dan ber-eksperimenlah ! Jangan mudah menyerah. Dibawah akan diperlihatkan contoh sederhana persentasi via html-css berserta sumber code. Upayakan tidak melihat sumber code, kecuali buntu !

Contoh halaman pertama:



Contoh halaman kedua:

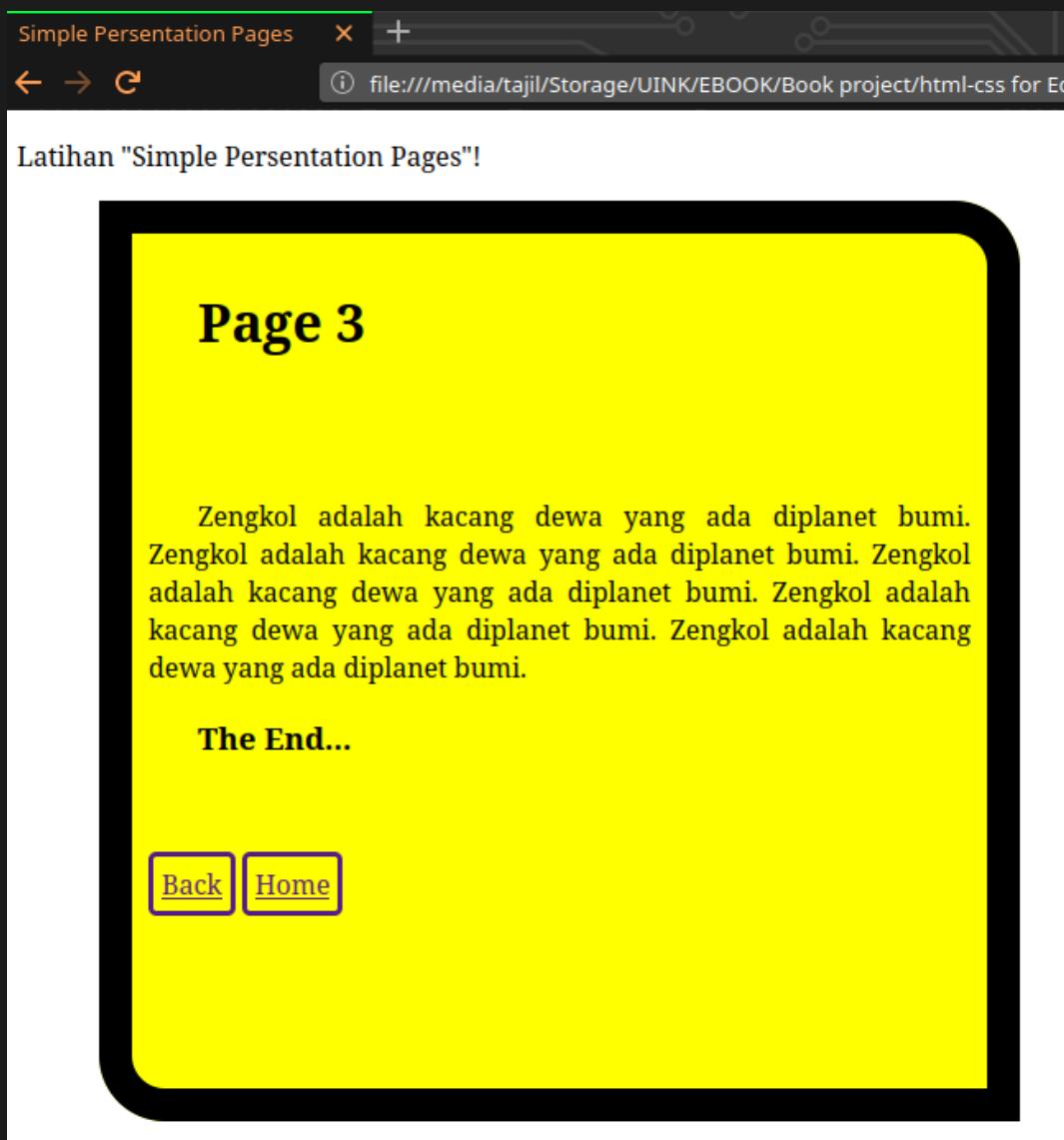
The screenshot shows a presentation slide titled "Page 2". The slide has a yellow background and a black border. At the top left, it says "Latihan 'Simple Persentation Pages'!". In the center, there is a photograph of several dark brown, shiny Zengkol (chestnut) fruits. To the right of the image, there is a block of text that repeats the sentence "Zengkol adalah kacang dewa yang ada diplanet bumi." five times. At the bottom right of the slide, there are two buttons: "Back" and "Next". The entire slide is set against a white background.

Notes:

Tautan halaman pertama hanya terdapat tombol "Next". Pada halaman kedua terdapat "Next" dan "Back". Tentu ini hal logis, karena pada halaman pertama tidak masuk akan ada tombol "Back" bukan ?

Halaman terakhir (page 3) seperti pada contoh dibawah nanti mengandung tombol "Back" dan "Home". Back untuk kembali satu halaman, sedangkan home untuk langsung kembali ke halaman pertama.

Contoh halaman ketiga (terakhir):



Notes:

Dokumen persentasi yang telah dibuat dapat dijadikan sebagai sebuah "template" ketika nanti kita akan membuat dokumen persentasi dengan materi berbeda.

Dengan cara ini tentu lebih mudah, karena hal-hal mendasar sudah tersedia, tinggal bagaimana kreativitas kita mendesain dan menghias halaman demi halaman.

Code html lat12.html:

```
lat12.html lat12.css

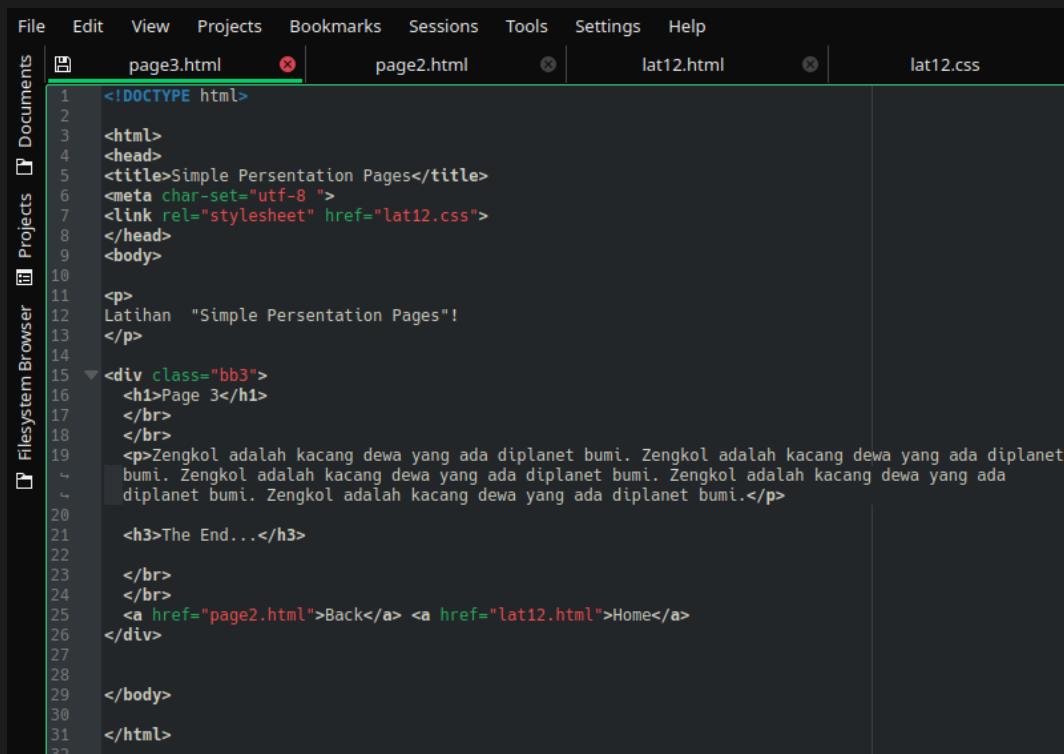
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5 <title>Simple Persentation Pages</title>
6 <meta charset="utf-8">
7 <link rel="stylesheet" href="lat12.css">
8 </head>
9 <body>
10
11 <p>
12 Latihan "Simple Persentation Pages"!
13 </p>
14
15 <div class="bb">
16   <h1>Page 1</h1>
17   <br>
18   <br>
19   <h3>Yuk, Cari Tau Si Zengkol !.</h3>
20
21   <br>
22   <br>
23   <a href="page2.html">Next</a>
24 </div>
25
26 </body>
27
28
29 </html>
```

Code html page2.html:

```
File Edit View Projects Bookmarks Sessions Tools Settings Help
Documents page2.html lat12.html lat12.css page3.html
Projects Filesystem Browser

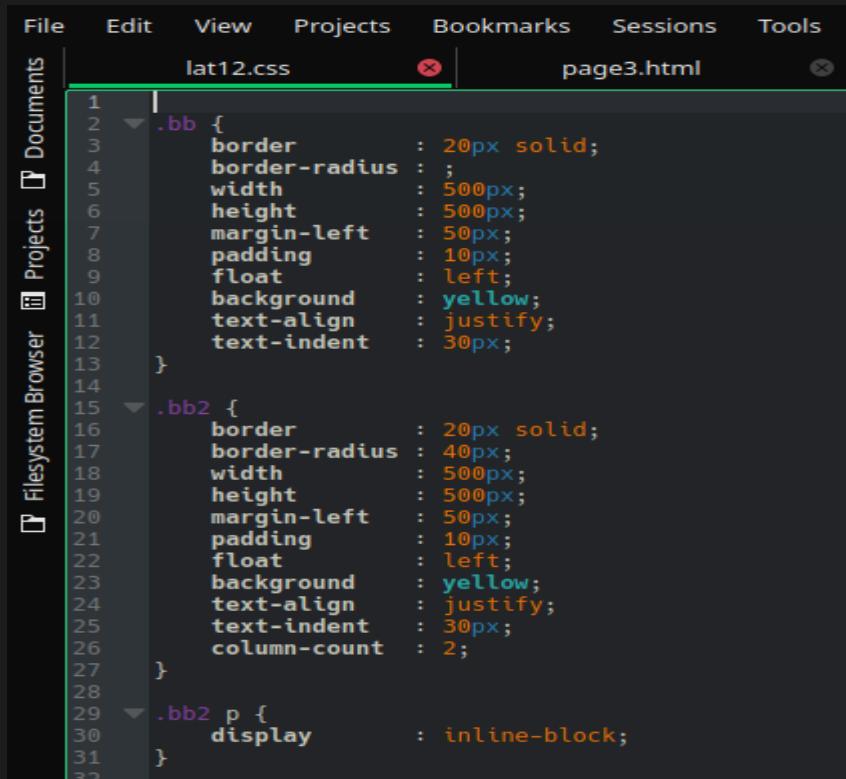
1 <!DOCTYPE html>
2
3 <html>
4 <head>
5 <title>Simple Persentation Pages</title>
6 <meta char-set="utf-8" >
7 <link rel="stylesheet" href="lat12.css">
8 </head>
9 <body>
10
11 <p>
12 Latihan "Simple Persentation Pages"!
13 </p>
14
15 <div class="bb2">
16 <h1>Page 2</h1>
17 
18 <p>Zengkol adalah kacang dewa yang ada diplanet bumi. Zengkol adalah kacang dewa yang ada diplanet bumi.</p>
19
20 <br>
21 <br>
22 <a href="lat12.html">Back</a> <a href="page3.html">Next</a>
23 </div>
24
25
26 </body>
27
28 </html>
```

Code html page3.html:



```
File Edit View Projects Bookmarks Sessions Tools Settings Help
Documents page3.html × | page2.html × | lat12.html × | lat12.css
Projects Filesystem Browser
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Simple Presentation Pages</title>
5 <meta charset="utf-8" >
6 <link rel="stylesheet" href="lat12.css">
7 </head>
8 <body>
9
10 <p>
11 Latihan "Simple Presentation Pages"!
12 </p>
13
14 <div class="bb3">
15   <h1>Page 3</h1>
16   <br>
17   <br>
18   <p>Zengkol adalah kacang dewa yang ada diplanet bumi. Zengkol adalah kacang dewa yang ada diplanet bumi.</p>
19
20   <h3>The End...</h3>
21
22   <br>
23   <br>
24   <a href="page2.html">Back</a> <a href="lat12.html">Home</a>
25 </div>
26
27
28 </body>
29
30 </html>
```

Code lat12.css part 1 :



```
File Edit View Projects Bookmarks Sessions Tools
Documents lat12.css × | page3.html ×
Projects Filesystem Browser
1 .bb {
2   border          : 20px solid;
3   border-radius   : ;
4   width           : 500px;
5   height          : 500px;
6   margin-left     : 50px;
7   padding         : 10px;
8   float           : left;
9   background      : yellow;
10  text-align      : justify;
11  text-indent     : 30px;
12 }
13
14 .bb2 {
15   border          : 20px solid;
16   border-radius   : 40px;
17   width           : 500px;
18   height          : 500px;
19   margin-left     : 50px;
20   padding         : 10px;
21   float           : left;
22   background      : yellow;
23   text-align      : justify;
24   text-indent     : 30px;
25   column-count    : 2;
26 }
27
28 .bb2 p {
29   display         : inline-block;
30 }
31
32 }
```

Code lat12.css part 2 :

```
32 
33   .bb3 {
34     border        : 20px solid;
35     border-radius: 0 40px 0 40px;
36     width         : 500px;
37     height        : 500px;
38     margin-left   : 50px;
39     padding       : 10px;
40     float         : left;
41     background    : yellow;
42     text-align    : justify;
43     text-indent   : 30px;
44   }
45 
46   a {
47     border        : 3px solid;
48     padding       : 5px;
49     border-radius: 5px;
50     transition    : 0.8s;
51   }
52 
53   a:hover {
54     border-radius: 30px;
55     transition    : 0.8s;
56   }
57 
```

PERTEMUAN 14 Box Sizing

Dahulu, sebelum CSS se-mature dewasa ini, para developer suka dibuat pusing untuk mengkondisikan ukuran elemen yang telah diberi ‘padding’. Elemen html tanpa padding dengan ukuran misal 200px x 200px , tidak sama dengan element html ber-padding pada ukuran yang sama.

Pembahasan kali ini juga otomatis melibatkan soal ‘margin’ dan ‘padding’. Secara simpel, margin adalah batas luar suatu elemen, sedangkan padding adalah batas dalam suatu elemen. Batas disini berlaku untuk empat arah; kanan, kiri, atas dan bawah. Baik margin maupun padding memiliki shortcut penulisan untuk sekaligus mengatur keempat arah tersebut atau mengatur secara individual.

Kembali pada paragraf awal, perubahan ukuran width dan height pada elemen karena telah diberi ‘padding’ cukup membuat para developer web memeras otak kala itu. Perubahan ukuran, berarti perubahan pada layout, perubahan pada layout sangat mungkin berdampak pada elemen dan konten lain pada halaman yang sama.

Beruntung, css dewasa ini telah mewadahi hal tersebut dengan properti “box-sizing”. Properti ini memiliki beberapa opsi value seperti border-box (yang akan dipraktekan disini), content-box dan lainnya. Untuk mengetahui info detail dan contoh praktik selain value “border-box, silahkan lihat pada web w3cschool.com atau sumber lain yang tersedia.

Persiapan praktik pertemuan 14 :

1. Buat dua dokumen; lat13.html dan lat13.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link pada head
4. Buat dua buah kotak dengan ukuran proporsional menggunakan elemen <div>. Beri ukuran yang sama atas kedua div tersebut.
5. Beri class, boleh class yang sama atau berbeda.
6. Atur ukuran mereka serta tampilan bordernya.

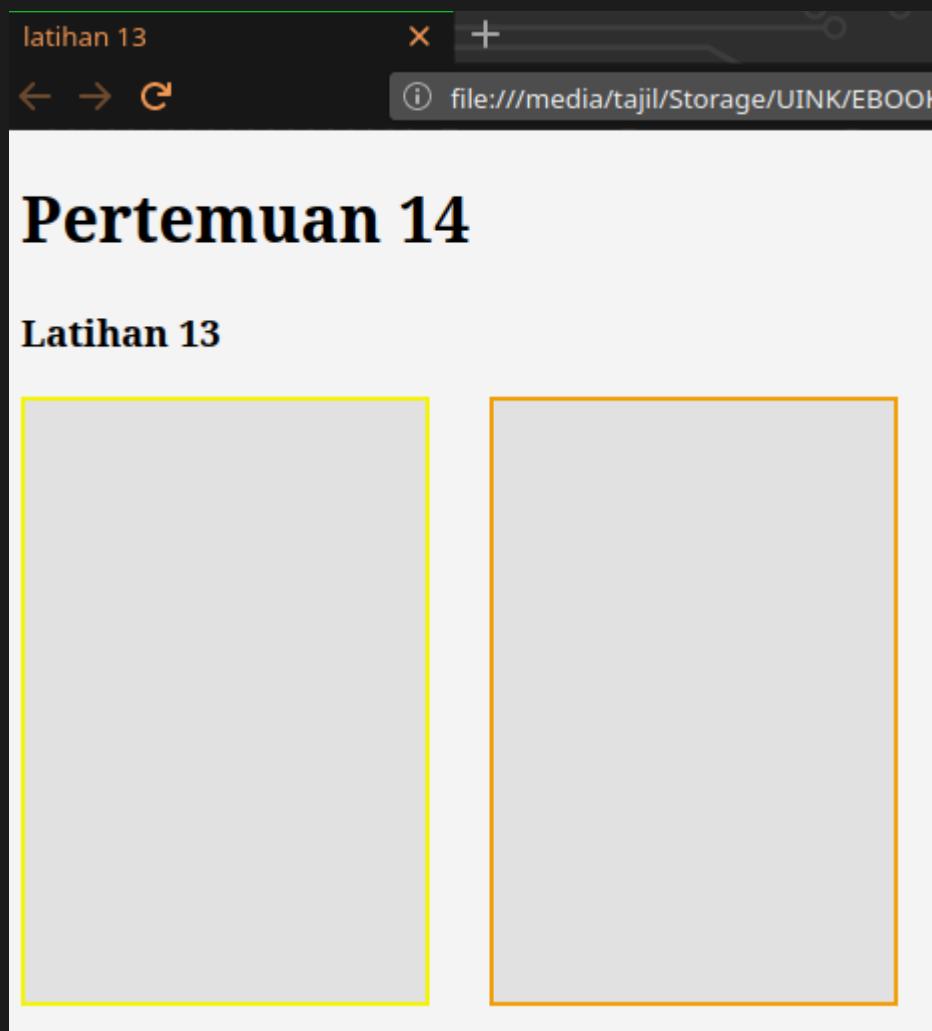
Contoh code html :

```
lat13.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 13</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat13.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 14</h1>
11  <h3>Latihan 13</h3>
12
13  <div class="wadah1"></div>
14  <div class="wadah2"></div>
15
16 </body>
17 </html>
18
```

Contoh code css :

```
lat13.html
1 .wadah1 {
2   border      : 2px solid yellow;
3   width       : 200px;
4   height      : 300px;
5   float       : left;
6   margin-right: 30px;
7   background   : #ececec;
8 }
9
10 .wadah2 {
11   border      : 2px solid orange;
12   width       : 200px;
13   height      : 300px;
14   float       : left;
15   background   : #ececec;
16 }
17
```

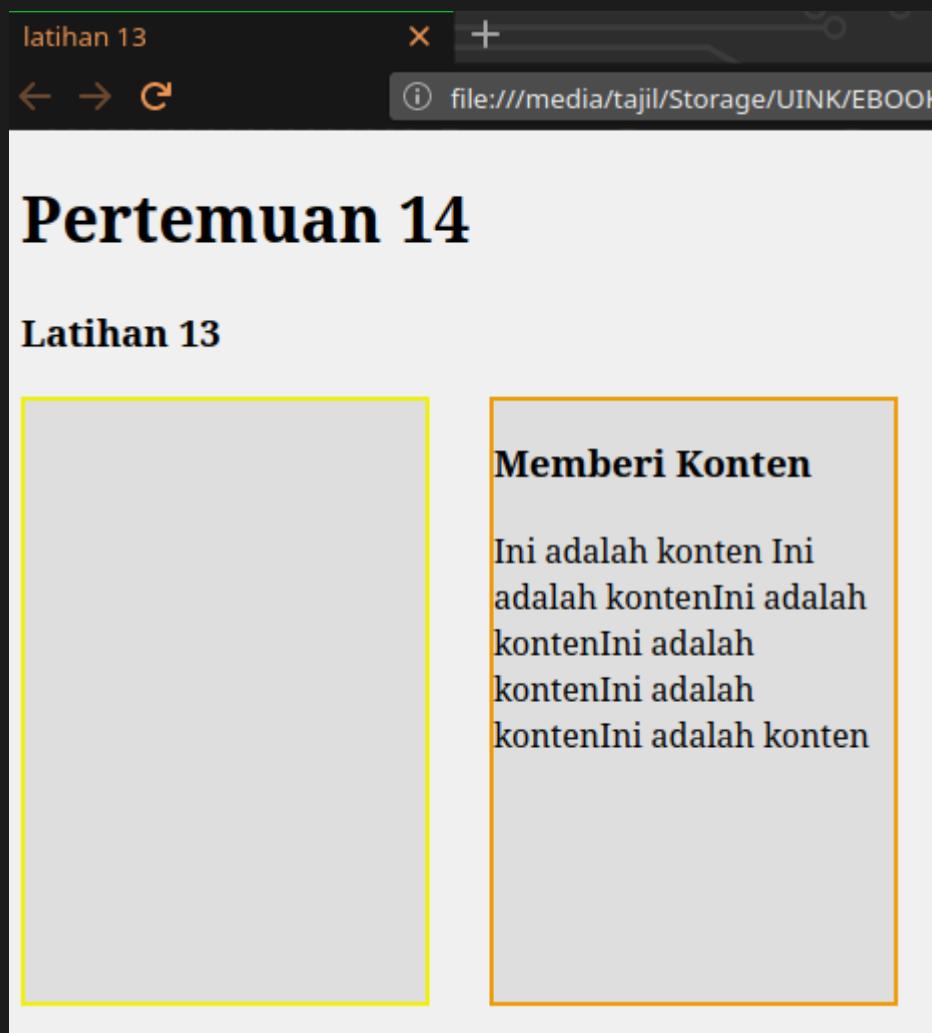
Tampilan pada browser :



Notes:

- Perhatikan dengan sekasama, ukuran tinggi dan lebar kedua elemen <div> adalah sama. Keduanya juga belum diberi konten (isi).
- Untuk itu, langkah selanjutnya adalah memberi konten pada salah satu <div> agar lebih mudah memahami konsep 'padding'.
- Mari kita beri konten div sebelah kanan (orange) dengan sejumlah paragraf seperti berikut:

Tampilan pada di browser :



Code html :

```
lat13.html
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>latihan 13</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="lat13.css">
7   </head>
8   <body>
9
10  <h1>Pertemuan 14</h1>
11  <h3>Latihan 13</h3>
12
13  <div class="wadah1"></div>
14
15  <div class="wadah2">
16    <h3>Memberi Konten</h3>
17    <p>Ini adalah konten Ini adalah kontenIni adalah kontenIni adalah kontenIni adalah konten</p>
18  </div>
19
20  </body>
21  </html>
```

```
lat13.css
```

```
.wadah1 {
  border: 2px solid yellow;
  padding: 10px;
}

.wadah2 {
  border: 2px solid orange;
  padding: 10px;
}
```

Notes:

Perhatikan, ukuran kedua div tidak berubah, walau div sebelah kanan sudah diberi sejumlah konten, ukuran width dan height keduanya tetap pada 200x200 pixel.

Hal berbeda akan terjadi apabila kita menerapkan “padding” pada salah satu div. Tentu target kita adalah div sebelah kanan (orange).

Mari kita beri propertis padding sebagai berikut:



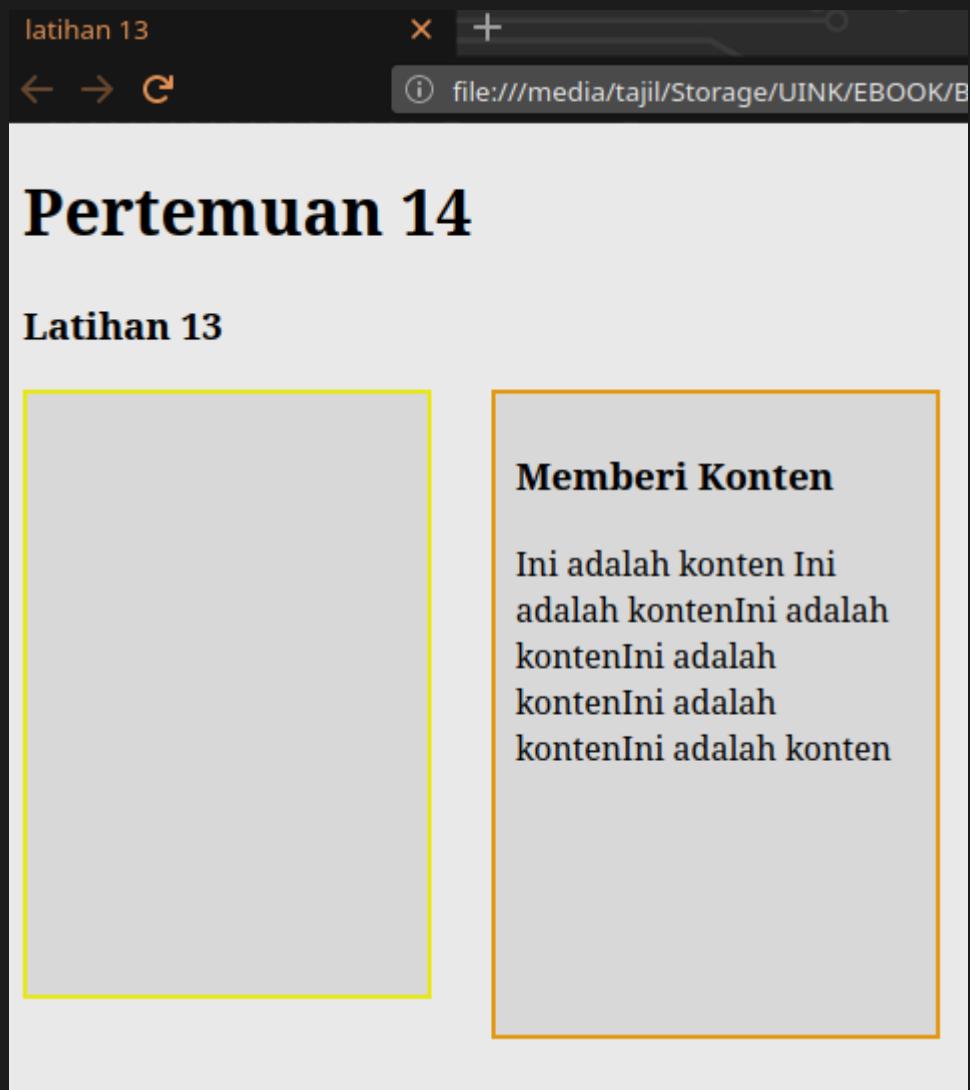
```
lat13.html
1 .wadah1 {
2   border      : 2px solid yellow;
3   width       : 200px;
4   height      : 300px;
5   float       : left;
6   margin-right: 30px;
7   background   : #ececec;
8 }
9
10 .wadah2 {
11   border      : 2px solid orange;
12   width       : 200px;
13   height      : 300px;
14   float       : left;
15   background   : #ececec;
16   padding     : 10px;
17 }
```

Notes:

Penambahan “padding” pada class ‘.wadah2’ artinya kita memberi batas dalam sebanyak 10 pixel dan berlaku untuk empat arah; kanan, kiri, atas dan bawah.

Sekarang save, lalu refresh browser. Perhatikan perubahan apa yang terjadi !

Tampilan pada browser :



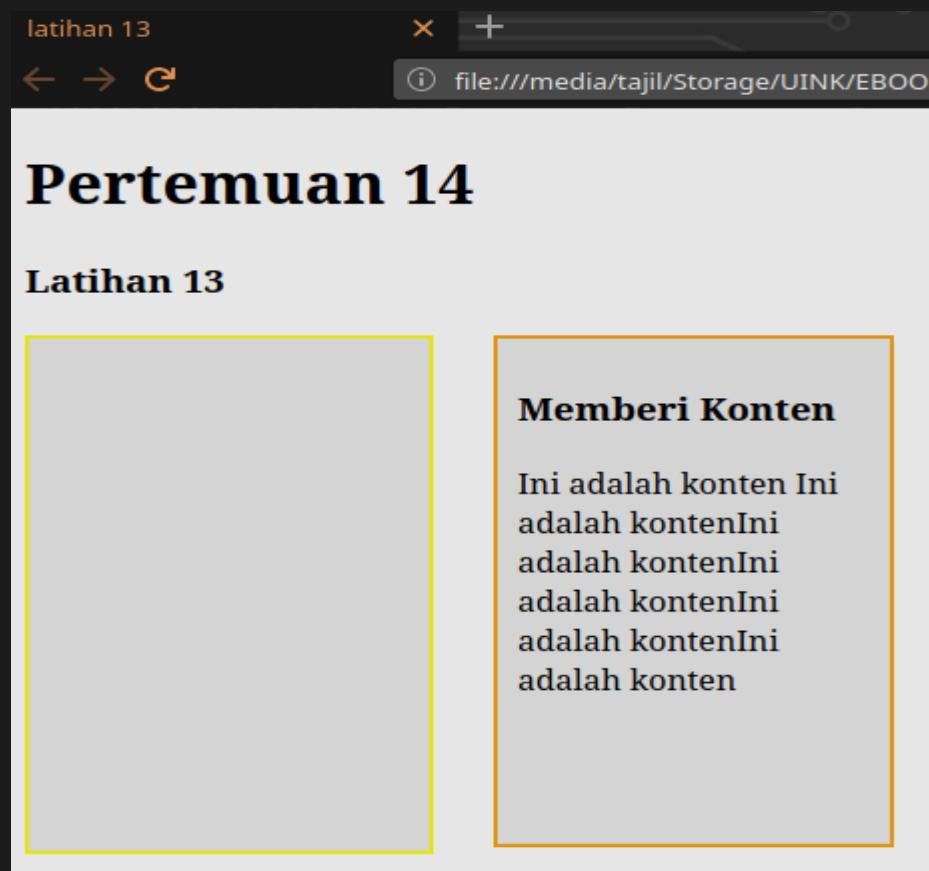
Notes:

- ◆ Terlihat jelas terdapat perubahan width dan height pada div yang telah diberi properti 'padding'. Untuk mengatasi hal tersebut, kita harus membubuhkan properti 'box-sizing' pada elemen yang bersangkutan.
- ◆ Memang, cara ini tidak 100% mengatasi kasus ini, tapi secara sekilas berhasil tidak membuat perbedaan yang sangat mencolok antara elemen tak ber-padding dengan elemen ber-padding.
- ◆ Mari kita tambah properti tersebut sebagai berikut :

code css :

```
File Edit View Projects Bookmarks Sessions
lat13.html
1 .wadah1 {
2   border : 2px solid yellow;
3   width : 200px;
4   height : 300px;
5   float : left;
6   margin-right : 30px;
7   background : #ececec;
8 }
9
10 .wadah2 {
11   border : 2px solid orange;
12   box-sizing : border-box;
13   width : 200px;
14   height : 300px;
15   float : left;
16   background : #ececec;
17   padding : 10px;
18 }
```

Tampilan di browser :



PERTEMUAN 15 Flexbox

‘Flex’ adalah istilah sekaligus kata kunci dalam dunia css bentuk dari kepanjangan flexible. Flexbox adalah istilah kontainer yang dapat diatur (diposisikan) dengan lebih mudah. Fitur ini hadir ketika CSS3 rilis.

Flexbox hadir semakin “memanjakan” para web developer dan web disainer. Dengan flexbox, kontainer yang terdiri atas sub-sub kontainer dengan lebih mudah diposisikan horizontal atau secara vertikal. Flexbox juga menyuguhkan beberapa opsi pensejajaran dan jarak antara sub-sub kontainer tersebut.

Pelajari detail flexbox pada sumber-sumber tersedia, buku ini hanya membahas porsi kecil dari propertis flexbox. Selain flexbox, css3 juga memiliki fasilitas grid yang mana lebih modern ketimbang flexbox. Grid juga tidak dibahas dibuku sederhana ini.

Langsung kita menuju tkp dengan melakukan langkah persiapan sebagai berikut:

1. Buat dua dokumen; lat14.html dan lat14.css.
2. Buat struktur utama html.
3. sesuaikan konten title dan link dibagian head.
4. Buat sebuah kontainer menggunakan <div> dengan ukuran 500x500 pixel, tampilkan bordernya.
5. Tambahkan empat sub kontainer didalam div tersebut, juga menggunakan elemen <div> dengan ukuran @100x100 pixel serta tampilkan bordernya.
6. Agar belajar lebih colorful, beri warna berbeda tiap kontainer!
7. Beri propertis “margin: auto;” pada setiap sub kontainer, agar tersedia jarak antara yang proporsional.

Code html:

```
lat14.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>latihan 14</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="lat14.css">
7 </head>
8 <body>
9
10    <h1>Pertemuan 15</h1>
11    <h3>Latihan 14</h3>
12
13    <div class="wadah">
14
15        <div class="sub"></div>
16        <div class="sub"></div>
17        <div class="sub"></div>
18        <div class="sub"></div>
19
20    </div>
21
22 </body>
23 </html>
```

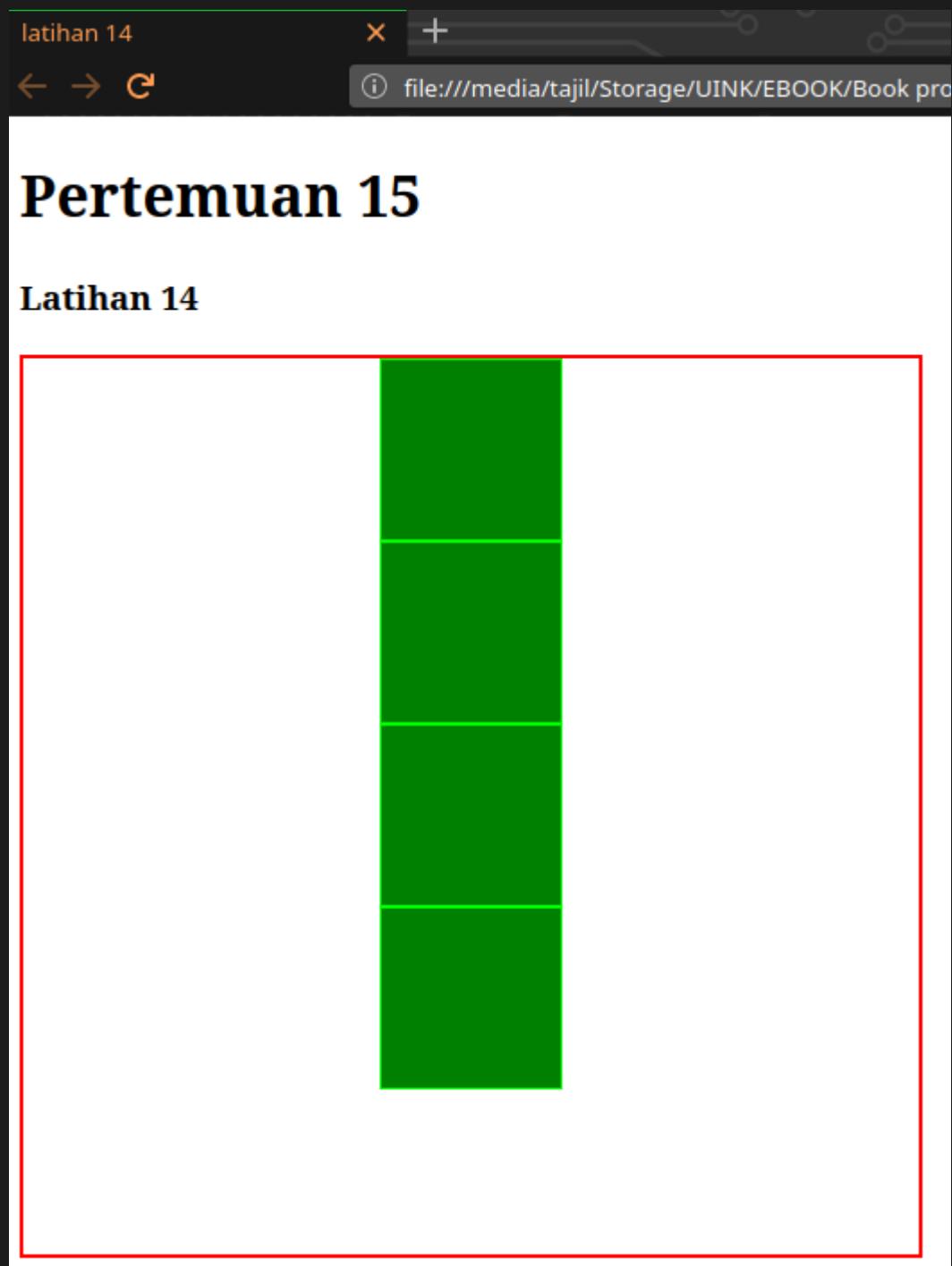
Code css:

```
lat14.css
1 .wadah {
2     border          : 2px solid red;
3     width           : 500px;
4     height          : 500px;
5 }
6
7 .sub {
8     border          : 1px solid lime;
9     width           : 100px;
10    height          : 100px;
11    background      : green;
12    margin          : auto;
13 }
14
```

Notes:

Biasakan selalu melakukan indentasi pada sub-sub elemen html, selain lebih rapih, akan lebih mudah untuk keperluan jangka panjang seperti pengecekan dan pengembangan.

Tampilan pada browser:



Notes:

properti "margin: auto;" membuat setiap sub elemen <div> diposisikan ditengah secara horizontal, tidak secara vertikal.

Untuk memberi propertis flexbox pada kontainer, kita harus membubuhkan propertis “display” dengan value “flex” di css. Setelah diberi propertis tersebut, kontainer dapat menggunakan “flex-direction” dan “flex-wrap”.

Flex direction memiliki dua opsi value, row atau column. Row berarti pensejajaran secara baris alias horizontal. Column berarti pensejajaran secara kolom alias vertikal. Secara default apabila kita tidak menentukan flex direction, maka direction akan di set sebagai ‘row’.

Flex wrap memiliki dua opsi value, wrap (default) dan nowrap. Apabila kita tidak membubuhkan propertis ini, maka flexbox secara default memiliki nilai ‘wrap’, artinya setiap sub elemen akan secara otomatis disimpan digaris baru ketika telah mencapai batas border kontainer induk. Value ‘wrap’ adalah nilai default apabila kita tidak mengatur secara manual.

Apabila value diset pada ‘nowrap’, maka sub-sub elemen didalam kontainer flexbox baik mode row atau mode column akan terus ditempatkan/ditambahkan kesamping kanan (row mode) dan kearah bawah (column mode), ketika telah mencapai batas pinggir kontainer flexbox, sub elemen dan sub elemen baru tidak akan diposisikan garis baru, melainkan tetap seperti itu dengan otomatis mengurangi size sub elemen.

Hal ini seperti ‘dual edge’ sword alias pedang bermata dua, disatu sisi baik karena auto memberikan pengaturan fleksibel otomatis ketika konten ditambah. Disisilain bagi yang ingin mempertahankan ukuran sub elemen akan sedikit terganggu.

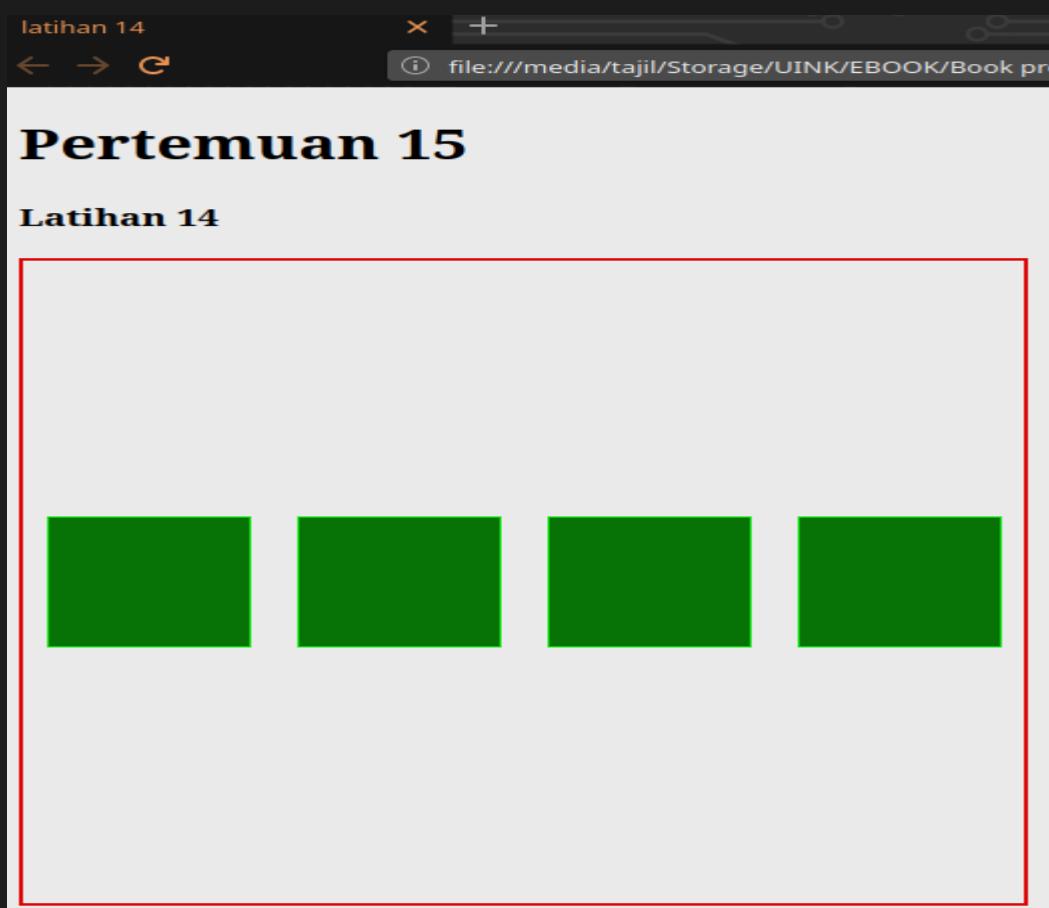
Mari kita buat kontainer menjadi “flexbox” dengan menambahkan beberapa propertis flexbox diatas pada css nya.

Code css :

The screenshot shows a code editor window with a dark theme. The menu bar includes File, Edit, View, Projects, Bookmarks, and Sessions. A sidebar on the left lists Documents, Projects, and System Browser. The main area displays the file 'lat14.css' containing the following CSS code:

```
1 .wadah {  
2     border: 2px solid red;  
3     width: 500px;  
4     height: 500px;  
5     display: flex;  
6     flex-direction: row;  
7     flex-wrap: wrap;  
8 }  
9  
10 .sub {  
11     border: 1px solid lime;  
12     width: 100px;  
13     height: 100px;  
14     background: green;  
15     margin: auto;  
16 }  
17
```

Hasil pada browser :



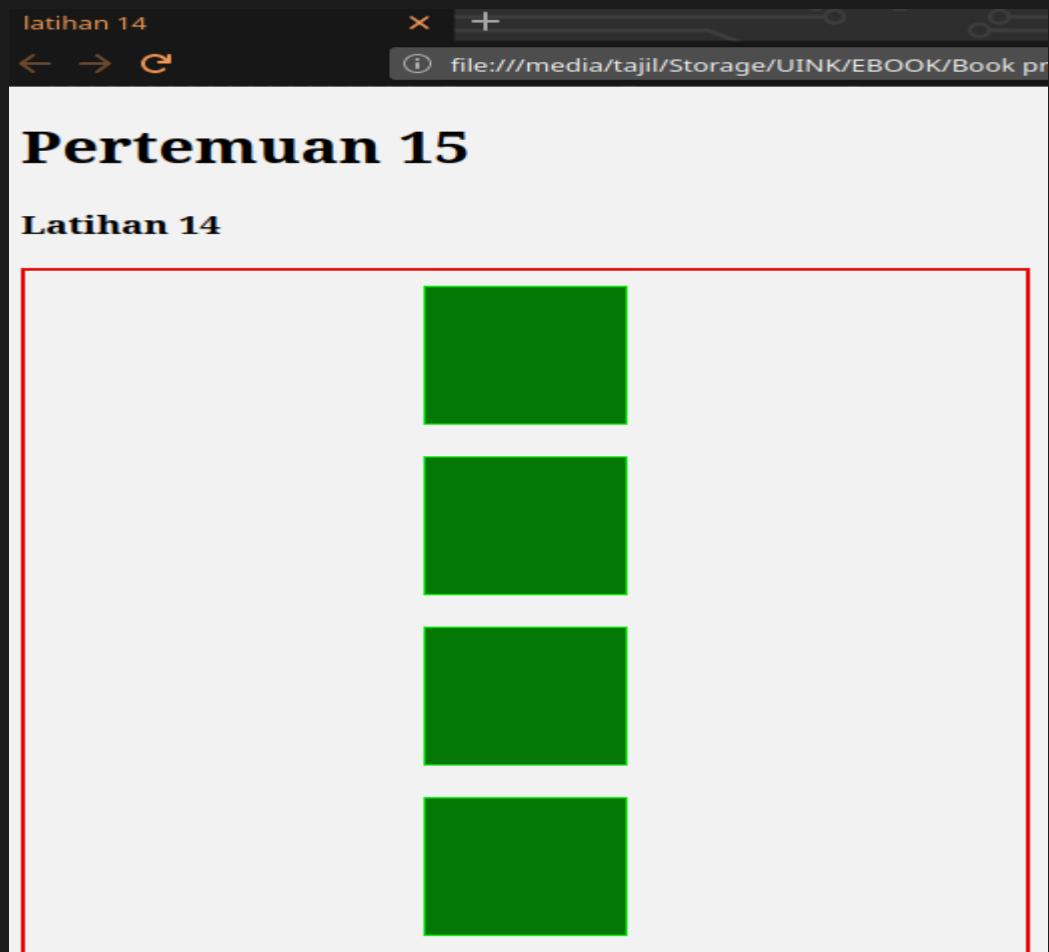
Notes:

Sekarang semua sub elemen <div> ditempatkan secara row mode dan memiliki jarak antara dari dan ke per elemen serta posisi mereka tepat ditengah kontainer flexbox.

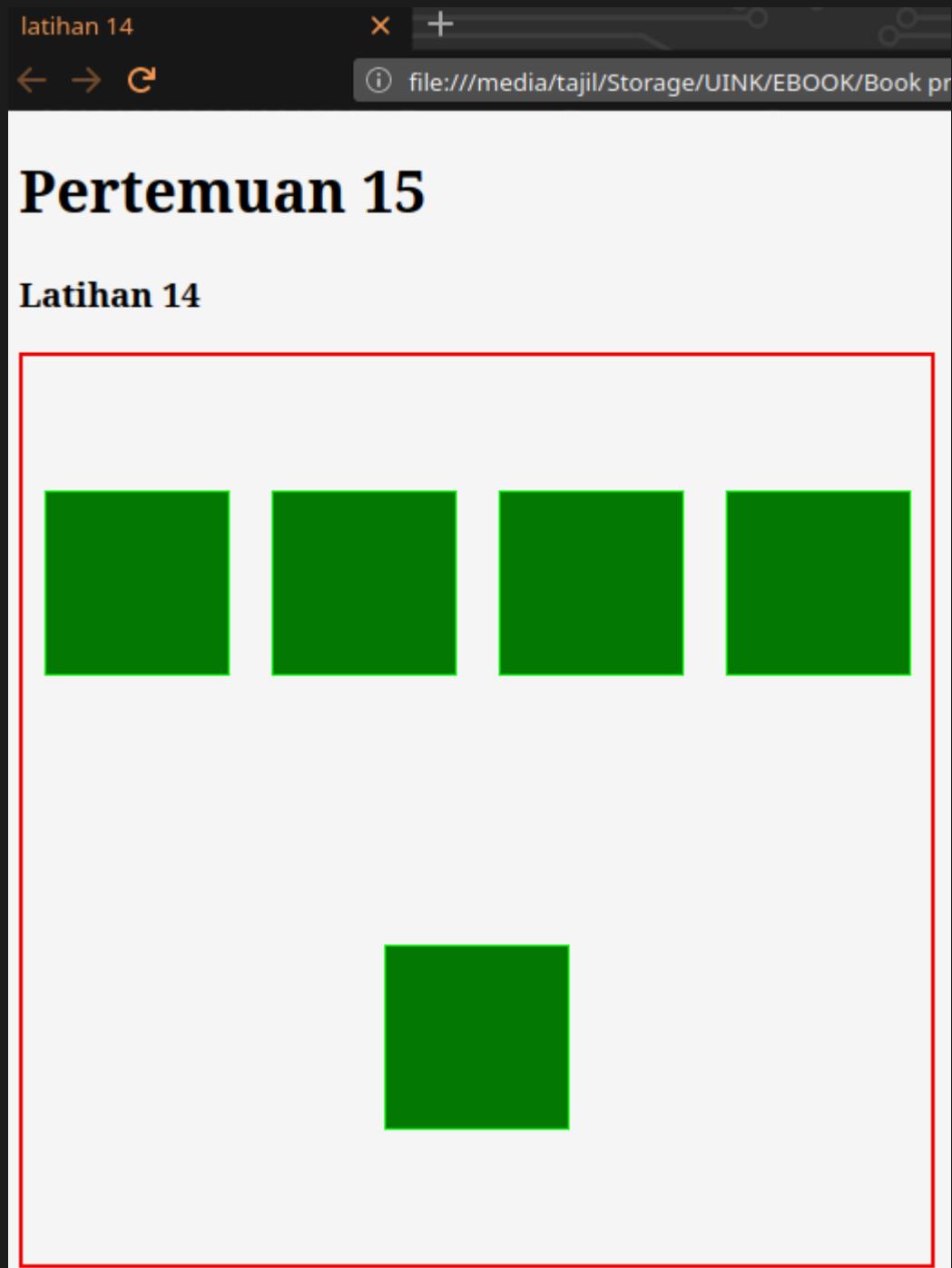
Tolong dicamkan! Yang membuat sub-sub elemen diposisikan persis diengah dan memiliki jarak antara per elemen adalah properti “margin: auto;”.

Posisi default tanpa properti “margin: auto;” adalah muali dari kiri atas layar. Ini berlaku untuk non flexbox dan bagi semua elemen html.

Selanjutnya cobalah merubah flex-direction menjadi ‘column’, sehingga menghasilkan tampilan sebagai berikut:

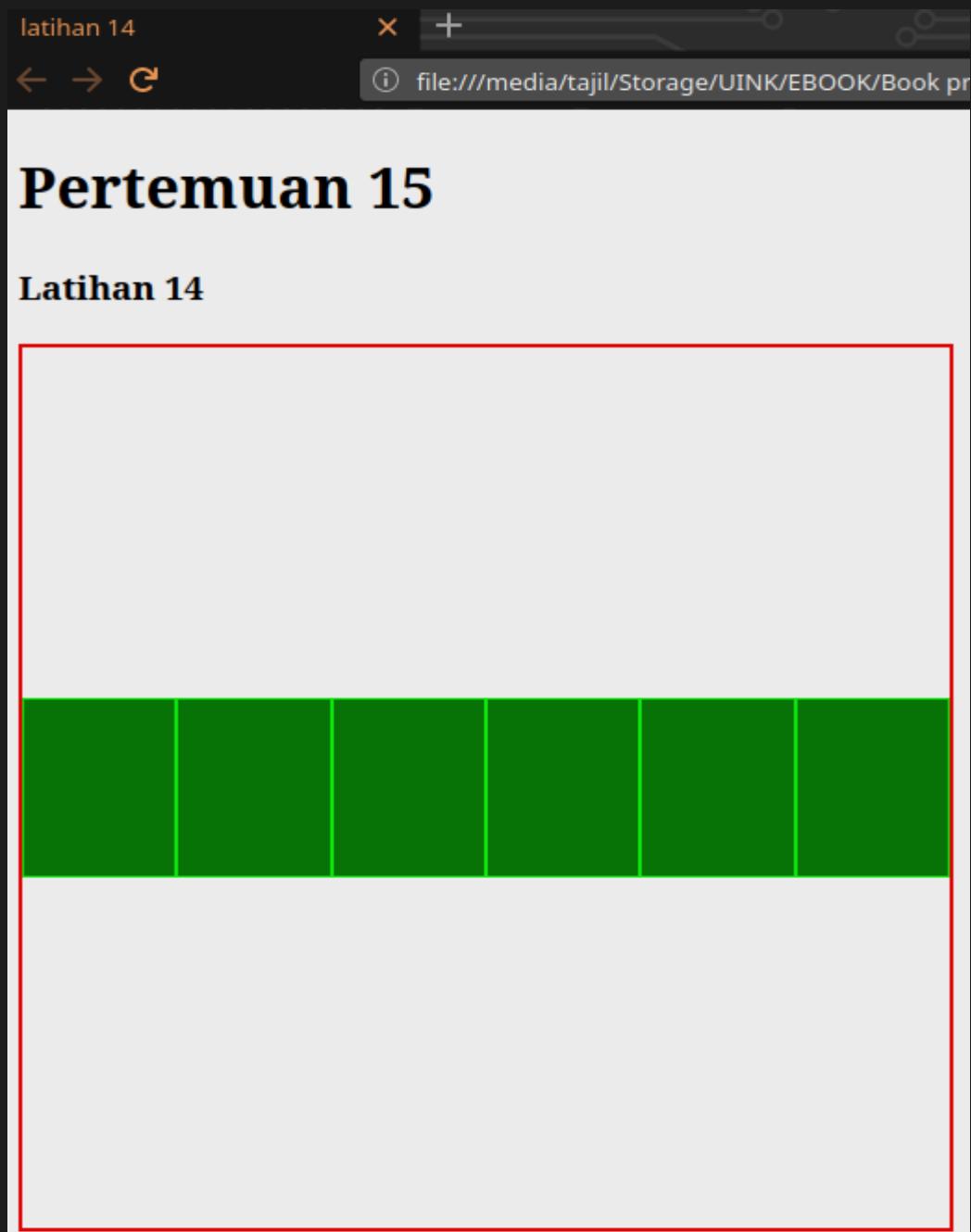


Kemudian, tambahkan satu sub elemen baru pada kontainer flexbox dengan properti yang sama seperti class “.sub”. Lihat apa yang terjadi pada sub elemen baru tersebut !



Dia ditempatkan pada garis baru karena sudah mencapai limit lebar kontainer flexbox (induk).

Sekarang coba ganti value flex-wrap dengan ‘nowrap’, lihat apa yang terjadi ! Tambahkan satu lagi sub elemen apabila belum paham apa yang terjadi. Contoh tampilan dengan ‘nowrap’ dan tambahan satu lagi sub elemen baru:



Coba terus tambah sub elemen baru, elemen tetap diposisi dengan auto resize pada propertis mereka.

PERTEMUAN 16 Responsive Layout

Hampir setiap insan telah memiliki unit komputer mini pada saku pakaian atau tas dalam keseharian. Smartphone atau telepon cerdas adalah komputer mini, dimana kapabilitasnya sudah tidak kalah dengan komputer pribadi (PC). Dahulu, akses website dilakukan via komputer dekstop atau laptop, saat ini akses berbagai web dan layanan komersial dilakukan melalui “jari” (smartphone).

Ukuran layar smartphone tentu beranekaragam. Ukuran mini sampai dengan layar jumbo tersedia dipasaran. Tidak seperti sebelumnya dimana developer website harus membuat dua layout, satu untuk komputer satu untuk handphone atau bahkan hanya membuat satu jenis ukuran layout saja. Kini, teknologi html dan css dewasa ini telah mewadahi varietas ukuran layar pengguna tersebut dengan fasilitas responsive design.

Responsive design artinya website dapat mendeteksi dia dimuat pada ukuran layar berapa kali berapa. Mirip seperti fungsi logika dalam pemrograman bahasa lain, css akan berperan sebagai “engine” cerdas tuk membedakan perbedaan ukuran layar, lalu memilih disain layout website berdasarkan hal tersebut.

Beruntung, dalam meracik responsive layout ini kita tidak perlu repot mencari dan install tools tambahan. Semua kebutuhan terpenuhi oleh browser dan editor tentunya. Brower modern menyediakan fasilitas web developer, salah satunya berupa responsive design mode. Setiap browser memiliki jalur menu dan shortcut berbeda untuk aktivasinya, silahkan cek dokumentasi browser yang digunakan.

Responsive design mode menyediakan kita berbagai simulasi ukuran layar untuk proses pengembangan website atau uji coba proyek. Tentu ini membuat proses development website menjadi lebih praktis.

Persiapan praktek pertemuan ke-16 :

1. Buat dua dokumen; lat15.html dan lat15.css.
2. Buat struktur utama html.
3. Sesuaikan konten title dan link bagian head.
4. Buat sebuah kontainer menggunakan elemen <div>, dengan ukuran 1000x500 pixel.
5. Tambahkan properti “flexbox” pada kontainer tersebut, sehingga dia menjadi flexbox.
6. Tambahkan empat sub <div> kedalam kontainer dengan ukuran 200x400 pixel.

Contoh tampilan layout:



Notes:

Halaman diatas memiliki width 1000px dan height 500px. Ini adalah tampilan ketika halaman web dimuat pada ukuran layar minimal lebar 1000px. Ukuran lebih dari 1000px tidak jadi masalah, tetapi seperti ini. Nah, yg memerlukan penyesuaian layout (bahkan konten) adalah ketika

ukuran layar kurang dari 1000px, seperti 720px atau 320px atau ukuran lainnya. Penyesuaian layar dilakukan secara otomatis oleh css sesuai konfigurasi ukuran layar yang tertera.

Code html:

```
File Edit View Projects Bookmarks Sessions Tool
lat15.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>latihan 15</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="lat15.css">
7   </head>
8   <body>
9
10    <h1>Pertemuan 16</h1>
11    <h3>Latihan 15</h3>
12
13    <div class="wadah">
14
15      <div class="sub"></div>
16      <div class="sub"></div>
17      <div class="sub"></div>
18      <div class="sub"></div>
19
20    </div>
21
22  </body>
23 </html>
24
```

Code css:

```
File Edit View Projects Bookmarks Sessions Tool
lat15.css
1 .wadah {
2   border         : 2px solid red;
3   width          : 1000px;
4   height         : 500px;
5   display        : flex;
6   flex-direction : row;
7   flex-wrap      : wrap;
8 }
9
10 .sub {
11   border         : 1px solid lime;
12   width          : 200px;
13   height         : 400px;
14   background     : green;
15   margin         : auto;
16 }
17
```

Hampir sama dengan konsep hover, responsive mode pada css juga melibatkan class target, kemudian membubuhkan propertis baru atau mengubah value propertis yang sama.

Syntax responsive design pada css adalah sebagai berikut:

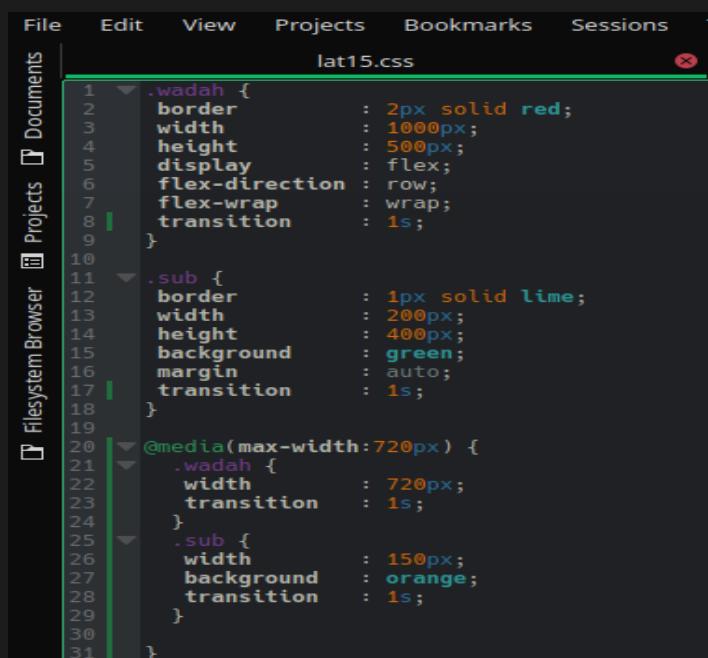
```
@media(max-width:720px) {  
    namaclass {  
        propertis class  
    }  
}
```

Notes:

@media adalah kata kunci utama

max-width adalah nilai lebar maximum. Ketika halaman website dimuat pada ukuran layar dengan lebar 720px, maka propertis dan nilai-nilai propertis yang tertera disini akan dieksekusi sesuai class yang memilikinya.

Mari kita tambahkan @media pada dokumen css sebagai berikut:



The screenshot shows a code editor window with a dark theme. The file is named "lat15.css". The code contains two main sections: ".wadah" and ".sub". The ".wadah" section has properties like border, width, height, display, flex-direction, flex-wrap, and transition. The ".sub" section has properties like border, width, height, background, margin, and transition. A third section, "@media(max-width:720px)", is nested under ".wadah". This media query contains a rule for ".wadah" setting width and transition, and another rule for ".sub" setting width, background, and transition. The code is numbered from 1 to 31.

```
.wadah {  
    border : 2px solid red;  
    width : 1000px;  
    height : 500px;  
    display : flex;  
    flex-direction : row;  
    flex-wrap : wrap;  
    transition : 1s;  
}  
.sub {  
    border : 1px solid lime;  
    width : 200px;  
    height : 400px;  
    background : green;  
    margin : auto;  
    transition : 1s;  
}  
@media(max-width:720px) {  
    .wadah {  
        width : 720px;  
        transition : 1s;  
    }  
    .sub {  
        width : 150px;  
        background : orange;  
        transition : 1s;  
    }  
}
```

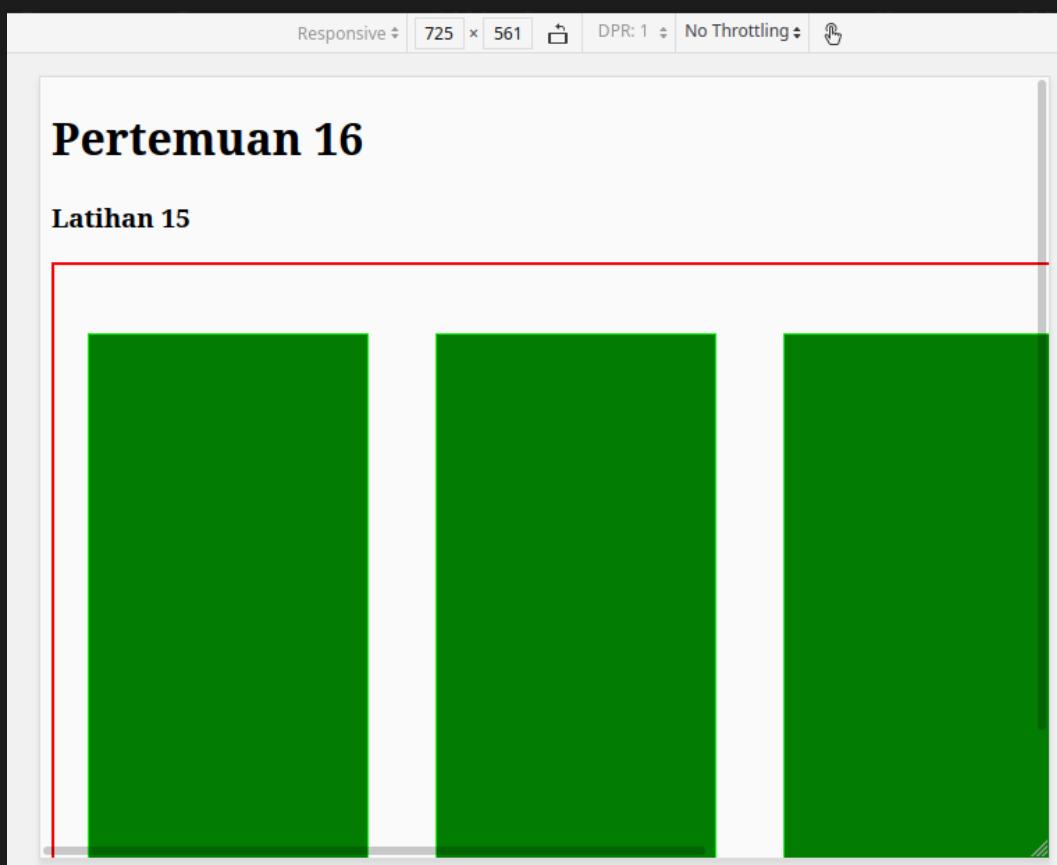
Notes:

penambahan ‘background’ dengan warna orange dan ‘transition’ pada setiap bagian dimaksudkan agar proses transisi dari layar berukuran 1000px menuju 720px dan sebaliknya dari 720px menuju 1000px terlihat jelas dan mencolok.

Pada bagian @media bubuhkan propertis yang nilainya hendak dirubah atau bahkan menambah dan mengurangi propertis. Propertis yang tidak berubah tidak perlu dicantumkan kembali.

Save dan refresh browser, lalu aktifkan responsive design mode. Browser modern dapat melalui menu baris tiga (humberger menu) lalu menuju developer tool/more tool, kemudian cari responsive design mode. Atau dapat melalui shortcut, firefox dengan ctrl+shift+m, chrome F12 lalu klik tab emulate device.

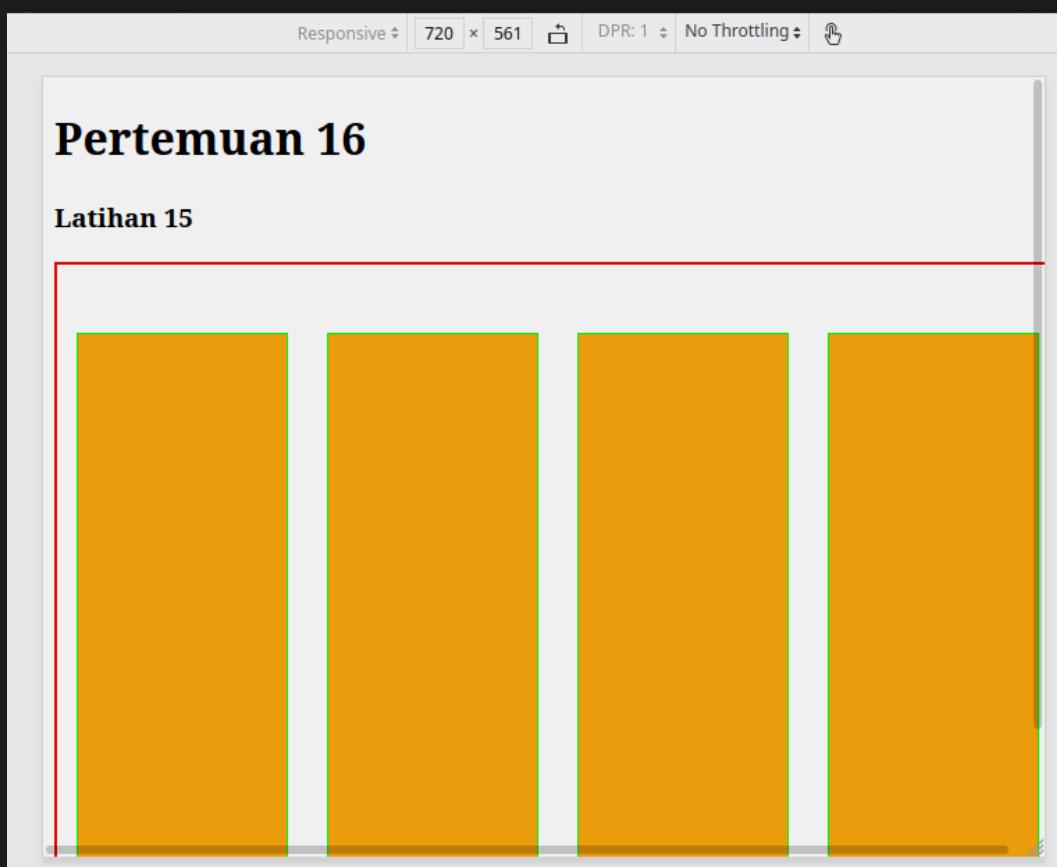
Contoh browser dengan responsive design mode aktif :



ukuran window dapat di drag apabila terlalu kecil. Bagian atas yang berlabel “Responsive : 725 x 561” adalah ukuran emulasi layar, width 725px dan height 561px. Nilai ukuran emulasi dapat dirubah sesuai kebutuhan. Disamping bagian ini terdapat icon handphone yang dapat diklik, untuk merubah posisi layar sebagai landscape atau potrait.

Fasilitas ini benar-benar memanjakan para web developer dan memudahkan pengetesan dan pengembangan web yang sedang dirancang.

Sekarang coba rubah ukuran width menjadi 720px, lalu tekan enter, lihat apa yang terjadi !



Sesuai dengan konfigurasi properti css pada bagian @media!

Warna latar sub elemen menjadi orange, ukuran kontainer dan sub elemen berkurang sesuai dengan perubahan ukuran layar. Selanjutnya mari tambah @media untuk ukuran layar 320 pixel seperti dicontohkan pada dokumen css berikut:

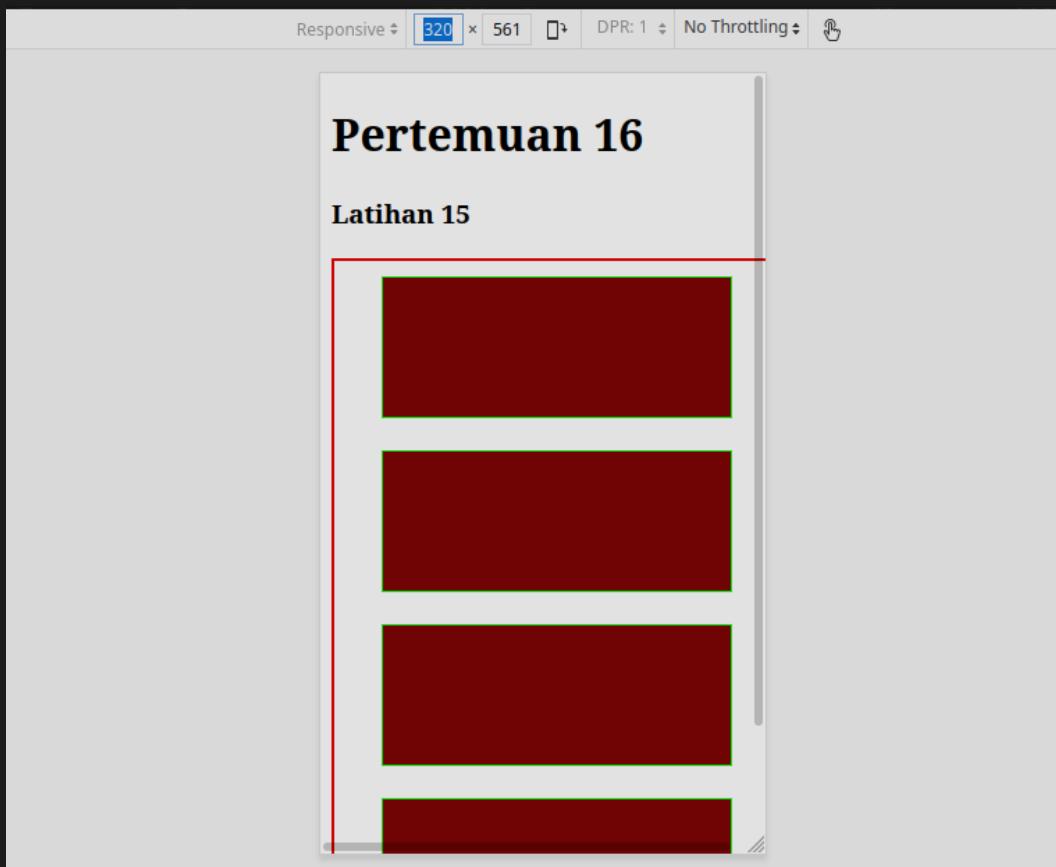
```
lat15.css
9 }
10
11 .sub {
12   border      : 1px solid lime;
13   width       : 200px;
14   height      : 400px;
15   background  : green;
16   margin      : auto;
17   transition  : 1s;
18 }
19
20 @media(max-width:720px) {
21   .wadah {
22     width      : 720px;
23     transition : 1s;
24   }
25   .sub {
26     width      : 150px;
27     background : orange;
28     transition : 1s;
29   }
30 }
31
32 @media(max-width:320px) {
33   .wadah {
34     width      : 320px;
35     transition : 1s;
36   }
37   .sub {
38     width      : 250px;
39     height     : 100px;
40     background : darkred;
41     transition : 1s;
42   }
43 }
44
45 }
46 |
```

Notes:

@media dapat dibubuhkan sebanyak mungkin sesuai kebutuhan target jumlah layar. Namun perlu dicamkan, @media harus berurut dari ukuran paling lebar menuju paling sempit.

@media ukuran 320 pixel diatas mengatur kembali lebar kontainer dan sub elemen serta kali ini mengatur height sub elemen agar sesuai dengan lebar layar.

Hasil pada browser :



Notes:

Pada kasus diatas kita tidak perlu merubah properti ‘flex-direction’, karena kita menggunakan ‘flex-wrap : wrap’. Dengan begini, ketika sub elemen telah mencapai lebar ukuran samping kontainer, ia akan langsung ditempatkan dibawah (garis baru).

Pada kasus tertentu kita membutuhkan perubahan ‘flex-direction’ dari row menuju column atau sebaliknya dari column menuju row. Hal ini tentu tergantung kebutuhan, disain layout dan target ukuran layar.

Apabila dibutuhkan pengaturan kembali ukuran konten seperti gambar atau logo pada ukuran layar tertentu, maka beri class elemen bersangkutan, lalu bubuhkan class tersebut pada bagian @media.

PERTEMUAN 17 Animation

Animation adalah istilah bahasa Inggris yang artinya semangat, gelora, semarak, kegembiraan. Namun, dalam dunia html dan css animation ya animasi. Efek animasi seperti objek yang bergerak dari suatu posisi menuju posisi lain dalam suatu transisi tertentu.

CSS3 hadir memfasilitasi animasi sehingga semakin menarik interaksi pada halaman website kita. Animasi walau menarik dan terlihat profesional, ia tidak boleh berlebihan dalam penerapannya. Sama seperti efek transition, animasi harus digunakan secara proporsional.

Animasi pada css melibatkan dua bagian yakni terdiri atas propertis “animation” dan “@keyframes” bagian isi dari propertis tersebut secara terpisah. Bagian “animation” memiliki lima value, secara runtut mereka adalah nama animasi, durasi, animation case, delay dan iteration count.

Syntax animasi css:

```
animation : contoh 4s ease 2s infinite;
```

Setiap penulisan bagian animasi harus dipisah dengan spasi kosong. Delay (bagian ke-4) dan iteration count (bagian terakhir) tidak wajib diisi, gunakan apabila diperlukan. Secara default bagian ke-5 (terakhir) yakni iteration count akan bernilai “finite” apabila tidak di set manual, finite artinya terbatas, dalam artian animasi berakhir ketika durasi telah habis.

Pada contoh diatas animasi di set dengan durasi 4s (empat detik), ketika nilai infinite tidak kita set, maka ia akan berakhir setelah empat detik berlalu. Animasi berjalan secara otomatis tanpa pemicu seperti kasus hover.

Propertis “animation” ini dibubuhkan pada elemen target yang akan diberi efek animasi. Adapun isi dari propertis ini dibubuhkan secara terpisah pada css. Contoh isi dari propertis animasi diatas :

```
@keyframes contoh {  
    from {background : red;}  
    to {background : yellow;}  
}
```

@keyframes adalah deklarasi bahwa ini adalah isi dari suatu properti animasi. “contoh” adalah nama animasi yang nanti dibubuhkan pada elemen target. Animasi harus diberi nama, satu dokumen css dapat menampung sebanyak-banyaknya material animasi dan setiap elemen dapat menerapkan lebih dari satu animasi. Jadi, penamaan animasi sangatlah krusial dalam hal ini.

Terdapat tiga model utama dalam merancang konten “animation”. Pertama, seperti pada contoh diatas menggunakan konsep “from...to”, dari dan ke. Perubahan terjadi dari satu titik/kondisi menuju titik/kondisi lain dalam selang durasi. Kedua, menggunakan konsep ratio persentasi dan ketiga metode matrix.

Buku ini hanya membahas cara pertama dan kedua, untuk metode matrix pembaca dapat merujuk referensi lain baik bentuk website atau media cetak.

Adapun persiapan pertemuan ke-17 adalah:

1. Buat dua dokumen; lat16.html dan lat16.css.
2. Buat struktur utama html (bagian ini diulang-ulang agar menjadi terbiasa sekaligus menghafal diluar kepala. Bagi pemula harap tidak melakukan copy-paste, sebelum terbiasa dan paham).
3. Sesuaikan konten title dan link bagian head.
4. Buat satu buah kotak menggunakan <div> berukuran 300x300 pixel.

Contoh code html dan css disediakan dibawah:

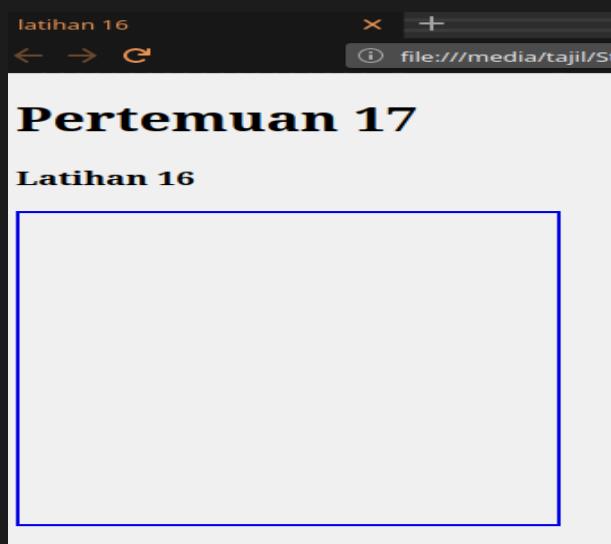
Code html:

```
lat16.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>latihan 16</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="lat16.css">
7 </head>
8 <body>
9
10    <h1>Pertemuan 17</h1>
11    <h3>Latihan 16</h3>
12
13    <div class="wadah"></div>
14
15 </body>
16 </html>
```

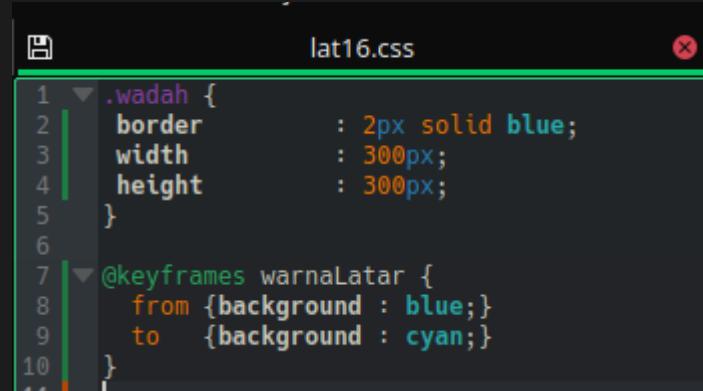
Code css:

```
lat16.css
File Edit View Projects Bookmarks Sessions 1
Documents
1 .wadah {
2     border         : 2px solid blue;
3     width          : 300px;
4     height         : 300px;
5 }
6
```

Hasil pada browser:

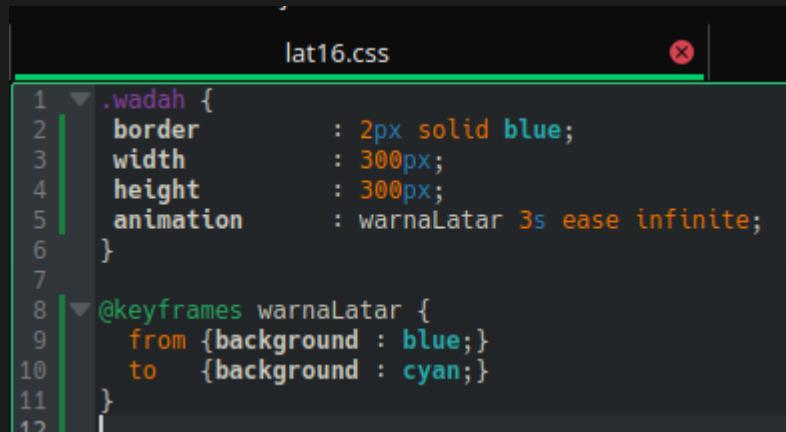


Selanjutnya mari kita rancang propertis animasi untuk elemen tersebut:



```
lat16.css
1 .wadah {
2     border: 2px solid blue;
3     width: 300px;
4     height: 300px;
5 }
6
7 @keyframes warnaLatar {
8     from {background: blue;}
9     to {background: cyan;}
10 }
```

Kemudian terapkan pada class elemen target, dalam hal ini adalah class “.wadah”, berikut contoh penerapan propertis animasi:



```
lat16.css
1 .wadah {
2     border: 2px solid blue;
3     width: 300px;
4     height: 300px;
5     animation: warnaLatar 3s ease infinite;
6 }
7
8 @keyframes warnaLatar {
9     from {background: blue;}
10    to {background: cyan;}
11 }
```

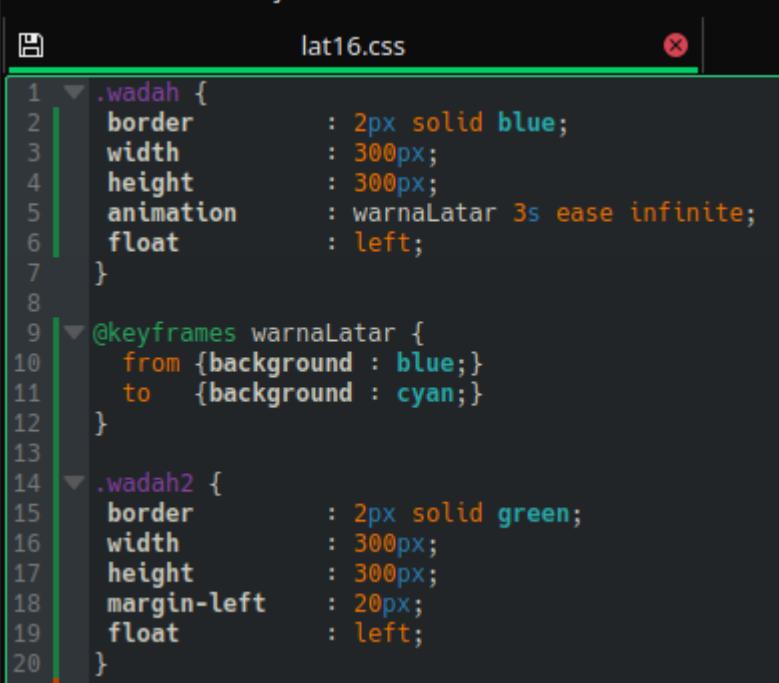
Notes:

- Pemberian nama animasi bebas, tetapi upayakan yang sederhana dan sesuai dengan konteks animasi.
- Animasi diatas bernama “warnaLatar” dengan durasi selama tiga detik, animasi bertipe “ease”, lihat sumber rujukan untuk melihat opsi atas kasus (case) animasi pada css. Bagian terakhir disematkan iteration count dengan value “infinite”, artinya animasi ini akan terus berjalan sampai tab atau browser ditutup.
- Pemberian animasi akan menghisap sumberdaya processor, maka gunakan animasi dengan bijak dan proporsional.
- Save, lalu refresh browser. Sekarang elemen div kita telah memiliki animasi perubahan warna dari blue menuju cyan dalam tiga detik.

Selanjutnya kita akan mempraktekan animasi dengan konsep kedua, yakni konsep proporsi persentasi. Langkah yang ditempuh adalah sebagai berikut:

1. Pada lat16.html, tambahkan satu elemen <div> baru.
2. Beri class berbeda dari <div> pertama.
3. Beri ukuran 300x300 pixel.
4. Atur margin dengan div pertama agar nyaman dilihat.
5. Posisikan tepat disebalahan kanan div pertama.
6. Tampilkan bordernya.

Contoh code CSS:

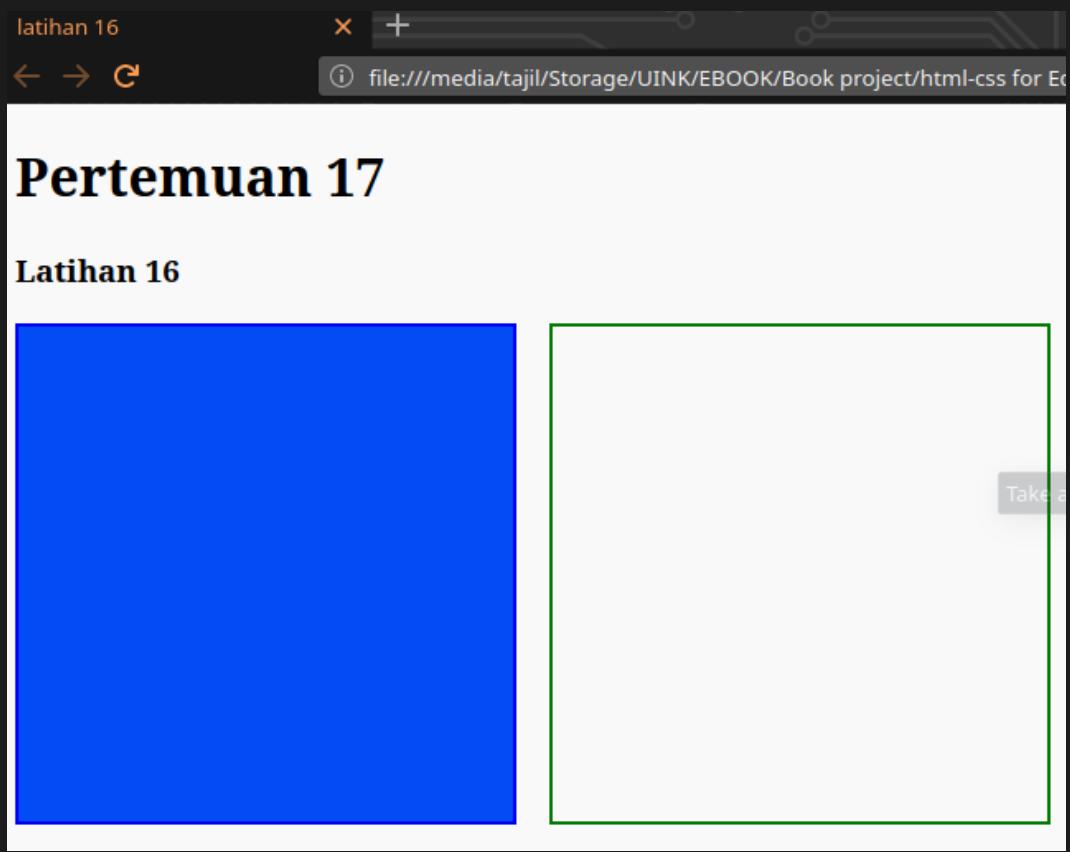


```
lat16.css
1 .wadah {
2     border      : 2px solid blue;
3     width       : 300px;
4     height      : 300px;
5     animation   : warnaLatar 3s ease infinite;
6     float       : left;
7 }
8
9 @keyframes warnaLatar {
10    from {background : blue;}
11    to   {background : cyan;}
12 }
13
14 .wadah2 {
15     border      : 2px solid green;
16     width       : 300px;
17     height      : 300px;
18     margin-left : 20px;
19     float       : left;
20 }
```

Notes:

Agar elemen tipe block dapat diposisikan side by side, maka salah satu metode dapat digunakan adalah dengan menggunakan properti “float”. Value “float” harus diterapkan pada kedua div, apabila hanya salah satu yang memiliki value ini, maka posisi tidak akan side by side. Hal ini dikarenakan <div> adalah tipe block secara default dan ukuran lebar 300px belum mengisi space kosong layar.

Hasil pada browser:



Berikutnya, kita rancang properti animasi untuk div kedua ini menggunakan ratio persentasi animasi sebagai berikut:

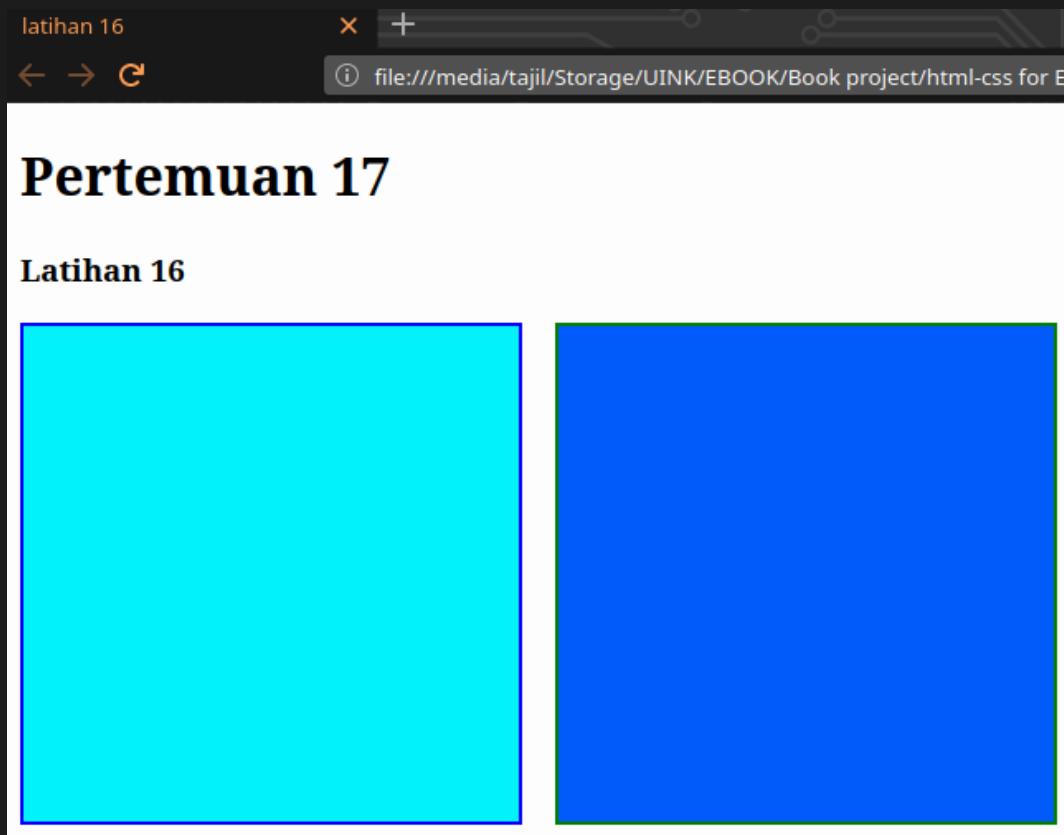
```
lat16.css
1 .wadah {
2     border      : 2px solid blue;
3     width       : 300px;
4     height      : 300px;
5     animation   : warnaLatar 3s ease infinite;
6     float       : left;
7 }
8
9 @keyframes warnaLatar {
10    from {background : blue;}
11    to   {background : cyan;}
12 }
13
14 .wadah2 {
15     border      : 2px solid green;
16     width       : 300px;
17     height      : 300px;
18     margin-left : 20px;
19     float       : left;
20     animation   : warnaLatar2 4s ease infinite;
21 }
22
23 @keyframes warnaLatar2 {
24    10% {background : lime;}
25    40% {background : green;}
26    70% {background : cyan;}
27    100% {background : blue;}
28 }
```

The image shows a code editor window with the file "lat16.css" open. The code defines two classes: ".wadah" and ".wadah2". The ".wadah" class has a blue border, a width of 300px, a height of 300px, and an infinite animation named "warnaLatar" that transitions from blue to cyan over 3 seconds using the ease timing function. The ".wadah2" class has a green border, a width of 300px, a height of 300px, a 20px margin-left, and an infinite animation named "warnaLatar2" that transitions through lime, green, cyan, and back to blue over 4 seconds using the ease timing function. Both classes have a float: left; property.

Notes:

- Langkah yang sama diawali membuat @keyframes.
- lalu beri nama, contoh diatas menggunakan “warnaLatar2”.
- Isi konten animasi per porsi dalam bentuk persentase angka, penulisan dimulai dari persentasi awal proses animasi berjalan sampai porsi terakhir. Rentetan persentase tersebut menjadi “frame” bagi css dalam mengeksekusi animasi.
- Terakhir, terapkan animasi pada elemen target.

Hasil pada browser:



PERTEMUAN 18 Transform Concept

Transform dalam bahasa Indonesia adalah perubahan atau pergantian bentuk. CSS modern juga menawarkan fasilitas transform dengan beberapa metode yang dapat digunakan. Kategori transform pada css berupa 2D dan 3D. Kita akan mempelajari versi 2D terlebih dahulu.

Beberapa metode transform diantaranya:

- ✓ scale
 - ✓ translate
 - ✓ rotate
 - ✓ skew
 - ✓ matrix (tidak dibahas pada buku ini)
- Scale, adalah metode merubah ukuran elemen, baik menambah atau mengurangi ukuran berdasarkan width dan height. Contoh:
`transform: scale(2,4);`
syntax tersebut memberi elemen dua kali lebih lebar (width) dan empat kali lebih tinggi (height) dari ukuran asal.
- Translate, ialah memindahkan posisi awal elemen berdasarkan koordinat X dan Y menggunakan syntax sebagai berikut:
`transform: translate(30px, 20px);`
translate tersebut memindahkan posisi elemen 30px ke arah kanan dan 20px ke arah bawah.
- Rotate, adalah memutar elemen searah jarum jam (clockwise) atau beralawanjarum jam (counter-clockwise), contoh:
`transform: rotate(30deg);`
elemen akan diputar searah jarum jam sebanyak 30 derajat. Apabila hendak menggunakan counter-clockwise, maka gunakan tanda negatif (-) pada ukuran derajat misalnya -30deg.

- Skew, adalah metode memberi kecondongan pada sudut pandang tertentu (angle). Contoh:
transform: skew(10deg);
Aritnya kita memberi elemen kecondongan X-axis sebesar 10 derajat. Apabila ingin memberi skew pada bagian axis Y, maka bubuhkan ukuran kedua umpamanya skew(10deg, 5deg);

Mari kita menuju TKP, berikut langkah-langkah pertemuan 18:

1. Buat dua dokumen; lat17.html dan lat17.css.
2. Buat struktur utama html.
3. Sesuaikan konten title dan link bagian head.
4. Buat empat kotak menggunakan <div> dengan ukuran @200x200px.
5. Beri masing-masing <div> class berbeda.
6. Tampilkan bordernya.
7. Beri div pertama transform: scale
8. Beri div kedua transform: translate
9. Beri div ketiga transform: rotate
10. Beri div keempat transform: skew

Value atas keempat macam metode transform diatas bebas. Silahkan ujicoba dan berlatih dengan berbagai value untuk dapat memahami dan mengenali keempat metode transform tersebut.

Penguasaan atas empat metode transform sangat berguna bagi aktivitas developmen website, melengkapi efek hover, transition dan animation sehingga halaman website lebih interaktif dan looking professional.

Contoh code html dan css disertakan dibawah untuk keperluan komparasi dan analisis pembelajaran.

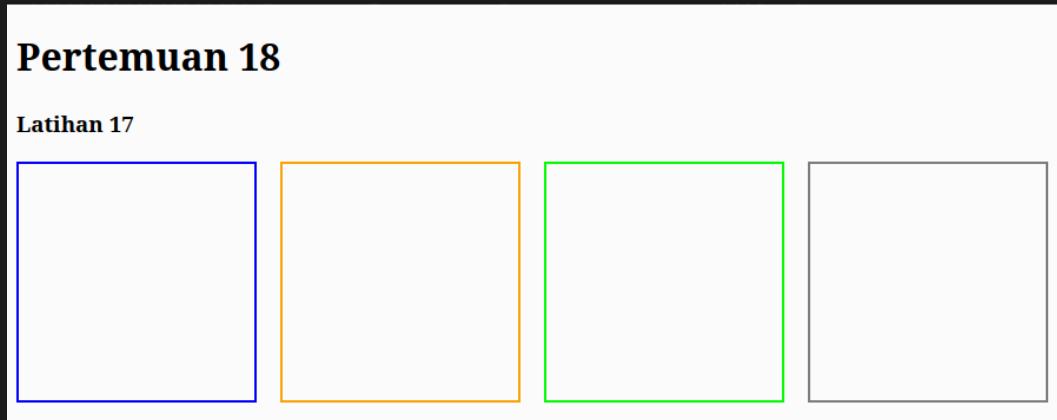
Contoh code html:

```
File Edit View Projects Bookmarks Sessions Tool
lat17.html ×
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>latihan 17</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="lat17.css">
7 </head>
8 <body>
9
10    <h1>Pertemuan 18</h1>
11    <h3>Latihan 17</h3>
12
13    <div class="wadah1"></div>
14    <div class="wadah2"></div>
15    <div class="wadah3"></div>
16    <div class="wadah4"></div>
17
18 </body>
19 </html>
20
```

Contoh code css:

```
File Edit View Projects Bookmarks Sessions
lat17.css ×
1 .wadah1 {
2     border      : 2px solid blue;
3     width       : 200px;
4     height      : 200px;
5     float       : left;
6     margin-right : 20px;
7 }
8
9 .wadah2 {
10    border      : 2px solid orange;
11    width       : 200px;
12    height      : 200px;
13    float       : left;
14    margin-right : 20px;
15 }
16
17 .wadah3 {
18     border      : 2px solid lime;
19     width       : 200px;
20     height      : 200px;
21     float       : left;
22     margin-right : 20px;
23 }
24
25 .wadah4 {
26     border      : 2px solid grey;
27     width       : 200px;
28     height      : 200px;
29     float       : left;
30     margin-right : 20px;
31 }
```

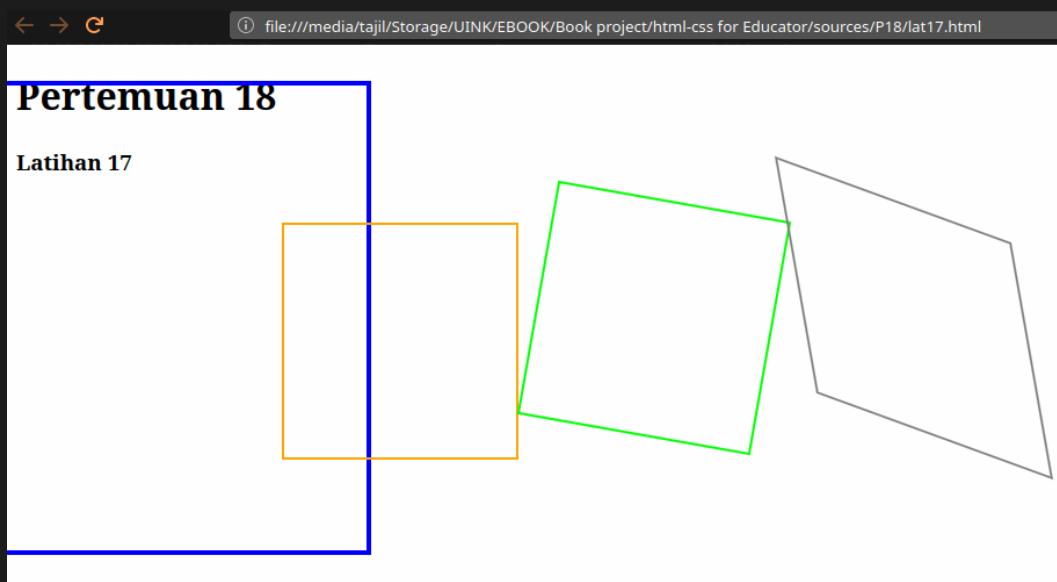
Hasil pada browser:



Selanjutnya, keempat elemen <div> diberi properti transform secara berurutan dari kotak biru, scale, translate, rotate dan skew. Maka code css kini sebagai berikut:

```
lat17.css
1  .wadah1 {
2      border      : 2px solid blue;
3      width       : 200px;
4      height      : 200px;
5      float       : left;
6      margin-right: 20px;
7      transform   : scale(2,2);
8  }
9
10 .wadah2 {
11     border      : 2px solid orange;
12    width       : 200px;
13    height      : 200px;
14    float       : left;
15    margin-right: 20px;
16    transform   : translate(5px,20px);
17  }
18
19 .wadah3 {
20     border      : 2px solid lime;
21    width       : 200px;
22    height      : 200px;
23    float       : left;
24    margin-right: 20px;
25    transform   : rotate(10deg);
26  }
27
28 .wadah4 {
29     border      : 2px solid grey;
30    width       : 200px;
31    height      : 200px;
32    float       : left;
33    margin-right: 20px;
34    transform   : skew(10deg,20deg);
35  }
```

Hasil pada browser:



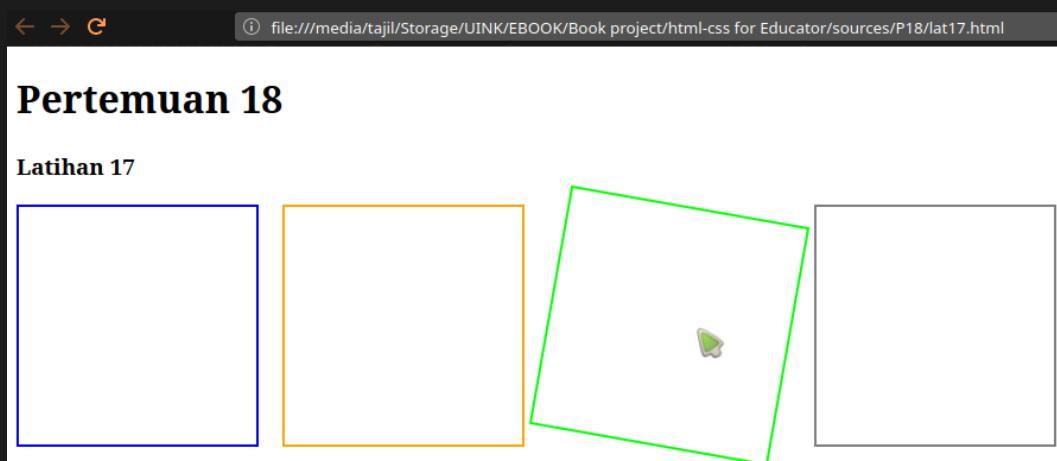
Notes:

Agar perubahan pada setiap metode transform terlihat jelas, sebaiknya kita manfaatkan “hover”, sehingga kondisi sebelum dan sesudah transform diterapkan dapat dibedakan secara lebih jelas.

```
lat17.css
1 .wadah1 {
2   border      : 2px solid blue;
3   width       : 200px;
4   height      : 200px;
5   float       : left;
6   margin-right: 20px;
7 }
8 .wadah1:hover {
9   transform   : scale(2,2);
10}
11
12 .wadah2 {
13   border      : 2px solid orange;
14   width       : 200px;
15   height      : 200px;
16   float       : left;
17   margin-right: 20px;
18 }
19 .wadah2:hover {
20   transform   : translate(5px,20px);
21 }
22
23 .wadah3 {
24   border      : 2px solid lime;
25   width       : 200px;
26   height      : 200px;
27   float       : left;
28   margin-right: 20px;
29 }
30 .wadah3:hover {
31   transform   : rotate(10deg);
32 }
33
34 .wadah4 {
35   border      : 2px solid grey;
36   width       : 200px;
37   height      : 200px;
38   float       : left;
39   margin-right: 20px;
40 }
41 .wadah4:hover {
42   transform   : skew(10deg,20deg);
43 }
44
45
46
47 }
```

Sekarang save dan refresh web browser, kemudian lakukan hover satu persatu pada empat tersebut!

Contoh tampilan pada browser, hover pada kotak ketiga (rotate):



Notes:

Coba bubuhkan properti “transition”, sehingga proses sebelum dan sesudah transform terjadi dalam durasi tertentu. Hal ini sekaligus menjadi kombinasi yang menarik antara hover, transform dan transition.

Transform 3D

CSS modern menyediakan fasilitas transform versi 3D dengan syntax yang tidak berbeda dengan versi 2D, memudahkan para pembelajar dan sangat berguna untuk para web designer. Namun, tingkat kompatibilitas tiap produk browser berbeda, boleh jadi ada bagian metode yang tidak kompatibel sama sekali pada jenis browser tertentu.

Oleh karena itu, hendaklah selalu menggunakan browser mainstream terlebih dahulu, agar problem seperti ini dapat dihindari. Bagi para expert, masalah kompatibilitas tentu tidak menjadi halangan dalam mendisain halaman website, tetapi bagi para pemula yang baru mengecap html dan css, tentu dapat menjadi ‘momok’ dan ‘batu sandungan’.

Perbedaan mendasar konsep 2D dan 3D adalah keberadaan axis-z atau kedalaman, sehingga objek terlihat seperti didunia nyata alias 3D. 2D hanya memiliki dua arah kedalaman, axis-x dan axis-y. 3D memiliki tiga arah kedalaman objek, axis-x, axis-y dan axis-z.

Cara penerapan metode transform 3D sangat mudah, cukup membubuhkan aksis ‘x’, ‘y’ atau ‘z’ pada kata kunci metode transform.

Contoh:

```
transform : rotateX(10deg); atau  
transform : rotateY(10deg); atau  
transform : rotateZ(10deg);
```

Kali ini kita hanya menggunakan metode rotate sebagai latihan transform 3D. Untuk metode lain silahkan diujicobakan secara mandiri dengan cara yang sama, yakni membubuhkan aksis ‘x’, ‘y’ atau ‘z’ .

Persiapan praktek transform 3D:

1. Buat dua dokumen; lat18.html dan lat18.css
2. Buat struktur utama html
3. Sesuaikan konten title dan link bagian head
4. Buat tiga buah kotak menggunakan <div> dengan ukuran @200x200 pixel
5. Tampilkan secara side by side menggunakan ‘float’
6. Beri margin-right secara proporsional
7. Beri class berbeda dan tampilkan border mereka
8. Bubuhkan masing-masing kotak dengan rotateX, rotateY dan rotateZ via “hover”
9. **Transform 3D wajib mengikutsertakan “transition”**. Agar terlihat smooth, ada baiknya disertakan baik pada class hover maupun non hover.

contoh code html:

```
lat18.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>latihan 18</title>
5   <meta charset="utf-8">
6   <link rel="stylesheet" href="lat18.css">
7 </head>
8 <body>
9
10  <h1>Pertemuan 18</h1>
11  <h3>Latihan 18</h3>
12
13  <div class="wadah1"></div>
14  <div class="wadah2"></div>
15  <div class="wadah3"></div>
16
17 </body>
18 </html>
19
```

Contoh code css:

```
lat18.css
1 .wadah1 {
2   border      : 2px solid blue;
3   width       : 200px;
4   height      : 200px;
5   float       : left;
6   margin-right: 20px;
7   transition   : 2s;
8 }
9
10 .wadah1:hover {
11   transform   : rotateX( 180deg );
12   transition   : 2s;
13 }
14
15 .wadah2 {
16   border      : 2px solid orange;
17   width       : 200px;
18   height      : 200px;
19   float       : left;
20   margin-right: 20px;
21   transition   : 2s;
22 }
23
24 .wadah2:hover {
25   transform   : rotateY( 180deg );
26   transition   : 2s;
27 }
28
29 .wadah3 {
30   border      : 2px solid lime;
31   width       : 200px;
32   height      : 200px;
33   float       : left;
34   margin-right: 20px;
35   transition   : 2s;
36 }
37
38 .wadah3:hover {
39   transform   : rotateZ( 40deg );
40   transition   : 2s;
41 }
```

Notes:

Pada metode rotate 3D, beri value angka 180 derajat, agar rotateX dan rotateY terlihat jelas!

Save dan refresh browser, kemudian lakukan hover satu persatu pada ketiga kotak. Perhatikan dengan sekasama transform 3D beraksi!

PERTEMUAN 19 Button Slider

Pertemuan ke-19 ini lebih mirip seperti tutorial html dan css ketimbang materi pertemuan pada umumnya. Button slider artinya membuat tombol yang mana tombol tersebut memiliki bagian slider (bergeser).

Konsep rancangan tombol seperti ini memanfaatkan metode tombol style dropdown (ketika diklik muncul sub-sub menu). Pembaca sekalian disarankan merujuk cara membuat tombol dropdown dengan html dan css.

Button slider memanfaatkan properti “display” serta elemen tipe inline. Tidak seperti elemen tipe block, inline elemen tidak memberi garis baru sebelum dan sesudah elemen tersebut di render. Element inline yang akan digunakan pada kesempatan ini adalah elemen .

Ada beberapa hal penting yang mesti dijelaskan terlebih dahulu, tentang penerapan properti css pada class dan non-class. Pertemuan sebelumnya, kita selalu menggunakan class dan menerapkan properti css pada class tersebut. Css dapat pula diterapkan langsung pada elemen bersangkutan dan pada sub elemen secara spesifik.

Contoh menerapkan langsung pada elemen tanpa class:

```
P {  
.....;  
}  
  
h1 {  
.....;  
}
```

Disini kita tidak membuat class, akan tetapi langsung menggunakan elemen sesuai dengan label tersebut. Penggunaan dengan metode seperti ini berdampak secara keseluruhan pada elemen <p> dan <h1> di dokumen yang digunakan.

Contoh menerapkan propertis css spesifik pada sub elemen (elemen induk tidak terdampak):

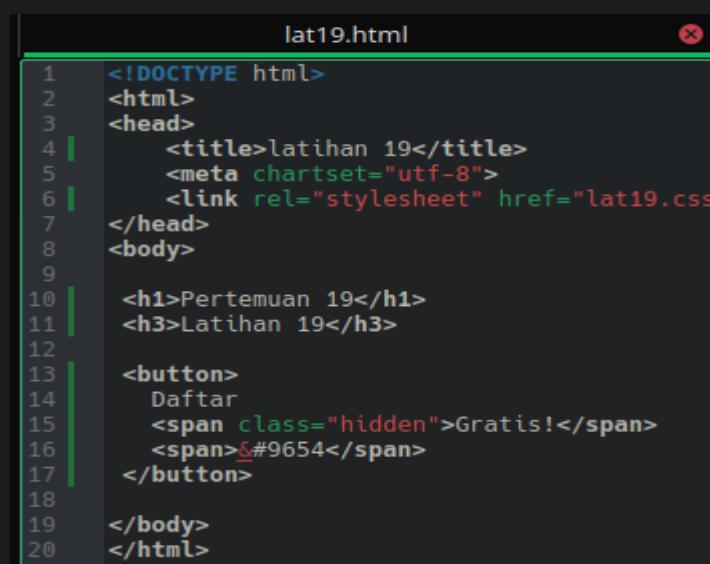
```
div p {  
.....;  
}  
  
h1 p{  
.....;  
}
```

Grup propertis pertama hanya berdampak pada elemen <p> yang terdapat dalam elemen <div>. Grup propertis kedua hanya berdampak pada elemen <p> dalam elemen <h1>.

Itulah beberapa metode lain dalam menerapkan propertis css pada elemen html. Namun demikian, penggunaan class sudah lebih dari cukup untuk menargetkan elemen secara spesifik.

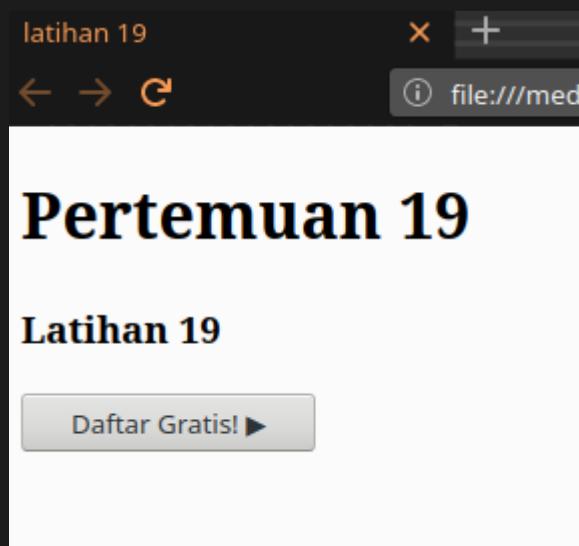
Adapun persiapan praktik pertemuan ke-19 ialah:

1. Buat dua dokumen; lat19.html dan lat19.css.
2. Buat struktur utama html.
3. Sesuaikan konten title dan link bagian head.
4. Isi bagian <body> dengan konten berikut:



```
lat19.html  
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4  <title>latihan 19</title>  
5  <meta charset="utf-8">  
6  <link rel="stylesheet" href="lat19.css">  
7  </head>  
8  <body>  
9  
10 <h1>Pertemuan 19</h1>  
11 <h3>Latihan 19</h3>  
12  
13 <button>  
14   Daftar  
15   <span class="hidden">Gratis!</span>  
16   <span>☰</span>  
17 </button>  
18  
19 </body>  
20 </html>
```

Hasil pada browser:

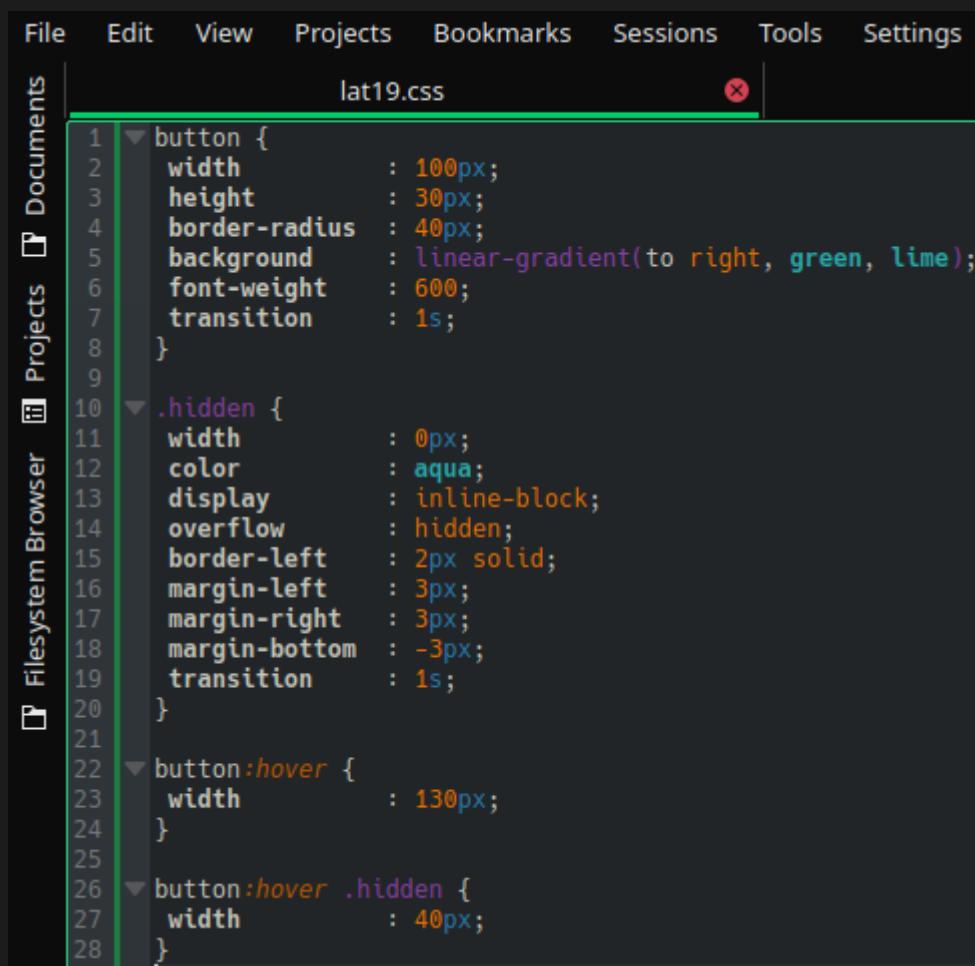


Notes:

- <button> adalah elemen html untuk keperluan tombol yang biasanya digunakan bersama tipe elemen input dalam suatu form. Pelabelan elemen button terdapat diantara tag buka dan tag tutup. Namun, dalam hal ini kita tidak memberi label, melainkan memberi beberapa konten text dan dua elemen didalamnya.
- ▶ adalah salah satu karakter entitas pada html. Karakter-karakter khusus dan spesial ditampilkan dengan cara ini pada html. Pelajari tentang html entity pada sumber rujukan website atau cetak, ada banyak list entitas html. ▶ adalah perwujudan dari panah segitiga arah kanan.

Rencananya, tulisan “gratis” pada tombol akan kita sembunyikan, sehingga tombol hanya menampilkan text “Daftar” dan bagian segitiga kecil tepat disebelah kanan. Baru ketika tombol di klik, tulisan gratis dimunculkan kembali dengan segitiga kecil bergerak ke arah kanan memberi space untuk text “Gratis!”.

Berikutnya, tambahkan propertis dibawah pada dokumen lat19.css:



The screenshot shows a code editor window with a dark theme. The menu bar includes File, Edit, View, Projects, Bookmarks, Sessions, Tools, and Settings. A tab bar at the top shows "lat19.css". The left sidebar has icons for Documents, Projects, and Filesystem Browser. The main area contains the following CSS code:

```
1  button {  
2      width      : 100px;  
3      height     : 30px;  
4      border-radius: 40px;  
5      background   : linear-gradient(to right, green, lime);  
6      font-weight  : 600;  
7      transition   : 1s;  
8  }  
9  
10 .hidden {  
11     width      : 0px;  
12     color       : aqua;  
13     display     : inline-block;  
14     overflow    : hidden;  
15     border-left : 2px solid;  
16     margin-left  : 3px;  
17     margin-right : 3px;  
18     margin-bottom: -3px;  
19     transition   : 1s;  
20 }  
21  
22 button:hover {  
23     width      : 130px;  
24 }  
25  
26 button:hover .hidden {  
27     width      : 40px;  
28 }
```

Notes:

background button diberi warna gradient tipe linear. “to right” artinya warna akan dimulai dari arah kiri menuju arah kanan (akhir). Daftar warna disamping “to right” adalah warna-warna gradient dan akan ditampilkan berdasarkan arah “to right”. Warna bisa ditambahkan sebanyak mungkin. Arah gradient dapat menggunakan left, right, bottom dan top. Bahkan kombinasi arah seperti “to bottom-right” dapat digunakan.

Font-weight adalah tingkat ketebalan text (boldness).

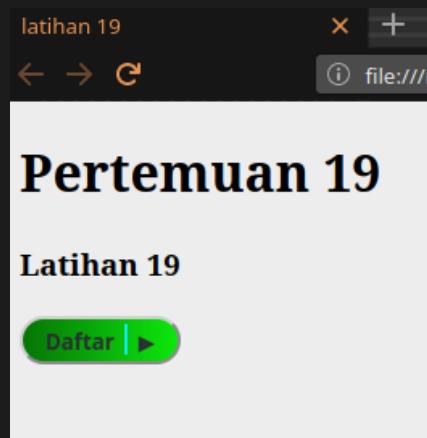
Class hidden memiliki propertis “overflow : hidden”, artinya tampilan diluar layar render akan disembunyikan, sehingga tidak memberi efek pada bagian yang ditampilkan.

Class hidden juga memiliki propertis “display : inline-block”, agar text “Gratis!” pada diposisikan side by side dengan text “Daftar”. Propertis “display” berfungsi merubah konfigurasi display bawaan elemen html.

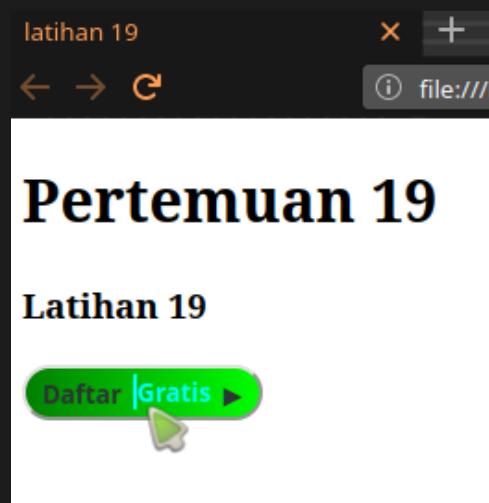
Class hidden diberi propertis width : 0px (nol pixel), agar tidak nampak sebelum di hover. Baru kemudian class hidden memiliki width: 40px ketika di hover.

Untuk mewadahi perubahan width class hidden, elemen <button> juga harus ditambah ukuran width saat dii hover, width <butto> bertambah menjadi 130px. Hal tersebut agar ukuran tombol dapat menampung text “Gratis!” sekaligus menjaga segitiga hitam tetap disebelah kanan.

Hasil pada browser sebelum hover:



Hasil pada browser sesudah hover:



PERTEMUAN 20 Compile The HTML & CSS ?

Compile aritnya menyusun, menghimpun atau mengumpulkan. Compile adalah istilah yang bermakna mengumpulkan objek dan sub objek yang berserakan menjadi satu kesatuan.

Dalam bahasa pemrograman, compile adalah aktivitas menerjemahkan sumber code (source code) menjadi bahasa mesin target atau umum. Dalam hal ini, bahasa pemrograman dibagi ke dalam dua kategori, kategori compiler dan kategori interpreter.

Contoh kategori compiler seperti bahasa C, C++, Java, Python dan lain sebagainya. Bahasa jenis ini mesti dikompilasi terlebih dahulu sebelum dapat dijalankan diplatform target (Linux, Windows, Mac atau mobile). Hasil dari kompilasi adalah paket executable, paket data yang dapat dieksekusi dan berjalan, contoh: deb, rpm, exe, msi, dmg, apk.

Adapun kategori bahasa interpreter seperti javascript, PHP dan lain-lain. Bahasa jenis ini tidak untuk dikompilasi, namun begitu ia membutuhkan “engine” untuk menerjemahkan ke tingkat bahasa mesin (platform), engine disini yang disebut sebagai interpreter.

Browser adalah “engine” atau “interpreter” atas javascript, html dan css. Sumber code diterjemahkan oleh interpreter, lalu ditampilkan pada layar. Tanpa interpreter, layar tidak akan “memahи” harus menampilkan apa dan bagaimana.

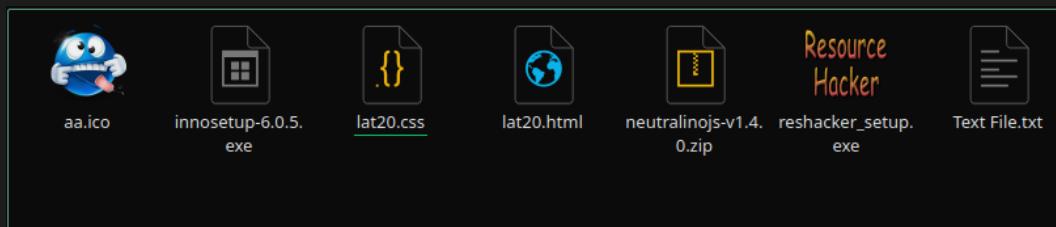
Nah, bagaimana seandainya projek html dan css kita dapat dikompilasi, sehingga menjadi satu bundel data executable dan menawarkan layar instalasi pada user (wizard) ?

Tentu akan membuat projek sederhana terlihat profesional bukan !

Kita akan memanfaatkan framework untuk proses encapsulation (membungkus) dokumen html dan css. Selanjutnya, memanfaatkan aplikasi “installer maker” untuk merancang layanan instalasi. Hasil akhir pertemuan ini berupa satu bundel data executable.

Persiapan pertemuan ke-20 ialah:

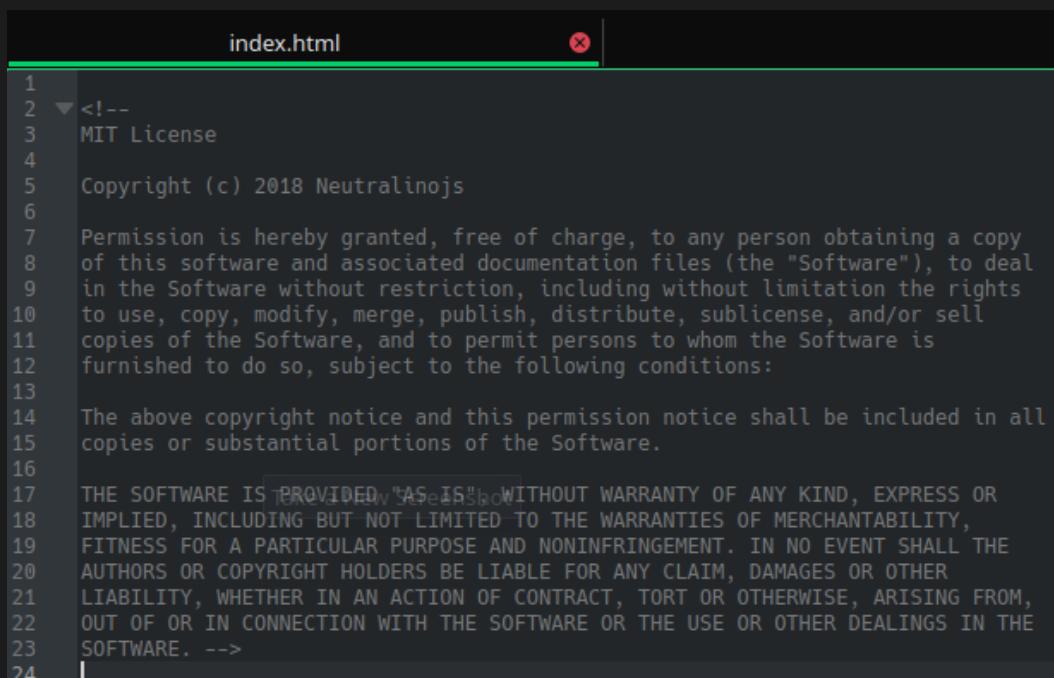
- ✓ Download dan install aplikasi Inno Setup.exe
- ✓ Download dan install aplikasi reshacker setup.exe
- ✓ Download dan ekstrak neuralinoJS.zip. Simpan file zip sebagai backup dan keperluan penggunaan akan datang.
- ✓ Siapkan sample projek html dan css, boleh menggunakan bekas latihan sebelumnya atau membuat yang baru.
- ✓ Siapkan sebuah icon berekstensi .ico.
- ✓ Buat sebuah dokumen berekstensi .txt, isi dengan kalimat sapaan sederhana.
- ✓ Simpan semua file diatas ke dalam satu folder, beri nama, misal “kompilasi html-css”. Contoh :



- ✓ Masuk ke folder neutralinoJS, terdapat beberapa file executable dan folder didalamnya. Executable untuk linux, windows dan mac. Disini kita akan menggunakan platform windows, jadi gunakan “neutralino-win.exe”, rename sesuai selera. Hapus dua file executable untuk linux dan mac.

Name	Size	Compressed
app	1 Folder, 4 Files	
assets	3 Files	
index.html	1,4 KiB	817 B
settings-browser.json	76 B	55 B
settings-cloud.json	137 B	92 B
settings.json	177 B	101 B
storage	1 File	
neutralino-linux	1,0 MiB	240,0 KiB
neutralino-mac	255,6 KiB	92,9 KiB
neutralino-win.exe	5,5 MiB	1,1 MiB

- ✓ Masih di folder neutralinoJS, hapus beberapa bagian yang tidak diperlukan seperti folder “storage”. Kemudian masuk ke folder “app”, hapus “settings-browser.json” dan “setting-cloud.json”.
- ✓ Buka dokumen “index.html”, hapus semua bagian, kecuali bagian MIT license, biarkan disitu. Kemudian copy dan paste semua konten html kita tepat dibawahnya. Cara kedua adalah sebaliknya copy lisensi MIT dibagian index.html, paste pada dokumen html kita, lalu hapus index.html dan rename dokumen html kita menjadi “index.html”.



```

index.html
 1
 2 <!--
 3 MIT License
 4
 5 Copyright (c) 2018 Neutralinojs
 6
 7 Permission is hereby granted, free of charge, to any person obtaining a copy
 8 of this software and associated documentation files (the "Software"), to deal
 9 in the Software without restriction, including without limitation the rights
10 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
11 copies of the Software, and to permit persons to whom the Software is
12 furnished to do so, subject to the following conditions:
13
14 The above copyright notice and this permission notice shall be included in all
15 copies or substantial portions of the Software.
16
17 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
18 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
19 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
20 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
21 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
22 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
23 SOFTWARE. -->
24

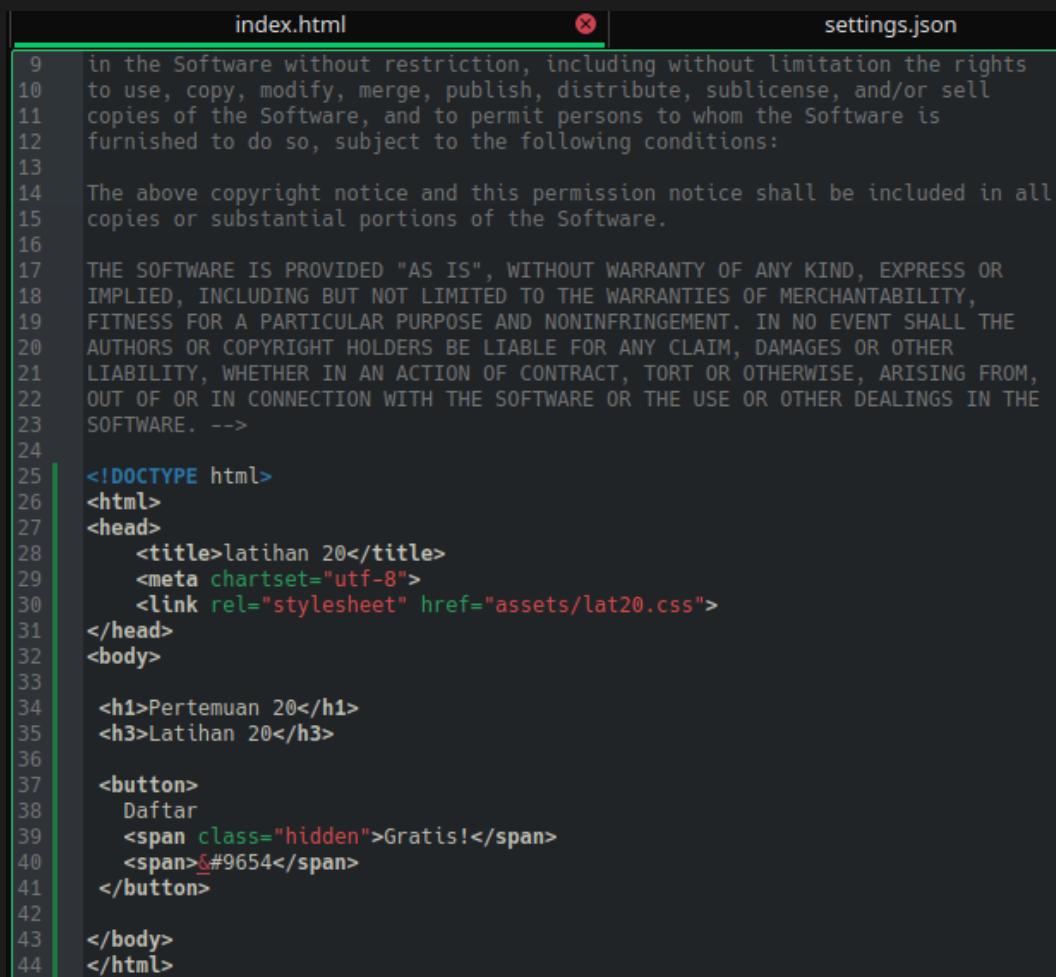
```

Notes:

Dokumen html yang di muat pertamakali oleh server harus dilabeli “index.html”, bagitupula pada framework neutralinoJS.

Mengapa lisensi diatas mesti kita bubuhkan pada projek kita ? Setiap porduk pasti menyedot waktu, biaya dan tenaga si pembuat. Keputusan memberi lisensi produk ada pada si pembuat tentu. Dalam hal ini, neutralinoJS diberi lisensi tipe “opensource”, secara lebih spesifik berlisensi MIT license. Untuk menghormati dan menghargai jerih payah si pembuat, maka kita harus bubuhkan lisensi tersebut.

Contoh dokumen index.html:



The screenshot shows a code editor with two tabs: "index.html" and "settings.json". The "index.html" tab is active, displaying the following code:

```
9  in the Software without restriction, including without limitation the rights
10 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
11 copies of the Software, and to permit persons to whom the Software is
12 furnished to do so, subject to the following conditions:
13
14 The above copyright notice and this permission notice shall be included in all
15 copies or substantial portions of the Software.
16
17 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
18 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
19 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
20 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
21 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
22 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
23 SOFTWARE. -->
24
25 <!DOCTYPE html>
26 <html>
27 <head>
28   <title>latihan 20</title>
29   <meta charset="utf-8">
30   <link rel="stylesheet" href="assets/lat20.css">
31 </head>
32 <body>
33
34   <h1>Pertemuan 20</h1>
35   <h3>Latihan 20</h3>
36
37   <button>
38     Daftar
39     <span class="hidden">Gratis!</span>
40     <span>=▶</span>
41   </button>
42
43 </body>
44 </html>
```

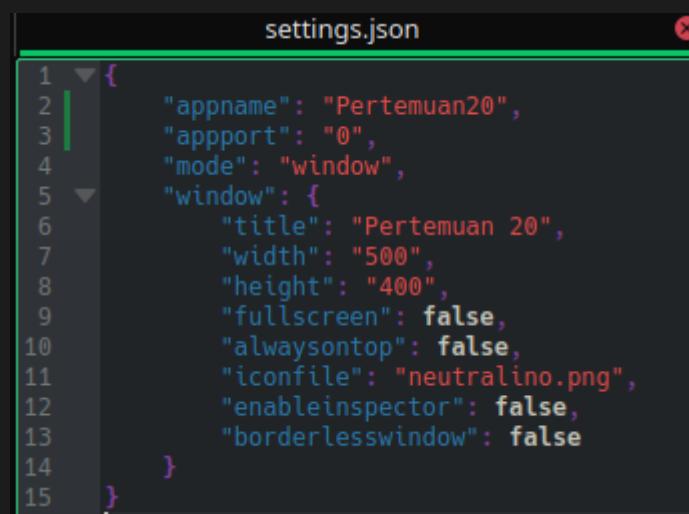
Notes:

Konten `<link>` pada bagian `<head>` harus mengikuti aturan neutralinoJS, yakni menyimpan dokumen css dan dokumen lain seperti javascript pada folder "assets". Sehingga href merujuk pada "assets/lat20.css".

Hapus dokumen lain pada folder "assets", saat ini kita tidak membutuhkannya.

Tepat disebelah folder "assets" dan dokumen "index.html" terdapat dokumen "settings.json". Buka dokumen tersebut pada editor seperti contoh dibawah:

Tampilan dokumen “setting.json” pada editor:



```
settings.json
1 { "appname": "Pertemuan20", "appport": "0", "mode": "window", "window": { "title": "Pertemuan 20", "width": "500", "height": "400", "fullscreen": false, "alwaysontop": false, "iconfile": "neutralino.png", "enableinspector": false, "borderlesswindow": false } }
```

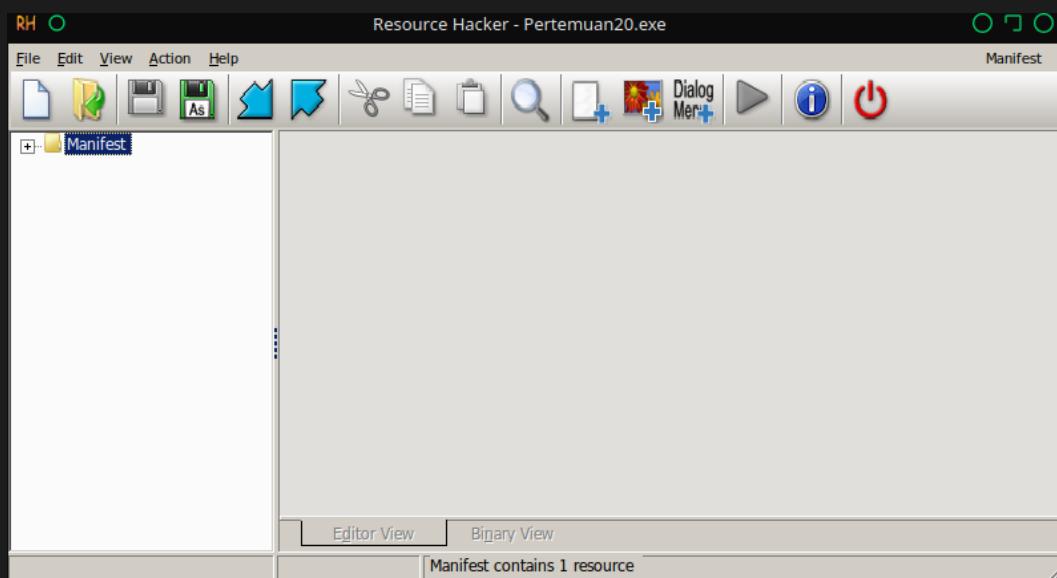
Notes:

- Dokumen ini adalah tempat pengaturan frontend bagi framework. Isinya terdapat beberapa bagian yang dapat diubah sesuai keperluan.
- “appname” adalah nama aplikasi. Tidak boleh menggunakan spasi kosong!!
- “apport” bagian penghubung menuju port di system komputer, biarkan saja bernilai “0”, apabila diperlukan dapat diisi secara manual seperti “8080”.
- “mode” ialah metode framework dalam menampilkan konten pada sistem. Terdapat tiga opsi, dengan browser, metode cloud (background) dan window atau secara native. Kita menggunakan metode native (window).
- “title” ialah judul pada jendela aplikasi.
- “width dan height” adalah ukuran layar aplikasi.
- Bagian lain dapat dirujuk pada dokumentasi neutralinoJS.

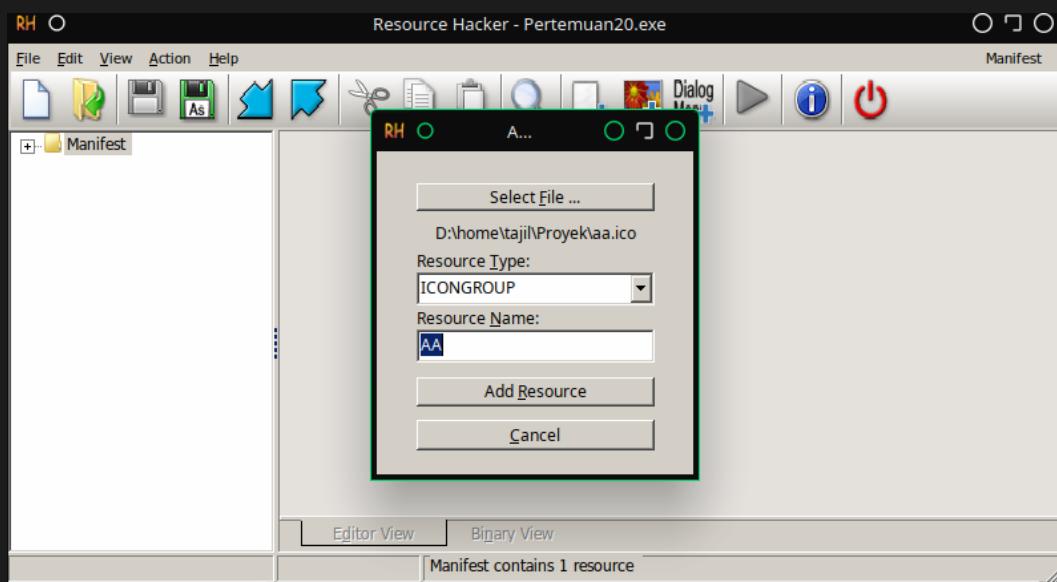
Untuk sementara sudah lebih dari cukup aspek minimal yang perlu kita ketahui. Tentu kedepannya dapat mempelajari dan menggunakan bagian framework yang tidak disinggung pada buku ini.

Berikutnya, buka aplikasi resource hacker, arahkan pada file executable dari neutralinoJS ('neutralino-win.exe' apabila belum anda rename). Kita harus mengganti icon file executable tersebut agar terlihat lebih profesional.

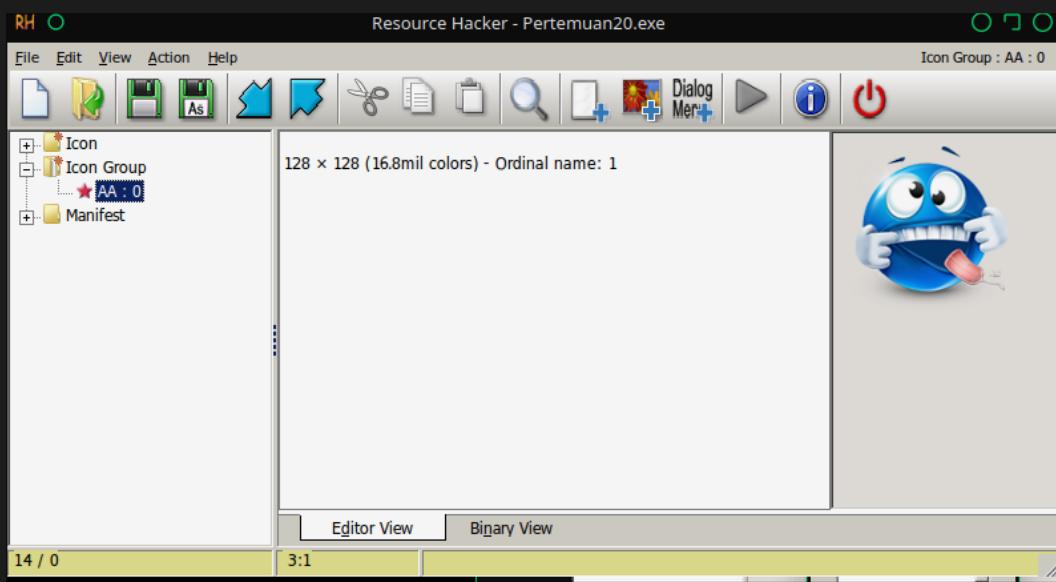
Apabila sudah terbuka, File-open-arahkan menuju file executable diatas. Contoh:



Kemudian, klik icon berlabel "add binary or resource icon", posisi keempat dari kanan. Klik Select_File, Arahkan menuju file icon yang telah kita persiapkan, lalu klik Add Resource. Contoh:

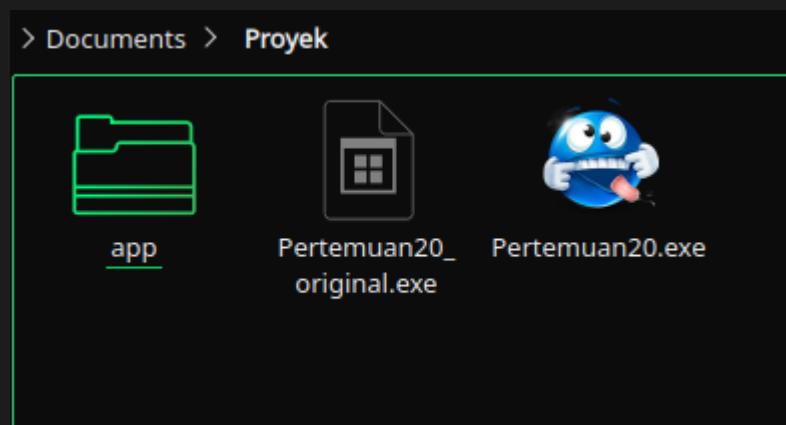


icon berhasil ditambahkan:



Langkah terakhir tahap ini : Save !

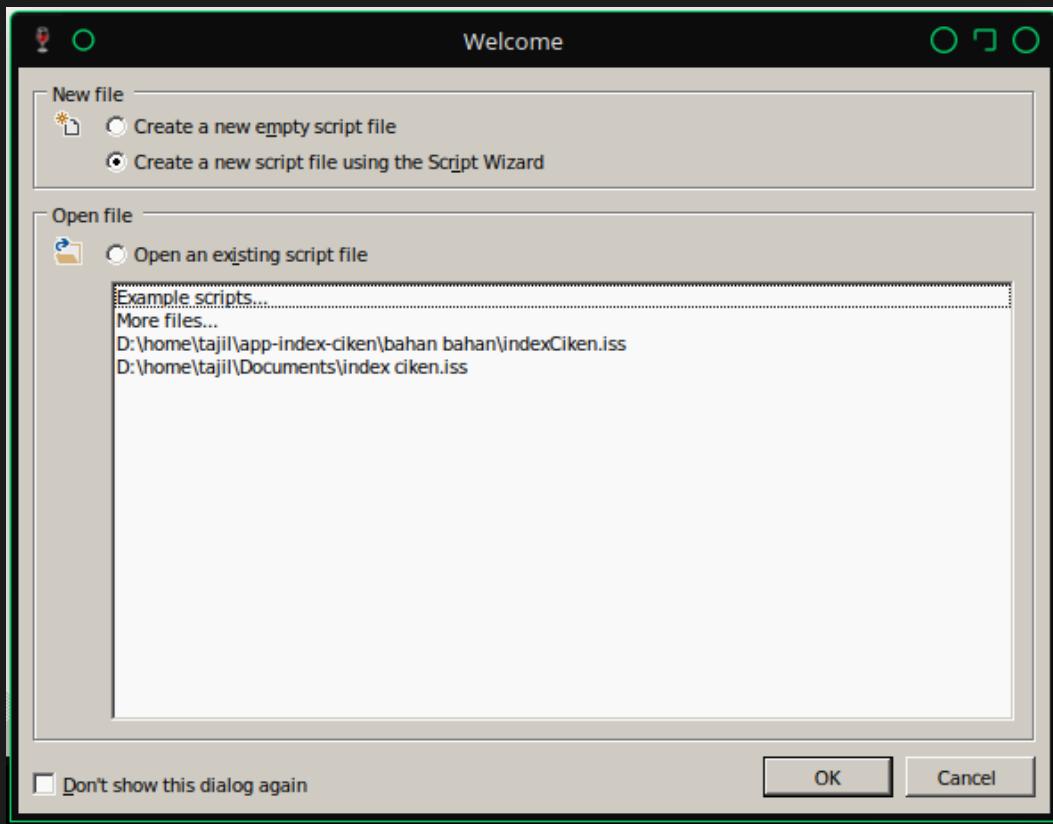
Proses ini menghasilkan dua file executable, satu file original dan satu lagi file executable yang telah kita tambahkan icon diatas.



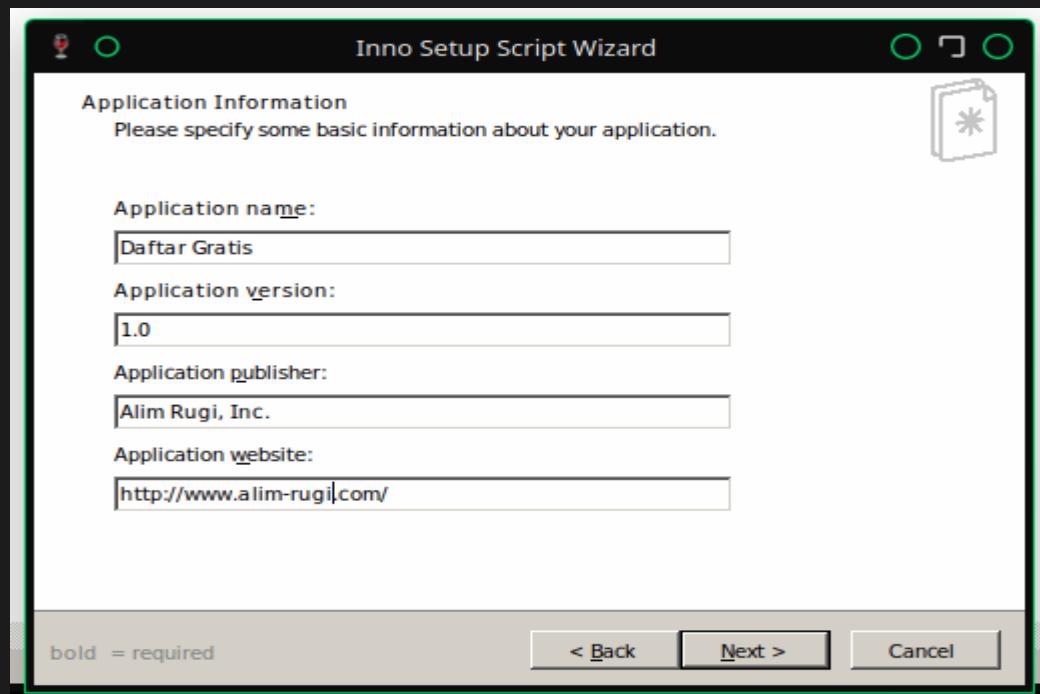
Agar ukuran aplikasi tidak “membengkak”, simpan file executable original dilokasi lain.

Tahap selanjutnya melakukan proses enkapsulasi menggunakan aplikasi “Inno setup.exe”.

Buka aplikasi inno setup, pilih “create a new script file using the script wizard”.

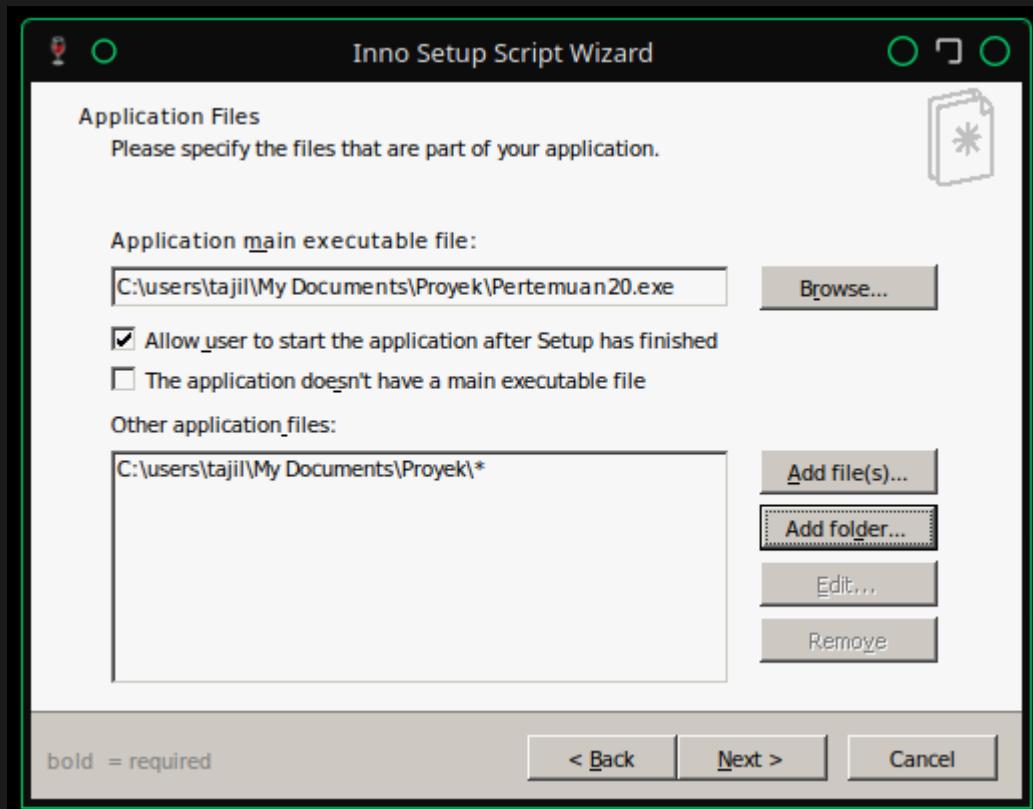


Kemudian klik “Next”, isi beberapa keterangan pada layar:



Klik “Next”, lakukan perubahan yang diperlukan, apabila dirasa cukup dapat langsung klik “Next”.

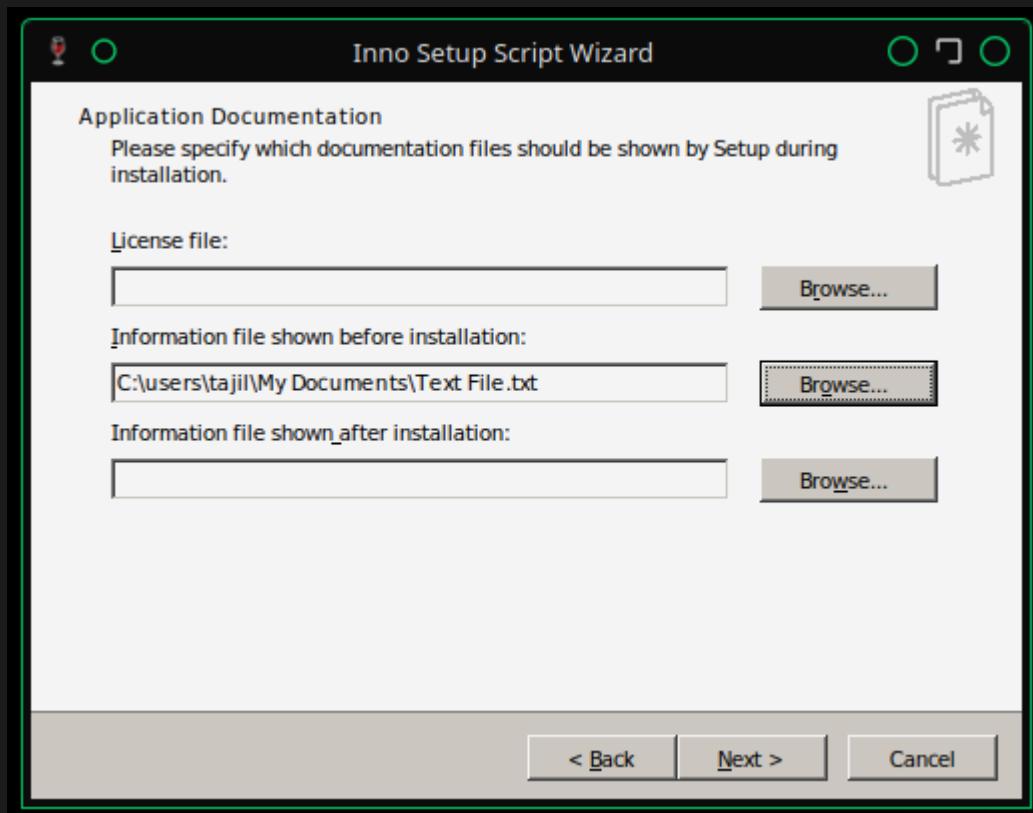
Layar selanjutnya, klik “Browse” untuk memilih application main executable file (file executable utama aplikasi), lalu pada bagian “Other application files”, klik “dd Folder”, arahkan pada lokasi dimana folder proyek kita berada. Perhatikan contoh dibawah:



Klik “Next”, apabila cukup klik kembali “Next”,

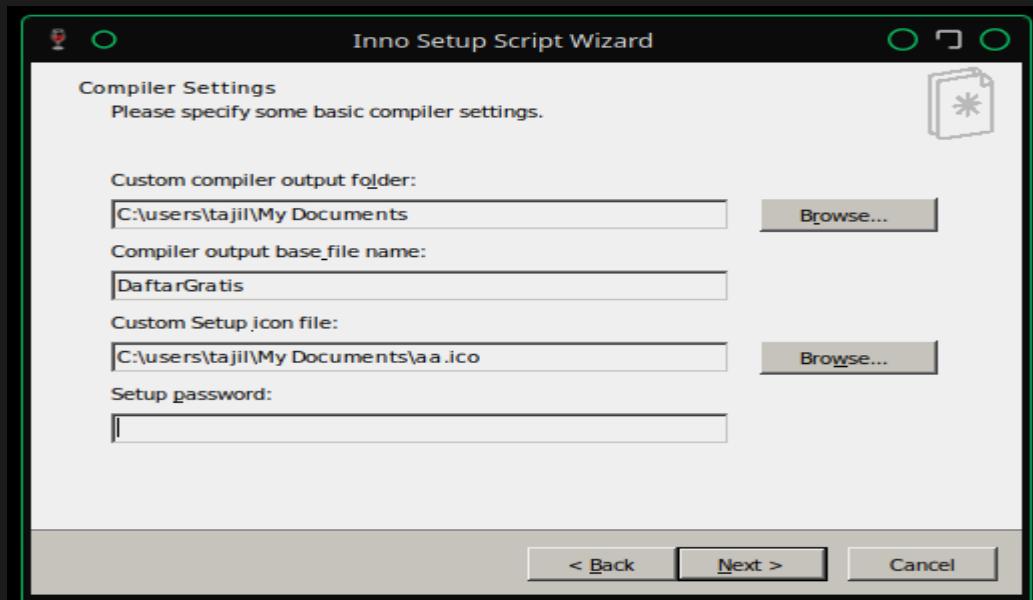
Kali ini ada tiga bagian, bagian dokumen lisensi apabila sudah menyediakan, kosongkan apabila tidak tersedia. Bagian kedua adalah notifikasi ketika proses instalasi akan dimulai, bagian ini akan kita isi dengan dokumen.txt yang telah kita persiapkan sebelumnya. Bagian ketiga adalah notifikasi ketika proses instalasi selesai, kosongkan apabila tidak meyiapkan.

Perhatikan contoh dibawah:



Klik "Next", kemudian atur konfigurasi sesuai kebutuhan, apabila dirasa cukup klik kembali "Next".

Layar bahasa (language) adalah memilih opsi bahasa lingkungan installer. Apabila diaktifkan lebih dari satu bahasa, user akan ditawarkan memilih bahasa ketika awal proses instalasi. Klik "Next" apabila selesai.

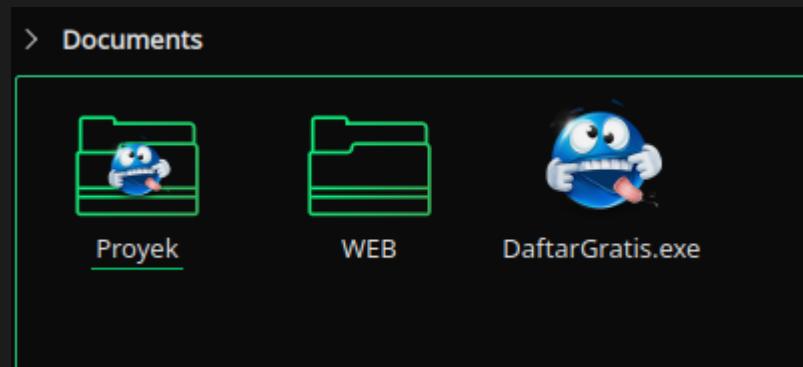


Notes:

Bagian ini terdiri atas empat konfigurasi. Bagian pertama pengaturan manual dimana output kompilasi disimpan. Bagian kedua label untuk file hasil kompilasi. Bagian ketiga pembubuhan icon untuk file executable, pada contoh diatas menggunakan icon yang sama untuk file executable, boleh menggunakan icon mana saja. Bagian terakhir adalah bagian password, kosongan apabila tidak ingin menggunakan password, isi password apabila hendak menambahkan fitur password ketika proses instalasi dimulai.

Klik “Next” sampai bertemu tombol “Finish”, klik “Finish”, lalu klik “Yes” untuk menjalankan proses kompilasi. Sebelum kompilasi dimulai aplikasi menawarkan apakah perlu menyimpan script atas langkah-langkah yang telah dilakukan, hal ini seperti template ketika nanti melakukan kompilasi dengan input data yang semisal. Pilih “Yes” untuk menyimpan script atau pilih “No” tak masalah.

Kini kita memiliki file executable hasil kompilasi:



Silahkan klik 2x untuk menjalankan aplikasi.....Enjoy !

EPILOG

Pembelajar yang budiman, jaminan sukses tidaknya belajar tidak datang dari lembaga atau buku yang digunakan. Persentase terbesar sukses berasal dari diri sendiri. Sejauh mana kita berupaya mendalami materi pembelajaran dan sejauh mana effort kita dalam melakukan latihan-latihan untuk mendalami dan menguasai materi pelajaran, berdampak besar bagi pertumbuhan “jiwa” sebagai seorang penuntut ilmu.

Langkah berikutnya yang harus dilakukan setelah melalui seluruh bab pada buku ini adalah mempelajari jenis sumber bacaan “exhausted” atas HTML dan CSS. Exhausted disini bermakna sumber-sumber yang mamasukan (menjubelkan) berbagai dokumentasi html dan css. Sumber seperti ini ditujukan untuk para developer web tingkat menengah dan advance (walaupun pemula nol besar juga dapat mengambil manfaat atas itu).

Kemudian, ambil langkah “fokus”, hendak mendalami secara serius bagian mana, apakah html dan css atau bahasa pemrograman lain seperti javascript, php dan python. Pilihan apapun, harus disertai niat tulus dan kesungguhan. Dalam pelaksanaan proses demi proses yang dilalui dibutuhkan fokus yang konsisten, agar materi pembelajaran dapat terserap efisien dalam waktu sesingkat-singkatnya.

Tentukan “Roadmap” belajar secara mandiri, seolah kita mendesain kurikulum untuk kita dan oleh kita. Rasakan manfaatnya ketimbang belajar tanpa arah jelas. Adapun contoh roadmap atas perjalanan setelah html dan css adalah mempelajari javascript, lalu pelajari kombinasi bahasa PHP + Per-Database-an seperti mysql misalnya.

RoadMap:

HTML → CSS → Javascript → PHP → MYSQL

REFERENSI

Media Digital (PDF):

Bill Sanders, 2014, Smashing HTML5, Smashing Magazine

Jina Bolton dkk, 2009, Fancy Form Design, SitePoint:Canada

Jon Duckett, 2011, HTML & CSS Design and Build Websites, John Wiley & Sons.Inc:Indianapolis

Mark Pilgrim, 2013, Dive Into HTML5, O'Reilly: Google Press

Matt West.,2014, HTML5 Foundations, Treehouse.

WCGs (Web Code Geeks), 2015, HTML5 Programming Cookbook, Exilisis Media P.C.

html cheat sheet.pdf

css cheat sheet.pdf

Media Website:

w3cschool.com

developer.mozilla.org

sitepoint.com

neutralinojs.org

webcodegeek.com

diveintohtml5.org