

PETUNJUK PELAKSANAAN PRAKTIKUM

WEB 1

HTML, CSS, JS dan Bootstrap

Oleh

Mirza Yogy Kurniawan

Muhammad Edya Rosadi

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ISLAM KALIMANTAN MUHAMMAD ARSYAD AL BANJARI
BANJARMASIN**

DAFTAR ISI

DAFTAR ISI.....	<i>i</i>
ATURAN UMUM.....	<i>iii</i>
KELENGKAPAN	<i>iii</i>
PRASYARAT.....	<i>iii</i>
KESELAMATAN DAN KESEHATAN KERJA (K3).....	<i>iii</i>
KONTAK.....	<i>iv</i>
SESI 1 Pengenalan HTML	1
TUJUAN PRAKTEK	1
PENDAHULUAN	1
PERCOBAAN 1.1 Mengenal HTML	2
PERCOBAAN 1.2 List.....	9
PERCOBAAN 1.3 Gambar.....	10
KESIMPULAN 1	12
LATIHAN 1.1	12
PERCOBAAN 1.4 Anchor.....	13
LATIHAN 1.2	14
SESI 2 Pengenalan CSS.....	16
TUJUAN PRAKTEK	16
PENDAHULUAN	16
PERCOBAAN 2.1 Mengenal CSS.....	17
PERCOBAAN 2.2 Class Selector dan ID Selector	20
PERCOBAAN 2.3 Descendant Selector dan Child Selector	22
PERCOBAAN 2.4 Chaining Selector dan Pseudo Selector	24
PERCOBAAN 2.5 Menambahkan CSS ke HTML.....	26
PERCOBAAN 2.6 Box Model	28
PERCOBAAN 2.7 Padding, Margin dan Border.....	30
PERCOBAAN 2.8 Flexbox.....	32
PERCOBAAN 2.9 Membuat Landing Page	35
KESIMPULAN 2	46
LATIHAN 2.1	46
SESI 3 Form dan Input	47
TUJUAN PRAKTEK	47
PENDAHULUAN	47
PERCOBAAN 3.1 Form, Label, dan Input	48

PERCOBAAN 3.2 DOM.....	52
PERCOBAAN 3.3 DOM Event	53
<i>SESI 4 Bootstrap dan Fontawesome.....</i>	<i>58</i>
TUJUAN PRAKTEK	58
PENDAHULUAN	58
PERCOBAAN 4.1 Bootstrap	59
PERCOBAAN 4.2 Fontawesome	65
PERCOBAAN 4.3 Form Input dengan Bootstrap dan Fontawesome.....	66
PENUTUP.....	70

ATURAN UMUM

KELENGKAPAN

1. Peserta praktikum wajib berpakaian sopan dan formal, mengenakan celana panjang/rok, kemeja, dan sepatu.
2. Peserta diwajibkan membawa laptop masing-masing dan stop kontak jika diperlukan.
3. Peserta menyiapkan kuota internet sebagai cadangan

PRASYARAT

Prasyarat yang harus dilakukan peserta sebelum pelaksanaan praktikum

1. Memasang dan memperbarui peramban (Google Chrome, Mozilla FireFox atau Microsoft Edge)
2. Mengunduh dan memasang Visual Studio Code
(<https://code.visualstudio.com/download>)

KESELAMATAN DAN KESEHATAN KERJA (K3)

1. Sebelum memulai praktikum, praktikan memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik (stop kontak dan circuit breaker)
3. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera lapor ke asisten terkait atau dapat langsung melapor ke laboran.
4. Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik (sengatan listrik) secara tidak sengaja, misal kabel yang terkelupas
5. Keringkan bagian tubuh yang basah, seperti keringat atau sisa air wudhu
6. Selalu waspada terhadap bahaya listrik pada setiap aktifitas praktikum.

7. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
 - a. Jangan panik
 - b. Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikum yang tersengat arus listrik.
 - c. Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik
 - d. Beritahukan dan minta bantuan kepada laboran, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik.
8. Jangan membawa benda-benda mudah terbakar (korek api, gas, dll) ke dalam ruangan laboratorium bila tidak disyaratkan dalam modul praktikum.
9. Jangan melakukan sesuatu yang menimbulkan api, percikan api, atau panas yang berlebihan.
10. Jangan melakukan sesuatu yang menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain.
11. Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
12. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai.
13. Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain.

KONTAK

Telegram: @mirzayogy

SESI 1

Pengenalan HTML

TUJUAN PRAKTEK

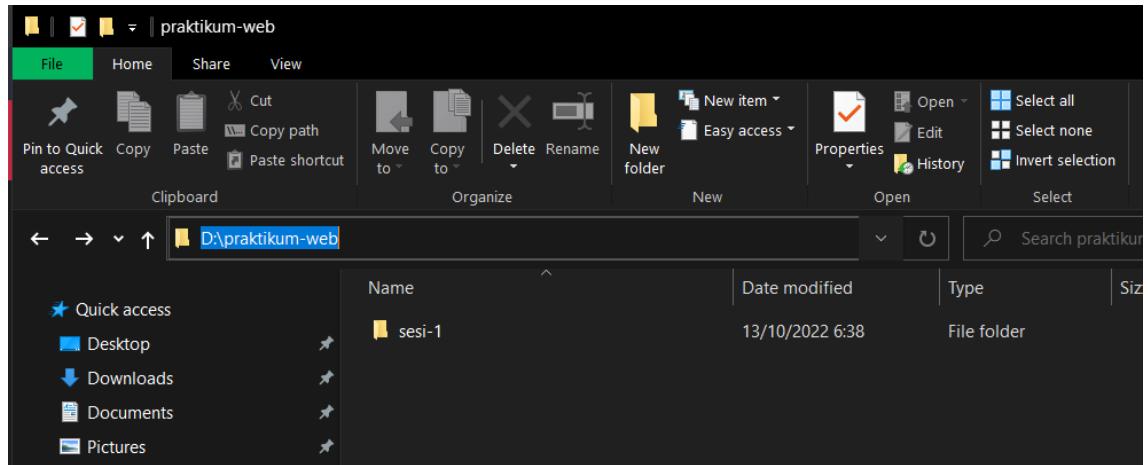
1. Mahasiswa mampu membuat berkas HTML beserta sintaks dasarnya
2. Mahasiswa mampu menambahkan asset gambar dan URL pada dokumen HTML

PENDAHULUAN

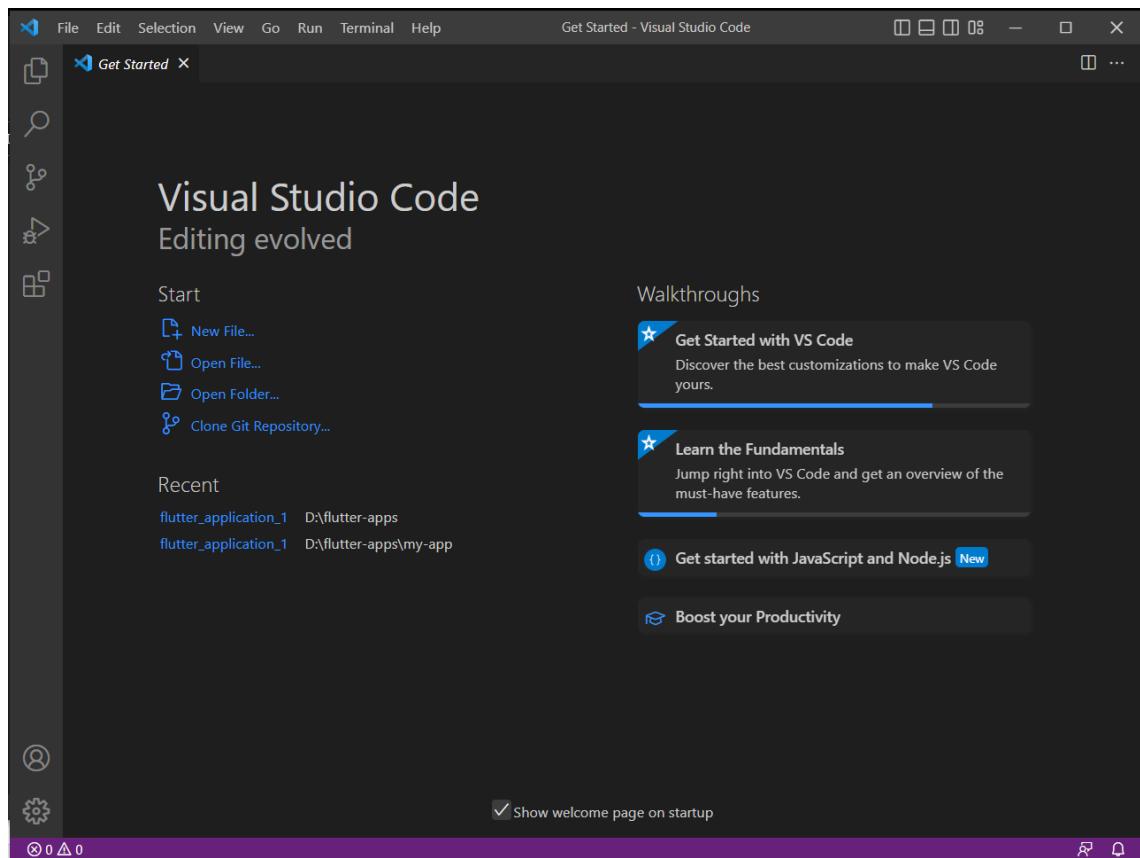
HTML adalah singkatan dari Hypertext Markup Language, yang merupakan bahasa standar yang digunakan untuk membuat halaman web. HTML menggunakan tag atau markup untuk menentukan cara menampilkan teks, gambar, link, dan elemen lainnya pada halaman web. Belajar HTML penting karena ini adalah bahasa dasar yang digunakan untuk membuat halaman web. Dengan mengetahui HTML, Anda dapat membuat struktur dan konten untuk halaman web Anda, seperti menambahkan heading, paragraf, atau link. Selain itu, HTML juga memberikan dasar yang baik untuk belajar bahasa pemrograman web lainnya, seperti CSS dan JavaScript. Selain itu, belajar HTML juga penting karena HTML adalah bahasa yang mudah dipelajari dan dapat dikembangkan sendiri. Anda dapat dengan mudah mencari sumber belajar HTML di internet, baik berupa tutorial, dokumentasi resmi, maupun komunitas yang akan membantu Anda belajar HTML. Kemampuan HTML juga sangat dibutuhkan dalam dunia kerja. Banyak perusahaan yang mencari pekerja yang memiliki kemampuan HTML, terutama dalam bidang pemrograman web dan pengembangan situs web.

PERCOBAAN 1.1 Mengenal HTML

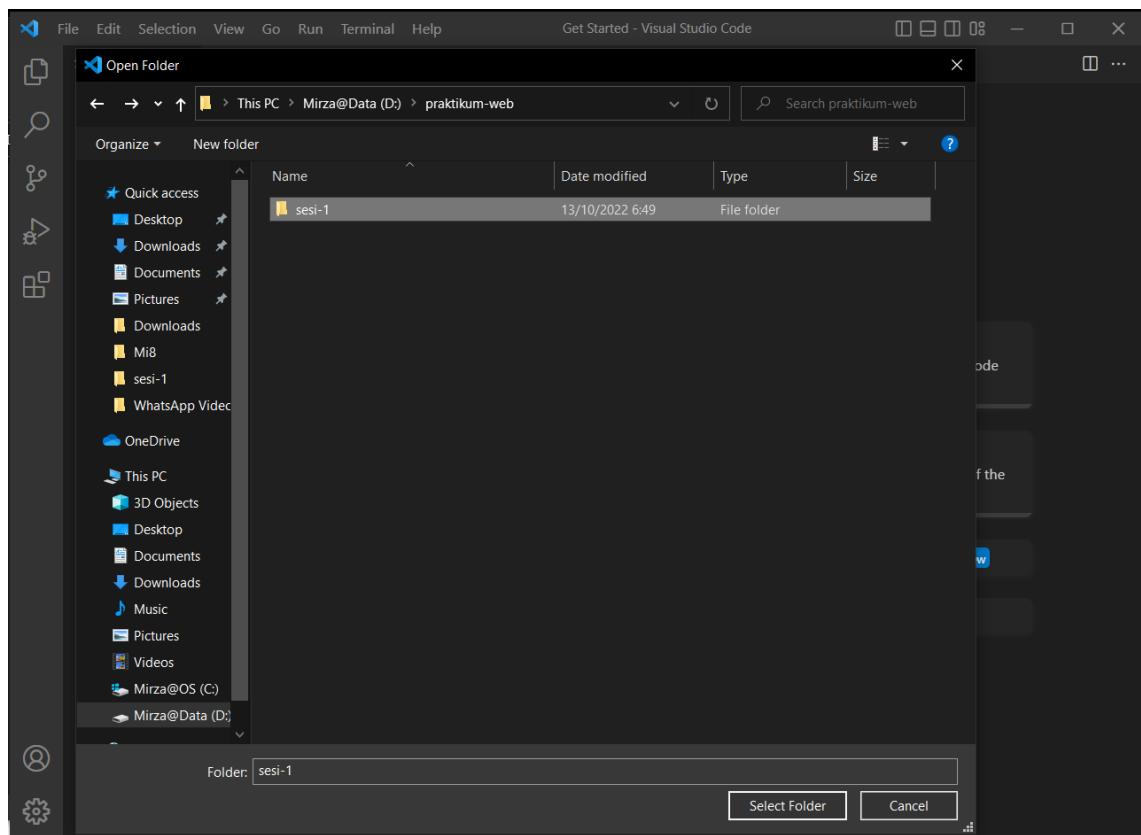
1. Buat folder 'praktikum-web' di dalamnya buat lagi folder sesi-1



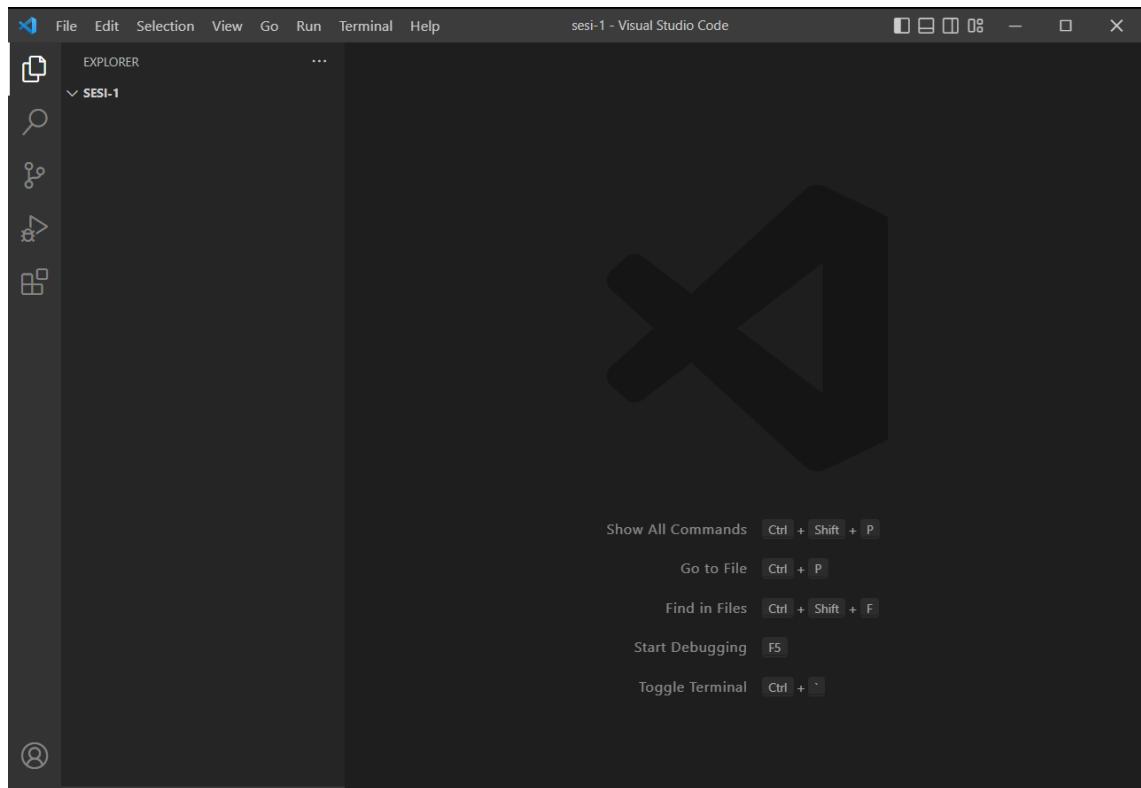
2. Buka Visual Studio Code dan klik Open Folder.



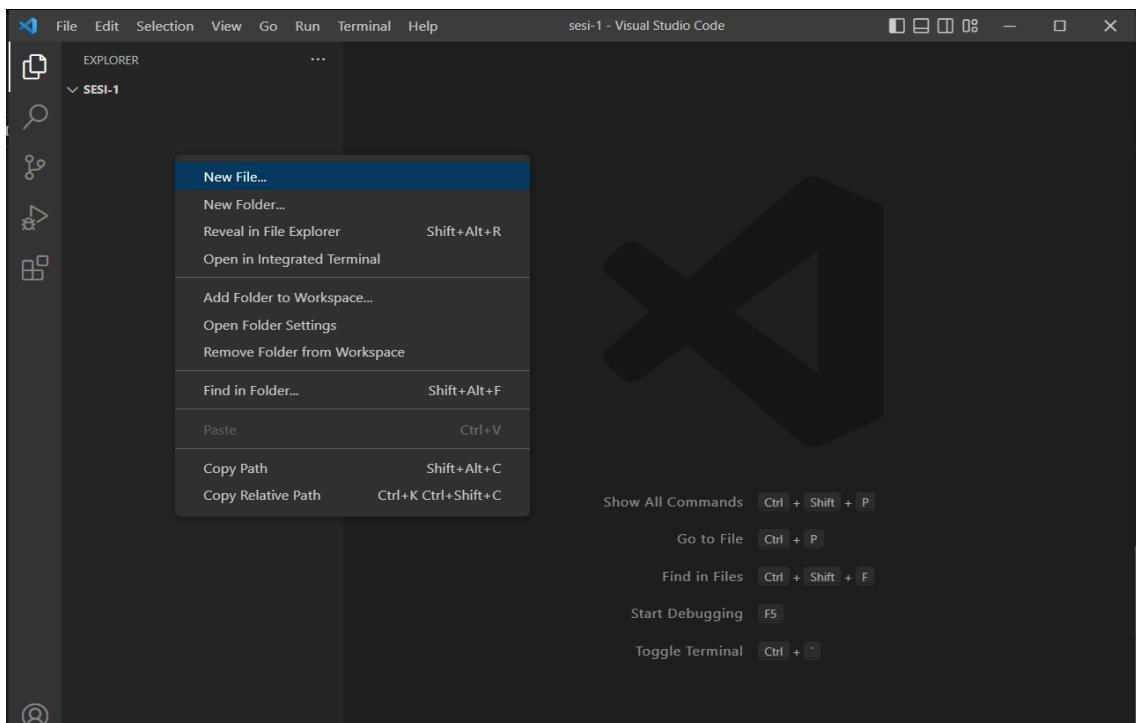
3. Buka folder 'sesi-1' yang sebelumnya dibuat.



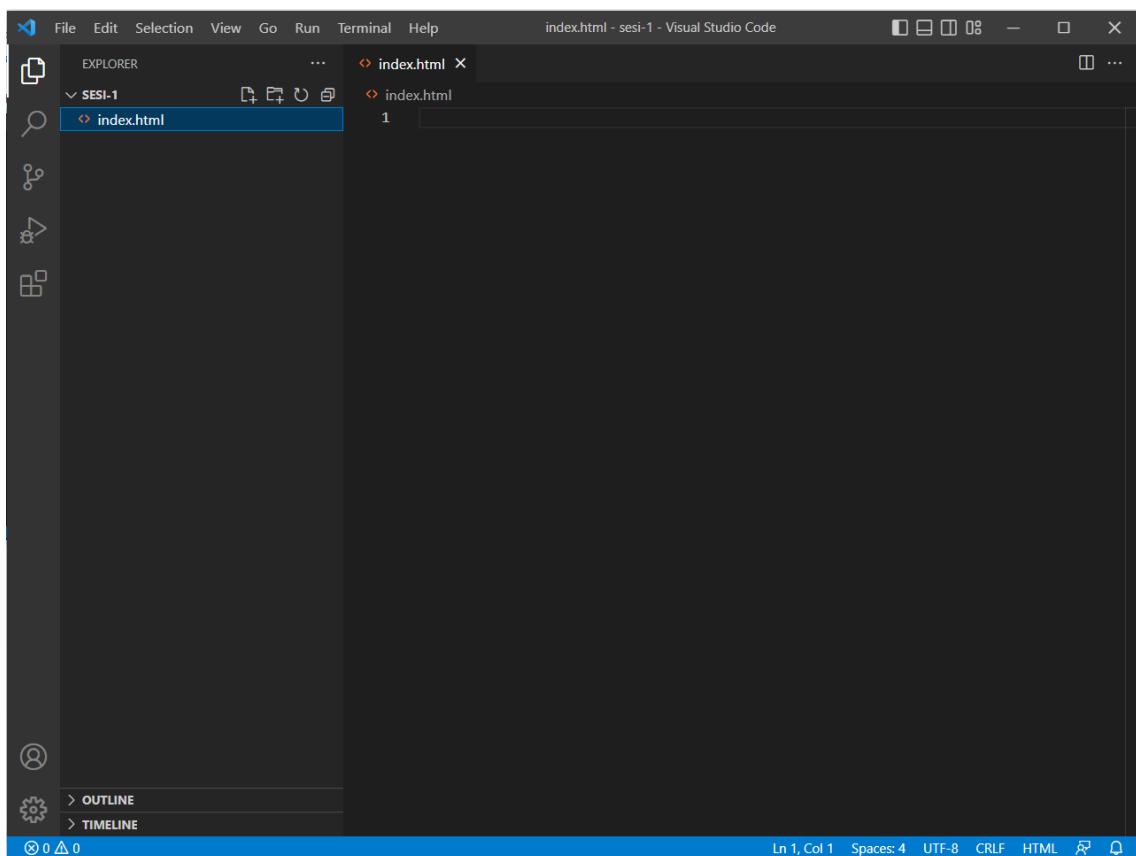
4. Berikut tampilan hasilnya



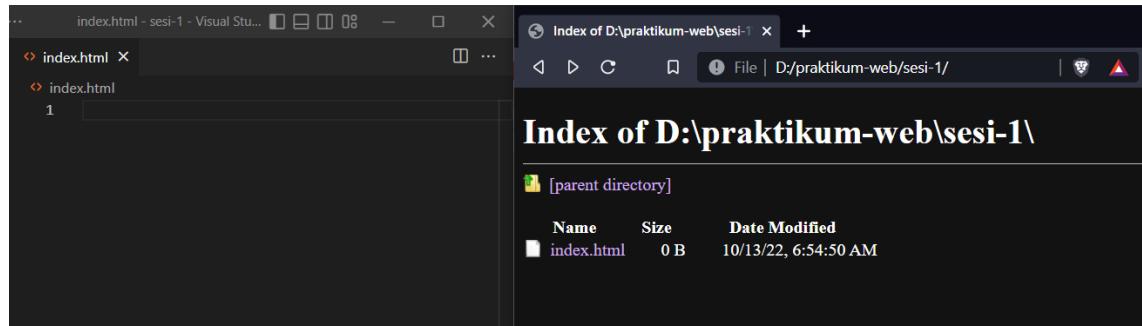
5. Klik kanan pada bagian bawah dari ‘sesi-1’ dan pilih ‘New File...’



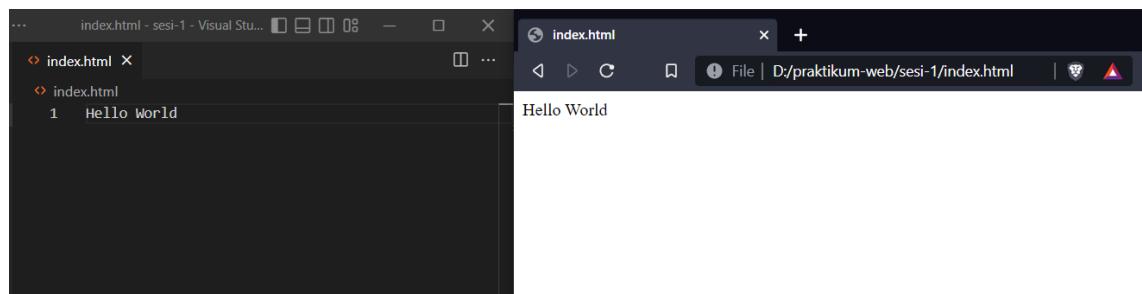
6. Beri nama index.html



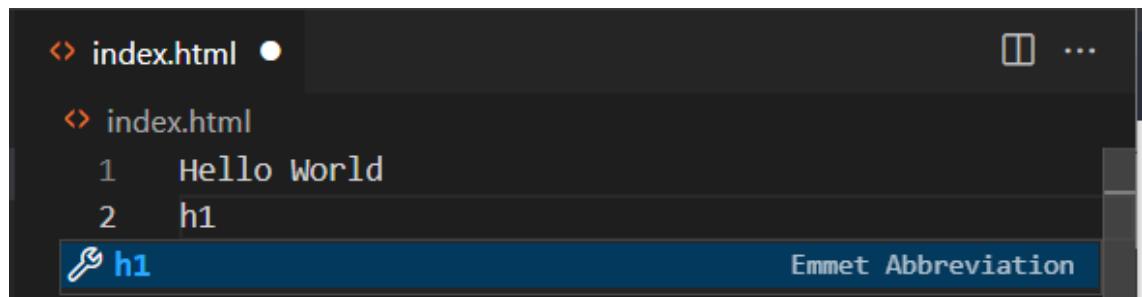
7. Buka peramban (browser) ketikkan alamat folder ‘sesi-1’ dan pilih file index.html



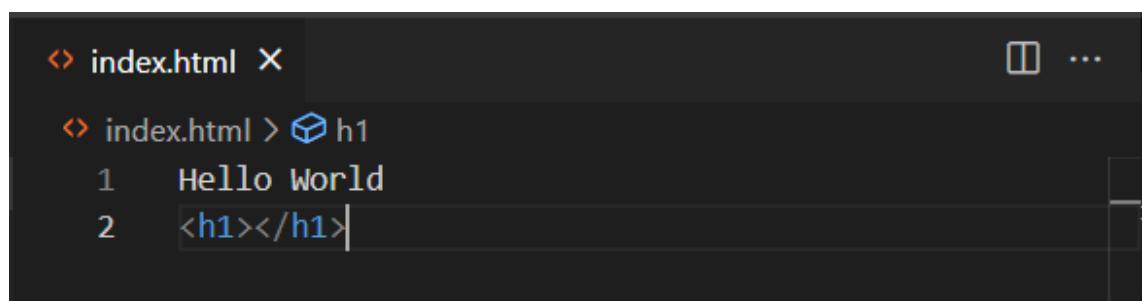
8. Ketikkan perintah, Save pada Code, dan Reload/Refresh pada browser



9. Ketikkan h1

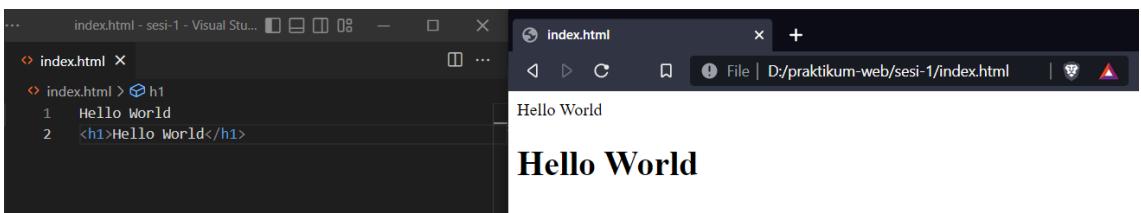


10. Visual Studio Code sudah memiliki Emmet yang membantu penulisan tag HTML, dengan hanya menuliskan h1 maka Emmet akan menuliskan tag <h1> beserta penutupnya </h1>



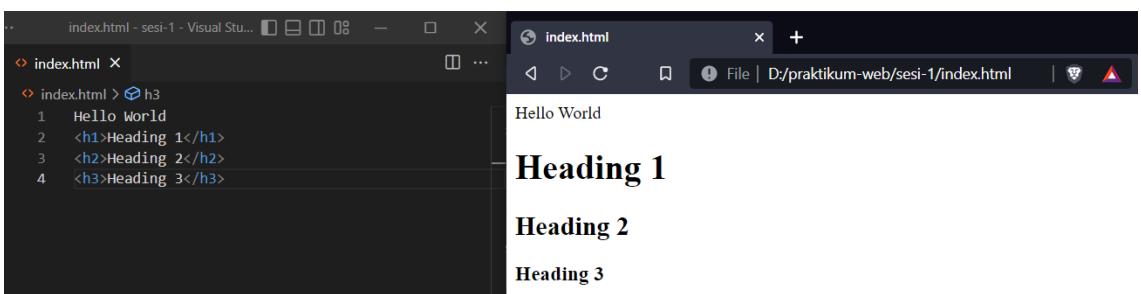
5

11. Tuliskan kembali Hello World diantara tag <h1> dan </h1>, Save pada Code dan Reload pada browser



The screenshot shows the Visual Studio Code interface with an open file named 'index.html'. The code contains a single line: '<h1>Hello World</h1>'. To the right, a browser window titled 'index.html' displays the text 'Hello World' in a large, bold font.

12. Berikutnya tuliskan tag h1, h2, dan h3, kemudian bandingkan hasilnya pada browser



The screenshot shows the Visual Studio Code interface with an open file named 'index.html'. The code contains four lines: '1 Hello World', '2 <h1>Heading 1</h1>', '3 <h2>Heading 2</h2>', and '4 <h3>Heading 3</h3>'. To the right, a browser window titled 'index.html' displays three levels of headings: 'Heading 1' in large bold font, 'Heading 2' in medium bold font, and 'Heading 3' in small bold font.

13. Tuliskan perintah berikut



The screenshot shows the Visual Studio Code interface with an open file named 'index.html'. The code is a multi-sectioned document. On the left, the code editor shows several sections with titles like 'Praktek Pemrograman Web 1', 'Sesi 1', and 'Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya.'. On the right, a browser window titled 'index.html' displays the content of the document, showing the titles and descriptive text.

Perhatikan bahwa meskipun pada Code sudah diberikan jarak dan ganti baris yang menunjukkan pergantian paragraf, tapi pada browser tidak tampak ada perbedaan, tulisan dituliskan begitu saja tanpa diberi jarak.

14. Permasalahan ini bisa diselesaikan dengan menuliskan isi paragraf ke dalam tag untuk paragraf yaitu <p></p>. Perhatikan bahwa cara penulisannya sama dengan seperti pada heading yaitu, teks ditulis di antara <p> dan </p> sehingga didapatkan hasil seperti pada gambar berikut:

The screenshot shows a Visual Studio Code interface with two panes. The left pane displays the HTML code for 'index.html' with line numbers 1 through 6. The right pane shows a browser window displaying the rendered HTML with the title 'Praktek Pemrograman Web 1' and a single paragraph of text.

```
<html>
  <head>
    <title>Praktek Pemrograman Web 1</title>
  </head>
  <body>
    <h1>Praktek Pemrograman Web 1</h1>
    <h2>Sesi 1</h2>
    <p>Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS, dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut menuliskan hasil kreasinya sendiri.</p>
    <p>Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya.</p>
  </body>
</html>
```

15. Penulisan file HTML memiliki struktur tersendiri yang sejatinya tidak seperti yang sebelumnya dituliskan. Pada Code tuliskan tanda seru (!) kemudian tekan tombol ‘tab’ atau ‘enter’

The screenshot shows a Visual Studio Code interface with the 'index.html' file open. The code has been modified to remove the standard HTML boilerplate, specifically the DOCTYPE declaration, the opening and closing html tags, and the head and body sections. Only the main content structure remains.

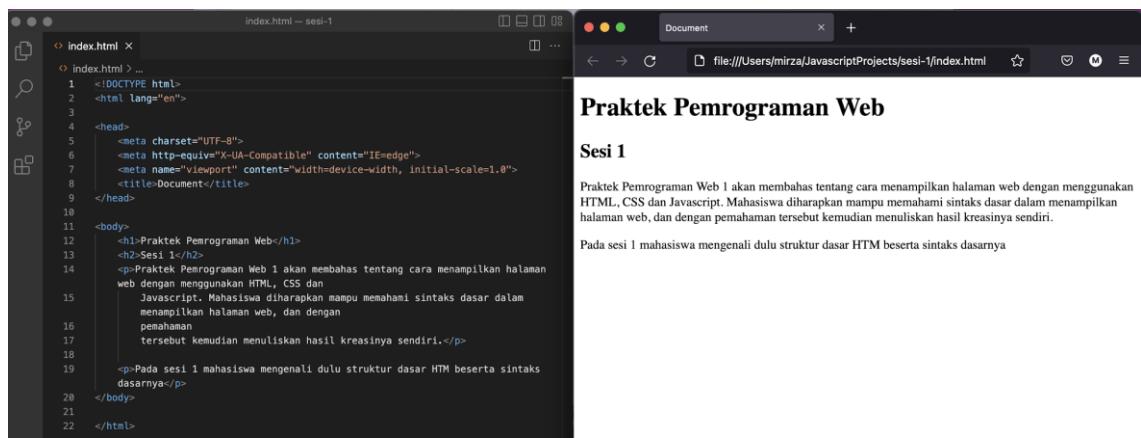
```
<h1>Praktek Pemrograman Web</h1>
<h2>Sesi 1</h2>
<p>Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasi sendiri.</p>
<p>Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya.</p>
```

Perintah tersebut menuliskan kode boilerplate atau kode yang sering dituliskan dan akan selalu dituliskan pada file HTML sehingga daripada selalu ditulis berulang maka dibuat shortcut dengan mengetikkan ! (tanda seru).

Perintah yang dihasilkan digunakan oleh browser untuk mengenali berbagai macam konfigurasi, di antaranya bahasa, device, charset yang mana bisa berbeda jika karakter yang ditulis di dalam halaman bukan merupakan huruf latin, misalkan Kanji atau Hangeul.

Kode ini juga mengelompokkan bagian menggunakan tag <head> dan <body>. Bagian <head> seringnya digunakan untuk menyatakan atau memanggil file CSS yang merupakan konfigurasi style pada halaman HTML, bisa juga memanggil jenis font yang akan digunakan. Terdapat pula tag <title> yang akan memunculkan tulisan pada bagian atas tab browser. Sedangkan isi dari laman website diletakkan pada tag <body>

16. Pindahkan isi laman ke dalam tag <body>



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Praktek Pemrograman Web</h1>
    <h2>Sesi 1</h2>
    <p>Praktek Pemrograman Web I akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasiya sendiri.</p>
    <p>Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTM beserta sintaks dasarnya</p>
</body>
</html>
```

The browser window shows the rendered HTML with the title "Praktek Pemrograman Web" and the content "Sesi 1". The content includes an H1 heading "Praktek Pemrograman Web", an H2 heading "Sesi 1", and two paragraphs explaining the course objectives.

Setelah kita reload, tidak nampak ada perbedaan namun disini HTML sudah ditulis dengan struktur yang lebih baik.

PERCOBAAN 1.2 List

List secara bahasa merupakan daftar, artinya list merupakan teks yang tidak ditulis berbentuk paragraf. List yang tidak ada pengurutnya bisa ditulis menggunakan unordered list , sedangkan yang berurutan dituliskan menggunakan ordered list

1. Tambahkan perintah berikut untuk menampilkan unordered list



The screenshot shows a web browser window with the title "Praktek Pemrograman Web". On the left, there is a code editor displaying the following HTML code:

```
<ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>Javascript</li>
</ul>
```

To the right of the code editor, the page content is displayed under the heading "Sesi 1". It contains the following text and bullet points:

Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasinya sendiri.

- HTML
- CSS
- Javascript

Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTM beserta sintaks dasarnya

2. Tambahkan perintah berikut untuk menampilkan ordered list



The screenshot shows a web browser window with the title "Praktek Pemrograman Web". On the left, there is a code editor displaying the following HTML code:

```
<ol>
    <li>Pengenalan HTML</li>
    <li>Tabel</li>
    <li>Form</li>
</ol>
```

To the right of the code editor, the page content is displayed under the heading "Sesi 1". It contains the following text and numbered list:

Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasinya sendiri.

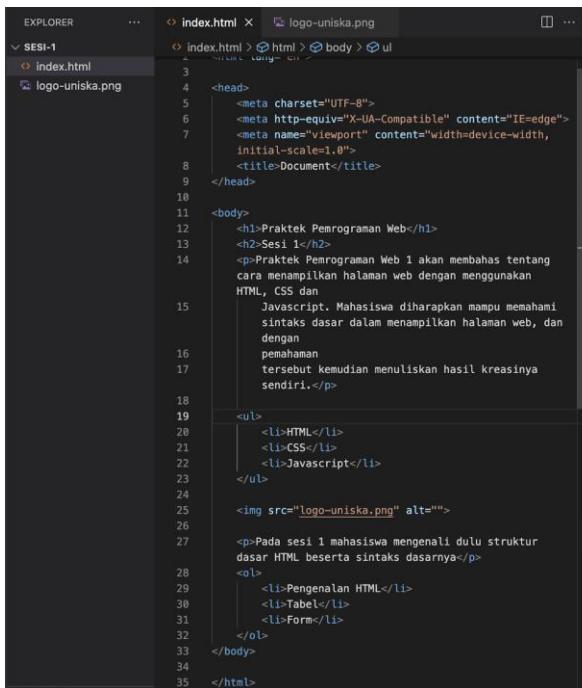
- HTML
- CSS
- Javascript

Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya

1. Pengenalan HTML
2. Tabel
3. Form

PERCOBAAN 1.3 Gambar

Meletakkan gambar pada HTML dapat dilakukan menggunakan tag , berbeda dengan tag yang sudah dibahas sebelumnya, tag ini tidak memiliki tag penutup seperti pada </html> </p> dan lainnya. Tag img memiliki atribut src yang fungsinya untuk menyatakan lokasi gambar yang akan ditampilkan. Misalkan pada gambar berikut:



The screenshot shows a code editor with an HTML file named index.html. The code includes a logo image tag:

```

```

The browser preview window shows the page title "Praktek Pemrograman Web" and the heading "Sesi 1". Below the heading is a paragraph of text and a bulleted list. A large watermark of the Universitas Islam Kalimantan Muhammadiyah Arsyad Al-Banjarmasin logo is overlaid on the page.

Perintah yang digunakan

```

```

Perintah ini memanggil gambar dengan nama logo-uniska.png yang ada dalam folder yang sama dengan file index.html, atribut alt digunakan untuk menampilkan alternatif berupa tulisan jika gambar yang disebutkan di src tidak berhasil ditemukan. Misalkan pada gambar berikut, atribut src diisi dengan logo.jpg yang mana tidak ada filenya, sedangkan atribut alt diisi dengan teks Logo Uniska, sehingga ketika tampil di halaman web adalah teks alt-nya karena file logo.jpg tidak ditemukan

```

index.html
1  <head>
2    <meta charset="UTF-8">
3    <meta http-equiv="X-UA-Compatible" content="IE=edge">
4    <meta name="viewport" content="width=device-width,
5      initial-scale=1.0">
6    <title>Document</title>
7  </head>
8
9  <body>
10
11    <h1>Praktek Pemrograman Web</h1>
12    <h2>Sesi 1</h2>
13    <p>Praktek Pemrograman Web 1 akan membahas tentang
14      cara menampilkan halaman web dengan menggunakan
15      HTML, CSS dan
16      Javascript. Mahasiswa diharapkan mampu memahami
17      sintaks dasar dalam menampilkan halaman web, dan
18      dengan
19      pemahaman
20      tersebut kemudian menuliskan hasil kreasi
21      sendiri.</p>
22
23    <ul>
24      <li>HTML</li>
25      <li>CSS</li>
26      <li>Javascript</li>
27    </ul>
28
29    

```

Praktek Pemrograman Web

Sesi 1

Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasi sendiri.

- HTML
- CSS
- Javascript

Logo Uniska

Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya

1. Pengenalan HTML
2. Tabel
3. Form

Atribut src juga bisa diisi dengan lokasi file gambar dari internet, sehingga asalkan komputer yang digunakan sedang terhubung ke internet maka halaman web akan menampilkan gambar berdasarkan lokasi tersebut. Jika tidak terhubung internet atau gagal mengambil gambar, maka akan muncul teks yang sudah diisikan pada atribut alt. Contoh akan dipanggil gambar melalui internet dengan atribut src diisikan alamat <http://fti.uniska-bjm.ac.id/wp-content/uploads/2018/08/cropped-logo-uniska-ok.png>. Hasilnya dapat dilihat pada gambar berikut:

The screenshot shows a browser window with the following details:

- Title Bar:** index.html
- Address Bar:** file:///Users/mirza/JavascriptProjects/sesi-1/index.html
- Content Area:**
 - Page Title:** Praktek Pemrograman Web
 - Section Title:** Sesi 1
 - Text Content:** Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasi sendiri.
 - List:** HTML, CSS, Javascript
 - Image:** Logo of Universitas Islam Kalimantan Muhammadiyah Arsyad Al-Banjarmasin, featuring a green book and a crescent moon.
 - Text:** Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya
 - List:** 1. Pengenalan HTML, 2. Tabel, 3. Form

KESIMPULAN 1

1. Menggunakan simbol tanda seru (!) untuk menuliskan boilerplate tag html
2. Tag heading (h1, h2, h3, h4, h5, h6) untuk menuliskan judul atau sub judul
3. Tag p untuk menuliskan paragraf
4. Menuliskan daftar dengan tag , jika tanpa pengurutan maka dibungkus dengan tag jika dengan urutan atau nomor maka dibungkus dengan
5. Memanggil gambar menggunakan tag dan atribut src

LATIHAN 1.1

Buat sebuah halaman hasil kreasi sendiri menggunakan tag html yang sudah dipelajari (html, heading, paragraf, list, img)

PERCOBAAN 1.4 Anchor

Tag anchor <a> yang memiliki atribut href, digunakan untuk navigasi baik pada halaman itu sendiri maupun ke halaman web lain. Misalkan list yang sebelumnya ditambahkan anchor seperti berikut

```
<ul>
  <li><a href="#html"> HTML</a></li>
  <li><a href="#css"> CSS</a></li>
  <li><a href="#javascript"> Javascript</a></li>
</ul>
```

Penggunaan tanda pagar (tagar/hashtag) merupakan penanda bahwa jika anchor ini di klik maka akan menuju pada elemen yang memiliki id sesuai dengan teks pada tagar tersebut. Sehingga agar anchor ini berfungsi bisa ditambahkan elemen berikut

```
<h2 id="html">HTML</h2>

<p>Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya</p>
<ol>
  <li>Pengenalan HTML</li>
  <li>Tabel</li>
  <li>Form</li>
</ol>
<h2 id="css">CSS</h2>

<p>Cascading Style Sheet (CSS) merupakan standar untuk mengatur tampilan yang ada pada suatu berkas HTML. CSS bukan bahasa pemrograman tetapi merupakan bahasa deklaratif yang dinyatakan langsung untuk menyatakan suatu tampilan, tidak terdapat logika, percabangan, ataupun perulangan.</p>
<h2 id="javascript">Javascript</h2>

<p>Javascript merupakan bahasa pemrograman yang berjalan disisi browser, disisi pengguna, atau di peramban bukan diserver</p>
```

Pada gambar, ada tiga h2 yang diberikan id, yaitu HTML, CSS dan Javascript, sehingga ketika list yang berisi anchor tadi diklik maka akan menggeser tampilan peramban kepada elemen dengan id tersebut.

Selain itu tentunya kita juga bisa membuat anchor yang memanggil halaman web online misalkan

```
<ul>
    <li><a href="#html"> HTML</a></li>
    <li><a href="#css"> CSS</a></li>
    <li><a href="#javascript"> Javascript</a></li>
    <li><a href="https://google.com"> Google</a></li>

</ul>
```

Anchor memiliki atribut yang bisa diatur agar jika pengguna mengkliknya maka URL tersebut akan dibuka pada tab baru, yaitu dengan menambahkan target="_blank"

```
<ul>
    <li><a href="#html"> HTML</a></li>
    <li><a href="#css"> CSS</a></li>
    <li><a href="#javascript"> Javascript</a></li>
    <li><a href="https://google.com" target="_blank"> Google</a></li>
</ul>
```

LATIHAN 1.2

Berikutnya buat halaman serupa dengan halaman berikut, gambar bisa mengambil langsung dari internet ataupun download dan letakkan bersama dalam folder sesi-1

Praktek Pemrograman Web

Sesi 1

H2

H1

Praktek Pemrograman Web 1 akan membahas tentang cara menampilkan halaman web dengan menggunakan HTML, CSS dan Javascript. Mahasiswa diharapkan mampu memahami sintaks dasar dalam menampilkan halaman web, dan dengan pemahaman tersebut kemudian menuliskan hasil kreasinya sendiri.

p

- [HTML](#)
- [CSS](#)
- [Javascript](#)
- [Google](#)

ul, li, dan anchor



img

HTML

H2



img

Pada sesi 1 mahasiswa mengenali dulu struktur dasar HTML beserta sintaks dasarnya

p

1. Pengenalan HTML
2. Tabel
3. Form

ol dan li

CSS

H2



img

Cascading Style Sheet (CSS) merupakan standar untuk mengatur tampilan yang ada pada suatu berkas HTML. CSS bukan bahasa pemrograman tetapi merupakan bahasa deklaratif yang dinyatakan langsung untuk menyatakan suatu tampilan, tidak terdapat logika, percabangan, ataupun perulangan.

p

Javascript

H2



JavaScript

img

Javascript merupakan bahasa pemrograman yang berjalan disisi browser, disisi pengguna, atau di peramban bukan diserver

p

SESI 2

Pengenalan CSS

TUJUAN PRAKTEK

1. Mahasiswa mampu membuat berkas CSS beserta sintaks dasarnya
2. Mahasiswa mampu membuat dan menghubungkan HTML dengan CSS
3. Mahasiswa mampu membuat dan menerapkan box model

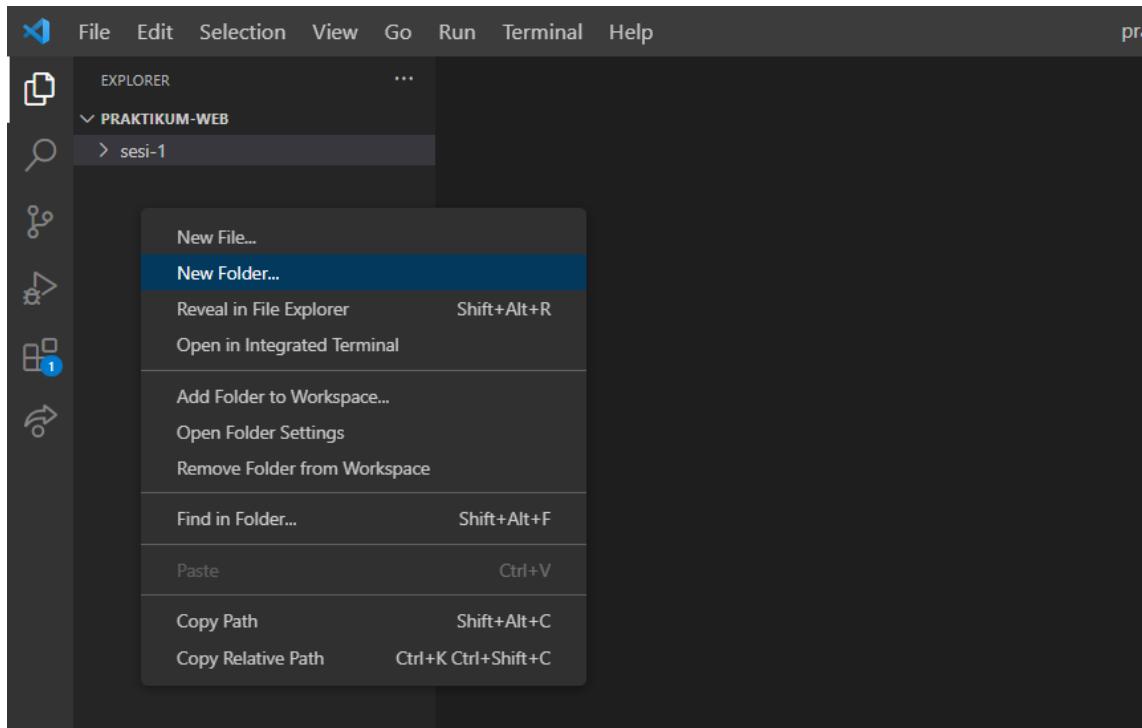
PENDAHULUAN

Pada pelajaran sebelumnya, kita telah mempelajari cara menulis HTML yang menentukan struktur halaman web. Langkah selanjutnya adalah membuat struktur itu terlihat bagus dengan beberapa gaya (style) dengan menggunakan CSS. Dalam sesi ini, kita akan fokus pada beberapa konsep dasar CSS.

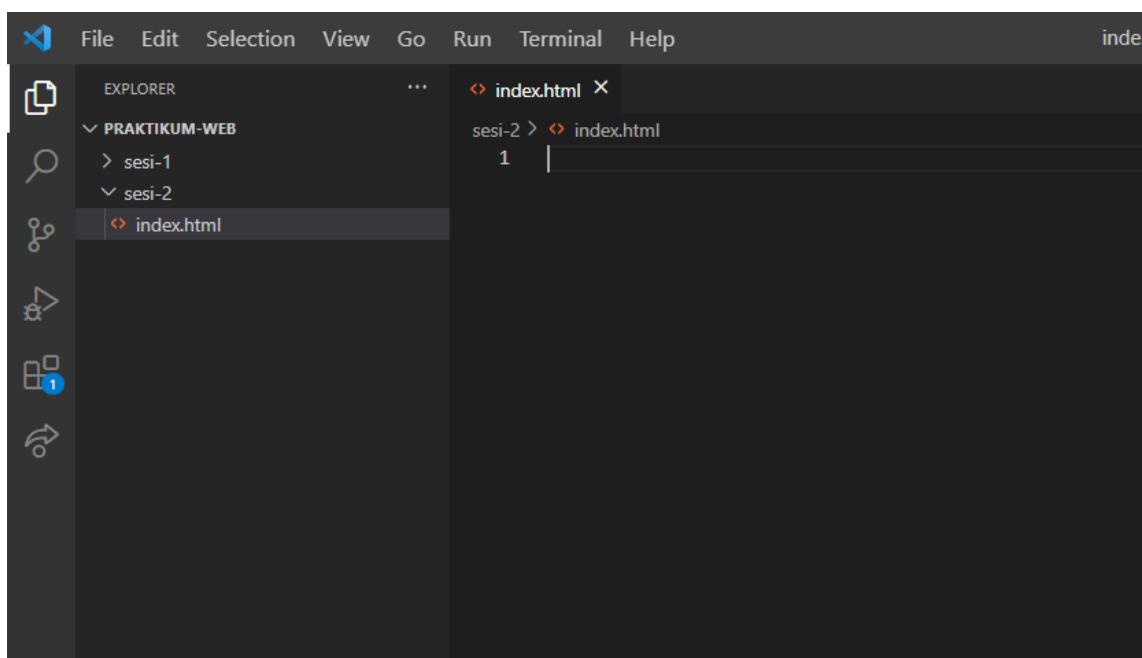
CSS adalah singkatan dari Cascading Style Sheets. Ini adalah bahasa desain yang digunakan untuk mengatur tampilan sebuah halaman web. CSS memungkinkan kita untuk mengatur elemen seperti font, warna, dan tata letak halaman web agar terlihat lebih menarik dan mudah dibaca. CSS sangat penting untuk dipelajari karena menjadi cara kita untuk membuat halaman web yang terlihat lebih profesional dan menarik. Dengan CSS, kita dapat mengontrol tampilan halaman web dengan lebih baik dan memberikan pengguna pengalaman yang juga lebih baik saat menjelajah situs kita. Selain itu, CSS juga membantu menjaga konsistensi tampilan di seluruh halaman web. kita dapat menuliskan aturan CSS yang sama untuk elemen yang sama di seluruh halaman web, sehingga membuatnya lebih mudah untuk mengelola dan memperbarui situs.

PERCOBAAN 2.1 Mengenal CSS

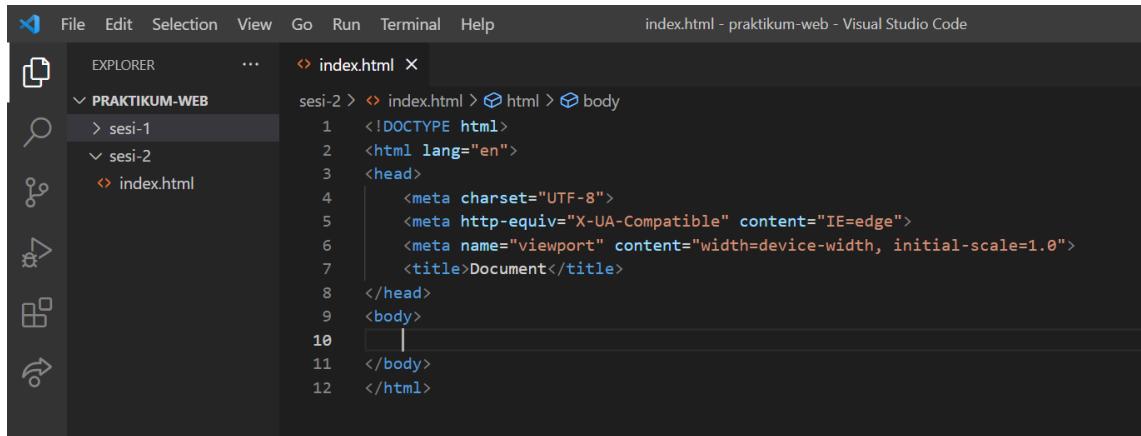
1. Klik kanan di bagian Explorer Visual Studio Code, dan buat folder sesi-2 di dalam folder ‘praktikum-web’



2. Buka folder ‘sesi-2’ yang sebelumnya dibuat dan buat file dengan nama index.html.



3. Buat boilerplate html dengan menuliskan tanda seru kemudian tekan tombol ‘tab’ atau ‘enter’.

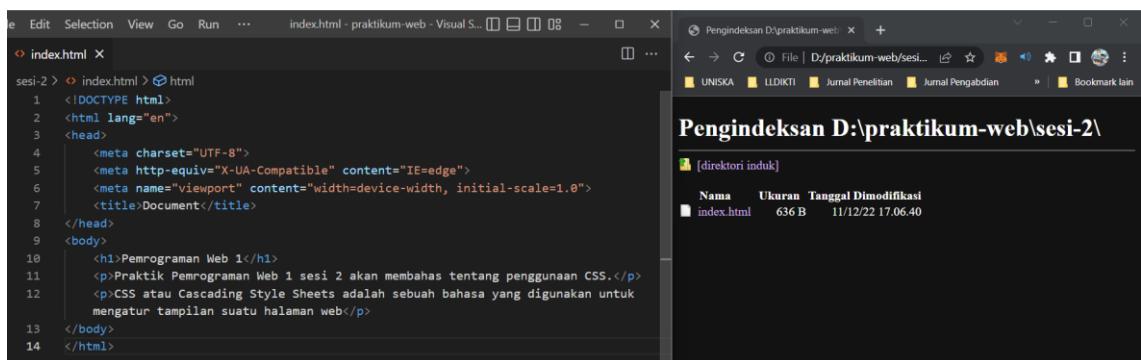


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
</body>
</html>
```

4. Ketikkan kode berikut pada body

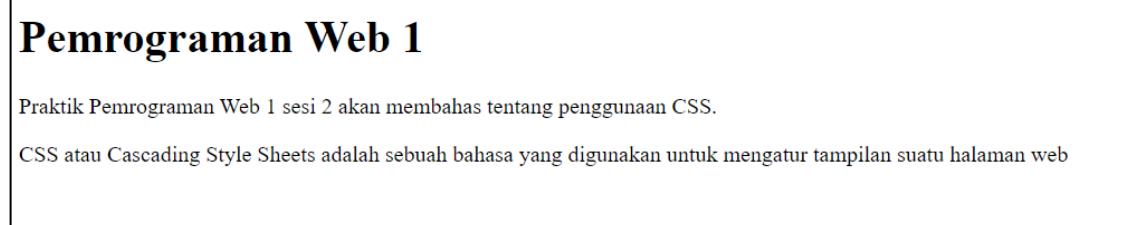
```
<body>
    <h1>Pemrograman Web 1</h1>
    <p>Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>
    <p>CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>
</body>
```

5. Buka peramban (browser) ketikkan alamat folder ‘sesi-2’ dan pilih file index.html



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>Pemrograman Web 1</h1>
    <p>Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>
    <p>CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>
</body>
</html>
```

6. Maka akan tampil seperti berikut



Pemrograman Web 1

Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.

CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web

Jika kita lihat, tulisan yang ditampilkan hanya tulisan biasa (plain text) berwarna hitam tanpa hiasan apapun. Dengan menggunakan CSS kita dapat menghias tulisan tersebut.

7. Tambahkan kode berikut di dalam ‘head’, kemudian refresh/reload

The screenshot shows a browser window with two main panes. The left pane is a code editor displaying an HTML file named 'index.html'. It contains the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        * {
            color: blue;
        }
    </style>
</head>
<body>
    <h1>Pemrograman Web 1</h1>
    <p>Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>
    <p>CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>
</body>
</html>
```

A red box highlights the CSS rule `* { color: blue; }`. The right pane shows a preview of the page with the heading 'Pemrograman Web 1' in blue.

Kode tersebut adalah kode CSS yang membuat semua warna tulisan di dalam halaman tersebut menjadi berwarna biru. Hal tersebut terjadi karena kita menggunakan selector universal yang ditandai dengan sintaks bintang (*) dengan properti warna bernilai biru (color: blue).

8. Tambahkan kode berikut di bagian style, kemudian refresh/reload dan lihat perbedaan dengan sebelumnya



```
<html>
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
      * {
        color: blue;
      }

      h1 {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1>Pemrograman Web 1</h1>
    <p>Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>
    <p>CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>
  </body>
</html>
```

Jika kita lihat warna tulisan heading 1 berubah menjadi warna hijau, ini karena kita menambahkan selector `<h1>` dengan warna hijau (`color: green`), namun selector `<p>` tetap berwarna biru karena pengaruh dari selector universal. Dari sini kita mengetahui bahwa salah satu cara untuk menghias komponen html adalah dengan memanggil tipe element html tersebut dan ini sering juga disebut sebagai tipe/elemen selector. Selain menggunakan tipe selector, kita juga dapat menggunakan class selector dan id selector.

PERCOBAAN 2.2 Class Selector dan ID Selector

Pada percobaan sebelumnya kita telah mengenal tipe/elemen selector, selain itu terdapat juga class selector dan ID selector. Perbedaan utama antara class selector dan ID selector adalah bahwa class selector digunakan untuk mengatur gaya untuk kelompok elemen HTML, sedangkan ID selector digunakan untuk mengatur gaya untuk elemen individu.

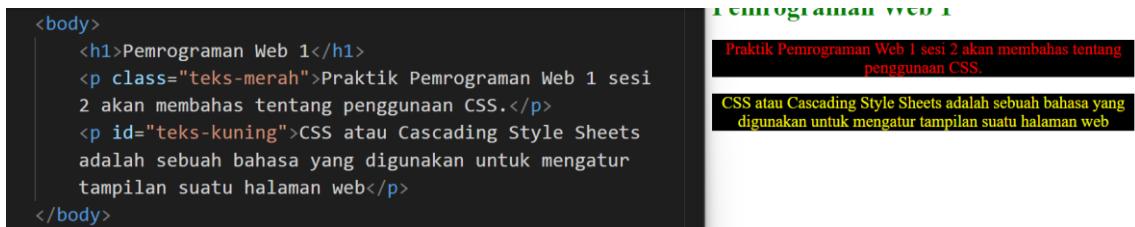
Sebuah elemen HTML dapat memiliki lebih dari satu class, tetapi hanya bisa memiliki satu ID. Ini berarti bahwa class selector lebih fleksibel daripada ID selector. Selain itu, ID selector lebih spesifik dari pada class selector, sehingga jika ada konflik antara style yang ditentukan oleh class selector dan ID selector, style yang ditentukan oleh ID selector akan dijalankan.

Untuk pemanggilan pada CSS, class selector ditulis dengan tanda titik (.) di depan nama class-nya, sedangkan ID selector ditulis dengan tanda kurung kurawal (#) di depan nama ID-nya. Sebagai contoh, class selector untuk elemen dengan class nama-class akan ditulis sebagai `.nama-class`, sedangkan ID selector untuk elemen dengan ID nama-id akan ditulis sebagai `#nama-id`.

1. Ketikkan kode ini di bagian style

```
.teks-merah {  
    font-size: 20px;  
    text-align: center;  
    background-color: black;  
    color: red;  
}  
  
.teks-kuning {  
    font-size: 20px;  
    text-align: center;  
    background-color: black;  
    color: yellow;  
}
```

2. Tambahkan class dan id pada elemen paragraf kemudian reload halaman



The screenshot shows a simple web page with an H1 header and two paragraphs. The first paragraph uses the 'teks-merah' class, and the second paragraph uses the 'teks-kuning' ID. The browser's developer tools are open, showing the applied CSS rules for each element.

```
<body>  
    <h1>Pemrograman Web 1</h1>  
    <p class="teks-merah">Praktik Pemrograman Web 1 sesi  
2 akan membahas tentang penggunaan CSS.</p>  
    <p id="teks-kuning">CSS atau Cascading Style Sheets  
adalah sebuah bahasa yang digunakan untuk mengatur  
tampilan suatu halaman web</p>  
</body>
```

Pemrograman Web 1
Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.
CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web

3. Jika kita lihat pada class teks-merah dan id teks-kuning mempunyai beberapa properti yang sama, untuk menghemat kode, kita dapat menggabungkan properti yang sama dengan membuat grup dipisahkan tanda koma. Ubah kode menjadi seperti ini

```
.teks-merah,  
#teks-kuning {  
    font-size: 20px;  
    text-align: center;  
    background-color: black;  
}  
  
.teks-merah {  
    color: red;  
}  
  
#teks-kuning {  
    color: yellow;  
}
```

Jika halaman kita reload, tidak akan terlihat perubahan, tetapi kode kita sekarang lebih efisien karena kita membuat grup untuk properti yang sama. Kita dapat membuat grup antara class dengan class, id dengan id ataupun class dengan id.

PERCOBAAN 2.3 Descendant Selector dan Child Selector

Descendant selector digunakan untuk memilih elemen HTML yang berada dalam elemen lain, sedangkan child selector digunakan untuk memilih elemen HTML yang merupakan anak langsung dari elemen lain. Perbedaannya adalah bahwa descendant selector memilih semua elemen yang berada di dalam elemen lain, termasuk elemen-elemen yang berada di dalam elemen anak dari elemen tersebut.

Sedangkan child selector hanya memilih elemen-elemen yang merupakan anak langsung dari elemen lain, tidak termasuk elemen-elemen yang berada di dalam elemen anak dari elemen tersebut. Mari kita buat contoh kode yang menggunakan descendant selector dan child selector. Tambahkan kode ini pada bagian style:

```
#teks-kuning {  
    color: yellow;  
}  
  
div p {  
    color: red;  
}  
  
div > div > p{  
    font-weight: bold;  
}  
/>style>
```

Kemudian tambahkan kode berikut ke bagian body:

```
<body>
  <h1>Pemrograman Web 1</h1>
  <p class="teks-merah">Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>
  <p id="teks-kuning">CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>

  <div>
    <p>Ini adalah paragraf pertama.</p>
    <div>
      <p>Ini adalah paragraf kedua.</p>
    </div>
  </div>

</body>
```

Reload/refresh halaman, hasilnya akan terlihat seperti ini:

Pemrograman Web 1

Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.

CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web

Ini adalah paragraf pertama.

Ini adalah paragraf kedua.

Dengan kode di atas, elemen paragraf pertama akan berubah menjadi warna merah, ini karena kita menggunakan descendant selector dengan menuliskan div p { ... } yang membuat semua paragraph di dalam sebuah div akan berubah menjadi merah, sedangkan elemen paragraf kedua akan berubah menjadi warna merah dan tebal karena kita menggunakan child selector dengan menuliskan div > div > p { ... } yang secara spesifik hanya akan membuat paragraf di dalam 2 buah div yang dicetak tebal.

Sebagai catatan di kode tersebut terdapat elemen <div>, elemen <div> adalah salah satu elemen HTML dasar yang merupakan sebuah container/wadah kosong yang digunakan untuk membagi dokumen HTML menjadi bagian-bagian yang lebih kecil. Elemen <div> tidak memiliki fungsi tertentu secara default, sehingga kita dapat menggunakannya untuk membagi dokumen menjadi bagian-bagian yang lebih kecil.

PERCOBAAN 2.4 Chaining Selector dan Pseudo Selector

Chaining selector adalah cara untuk menggabungkan beberapa selector dalam satu baris kode dengan menggunakan spasi. Ini memungkinkan kita untuk memberikan gaya yang sama untuk elemen HTML yang berada dalam struktur elemen tertentu. Ketikkan kode ini di bagian style:

```
.subbagian {  
    font-size: 30px;  
}  
  
.subbagian.header {  
    color: blueviolet;  
}  
  
.subbagian#teks {  
    color: brown;  
}
```

Dan ketikkan kode ini pada bagian body:

```
<body>  
    <h1>Pemrograman Web 1</h1>  
    <p class="teks-merah">Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.</p>  
    <p id="teks-kuning">CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web</p>  
  
    <div>  
        <p>Ini adalah paragraf pertama.</p>  
        <div>  
            <p>Ini adalah paragraf kedua.</p>  
        </div>  
    </div>  
  
    <div>  
        <div class="subbagian header">Bagian header</div>  
        <p class="subbagian" id="teks">Ini adalah paragraf</p>  
    </div>  
  
</body>  
</html>
```

Pemrograman Web 1

Praktik Pemrograman Web 1 sesi 2 akan membahas tentang penggunaan CSS.

CSS atau Cascading Style Sheets adalah sebuah bahasa yang digunakan untuk mengatur tampilan suatu halaman web

Ini adalah paragraf pertama.

Ini adalah paragraf kedua.

Bagian header

Ini adalah paragraf

Kita lihat pada kode CSS terdapat Chaining Selector dengan nama .subbagian.header, yang artinya stylenya akan aktif pada elemen yang mempunyai class="subbagian header". Selain itu terdapat juga .subbagian#teks yang stylenya akan aktif pada elemen yang mempunyai class="subbagian" id="teks". Perlu

diperhatikan untuk penulisan Chaining Selector di CSS tidak perlu spasi karena kalau menggunakan spasi akan menjadi Descendant Selector.

Pseudo-selector adalah sebuah fitur di CSS yang memungkinkan kita untuk memilih elemen HTML berdasarkan kondisi tertentu yang tidak dapat dinyatakan dengan menggunakan selektor biasa. Pseudo-selector sering digunakan untuk mengubah gaya elemen berdasarkan interaksi pengguna dengan elemen tersebut, seperti ketika kita menunjuk elemen dengan mouse atau menekan tombol pada elemen tersebut.

Beberapa contoh pseudo-selector yang umum digunakan adalah:

- `:hover`: digunakan untuk mengubah gaya elemen ketika pengguna menunjuk elemen tersebut dengan mouse.
- `:active`: digunakan untuk mengubah gaya elemen ketika pengguna menekan tombol pada elemen tersebut.
- `:focus`: digunakan untuk mengubah gaya elemen ketika elemen tersebut mendapat fokus, misalnya ketika pengguna mengklik pada elemen tersebut atau menggunakan tombol panah untuk berpindah ke elemen tersebut.
- `:before` dan `:after`: digunakan untuk menambahkan konten sebelum atau setelah elemen HTML tertentu.

Untuk mencoba salah satu pseudo selector tambahkan kode berikut pada `<style>`:

```
p:hover {  
    background-color: #aqua;  
}
```

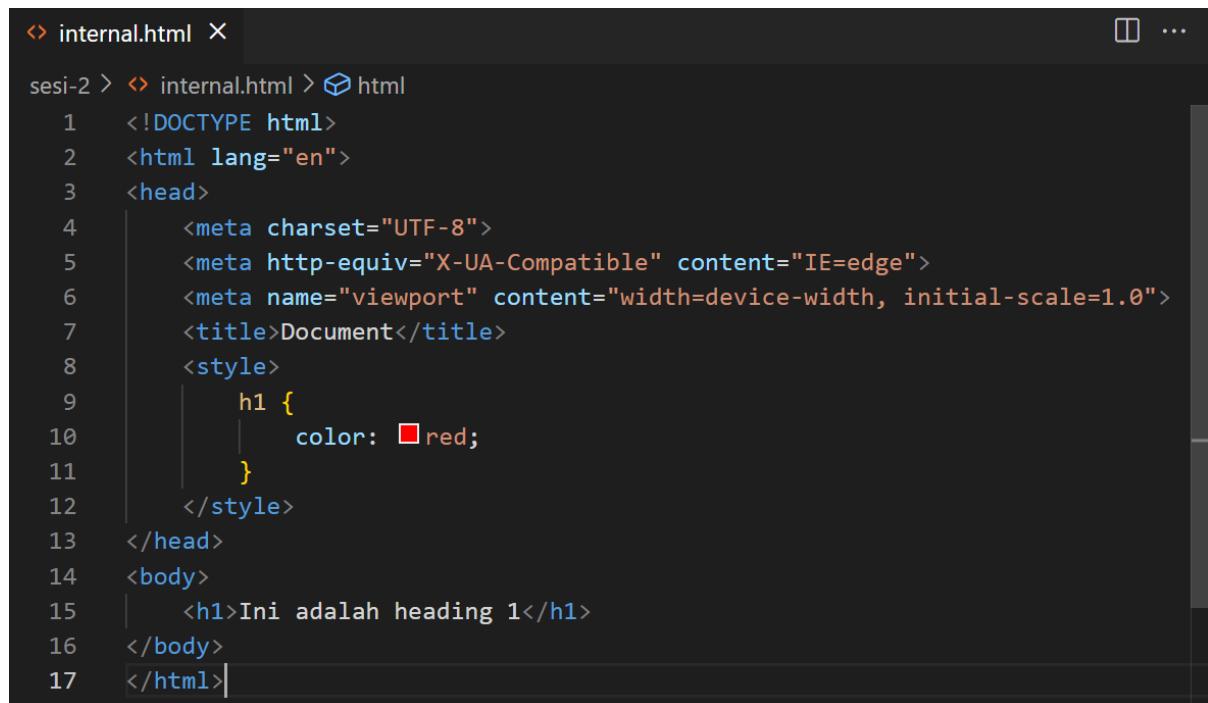
Reload/refresh index.html dan arahkan mouse ke salah satu paragraf, maka warna latar paragraf tersebut akan berubah menjadi warna aqua, ini disebabkan oleh pseudo selector :hover yang aktif saat kita mengarahkan mouse ke elemennya.

PERCOBAAN 2.5 Menambahkan CSS ke HTML

Untuk menambahkan kode CSS ke dalam HTML, kita dapat menggunakan beberapa cara, antara lain:

1. Internal CSS

Menambahkan CSS secara internal adalah cara yang kita lakukan pada percobaan-percobaannya sebelumnya, yaitu dengan menambahkan kode CSS di dalam tag style pada head HTML. Untuk lebih memahami, buat sebuah file dengan nama internal.html dan isikan kode berikut:

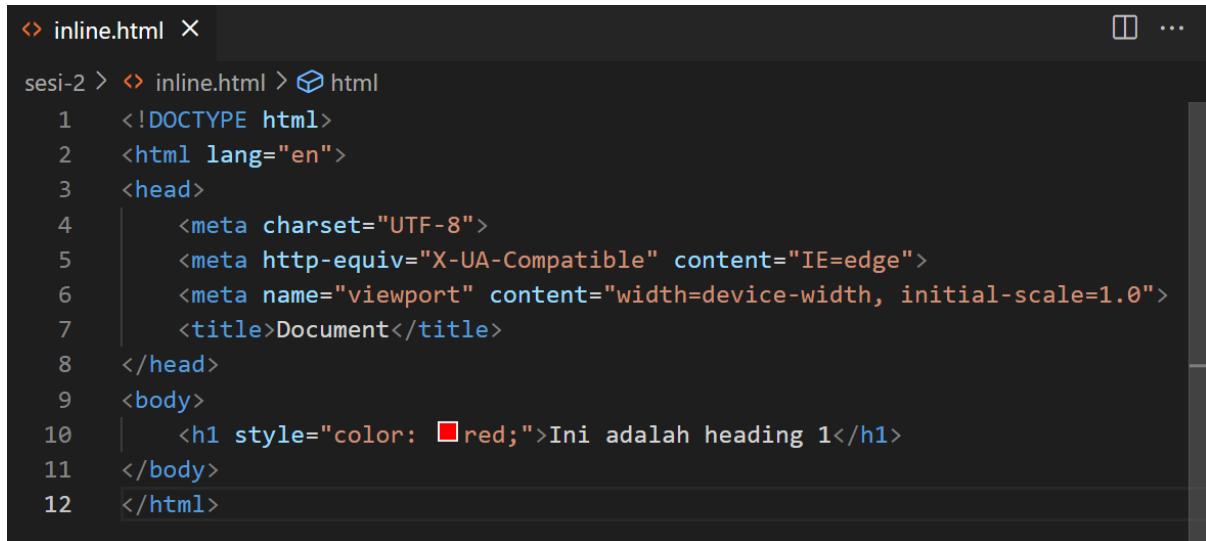


```
internal.html X
sesi-2 > internal.html > html
1  !DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <style>
9          h1 {
10              color: red;
11          }
12      </style>
13  </head>
14  <body>
15      <h1>Ini adalah heading 1</h1>
16  </body>
17  </html>
```

2. Inline CSS

Inline CSS adalah cara yang dilakukan dengan menambahkan style langsung di dalam elemen HTML yang ingin kita ubah tampilannya. Atribut style tersebut berisi

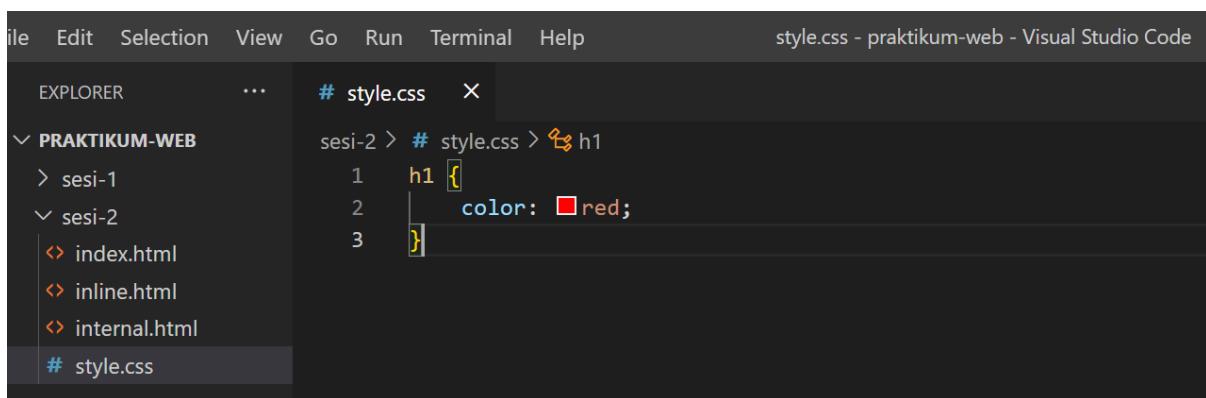
kode CSS yang akan diterapkan pada elemen tersebut. Uji coba dengan membuat file baru bernama inline.html dan ketikkan kode berikut:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1 style="color: red;">Ini adalah heading 1</h1>
</body>
</html>
```

3. Eksternal CSS

Eksternal CSS adalah sebuah file CSS yang terpisah dari file HTML dan dihubungkan ke dalam file HTML menggunakan tag `<link>` di dalam `<head>` element. Untuk mencoba eksternal CSS, buat file baru dengan nama style.css dan isikan kode berikut:



```
h1 {
    color: red;
```

Selanjutnya buat file eksternal.html dan panggil style.css tersebut pada bagian `<head>` elemen dengan cara mengetikkan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Document</title>
</head>
<body>
    <h1>Ini adalah heading 1</h1>
</body>
</html>
```

Dari ketiga cara di atas, menggunakan file CSS eksternal dengan tag `<link>` dianggap sebagai cara yang direkomendasikan untuk memasukkan CSS ke dalam HTML. Hal ini karena menggunakan file CSS eksternal akan memudahkan kita untuk mengelola dan memperbarui aturan CSS, terutama jika dokumen HTML kita cukup besar. Ini akan membuat kode HTML kita lebih terstruktur dan mudah dibaca. Sementara itu, cara lain seperti menambahkan tag `<style>` di dalam tag `<head>` atau menambahkan aturan CSS langsung ke elemen HTML bisa menjadi pilihan yang tepat jika kita hanya ingin menambahkan aturan CSS sederhana dan tidak memiliki banyak aturan CSS yang perlu dikelola.

PERCOBAAN 2.6 Box Model

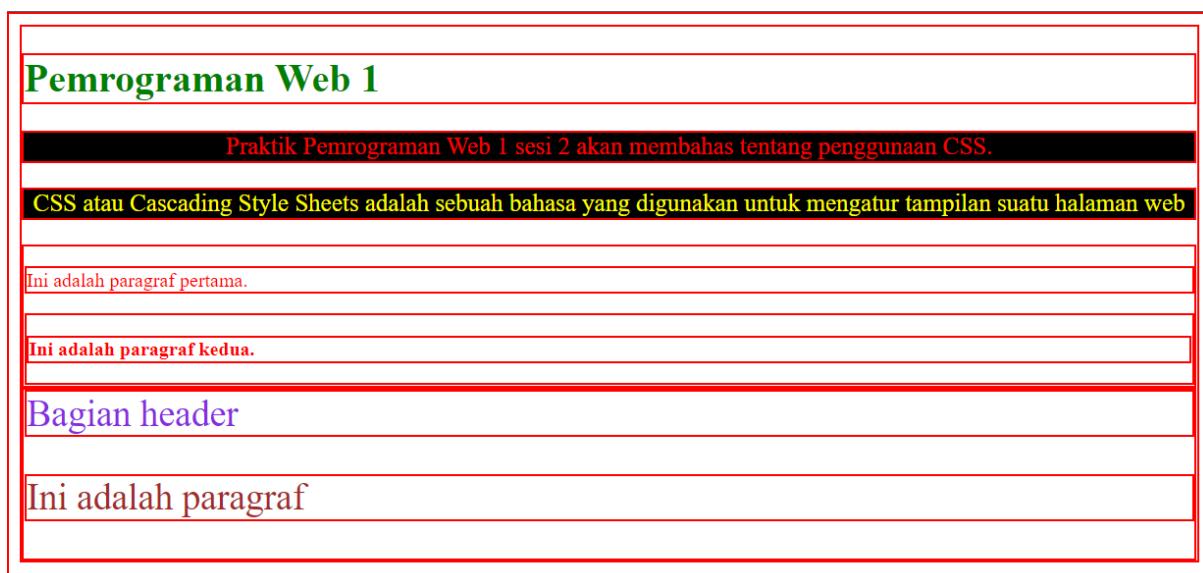
Kita telah memahami sintaks dasar HTML dan CSS, selanjutnya kita akan mencoba memahami tentang box model. Salah satu keahlian yang kita perlukan untuk menguasai CSS adalah positioning dan layout. Mengubah jenis huruf, warna dan ukuran adalah keahlian penting, tetapi kemampuan untuk menempatkan elemen tepat di tempat yang diinginkan di halaman web lebih penting, tanpa kemampuan tersebut kita akan frustrasi mengatur elemen yang akan bertumpuk satu dengan yang lain.

Belajar memosisikan elemen di halaman web tidak terlalu sulit saat kita sudah memahami beberapa konsep kunci.

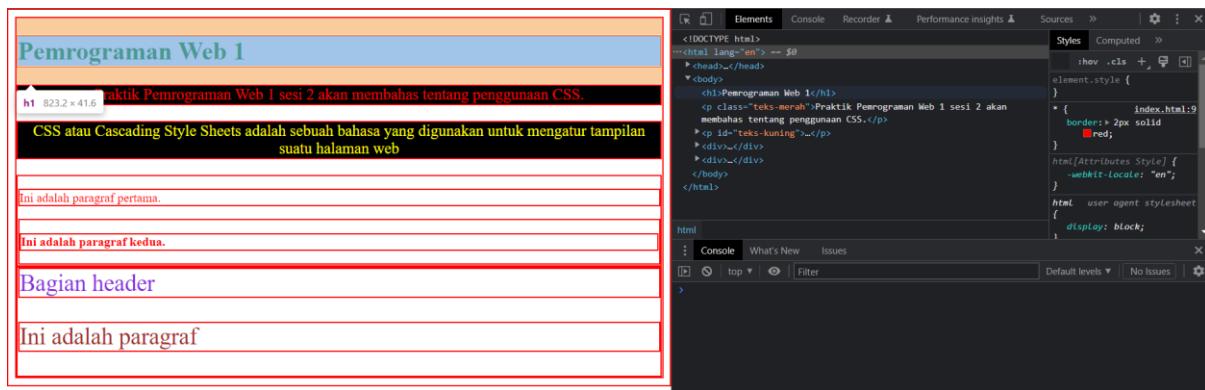
Konsep pertama disebut sebagai box model, kita perlu melihat semua yang ada di sebuah halaman web adalah sebuah kotak/persegi panjang, kotak tersebut dapat memiliki kotak lain di dalamnya dan dapat memiliki posisi berdampingan satu sama lain. Untuk mendapat gambaran yang lebih jelas, buka kembali index.html yang ada di folder sesi-2 dan tambahkan kode berikut pada universal selectornya:

```
* {  
    color: blue;  
    border: 2px solid red;  
}
```

Buka kembali halaman index.html tersebut dan kita akan mendapatkan halaman seperti ini:



Terlihat bahwa elemen yang tersusun terdiri dari kotak/persegi panjang yang tersusun dan saling bertumpuk. Untuk mengetahui kotak tersebut dimiliki oleh elemen tertentu, kita dapat klik kanan peramban dan memilih inspeksi (inspect element) kemudian memilih pada elemennya.



Pada akhirnya, menata halaman web dan memosisikan semua elemennya adalah cara kita menyusun dan menumpuk kotak-kotak ini. Untuk mengubah ukuran kotak-kotak tersebut dan mengatur jarak di antaranya dapat menggunakan padding, margin dan border.

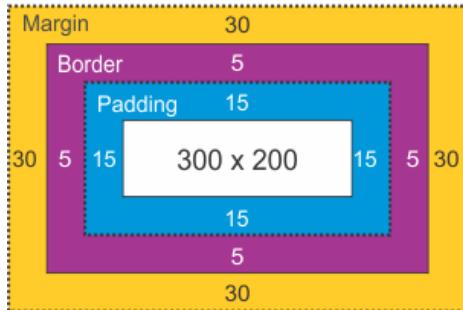
PERCOBAAN 2.7 Padding, Margin dan Border

Padding, margin, dan border adalah properti yang digunakan dalam tata letak (layout) CSS untuk mengatur ruang di sekitar elemen HTML.

1. Padding adalah ruang di dalam elemen, di antara isi elemen dan border. Padding dapat diatur untuk setiap sisi elemen secara terpisah, misalnya padding-top, padding-right, padding-bottom, dan padding-left.
2. Margin adalah ruang di luar elemen, di antara border dan elemen lain atau sisi dokumen. Margin juga dapat diatur untuk setiap sisi elemen secara terpisah, misalnya margin-top, margin-right, margin-bottom, dan margin-left.
3. Border adalah garis di sekitar elemen yang dapat diatur lebar, gaya, dan warna. Seperti padding dan margin, border juga dapat diatur untuk setiap sisi elemen secara terpisah, misalnya border-top, border-right, border-bottom, dan border-left.

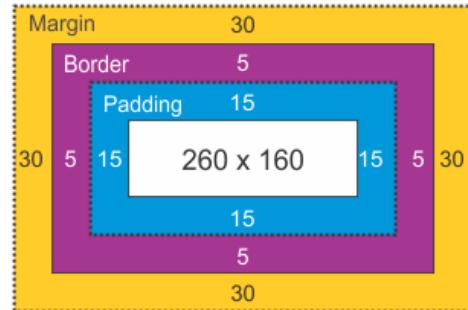
Berikut gambar yang dapat digunakan untuk melihat posisi padding, margin dan border:

Box Model is content-box



```
div{  
    width: 300px;  
    height: 200px;  
    padding: 15px;  
    border: 5px solid grey;  
    margin: 30px;  
    -moz-box-sizing: content-box;  
    -webkit-box-sizing: content-box;  
    box-sizing: content-box;  
}
```

Box Model is border-box



```
div{  
    width: 300px;  
    height: 200px;  
    padding: 15px;  
    border: 5px solid grey;  
    margin: 30px;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

Pada gambar di atas dapat dilihat posisi padding, margin dan border di dua jenis box model yaitu content-box dan border-box. Dengan content box, padding dan border tidak termasuk dalam ukuran elemen, sedangkan dengan border box, padding dan border termasuk dalam ukuran elemen.

Mari kita coba membuat kodennya, buka kembali file style.css di folder sesi-2 dan tambahkan kode berikut:

```
.kotak-satu {  
    padding: 20px;  
    margin: 10px;  
    border: 2px;  
    background-color: lightblue;  
}  
  
.kotak-dua {  
    padding: 20px;  
    margin: 10px;  
    border: 2px solid red;  
    background-color: palegreen;  
    max-width: 50%;  
}  
  
.kotak-tiga [  
    padding-left: 50px;  
    margin-top: 50px;  
    margin-left: 50px;  
    border-left: 2px dotted red;  
    background-color: burlywood;  
    width: 300px;  
    height: 200px;  
]
```

Buka file eksternal.html, tambahkan tiga buah <div> dan panggil class seperti berikut:

```
<body>
  <h1>Ini adalah heading 1</h1>

  <div class="kotak-satu">
    Kotak satu
  </div>

  <div class="kotak-dua">
    Kotak dua
  </div>

  <div class="kotak-tiga">
    Kotak tiga
  </div>
</body>
```

Reload/refresh halaman eksternal.html dan lihat pengaruh padding, margin, dan border pada kotak-kotak tersebut.

PERCOBAAN 2.8 Flexbox

Flexbox atau flexible box adalah sebuah modul layout di CSS yang memungkinkan kita mengelola elemen HTML sebagai sebuah grup yang dapat diatur secara fleksibel. Ini berguna ketika kita ingin mengelola elemen dengan lebih baik dalam layout responsif, cara membuat elemen diatur secara vertikal atau horizontal, atau menentukan elemen harus mengalokasikan ruang yang tersedia. Dengan Flexbox kita mendapatkan beberapa fitur yang berguna, seperti:

- Flex container: elemen yang mengelola elemen anaknya sebagai sebuah grup.
- Flex item: elemen anak dari flex container.
- Flex direction: menentukan arah elemen anak dari flex container (horizontal atau vertikal).
- Flex wrap: menentukan apakah elemen anak dapat dibungkus ke baris baru atau tidak.

- Flex flow: sebuah shorthand untuk mengatur flex direction dan flex wrap sekaligus.
- Justify content: mengatur bagaimana elemen anak dari flex container disebar pada sumbu main/horisontal (sumbu yang sejajar dengan arah flex).
- Align items: mengatur bagaimana elemen anak dari flex container disebar pada sumbu cross/vertikal (sumbu yang tegak lurus dengan arah flex).

Untuk mencoba flexbox, buat file `flexbox.html` dan isikan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="flex.css">
<title>Document</title>
</head>
<body>
  <div class="container">
    <div class="satu">Kotak 1</div>
    <div class="dua">Kotak 2</div>
    <div class="tiga">Kotak 3</div>
  </div>
</body>
</html>
```

Lanjutkan dengan membuat file `flex.css` yang sebelumnya telah kita hubungkan, isi dengan kode berikut:

```
.container div {
  background: #peachpuff;
  border: 5px solid #brown;
  height: 100px;
}
```

Jalankan kode tersebut dan hasilnya akan seperti ini:



Terlihat bahwa elemen <div> yang kita buat di dalam class container akan bertumpuk dan memenuhi halaman secara vertikal. Untuk melihat perbedaan saat menggunakan flexbox, sesuaikan kode flex.css menjadi seperti berikut:

```
.container {  
    display: flex;  
}  
.container div {  
    background: #peachpuff;  
    border: 5px solid #brown;  
    height: 100px;  
    flex: 1;  
}
```

Pada kode di atas kita menambahkan display: flex pada bagian class container yang menandakan class container akan menjadi flex container, kemudian kita menambahkan juga flex: 1 di semua elemen div yang artinya div di dalam class container akan menjadi flex item, sehingga semua div akan memenuhi sumbu horizontal container dan ditampilkan dengan ukuran yang sama. Hasilnya dapat dilihat seperti berikut:



Dari contoh sederhana itu, kita dapat melihat perbedaan sebelum dan sesudah menggunakan flexbox, flexbox dapat menjadi cara utama kita untuk membangun layout web yang rumit, tentu ada beberapa properti lain yang perlu kita pelajari, properti flexbox seperti flex direction, flex wrap, flex flow, justify content, align items dan sebagainya.

Dinginkan kepala dulu, kita akan coba belajar beberapa properti tersebut dengan bermain game sederhana, buka web <https://flexboxfroggy.com> dan coba selesaikan semua levelnya.

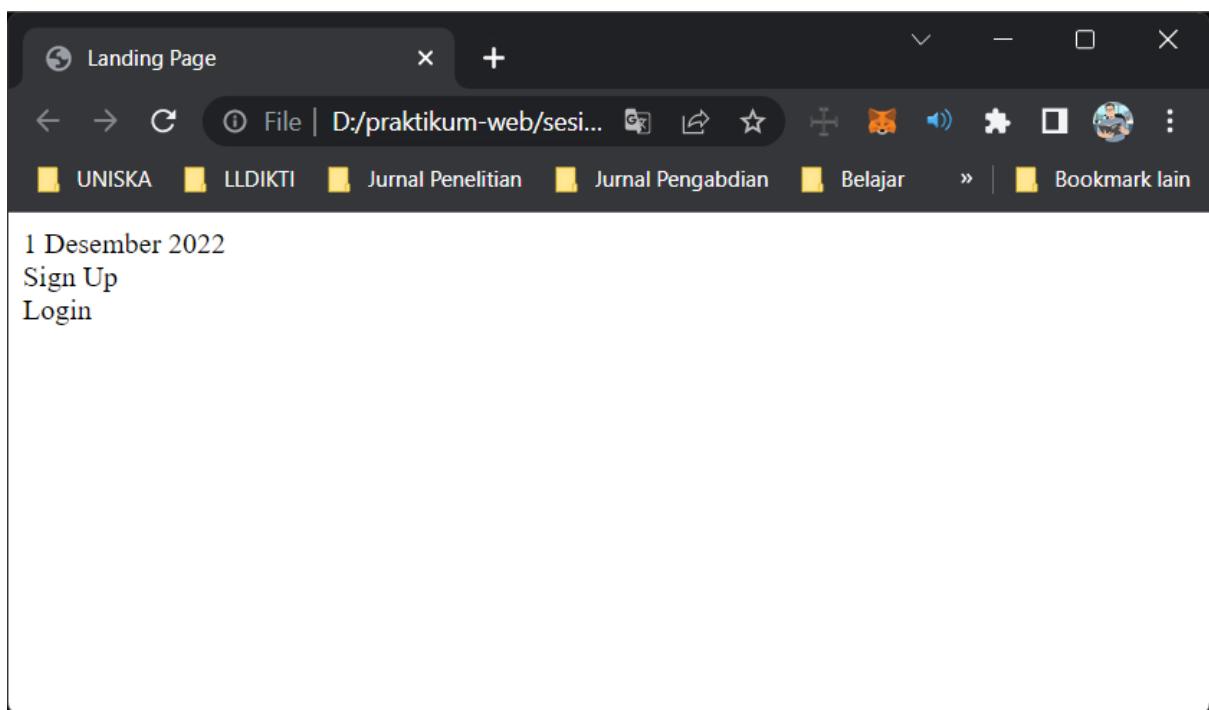
PERCOBAAN 2.9 Membuat Landing Page

Pada percobaan ini kita akan membuat sebuah halaman web sederhana, tetapi cukup jelas memberikan contoh semua penggunaan properti-properti flexbox.

- Untuk memulai, buat file landingpage.html yang hanya berisi menu bar, tambahkan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="styles.css">
    <title>Landing Page</title>
</head>
<body>
    <div class="menu-container">
        <div class="menu">
            <div class="date">1 Desember 2022</div>
            <div class="signup">Sign Up</div>
            <div class="login">Login</div>
        </div>
    </div>
</body>
</html>
```

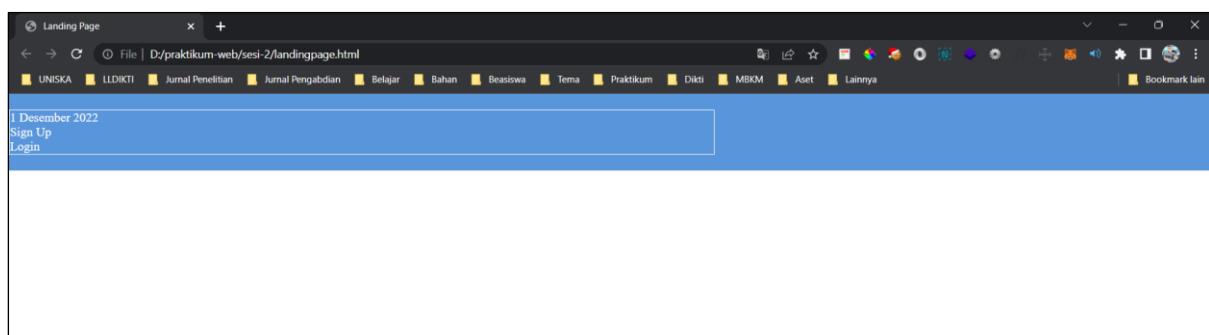
Buka landingpage.html melalui peramban, berikut tampilan yang seharusnya muncul:



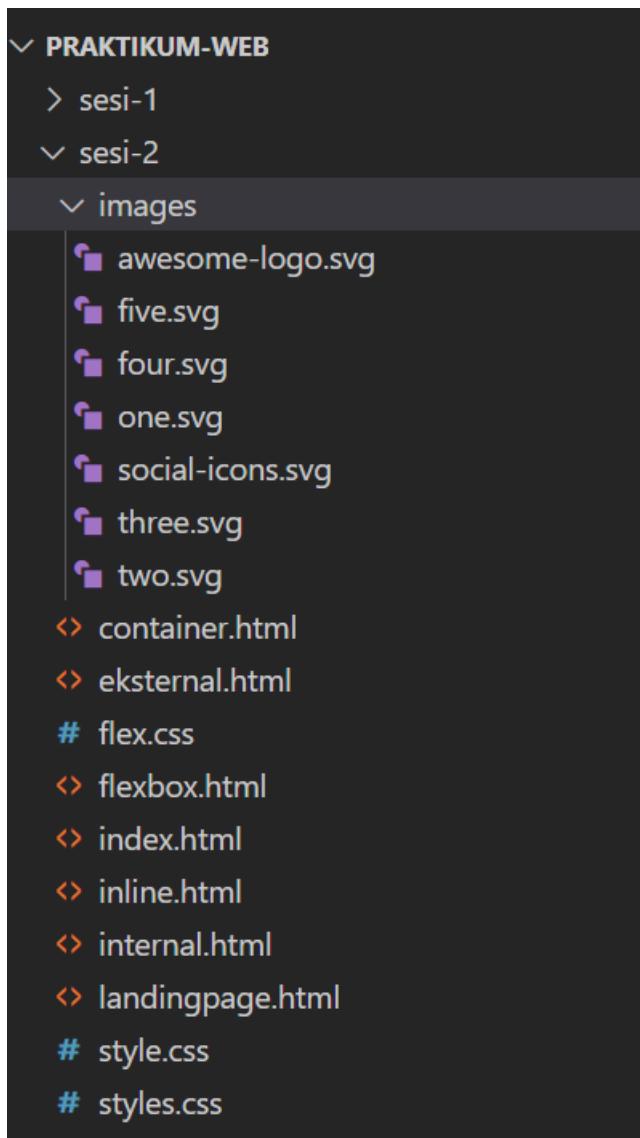
2. Selanjutnya kita perlu membuat styles.css yang sebelumnya telah kita panggil di landingpage.html. Kita akan mengisi file tersebut agar dapat membuat menu bar berwarna biru dengan border putih. Ketikkan kode berikut pada styles.css:

```
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}  
  
.menu-container {  
    color: #fff;  
    background-color: #5995DA;  
    padding: 20px 0;  
}  
  
.menu {  
    border: 1px solid #fff;  
    width: 900px;  
}
```

Reload/refresh halaman landingpage, berikut tampilan yang harusnya muncul:



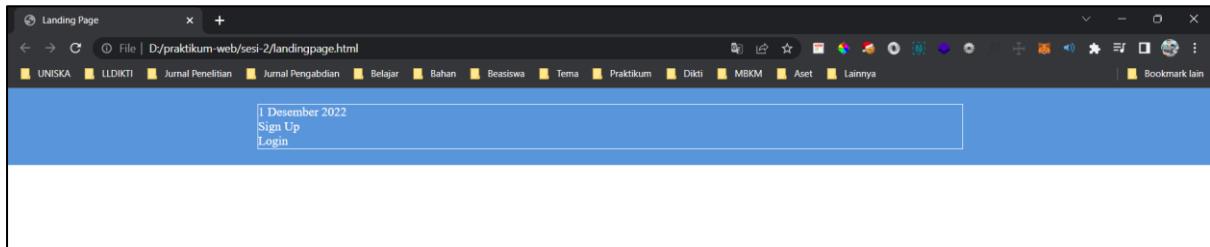
3. Sebelum melanjutkan, unduh gambar di tautan <https://github.com/sikidal/web1-2022/raw/main/images.rar> yang akan kita gunakan pada halaman kita. Ekstrak dan masukkan ke dalam folder sesi-2, pastikan gambar tetap di dalam folder images. Folder proyek kita sekarang, harusnya seperti gambar di bawah ini:



4. Kita akan menggunakan flexbox di dalam landingpage dan kita sudah mengetahui bahwa flexbox menggunakan dua buah tipe box, yaitu flex container dan flex item. Pertama kita buat menu-container menjadi flex container dan kita posisikan di tengah. Tambahkan kode berikut di class menu-container:

```
.menu-container {  
    color: #fff;  
    background-color: #5995DA;  
    padding: 20px 0;  
    display: flex;  
    justify-content: center;  
}
```

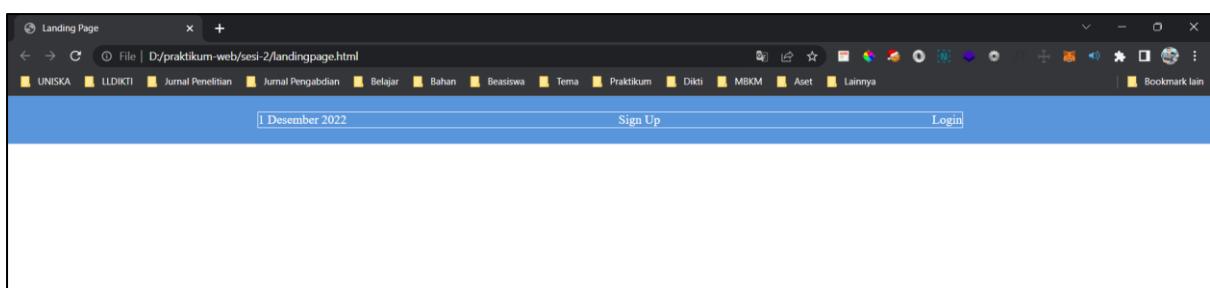
Reload halaman landingpage dan kita akan melihat kotak menu akan bergeser ke tengah:



5. Jika kita lihat tulisan tanggal dan menu masih bertumpuk, kita buat agar tersusun secara horisontal dan memenuhi container menu. Tambahkan kode berikut pada class menu:

```
.menu {  
    border: 1px solid #fff;  
    width: 900px;  
    display: flex;  
    justify-content: space-between;  
}
```

Reload halaman landingpage dan terlihat seperti berikut:



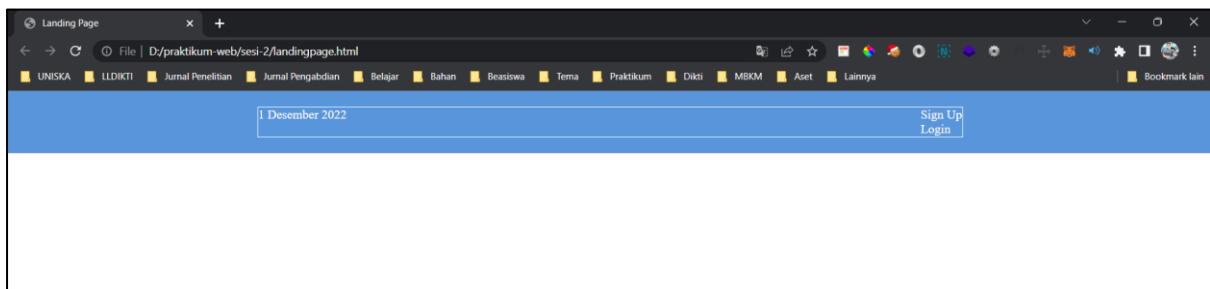
6. Selanjutnya kita rapikan agar tanggal berada di sebelah kiri, sedangkan menu Sign Up dan Login kita pindah ke sebelah kanan. Untuk mengelompokkan menu Sign Up dan Login kita masukkan keduanya ke dalam sebuah class yang kita beri nama links. Sesuaikan kode landingpage menjadi seperti berikut:

```

<body>
  <div class="menu-container">
    <div class="menu">
      <div class="date">1 Desember 2022</div>
      <div class="links">
        <div class="signup">Sign Up</div>
        <div class="login">Login</div>
      </div>
    </div>
  </div>
</body>

```

Reload halaman landingpage dan terlihat seperti berikut:



7. Menu Sign Up dan Login menjadi bertumpuk karena keduanya masuk ke dalam class links, kita perbaiki dengan menjadikan class links sebagai flex container. Tambahkan kode berikut ke styles.css:

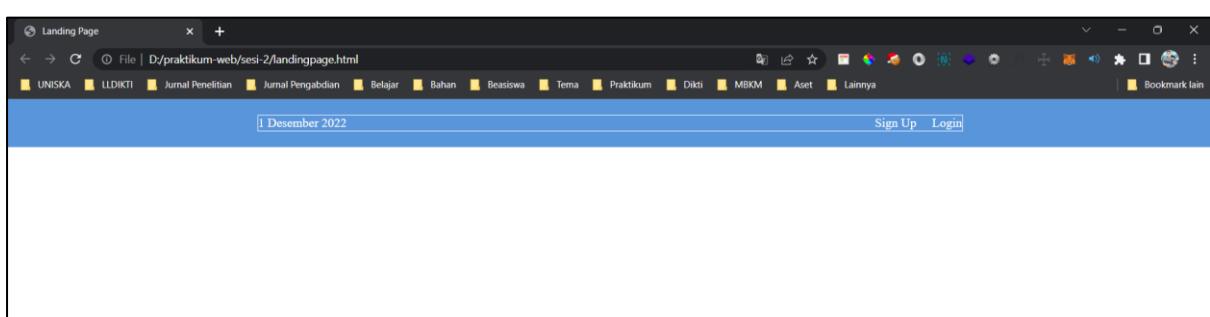
```

.links {
  display: flex;
  justify-content: flex-end;
}

.login {
  margin-left: 20px;
}

```

Reload halaman landingpage dan harusnya tampil seperti berikut:

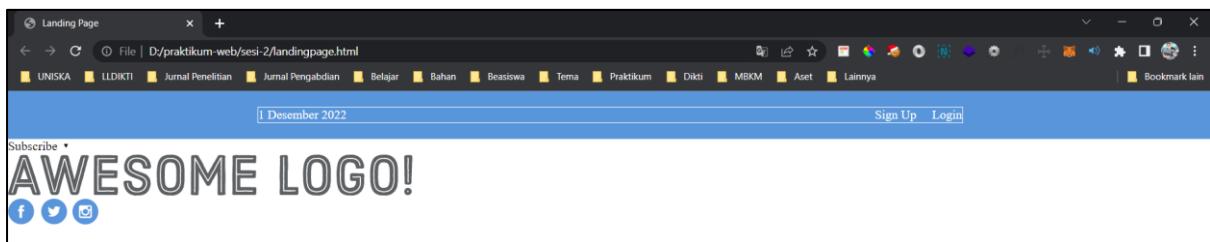


8. Class menu-container sudah selesai, kita lanjutkan dengan menambahkan header-container. Tambahkan kode berikut di bawah menu-container pada landingpage.html:

```
<body>
  <div class="menu-container">
    <div class="menu">
      <div class="date">1 Desember 2022</div>
      <div class="links">
        <div class="signup">Sign Up</div>
        <div class="login">Login</div>
      </div>
    </div>
  </div>

  <div class="header-container">
    <div class="header">
      <div class="subscribe">Subscribe &#9662;</div>
      <div class="logo"></div>
      <div class="social"></div>
    </div>
  </div>
</body>
```

Reload halaman landingpage dan harusnya tampil seperti berikut:

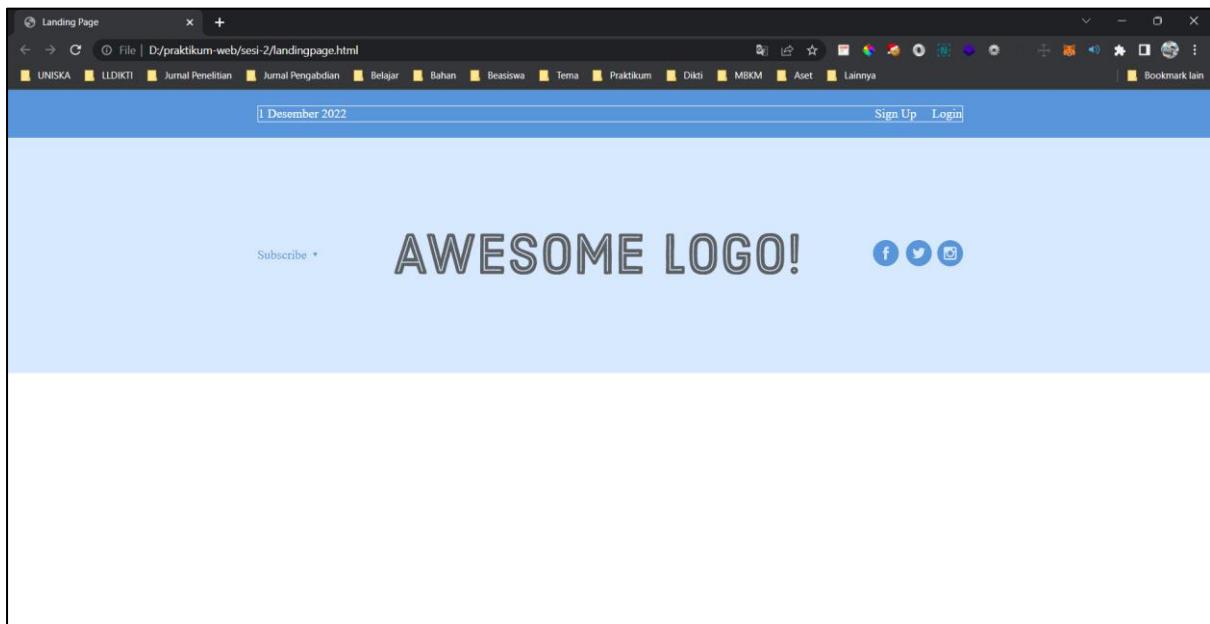


9. Tentu saja tampilannya masih berantakan, lanjut dengan menambahkan style untuk mengatur header yang kita buat. Tambahkan kode berikut ke styles.css:

```
.header-container {
  color: #5995DA;
  background-color: #D6E9FE;
  display: flex;
  justify-content: center;
}

.header {
  width: 900px;
  height: 300px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

Reload halaman landingpage dan akan terlihat seperti berikut:



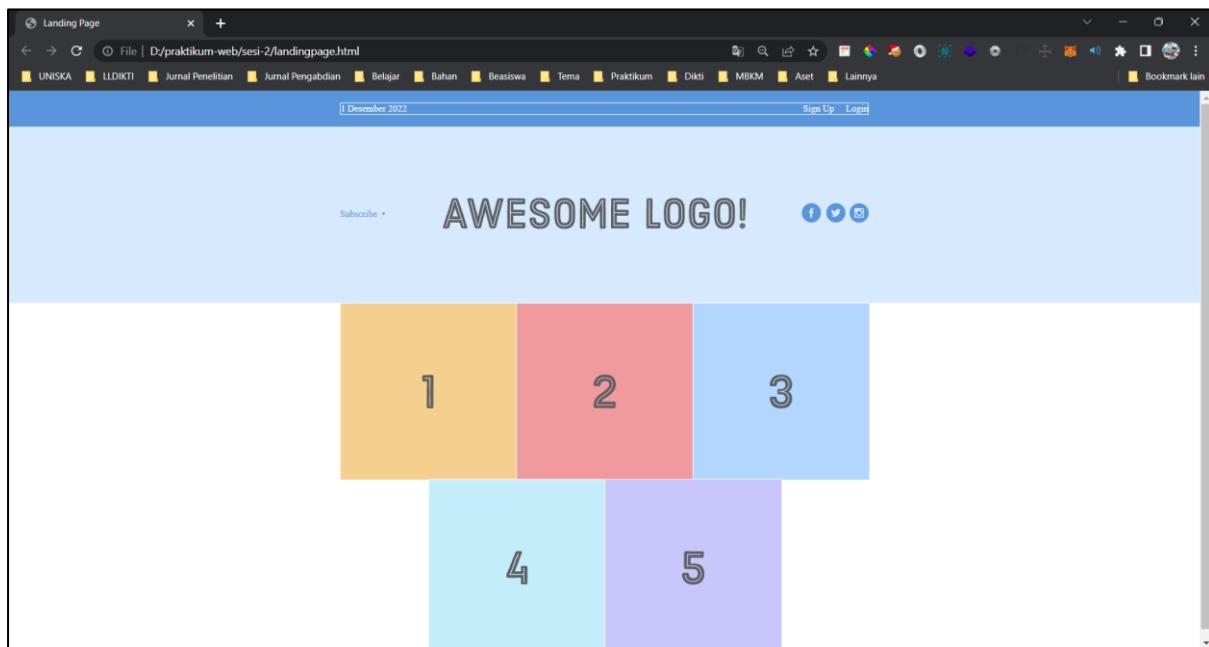
10. Sudah mulai terlihat bagian utama dari halaman web yang kita buat, mari kita lengkapi lagi dengan menambahkan beberapa baris foto. Tambahkan kode berikut di bawah header-container:

```
<div class="photo-grid-container">
  <div class="photo-grid">
    <div class="photo-grid-item">
      
    </div>
    <div class="photo-grid-item">
      
    </div>
    <div class="photo-grid-item">
      
    </div>
    <div class="photo-grid-item">
      
    </div>
    <div class="photo-grid-item">
      
    </div>
  </div>
</div>
```

11. Jika kita reload halaman landingpage maka foto-foto itu akan tersusun secara vertikal karena masih belum menggunakan flexbox. Kita tambahkan flexbox dan beberapa properti lainnya. Ketikkan kode berikut ke dalam styles.css:

```
.photo-grid-container {  
    display: flex;  
    justify-content: center;  
}  
  
.photo-grid {  
    width: 900px;  
    display: flex;  
    justify-content: center;  
    flex-wrap: wrap;  
    align-items: center;  
}  
  
.photo-grid-item {  
    border: 1px solid #fff;  
    width: 300px;  
    height: 300px;  
}
```

Reload halaman landingpage dan seperti ini lah seharusnya yang tampil:



12. Komponen selanjutnya adalah footer yang biasanya terletak di bagian bawah halaman. Footer sering kali berisi informasi tentang situs web, hak cipta, kontak, atau

tautan ke halaman lain yang mungkin bermanfaat bagi pengunjung. Tambahkan kode berikut ke bawah photo-grid-container:

```
<div class="photo-grid-item">
|   
</div>
<div class="photo-grid-item">
|   
</div>
</div>

<div class='footer'>
<div class='footer-item footer-one'></div>
<div class='footer-item footer-two'>Nama - NPM</div>
<div class='footer-item footer-three'></div>
</div>
```

Ubah Nama – NPM pada class footer-two dengan nama dan NPM masing-masing.

Lanjutkan dengan menambahkan kode berikut ke styles.css:

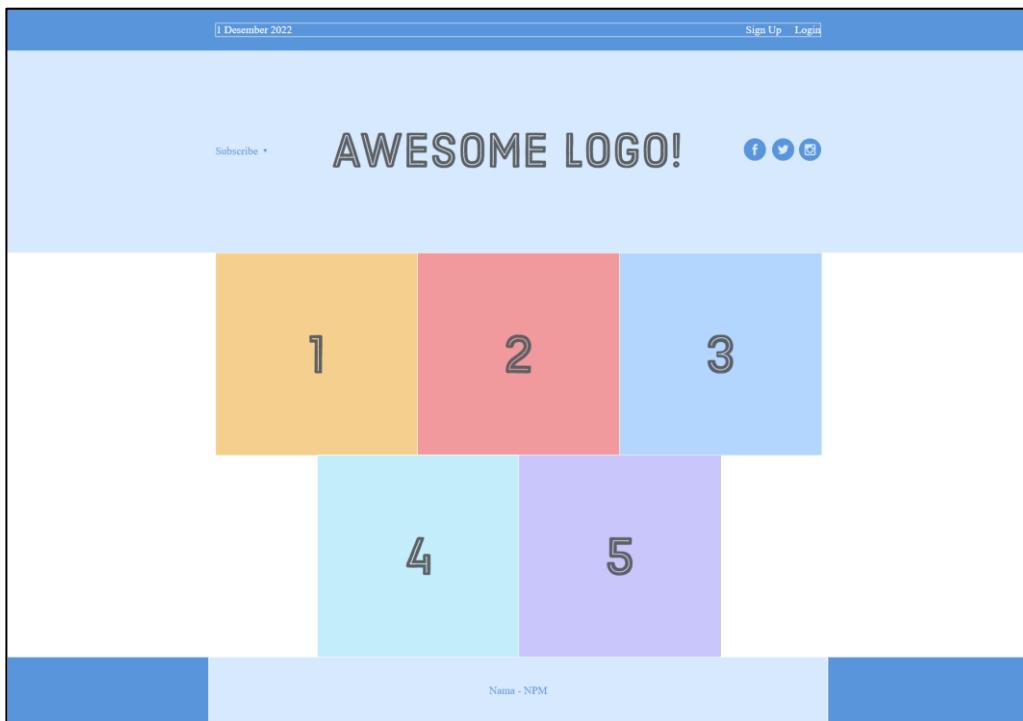
```
.footer {
    display: flex;
    justify-content: space-between;
}

.footer-item {
    border: 1px solid #ffff;
    background-color: #D6E9FE;
    height: 100px;
    flex: 1;
}

.footer-one,
.footer-three {
    background-color: #5995DA;
    flex: initial;
    width: 300px;
}

.footer-two {
    color: #5995DA;
    display: flex;
    justify-content: center;
    align-items: center;
}
```

Reload halaman landingpage dan seperti ini lah seharusnya yang tampil:



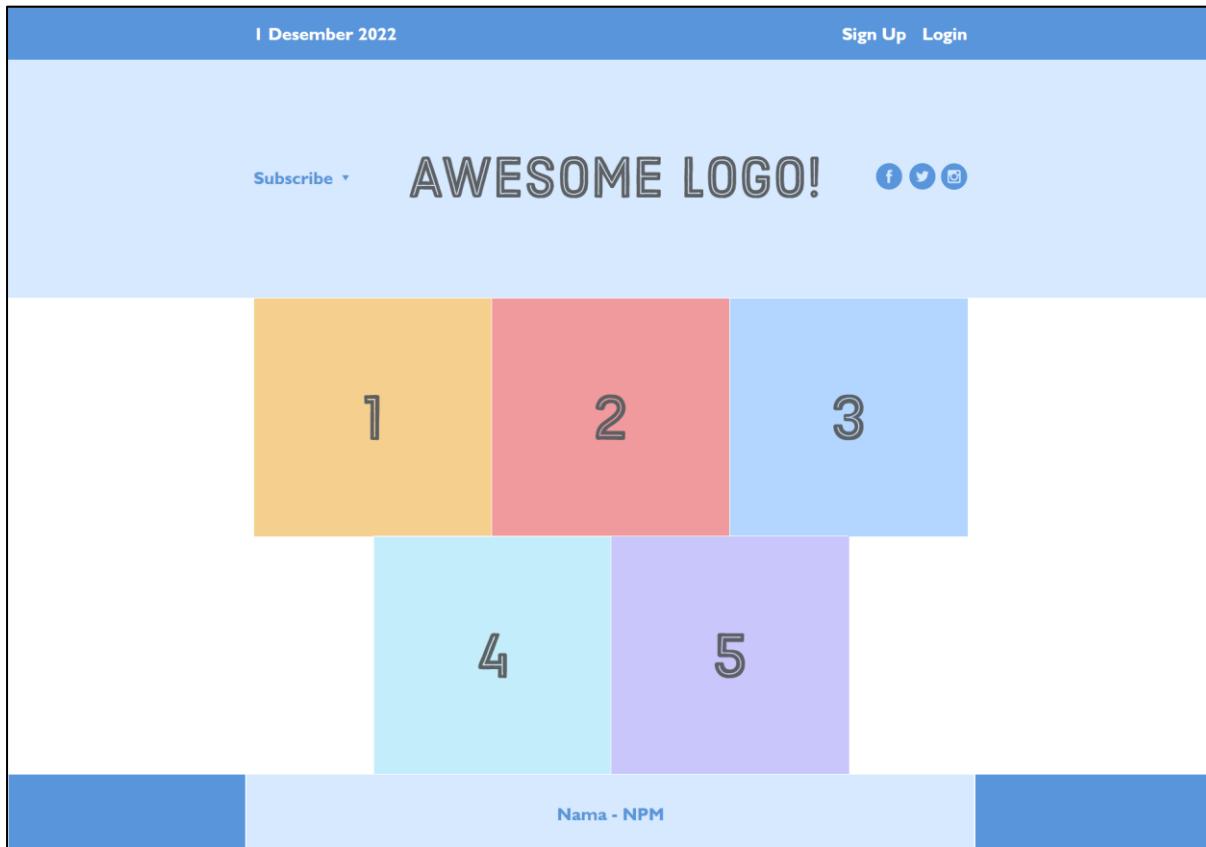
13. Web landingpage kita sudah tergambar sepenuhnya, tinggal sentuhan terakhir. Mari kita hilangkan garis bantu yang ada di tanggal dan tautan, serta kita sesuaikan beberapa properti huruf agar terbaca lebih jelas. Sesuaikan kode styles.css menjadi seperti ini:

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    font-size: 22px;
    font-weight: bold;
}

.menu-container {
    color: #fff;
    background-color: #5995DA;
    padding: 20px 0;
    display: flex;
    justify-content: center;
}

.menu {
    /* border: 1px solid #fff; */
    width: 900px;
    display: flex;
    justify-content: space-between;
}
```

Reload/refresh landingpage dan ini lah tampilan akhir dari halaman yang kita buat:



Selamat, kita sudah berhasil membuat tampilan landing page sederhana. Kita dapat menyesuaikan halaman yang telah dibuat dengan menghubungkan tautan, media sosial atau mengganti warna dan gambar. Masih banyak materi CSS yang perlu dicoba dan dipelajari, tidak perlu terburu-buru, mulai lah dari membuat tampilan-tampilan sederhana sampai terus berkembang dan memahami layouting dengan CSS.

Inginlah bahwa bagian tersulit sebenarnya bukanlah menuliskan kode HTML dan CSS, tetapi mendesain secara konseptual (di selembar kertas) dan menyusun tata letak kotak-kotak yang diperlukan sesuai desainnya.

KESIMPULAN 2

1. CSS adalah bahasa desain yang digunakan untuk mengatur tampilan (style) sebuah halaman web
2. Sintaks CSS terbagi menjadi selector, properti dan nilai
3. Terdapat banyak properti dan nilai yang dapat digunakan, kita dapat belajar dari dokumentasi resmi CSS di situs web W3C: <https://www.w3.org/Style/CSS/>
4. Ada tiga cara untuk menambahkan kode CSS ke halaman HTML antara lain inline, internal dan eksternal
5. Flexbox adalah modul layout CSS yang luar biasa untuk menata halaman web

LATIHAN 2.1

Di dalam folder sesi-2, buat file html dengan nama ‘profil.html’ dan file css dengan nama ‘style-profil.css’. Isikan file profil.html dengan data profil Anda sendiri kemudian hubungkan dengan file style-profil.css yang berisi style untuk menghias halaman profil tersebut.

SESI 3

Form dan Input

TUJUAN PRAKTEK

1. Mahasiswa mampu membuat form pada lama web
2. Mahasiswa mampu menangani berbagai macam tipe data dan mengekspresikannya dalam komponen input

PENDAHULUAN

Form HTML digunakan untuk mengumpulkan hasil input dari pengguna. Nilai yang diinput pengguna seringkali dikirim ke server untuk diproses, atau bisa juga diproses langsung pada peramban dengan menggunakan Javascript

PERCOBAAN 3.1 Form, Label, dan Input

1. Buat dokumen HTML, tambahkan tag form kedalam body

```
<index.html> <html> <body> <form>
1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Form dan Input</title>
8   </head>
9   <body>
10  <form action="">
11
12  </form>
13 </body>
14 </html>
```

2. Jika diperhatikan pada hasilnya, maka belum ada tampak perubahan pada laman, karena memang form tidak terlihat secara langsung, tapi hanya mengelompokkan inputan. Atribut action masih kosong, nantinya atribut ini bisa diisi dengan berkas PHP yang mengeksekusi hasil inputan
3. Tambahkan sebuah input

```
<form action="">
| <input type="text">
</form>
```

4. Pada laman sudah muncul komponen input berupa teks yang secara default memiliki lebar 20 karakter. Agar komponen input ini memiliki konteks maka bisa ditambahkan label

```
<form action="">
| <label for="">Nama :</label> <br>
| <input type="text">
</form>
```

5. Perhatikan bahwa label memiliki atribut for yang harusnya diisi dengan id dari komponen pasangannya. Maka tambahkan id untuk komponen input dan id tersebut dimasukkan pada atribut for milik label

```
<form action="">
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama">
</form>
```

6. Tambahkan sepasang input label, dan sepasang lagi dengan atribut type date

```
<form action="">
    <label for="nik">NIK :</label> <br>
    <input type="text" id="nik"><br>
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama"><br>
    <label for="tanggal_mulai">Tanggal Mulai :</label><br>
    <input type="date" id="tanggal_mulai"><br>
</form>
```

7. Perhatikan bahwa HTML memiliki komponen input selain teks, diantaranya date yang berfungsi untuk menginput nilai tanggal, format tanggal yang ditampilkan akan menyesuaikan pengaturan regional dari komputer yang digunakan.
8. Selain tipe tanggal, input juga bisa dibuat spesifik untuk inputan berupa angka dengan type number

```
<form action="">
    <label for="nik">NIK :</label> <br>
    <input type="text" id="nik"><br>
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama"><br>
    <label for="tanggal_mulai">Tanggal Mulai :</label><br>
    <input type="date" id="tanggal_mulai"><br>
    <label for="gaji_pokok">Gaji Pokok :</label><br>
    <input type="number" id="gaji_pokok"><br>
</form>
```

9. Agar isi nilai yang diinput nantinya bisa dikirim ke server, maka komponen input harus diberi atribut name, bisa diberikan name yang sama dengan id nya

```
<form action="">
    <label for="nik">NIK :</label> <br>
    <input type="text" id="nik" name="nik"><br>
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama" name="nama"><br>
    <label for="tanggal_mulai">Tanggal Mulai :</label><br>
    <input type="date" id="tanggal_mulai" name="tanggal_mulai"><br>
    <label for="gaji_pokok">Gaji Pokok :</label><br>
    <input type="number" id="gaji_pokok" name="gaji_pokok"><br>
</form>
```

10. Input dengan tipe radio, menggunakan name yang sama untuk tiap radio tapi dengan id dan value yang berbeda

```
<form action="">
    <label for="nik">NIK :</label> <br>
    <input type="text" id="nik" name="nik"><br>
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama" name="nama"><br>
    <label for="tanggal_mulai">Tanggal Mulai :</label><br>
    <input type="date" id="tanggal_mulai" name="tanggal_mulai"><br>
    <label for="gaji_pokok">Gaji Pokok :</label><br>
    <input type="number" id="gaji_pokok" name="gaji_pokok"><br>
    <label for="">Status Karyawan</label><br>
    <input type="radio" id="tetap" name="status_karyawan" value="TETAP">
    <label for="tetap">TETAP</label><br>
    <input type="radio" id="kontrak" name="status_karyawan" value="KONTRAK">
    <label for="kontrak">KONTRAK</label><br>
    <input type="radio" id="magang" name="status_karyawan" value="MAGANG">
    <label for="magang">MAGANG</label><br>
</form>
```

NIK :

Nama :

Tanggal Mulai :

Gaji Pokok :

Status Karyawan
○ TETAP
○ KONTRAK
○ MAGANG

11. Komponen yang digunakan untuk mengirimkan nilai inputan adalah komponen input type submit, yang biasanya disandingkan dengan input type reset

```
<form action="">
    <label for="nik">NIK :</label> <br>
    <input type="text" id="nik" name="nik"><br>
    <label for="nama">Nama :</label> <br>
    <input type="text" id="nama" name="nama"><br>
    <label for="tanggal_mulai">Tanggal Mulai :</label><br>
    <input type="date" id="tanggal_mulai" name="tanggal_mulai"><br>
    <label for="gaji_pokok">Gaji Pokok :</label><br>
    <input type="number" id="gaji_pokok" name="gaji_pokok"><br>
    <label for="">Status Karyawan</label><br>
    <input type="radio" id="tetap" name="status_karyawan" value="TETAP">
    <label for="tetap">TETAP</label><br>
    <input type="radio" id="kontrak" name="status_karyawan" value="KONTRAK">
    <label for="kontrak">KONTRAK</label><br>
    <input type="radio" id="magang" name="status_karyawan" value="MAGANG">
    <label for="magang">MAGANG</label><br>
    <input type="submit" id="btn_submit" name="btn_submit" value="Submit">
    <input type="reset" id="btn_reset" name="btn_reset" value="Reset">
</form>
```

NIK :

Nama :

Tanggal Mulai :

Gaji Pokok :

Status Karyawan
○ TETAP
○ KONTRAK
○ MAGANG

12. Komponen input memiliki berbagai macam atribut yang bisa digunakan, diantaranya yang sering digunakan adalah:

- a. value untuk memberikan nilai terhadap atribut

- b. readonly untuk memblokir user dari menginput
- c. disabled sama dengan readonly tapi nilainya juga tidak bisa digunakan
- d. size untuk menentukan secara spesifik lebar dari inputan
- e. maxlength untuk membatasi panjang inputan
- f. min dan max untuk membatas nilai minimal dan maksimal pada type number dan date
- g. placeholder untuk memberi contoh atau keterangan inputan
- h. required untuk memberikan validasi data kosong

13. Tambahkan atribut berikut dan perhatikan dampaknya

```
<label for="nik">NIK :</label> <br>
<input type="text" id="nik" name="nik" size="10" maxlength="10" required placeholder="misal: 123"><br>

<label for="nama">Nama :</label> <br>
<input type="text" id="nama" name="nama" required placeholder="misal: Fulan"><br>
```

14. Tambahkan atribut berikut untuk komponen type date

```
<label for="tanggal_mulai">Tanggal Mulai :</label><br>
<input type="date" id="tanggal_mulai" name="tanggal_mulai" required min="2022-01-11"><br>
```

Hasilnya adalah ketika belum diisi dan ditekan submit maka muncul pesan “Please fill out this field.” Yang muncul dari atribut required, sedangkan ketika dipilih tanggalnya tidak bisa dipilih sebelum 11 Januari 2022

15. Tambahkan atribut berikut untuk komponen type number

```
<label for="gaji_pokok">Gaji Pokok :</label><br>
<input type="number" id="gaji_pokok" name="gaji_pokok" required min="1000000" step="1000"><br>
```

Hasilnya sama seperti sebelumnya yang memiliki atribut required, sedangkan untuk angka minimal adalah 1.000.000 dan ketika diisi kurang dari angka tersebut akan muncul pemberitahuan bahwa inputan lebih kecil dari nilai minimal. Atribut step membantu proses inputan ketika diklik tanda panah maka nilai akan bertambah sesuai nilai yang dituliskan pada atribut step tersebut.

PERCOBAAN 3.2 DOM

Document Object Model (DOM) merupakan objek yang diciptakan oleh peramban ketika laman web berjalan. Menggunakan objek ini menyebabkan JavaScript mampu memanipulasi HTML, baik itu elemen, atribut, CSS, menghapus dan menambahkan elemen, dan bereaksi terhadap event.

1. Buat halaman seperti berikut

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Belajar DOM</title>
</head>
<body>
    <h1 id="uji_heading"></h1>
    <p id="uji_paragraf"></p>

<script>
    document.getElementById('uji_heading').innerHTML = "Judul";
    document.getElementById('uji_paragraf').innerHTML = "Paragraf";
</script>
</body>
</html>
```

Halaman ini menunjukkan bahwa penulisan nilai pada heading dan paragraf tidak langsung pada elemen HTML nya melainkan melalui JavaScript.

2. Tambahkan komponen input

```
<body>
    <h1 id="uji_heading"></h1>
    <p id="uji_paragraf"></p>
    <input type="text" id="uji_input">

    <script>
        document.getElementById('uji_heading').innerHTML = "Judul";
        document.getElementById('uji_paragraf').innerHTML = "Paragraf";
        document.getElementById('uji_input').value = "Input";
    </script>
</body>
```

Dapat dilihat hasilnya adalah komponen input text berisikan tulisan “Input”

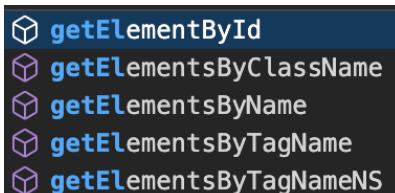
Dari source code yang sudah dituliskan dapat diamati bahwa **document** merupakan objek yang ada pada laman web. Perintah **getElementById** merupakan **method** yang digunakan untuk menemukan elemen HTML dengan id tertentu, sedangkan **innerHTML** dan **value** merupakan **property** dari elemen yang sebelumnya sudah ditemukan.

Jika ingin dipisah antara **method** dan **property**-nya maka source code tadi bisa diubah menjadi

```
ujiHeading = document.getElementById('uji_heading');
ujiParagraf = document.getElementById('uji_paragraf');
ujiInput = document.getElementById('uji_input');

ujiHeading.innerHTML = "Judul";
ujiParagraf.innerHTML = "Paragraf";
ujiInput.value = "Input";
```

Method lain yang bisa digunakan untuk memilih elemen antara lain



- ⌚ **getElementById**
- ⌚ **getElementsByClassName**
- ⌚ **getElementsByTagName**
- ⌚ **getElementsByTagName**
- ⌚ **getElementsByTagNameNS**

PERCOBAAN 3.3 DOM Event

Event bisa ditambahkan pada DOM yang menjadikannya jauh lebih interaktif. Beberapa event yang sering digunakan diantaranya onclick, onmouseup, onmousedown, onkeyup, onkeydown. Event ini akan dihandle dengan menggunakan function JavaScript.

1. Buat halaman dengan isi sebagai berikut

```
<body>
  <h1 id="uji_heading">
    | Sorot dong!
  </h1>

  <script>
  </script>
</body>
```

2. Pada elemen uji_heading tambahkan sebuah event yaitu onmouseover yang akan dihandle dengan function javascript sorot()

```
<body>
  <h1 id="uji_heading" onmouseover="sorot(this)">
    | Sorot dong!
  </h1>

  <script>
    function sorot(obj){
      | obj.innerHTML = "Sudah disorot";
    }
  </script>
</body>
```

Maka ketika mouse diarahakan ke uji_heading, tulisannya akan berubah menjadi ‘Sudah disorot’.

3. Tambahkan lagi event onmouseout dengan handle function kelarSorot()

```
<body>
  <h1 id="uji_heading"
    onmouseover="sorot(this)"
    onmouseout="keluarSorot(this)">
    | Sorot dong!
  </h1>

  <script>
    function sorot(obj){
      | obj.innerHTML = "Sudah disorot";
    }
    function keluarSorot(obj){
      | obj.innerHTML = "Sorot lagi dong!";
    }
  </script>
</body>
```

Dengan perintah ini maka tulisan pada uji_heading menjadi interaktif dan berganti setiap kali ada mouse yang menyorotnya.

4. Berikutnya tambahkan sebuah h2 dengan id uji_klik dan didalamnya ditambahkan event onmousedown yang dihandle function tahan() dan onmouseup yang dihandle oleh function lepas()

```

<body>
    <h1 id="uji_heading"
        onmouseover="sorot(this)"
        onmouseout="keluarSorot(this)">
        |   Sorot dong!
    </h1>

    <h2 id="uji_klik" onmousedown="tahan(this)" onmouseup="lepas(this)">
        |   Klik dong!
    </h2>

    <script>
        function sorot(obj){
            obj.innerHTML = "Sudah disorot";
        }
        function keluarSorot(obj){
            obj.innerHTML = "Sorot lagi dong!";
        }
        function tahan(obj){
            obj.innerHTML = "Lepasin..."
        }
        function lepas(obj) {
            |   obj.innerHTML = "Klik lagi dong!"
        }

    </script>
</body>

```

Element h2 ini awalnya bertuliskan ‘Klik dong!’ ketika diklik dan posisi mouse masih ditahan atau dalam istilah event ‘mousedown’ maka element ini akan berganti tulisan menjadi ‘Lepasin...’ dan ketika dilepas, atau secara event disebut ‘mouseup’ maka tulisan pada elemen tersebut akan berubah menjadi ‘Klik dong!’

5. Berikutnya tambahkan label, input, span, dan dua buah button, button Klik dan button Ulangi. Pada button Klik tambahkan atribut onclick dengan function klik(). Function tersebut diisi dengan perintah yang isinya mengambil nilai yang ada pada komponen input dan menuliskannya kedalam komponen span.

```

<body>
    <h1 id="uji_heading"
        onmouseover="sorot(this)"
        onmouseout="keluarSorot(this)">
        |   Sorot dong!
    </h1>

    <h2 id="uji_klik" onmousedown="tahan(this)" onmouseup="lepas(this)">
        |   Klik dong!
    </h2>

    <label for="uji_input">Uji Coba Input</label><br>
    <input type="text" id="uji_input"><br>
    <span id="uji_span"></span><br>
    <button onclick="klik()">Klik</button>
    <button>Ulangi</button>

    <script>
        function sorot(obj){
            |   obj.innerHTML = "Sudah disorot";
        }
        function keluarSorot(obj){
            |   obj.innerHTML = "Sorot lagi dong!";
        }
        function tahan(obj){
            |   obj.innerHTML = "Lepasin..."
        }
        function lepas(obj) {
            |   obj.innerHTML = "Klik lagi dong!"
        }
        function klik(){
            |   valueInput = document.getElementById('uji_input').value;
            |   document.getElementById('uji_span').innerHTML = valueInput;
        }

    </script>
</body>

```

6. Tambahkan juga untuk button Ulangi dengan function ulangi() yang isinya mengosongkan nilai dan mengembalikan komponen seperti semula

```
<button onclick="ulangi()">Ulangi</button>
```

```

function ulangi() {
    |   document.getElementById('uji_input').value = "";
    |   document.getElementById('uji_span').innerHTML = "";
    |   document.getElementById('uji_heading').innerHTML = "Sorot dong!";
    |   document.getElementById('uji_klik').innerHTML = "Klik dong!";
}

```

7. Modifikasi function klik() agar jika nilai yang diinput masih kosong maka akan menuliskan tulisan ‘Inputan kosong’ pada span dengan teks berwarna merah, sedangkan jika tidak kosong maka tuliskan dengan teks berwarna biru

```
function klik() {
    valueInput = document.getElementById('uji_input').value;
    span = document.getElementById('uji_span');

    if (valueInput == "") {
        span.innerHTML = "Inputan kosong";
        span.style = "color:red";
    } else {
        span.innerHTML = "Input : '"+valueInput+"'";
        span.style = "color:blue";
    }
}
```

8. Berikutnya ujicoba event onkeyup dengan function tiru() yang isinya adalah ketika diketikkan sesuatu pada komponen input, maka span akan memiliki tulisan yang sama.

```
<input type="text" id="uji_input" onkeyup="tiru()"><br>
```

```
function tiru(){
    valueInput = document.getElementById('uji_input').value;
    document.getElementById('uji_span').innerHTML = valueInput;
}
```