

MODUL PEMROGRAMAN WEB

HTML, CSS, & JAVASCRIPT



**DIVISI PENDIDIKAN
DAN PELATIHAN**

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Puji syukur kehadirat Allah SWT karena atas izinnya kami berkesempatan untuk membuat Modul Pemrograman Web ini. Shalawat serta salam kepada Nabi Muhammad SAW yang telah membawa kita dari zaman jahiliah menuju zaman terang-benderang yang penuh dengan ilmu pengetahuan.

Tak lupa penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu dengan tenaga dan pikirannya, terima kasih juga kepada teman-teman angkatan 20, rekan-rekan instruktur dan anggota Divdik, kakak tingkat angkatan 18 dan 19, Bapak Abdul Hafiz Tanjung S.E, M.Si, Ak., C.A, selaku pembina dan semuanya yang tidak bisa disebutkan satu persatu, yang selalu mendukung penulis sehingga modul ini sehingga dapat selesai sesuai yang kita inginkan semua

Modul ini disusun untuk memenuhi kebutuhan peserta pelatihan Web Angkatan 21 dan angkatan seterusnya. Sesuai dengan kebutuhan peserta, maka modul ini disusun dari beberapa sumber terpercaya seperti W3Schools, Petani Kode, Web Programming UNPAS, dan berbagai sumber lainnya. Modul ini terdiri 20 bab disertai dengan contoh source code serta berbagai output dalam bentuk gambar sehingga memudahkan para pembaca dalam membuat dan melihat hasil yang sesuai dengan arahan penulis.

Penulis menyadari masih banyak kekurangan dalam penyusunan modul ini. Untuk itu saran dan kritik yang membangun sangat penulis harapkan guna perbaikan dan pengembangan modul ini ke depan.

Akhir kata penulis berharap semoga Modul Pemrograman Web ini dapat dipergunakan sebaik-baiknya dan dapat dijadikan referensi untuk mahasiswa PUB yang ingin mempelajari bidang pemrograman web.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Bandung, 24 Juni 2023

Penulis

HAK CIPTA

Modul ini diterbitkan oleh Divisi Pendidikan dan Pelatihan Program Beasiswa Pemberdayaan Umat Berkelanjutan Periode 2022 – 2023 pada bulan Juni 2023 sehubungan dengan diadakannya pelatihan WEB untuk mahasiswa PUB Angkatan 21.

TIM PENYUSUN

A. Penanggung Jawab

- Muhammad Fadli Fathurrahman *(Koordinator Divdik)*

B. Penulis

- Dimas Firmansyah *(Sekretaris Divdik)*

TIM REVISI

- Nailly Rina Pribawa *(Wakil Koordinator Divdik)*
- Achmad Farhan Fauzan *(Sekretaris Divdik)*
- Aliya Rohaya Siregar *(Bendahara Divdik)*
- M. Azriel Pajarudin *(Humas Divdik)*
- Sawaludin *(Humas Divdik)*
- Tigana Reymansyah *(Humas Divdik)*
- Anggi Permana *(Instruktur Pelatihan Web)*
- Wahyu *(Instruktur Pelatihan Web)*

Dilarang menyalin atau memperbanyak sebagian atau seluruh isi buku tanpa izin tertulis dari pemegang hak cipta. Setiap pelanggaran hak cipta akan dituntut dan berhadapan dengan Undang - Undang yang berlaku.

Bandung, 24 Juni 2023

Penulis

SILABUS

Pertemuan ke	Materi	Durasi
1	BAB I : Pengenalan WEB BAB II : Dasar – Dasar HTML	120 Menit
2	BAB III : Menulis HTML BAB IV : Membuat Artikel Sederhana	120 Menit
3	BAB V : Tautan BAB VI : Gambar	120 Menit
4	BAB VII : Form BAB VIII : Tabel	120 Menit
5	BAB IX : Mengenal CSS BAB X : Properti Yang Sering Digunakan	120 Menit
6	BAB XI : Margin & Padding BAB XII : Dekorasi Teks & Dekorasi Tabel	120 Menit
7	BAB XIII : Posisi	120 Menit
8	Ujian Tengah Semester	120 Menit
9	BAB XIV : Display	120 Menit
10	BAB XV : Mengenal JavaScript BAB XVI : Variabel dan Fungsi	120 Menit
11	BAB XVII : Array	120 Menit
12	BAB XVIII : Perulangan	120 Menit
13	BAB XIX: DOM & Event Handler	120 Menit
14	BAB XX : Object di JavaScript	120 Menit
15	BAB XXI : JSON	120 Menit
16	Ujian Akhir Semester	10 Hari

BOBOT PENILAIAN

Tugas	10%
Kuis	20%
Ujian Tengah Semester	30%
Ujian Akhir Semester	40%

DAFTAR ISI

KATA PENGANTAR.....	i
HAK CIPTA.....	ii
SILABUS	iii
BOBOT PENILAIAN	iv
DAFTAR ISI.....	v
BAB I	1
PENGENALAN PEMROGRAMAN WEB	1
A. WEBSITE	1
B. JENIS – JENIS WEBSITE.....	2
C. CARA KERJA WEBSITE.....	2
D. WEB SERVER.....	2
BAB II	4
DASAR HTML	4
A. PENGERTIAN HTML	4
B. SEJARAH HTML	4
C. CONTOH HTML.....	4
D. BROWSER WEB.....	5
E. STRUKTUR HTML.....	6
BAB III	7
MENULIS HTML.....	7
A. EDITOR HTML.....	7
B. LANGKAH – LANGKAH MEMBUAT PROGRAM.....	7
C. TAG, ELEMEN, DAN ATRIBUT DALAM HTML	9
BAB IV	13
MEMBUAT ARTIKEL SEDERHANA	13

A. PARAGRAF	13
B. HEADING	15
C. TEXT FORMATING	17
D. LIST	18
E. KOMENTAR	24
F. ELEMEN SEMANTIK	24
BAB V	28
TAUTAN HTML.....	28
A. APA ITU LINK	28
B. JENIS – JENIS LINK.....	30
BAB VI.....	38
GAMBAR DI HTML	38
A. MENAMBAHKAN GAMBAR DI HTML.....	38
B. FORMAT FILE GAMBAR UNTUK HTML.....	41
C. ATRIBUT UNTUK TAG	41
D. TAG TAMBAHAN UNTUK GAMBAR.....	43
BAB VII	47
FORM HTML.....	47
A. CARA MEMBUAT FORM	47
B. FIELD	47
C. ELEMEN FORM	48
BAB VIII	52
TABEL HTML	52
A. MEMBUAT TABEL.....	52
B. MENGATUR JARAK DENGAN CELLPADDING	53
C. MENGGABUNGKAN SEL TABEL.....	54
D. MENYISIPKAN ELEMEN LAIN KE DALAM SEL	55

BAB IX	58
MENGENAL CSS	58
A. APA ITU CSS	58
B. SEJARAH CSS	58
C. CARA MENULIS CSS.....	59
D. SINTAKS DASAR CSS	61
E. MEMAHAMI 6 JENIS SELEKTOR	64
BAB X	70
PROPERTI YANG SERING DIPAKAI	70
A. COLOR	70
B. BACKGROUND.....	75
C. BORDER.....	79
BAB XI	81
MARGIN & PADDING	81
A. MARGIN.....	81
B. PADDING	83
BAB XII.....	85
DEKORASI TEKS	85
A. MENGATUR PERATAAN TEKS (TEXT ALIGN).....	85
B. TEXT DIRECTION.....	86
C. TEXT DECORATION.....	87
D. TEXT TRANSFORM.....	89
E. TEXT SPACING	89
F. TEXT SHADOW	92
G. MENGATUR UKURAN TEXT (FONT SIZE)	93
H. MENGATUR KETEBALAN (FONT WEIGHT)	93
I. TEKS MIRING (FONT STYLE).....	95

J. MENGATUR JENIS FONT (FONT FAMILY)	95
K. MENGGUNAKAN FONT DARI GOOGLE	97
BAB XIII.....	100
DEKORASI TABEL	100
A. BATAS TABEL.....	100
B. PADDING TABEL	102
C. PENYELARASAN TABEL (ALIGNMENT)	102
D. STRIPED TABEL	104

BAB I

PENGENALAN PEMROGRAMAN WEB



A. WEBSITE

Website adalah kumpulan halaman web yang menyajikan informasi tertentu berdasarkan topik atau jenisnya, dan bisa diakses oleh siapa saja, kapan saja di internet melalui aplikasi web browser.

Dalam bahasa Indonesia, pengertian website adalah situs web, yang berarti sebuah situs atau ‘lokasi’ di web. Website terdiri dari beberapa halaman web yang saling terkait dibawah sebuah nama domain, biasanya memuat konten seperti teks, video, gambar, audio, dan lain sebagainya.

Lalu, apa saja sih halaman yang ada pada website? Jadi, umumnya website memiliki satu halaman utama yang disebut *homepage*. *Homepage* adalah halaman beranda yang akan terbuka pertama kali saat Anda mengakses nama domain website.

Dari sini, Anda bisa menjelajahi halaman web lain, biasanya halaman produk atau jasa untuk e-commerce, kontak, *support*, dan lain sebagainya.

Setiap website dibuat untuk tujuan tertentu, misalnya sebagai wajah sebuah brand atau bisnis, media berita, hiburan, toko online, sarana pendidikan, hingga media sosial.

Untuk mengunjungi website, Anda membutuhkan sebuah perangkat seperti *smartphone* atau komputer, link URL (*Uniform Resource Locator*), alamat website atau nama domainnya, koneksi internet, dan aplikasi web browser seperti Google, Chrome, Mozilla Firefox, Bing, Microsoft Edge, dll.

B. JENIS - JENIS WEBSITE

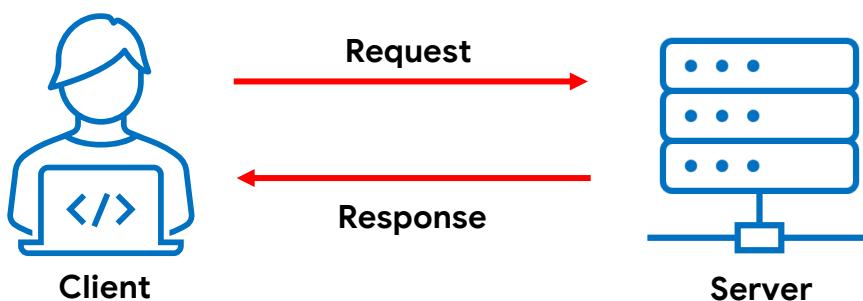
Secara garis besar, website dikelompokkan menjadi dua jenis yaitu website dinamis dan statis. Website dinamis memiliki konten yang berubah-ubah sesuai preferensi pengunjungnya, sementara website statis selalu menampilkan konten yang sama.

Dari sini, jenis website dikelompokkan lagi menurut tujuan dan kegunaannya. Yang termasuk dalam jenis-jenis website adalah :

1. Company Profile
2. Toko Online
3. Web Portal
4. Search Engine
5. Sosial Media
6. E-commerce
7. Dan lain-lain

C. CARA KERJA WEBSITE

Ada 2 bagian utama dalam menjalankan website yaitu client dan server. Client bertujuan untuk mengirimkan perintah (request) dari apa yang diinginkan oleh user dengan cara mengakses web browser seperti Google Chrome, Mozilla Firefox, dll. Sedangkan server digunakan untuk menyimpan data – data dari hasil request si client kemudian data tersebut diolah sedemikian rupa agar bisa bermanfaat untuk client (response).



D. WEB SERVER

Web server adalah sebuah software (perangkat lunak) yang memberikan layanan berupa data. Berfungsi untuk menerima permintaan HTTP atau HTTPS dari klien atau kita kenal dengan web browser (Chrome, Firefox). Selanjutnya ia akan mengirimkan response

atas permintaan tersebut kepada client dalam bentuk halaman web. Terdapat dua jenis web server, yaitu server online (web hosting) dan server offline (localhost).

Perbedaan web hosting dan localhost

LOCALHOST	WEB HOSTING
Localhost memiliki server yang disebut sebagai server offline	Web hosting atau bisa disebut server online merupakan server yang digunakan dalam menyimpan data-data dari sebuah website yang telah dikembangkan pada server localhost
Setiap data yang dimasukkan ke dalam server ini tidak bisa untuk diakses dari luar.	Website dapat diakses user mana pun melalui internet

BAB II

DASAR HTML

A. PENGERTIAN HTML

HTML merupakan singkatan dari Hypertext Markup Language adalah sebuah bahasa markup standar untuk membuat halaman website. Bahasa markup sendiri adalah bahasa komputer yang digunakan untuk menyusun, memformat, atau mendefinisikan hubungan antara berbagai bagian dokumen teks dengan bantuan simbol atau tag yang disisipkan dalam dokumen.

Elemen HTML digambarkan oleh tag, ditulis menggunakan kurung sudut (<>). Tag seperti `` dan `<input/>` langsung diubah menjadi sebuah konten ke dalam halaman website.

B. SEJARAH HTML

HTML pertama kali dibuat pada tahun 1990 oleh Tim Berners-Lee, seorang ilmuwan komputer di CERN (Organisasi Eropa untuk Riset Nuklir). Pada awalnya, HTML hanya terdiri dari beberapa elemen sederhana seperti teks, gambar, dan hyperlink. Namun, seiring waktu, HTML berkembang menjadi sebuah bahasa pemrograman yang lebih kompleks dengan kemampuan yang lebih besar.



Setiap versi HTML memiliki perbedaan dan peningkatan dari versi sebelumnya. Versi HTML yang paling awal adalah HTML 1.0, yang kemudian diikuti oleh HTML 2.0, HTML 3.2, HTML 4.01, dan HTML 5.0 yang terakhir.

HTML 5.0 dirilis pada tahun 2014 dan merupakan versi HTML terbaru yang lebih canggih dan fleksibel dibandingkan dengan versi sebelumnya. HTML 5.0 mendukung berbagai jenis media, seperti audio dan video, serta memungkinkan pengembangan aplikasi web yang lebih kompleks.

C. CONTOH HTML

Berikut ini adalah contoh source code penulisan HTML sederhana :

```
<!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>Modul Web</title>
</head>
<body>
    <h1>Contoh HTML Sederhana</h1>
    <p>Ini adalah paragraf pertama saya</p>
</body>
</html>
```

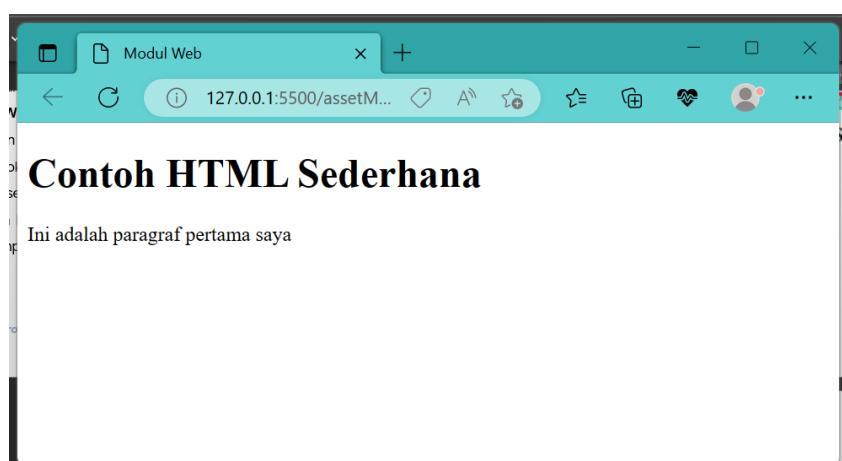
Keterangan :

- <!DOCTYPE html> , mendefinisikan jenis dokumen menjadi HTML
- <meta> , menyediakan metadata tentang dokumen HTML
- Teks antara <html> dan </html> menggambarkan sebuah dokumen HTML
- Teks antara <head> dan </head> memberikan informasi tentang dokumen
- Teks antara <title> dan </title> menyediakan judul untuk dokumen
- Teks antara <body> dan </body> menggambarkan isi dari halaman yang terlihat
- Teks antara <h1> dan </h1> menggambarkan sebuah heading
- Teks antara <p> dan </p> menggambarkan sebuah paragraf

D. BROWSER WEB

Tujuan dari web browser (Chrome, Firefox, Safari, Ms. Edge, dll) adalah untuk membaca dokumen HTML dan menampilkannya.

Browser tidak akan menampilkan tag HTML, tetapi web menggunakannya untuk menentukan bagaimana menampilkan dokumen. Dari contoh source code diatas maka akan menampilkan output sebagai berikut.

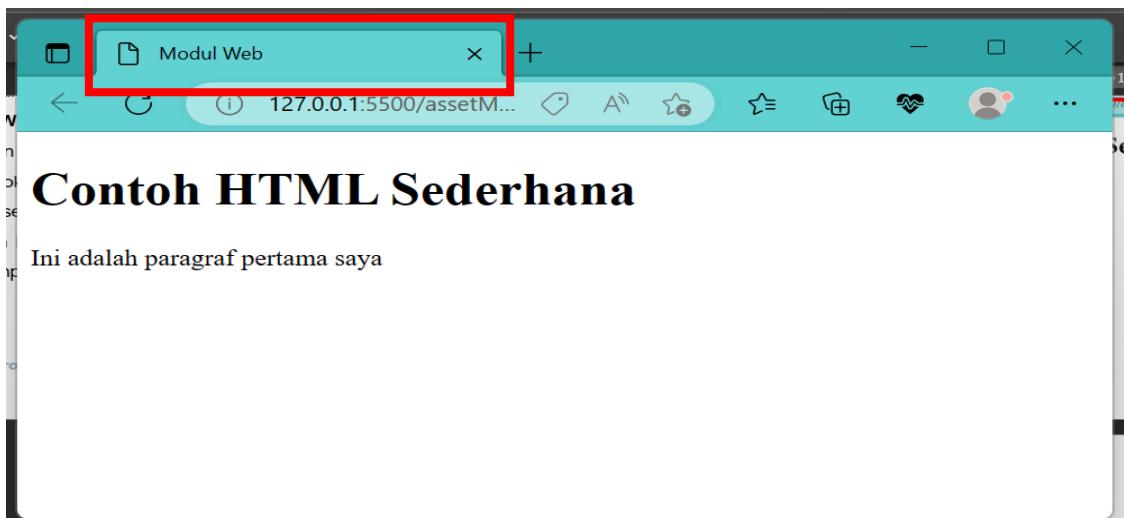


E. STRUKTUR HTML

HTML terdiri dari 2 bagian yaitu :

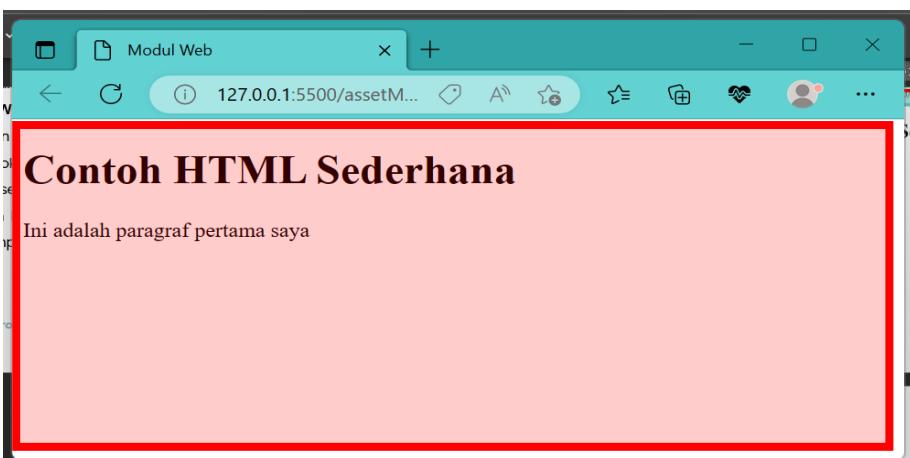
1) Tag `<head>`

Tag head merupakan tag yang berisi informasi tentang halaman yang **tidak ditampilkan** di halaman web browser. Misalnya: source css, js atau lainnya yang berasal dari luar, informasi meta, title, dan lainnya. Namun, khusus untuk tag `<title>` akan ditampilkan di title bar pada web browser. Tag `<title>` adalah tag yang berada di dalam head HTML yang berfungsi untuk menampilkan judul halaman web pada title bar web browser.



2) Tag `<body>`

Tag body merupakan tag yang berisi elemen-elemen yang **ditampilkan** di halaman web. Misalnya teks yang berupa paragraph `<p>`, heading `<h1>` hingga `<h6>`, menampilkan gambar ``, membuat tabel `<table>`, dan lainnya. Di dalam tag body inilah konten visual dari halaman web ditempatkan.



BAB III

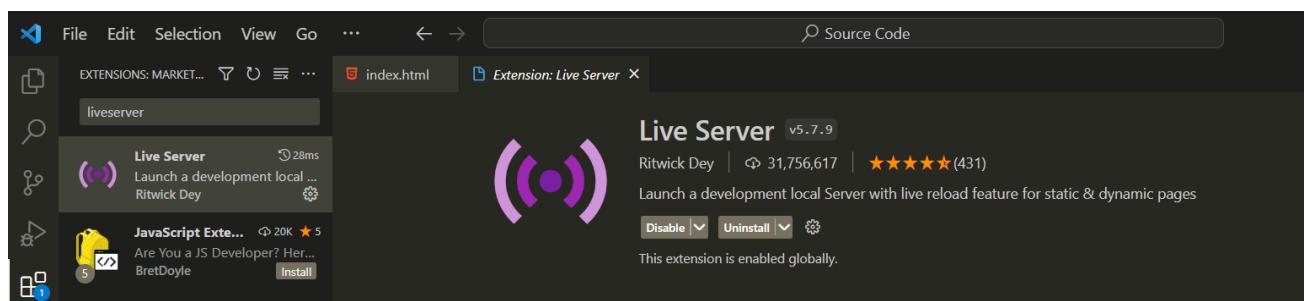
MENULIS HTML

A. EDITOR HTML

HTML dapat di edit menggunakan editor HTML profesional seperti :

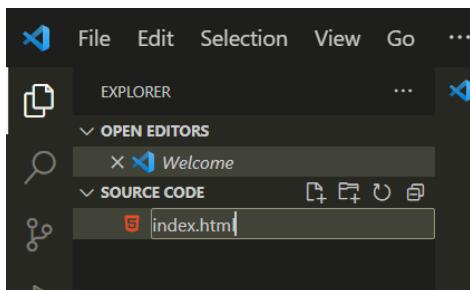
- Microsoft WebMatrix
- Sublime Text
- NotePad (PC)
- TextEdit (Mac)

Namun, pada pelatihan kali ini kita akan menggunakan IDE Visual Studio Code. Untuk saat ini belum ada update terbaru dari PUBCode, namun pihak Divdik akan terus mengembangkan IDE tersebut untuk pelatihan kedepannya. Jika sudah menginstall VSCode di device kalian maka install extensi *Live Server* di VSCode kalian.

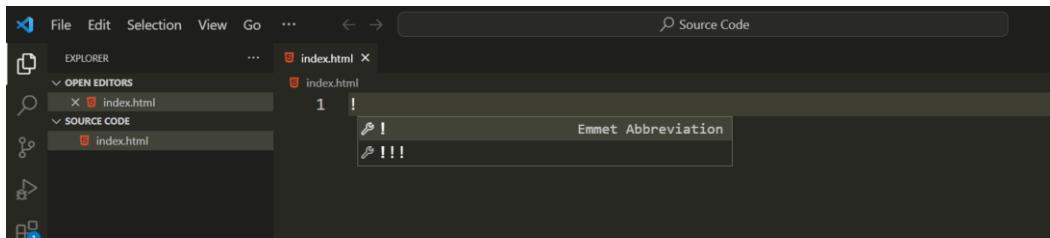


B. LANGKAH – LANGKAH MEMBUAT PROGRAM

1. Buka VSCode dan buat file dengan format .html atau bisa mengikuti contoh berikut, anggaplah kita membuat file dengan indeks.html (sebenarnya namanya bebas).



2. Kemudian ketik tanda seru (!) pada file yang dibuka, maka akan menampilkan sugestion seperti pada gambar dibawah ini, kemudian tekan enter.



3. Jika kita menekan tombol enter maka akan menampilkan struktur HTML 5 yang telah dibuat otomatis oleh komputer sehingga kita tidak perlu menuliskannya secara manual

```
... < - > Source Code index.html x index.html > html > body 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 </head> 9 <body> 10 11 </body> 12 </html>
```

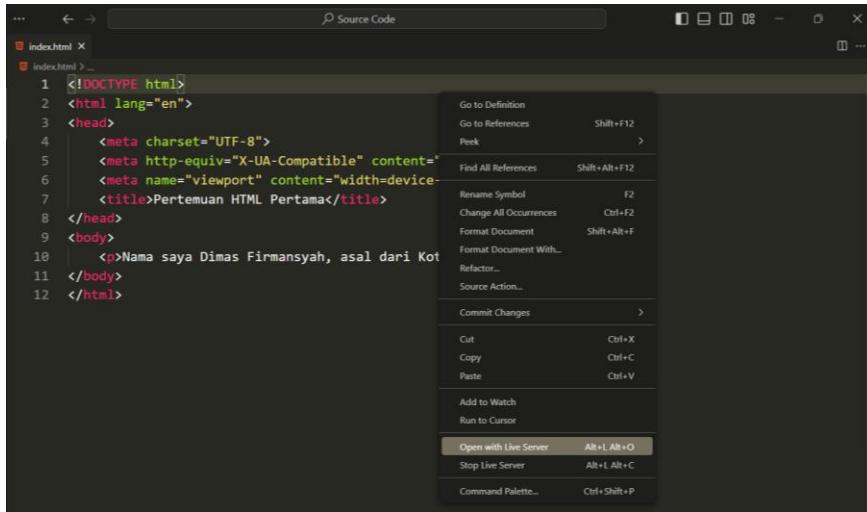
A screenshot of the Visual Studio Code interface. The title bar says 'Source Code'. The code editor shows the full HTML5 structure generated by Emmet. The code includes a DOCTYPE declaration, HTML and head tags, meta tags for charset, X-UA-Compatible, and viewport, and a title tag with the text 'Document'. The body and html tags are also present.

4. Selanjutnya kita ubah judulnya yang ada didalam tag <title> menjadi "Pertemuan HTML Pertama", dan isikan paragraf di dalam tag <body>.

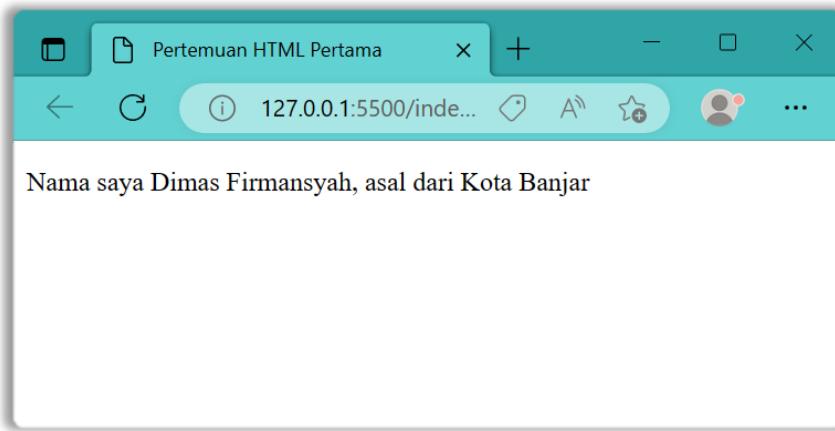
```
index.html x index.html > html > body > p 1 <!DOCTYPE html> 2 <html lang="en"> 3 <head> 4 <meta charset="UTF-8"> 5 <meta http-equiv="X-UA-Compatible" content="IE=edge"> 6 <meta name="viewport" content="width=device-width, initial-scale=1.0"> 7 <title>Pertemuan HTML Pertama</title> 8 </head> 9 <body> 10 <p>Nama saya Dimas Firmansyah, asal dari Kota Banjar</p> 11 </body> 12 </html>
```

A screenshot of the Visual Studio Code interface. The code editor shows the modified HTML. The title tag now contains the text 'Pertemuan HTML Pertama'. A new paragraph tag with the text 'Nama saya Dimas Firmansyah, asal dari Kota Banjar' has been added to the body.

5. Simpan kode tersebut dengan menekan tombol ctrl + s kemudian buka di browser kalian. Caranya, klik kanan, pilih Open with Live Server



6. Tunggu beberapa saat kemudian kode yang kita buat tadi akan muncul di browser kalian



C. TAG, ELEMEN, DAN ATRIBUT DALAM HTML

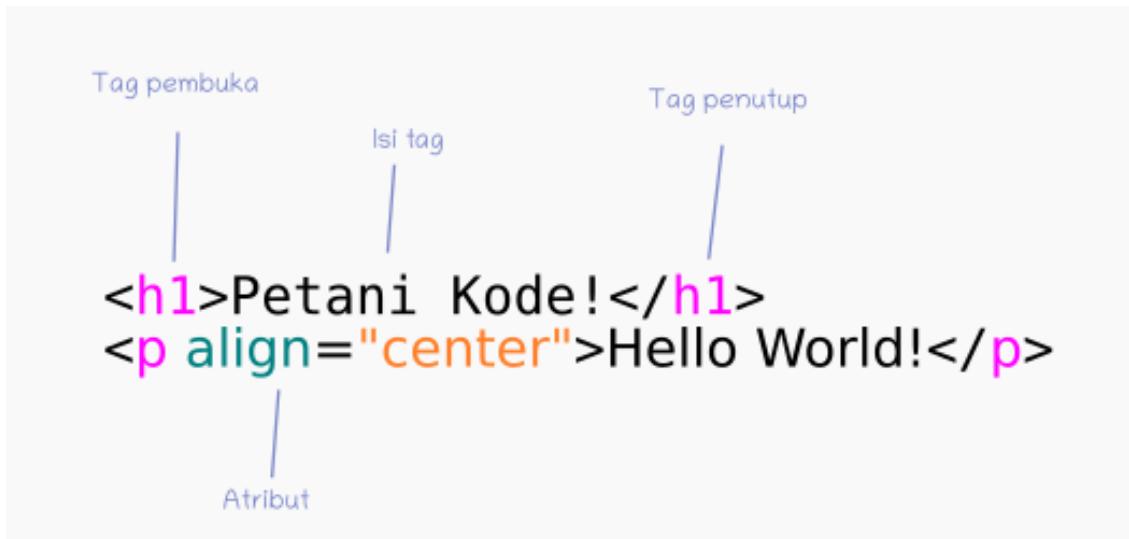
1. Apa itu Tag?

Tag adalah sebuah penanda awalan dan akhiran dari sebuah elemen di HTML.

Tag dibuat dengan kurung sudut (<.....>), lalu di dalamnya berisi nama tag dan kadang juga ditambahkan dengan atribut.

Contoh: <p>, <a>, <body>, <head>, dan sebagainya.

Tag selalu ditulis berpasangan. Ada tag pembuka dan ada tag penutupnya. Namun, ada juga beberapa tag yang tidak memiliki pasangan penutup. Tag penutup ditulis dengan menambahkan garis miring (/) di depan nama tag.

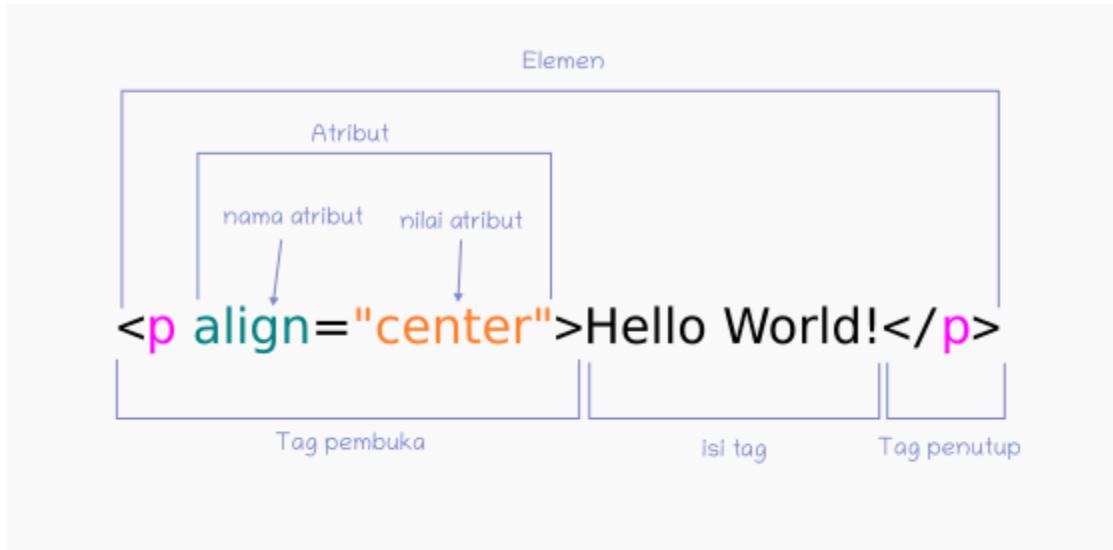


Sekarang HTML sudah memiliki 250 tag, namun kalian cukup mengetahui tag dasarnya saja. Berikut ini daftar tag – tag dasar, yang harus kalian hafal.

Tag	Fungsi
<html>	Untuk memulai dokumen HTML
<head>	untuk membuat bagian head (kode pelengkap yang tidak akan ditampilkan di browser)
<body>	Untuk membuat bagian body (kode yang akan ditampilkan di browser)
<h1> sampai <h6>	Untuk membuat heading atau judul sebuah artikel
<p>	Untuk membuat paragraf
<!-- -->	Untuk membuat komentar

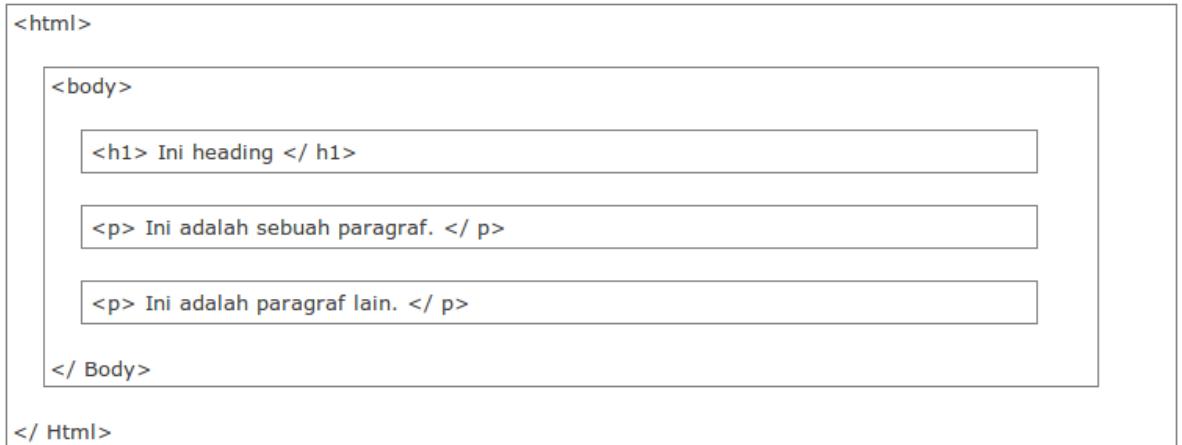
2. Apa itu Elemen?

Elemen dalam HTML adalah sebuah komponen yang menyusun dokumen HTML. Elemen dibuat dari tag pembuka, isi tag, dan tag penutup. Kadang juga ditambahkan beberapa atribut.



Elemen tidak selalu berisi teks, kadang ia juga akan berisi elemen lain. Ini biasanya kita sebut dengan *nested element* atau elemen di dalam elemen.

Bila digambarkan dalam bentuk kotak persegi, maka akan terlihat seperti ini :



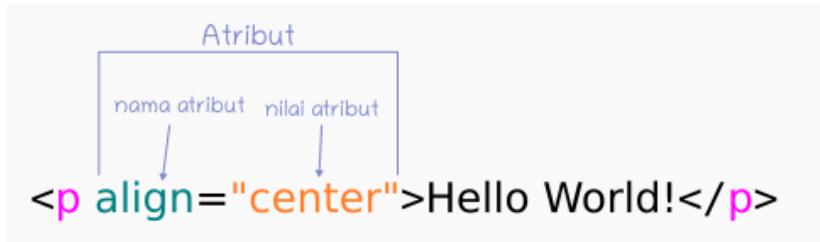
Elemen HTML ada banyak jenisnya. Ada elemen khusus untuk teks, ada elemen untuk multimedia, script, tabel, metadata, dll.

Elemen HTML terbagi menjadi dua yaitu elemen **block** dan elemen **inline**.

- Elemen Block akan membuat baris baru dan memisahkan diri dari tag sekitarnya. Contohnya adalah `<div>`, `<main>`, `<p>`, `<h1>` - `<h6>`, `<figure>`, dll.
- Elemen Inline tidak membuat baris baru. Contohnya adalah `<a>`, ``, ``, `<button>`, `<input>`, dll

3. Apa itu Atribut?

Atribut adalah kata khusus yang berada di dalam tag pembuka. Atribut juga disebut sebagai modifier yang akan menentukan perilaku dari elemen.



Contoh :

```
<a href="https://www.youtube.com/">Contoh Atribut pada link</a>
```

Tag **<a>** adalah tag untuk membuat link. Tag ini wajibkan menambahkan atribut href untuk menyatakan halaman tujuan dari link. Jumlah atribut pada elemen bisa lebih dari satu.

Contoh :

```

```

Tiap-tiap elemen kadang memiliki atribut khusus yang hanya bisa digunakan pada elemen tersebut. Ada juga atribut yang bersifat global dan bisa ditambahkan ke semua elemen. Berikut ini jenis-jenis atribut yang harus diketahui :

- Atribut Global, adalah atribut yang bisa ditambahkan pada semua elemen HTML. Berikut adalah daftar atribut global dan fungsinya.

Nama Atribut	Deskripsi atau Fungsi
acceskey	Menentukan tombol shortcut untuk mengaktifkan atau fokus pada elemen
class	Menentukan class CSS yang akan digunakan
contenteditable	Menentukan isi elemen bisa diedit atau tidak
id	menentukan id unik untuk elemen
style	menentukan <i>inline</i> CSS untuk elemen
title	menentukan informasi tambahan tentang elemen

- Atribut Event, adalah atribut yang digunakan untuk menentukan aksi yang akan dilakukan saat terjadi sesuatu pada elemen. Atribut ini nanti akan banyak digunakan pada pemrograman Javascript.

- c) Atribut Khusus, atribut yang hanya bisa dipakai pada elemen tertentu dan kadang atribut ini tidak bisa digunakan pada elemen yang lain.

Nama Atribut	Bisa dipakai di tag
<i>src</i>	<audio>, <embed>, <iframe>, , dll.
<i>href</i>	<a>, <link>
<i>action</i>	<form>
<i>autoplay</i>	<audio>, <video>

BAB IV

MEMBUAT ARTIKEL SEDERHANA

A. PARAGRAF

Paragraf adalah kumpulan dari beberapa kalimat. Pada web, paragraf biasanya digunakan untuk menampilkan teks atau artikel. Paragraf pada HTML dibuat dengan tag **<p>**.

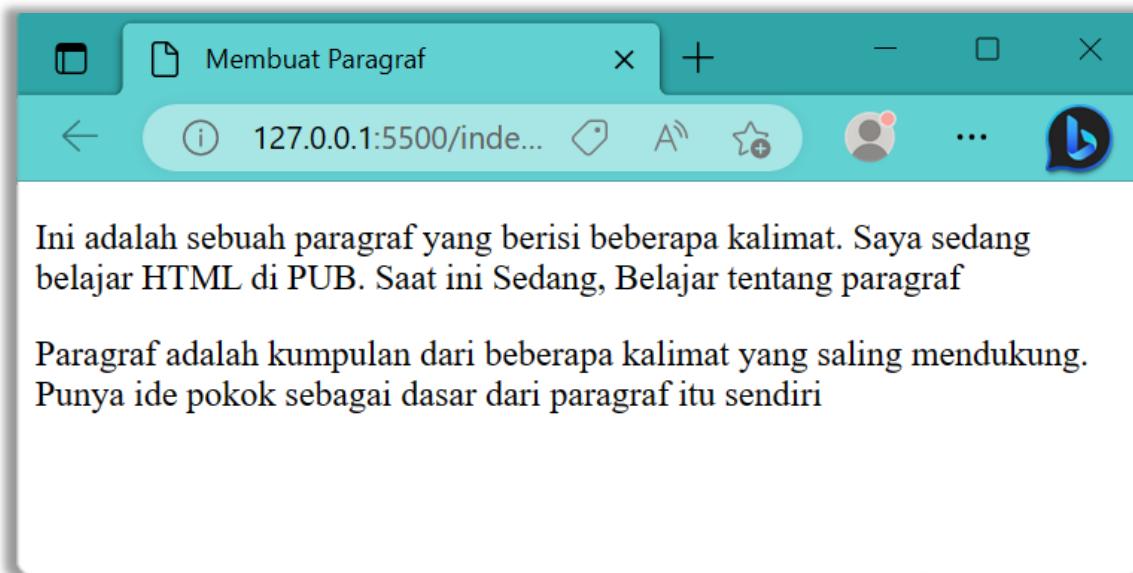
Contoh kode :

```

10
11  <body>
12      <p>Ini adalah sebuah paragraf yang berisi beberapa kalimat.
13          Saya sedang belajar HTML di PUB. Saat ini Sedang,
14          Belajar tentang paragraf
15      </p>
16      <p>Paragraf adalah kumpulan dari beberapa kalimat yang saling
17          mendukung. Punya ide pokok sebagai dasar dari
18          paragraf itu sendiri
19      </p>
20  </body>
21

```

Maka outputnya adalah :



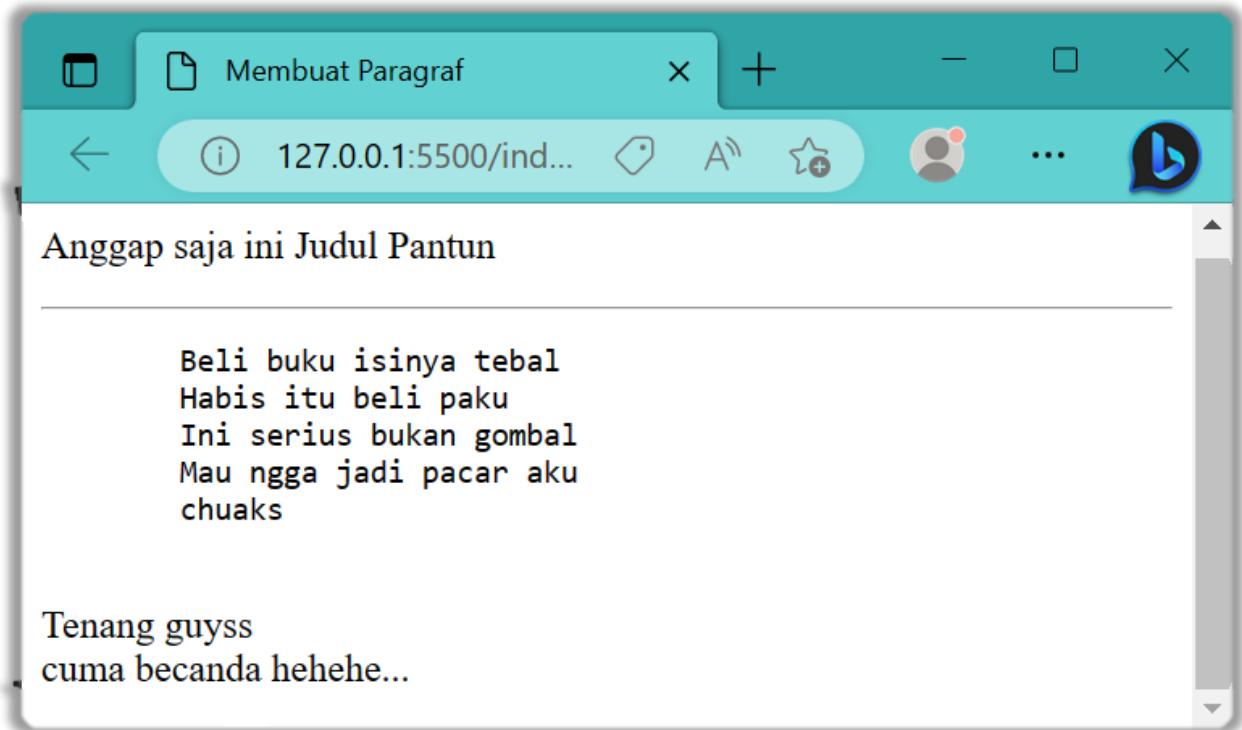
Selain tag ini, ada juga tag pendukung lain seperti `
`, `<hr>`, dan `<pre>`.

Tag	Deskripsi
<code><p></code>	Untuk membuat sebuah paragraf
<code>
</code>	Untuk membuat baris baru
<code><hr></code>	Untuk membuat garis horizontal
<code><pre></code>	mendefinisikan pra di format teks

Sebagai contoh dari penggunaan tag pendukung diatas maka cobalah source code berikut ini.

```
<body>
  <p>Anggap saja ini Judul Pantun</p>
  <hr>
  <pre>
    Beli buku isinya tebal
    Habis itu beli paku
    Ini serius bukan gombal
    Mau ngga jadi pacar aku
    chuaks
  </pre>
  <p>
    Tenang guyss <br> cuma becanda hehehe...
  </p>
</body>
```

Maka outputnya akan seperti gambar dibawah ini :



B. HEADING

Heading adalah sebuah judul yang biasanya diberikan pada halaman atau beberapa bagian dari artikel. Judul pada HTML dapat kita buat dengan tag `<h1>` sampai `<h6>`. Tag `<h1>` digunakan pada judul level pertama. Lalu tag lainnya digunakan pada sub heading atau level berikutnya. Masing-masing judul akan ditampilkan dengan ukuran teks yang berbeda. Tag `<h1>` adalah yang paling besar, dan tag `<h6>` paling kecil.

Contoh :

```
<body>
  <h1>Ini Judul Level 1</h1>
  <h2>Ini Judul Level 2</h2>
  <h3>Ini Judul Level 3</h3>
  <h4>Ini Judul Level 4</h4>
  <h5>Ini Judul Level 5</h5>
  <h6>Ini Judul Level 6</h6>
</body>
```

Outputnya adalah :

Membuat Paragraf

127.0.0.1:5500/ind...

Ini Judul Level 1

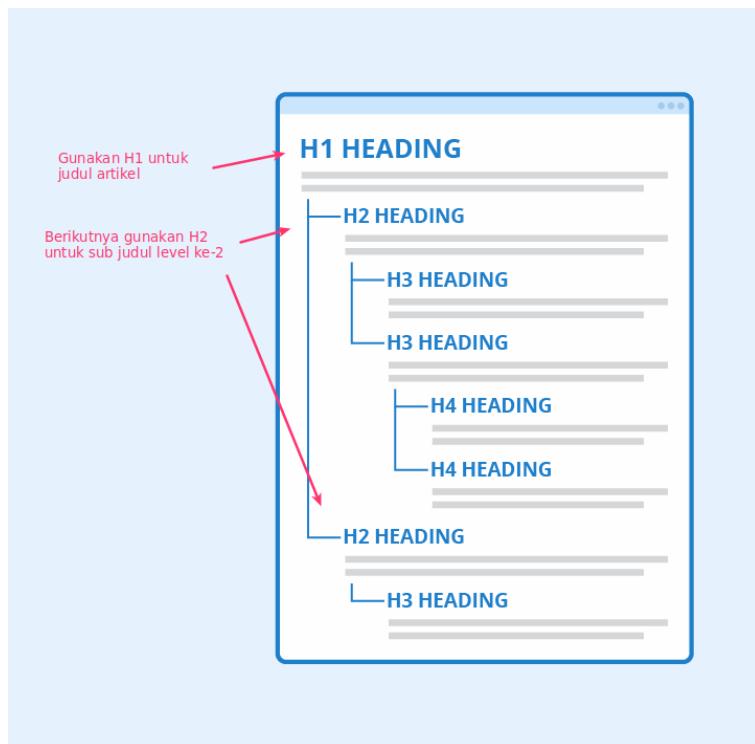
Ini Judul Level 2

Ini Judul Level 3

Ini Judul Level 4

Ini Judul Level 5

Apakah boleh menulis judul yang tidak urut? Maksudnya.. seperti menggunakan `<h6>` untuk judul awal, lalu berikutnya `<h4>` untuk sub judul. Ini boleh-boleh saja, tapi kurang bagus untuk SEO. Penulisan judul yang bagus adalah judul yang mengikuti levelnya.



Judul `<h1>` pada level pertama, `<h2>` pada level ke-2, dan seterusnya

C. TEXT FORMATING

Teks yang terformat akan lebih mudah terbaca karena ada penegasan seperti **teks tebal**, *teks miring*, dan garis bawah. Tujuan dari teks yang terformat adalah memberi penegasan pada kata atau kalimat.

1. Teks Tebal (Bold)

Tag yang biasa digunakan adalah `` dan ``. Berikut adalah contohnya:

```
<b>Ini adalah teks tebal</b> <br>
<strong>Ini menggunakan tag strong</strong>
```

Hasilnya akan menampilkan output yang sama namun sejak adanya HTML 5 (versi terbaru saat ini) tag `` yang awalnya berfungsi sama dengan tag `` berubah fungsi menjadi “Important Text” yaitu membuat kata penting sekaligus membuat teks tebal. Tapi penggunaan tag `` pada pelatihan ini tidak terlalu disarankan, jika ingin tahu alasan lengkapnya maka bisa mempelajari materi tentang SEO (Search Engine Optimizer).

2. Teks Miring (Italic)

Biasanya digunakan untuk menegaskan sebuah istilah baru. Biasanya menggunakan tag `<i>` (italic) atau `` (empharis). Berikut contohnya :

```
<i>Ini adalah teks miring</i> <br>
<em>Kalo ini menggunakan empharis</em>
```

3. Garis Bawah pada Teks (Underline)

Biasanya digunakan untuk menandai teks yang disisipkan atau teks yang memiliki arti penting dibandingkan dengan teks lainnya. Biasanya menggunakan tag `<u>` (underline) atau tag `<ins>` (insert). Berikut adalah contohnya :

```
<u>Ini adalah teks bergaris bawah</u> <br>
<ins>Kalo ini menggunakan insert</ins>
```

Tiga tag diatas adalah tag yang biasa digunakan atau yang paling populer namun ada beberapa tag lain yang harus diketahui juga. Berikut adalah tabelnya:

Tag	Kegunaan	Tindakan yang Dihasilkan
	Memberi penegasan pada teks	Teks menjadi tebal
	Membuat kata penting	
<i> dan 	Menegaskan sebuah kata atau istilah baru	Teks menjadi miring
<u> dan <ins>	Menandai teks yang disisipkan	Teks memiliki garis bawah
 dan <s>	Menandai teks yang dihapus	Teks tercoret
<sup>	Membuat pangkat atas	Terdapat sebuah pangkat di atas teks
<sub>	Membuat pangkat bawah	Terdapat sebuah pangkat di bawah teks
<mark>	Menandai kata kunci yang penting	Memberi efek stabilo

Tag – tag tersebut juga bisa digunakan secara bersamaan. Contohnya jika ingin membuat kata tebal dan miring maka harus menggabungkan tag **** dan tag **<i>**.

Contohnya :

```
<i><b>Ini teks tebal dan miring</b></i>
```

D. LIST

HTML sudah menyediakan elemen untuk membuat list. Ada tiga macam jenis list yang bisa dibuat di HTML:

1. Ordered List di HTML

Ordered list dibuat dengan tag ****. Lalu di dalamnya diisi dengan item-item yang akan dimasukkan ke dalam list. Item dibuat dengan tag **** (list item). Contohnya adalah kode berikut.

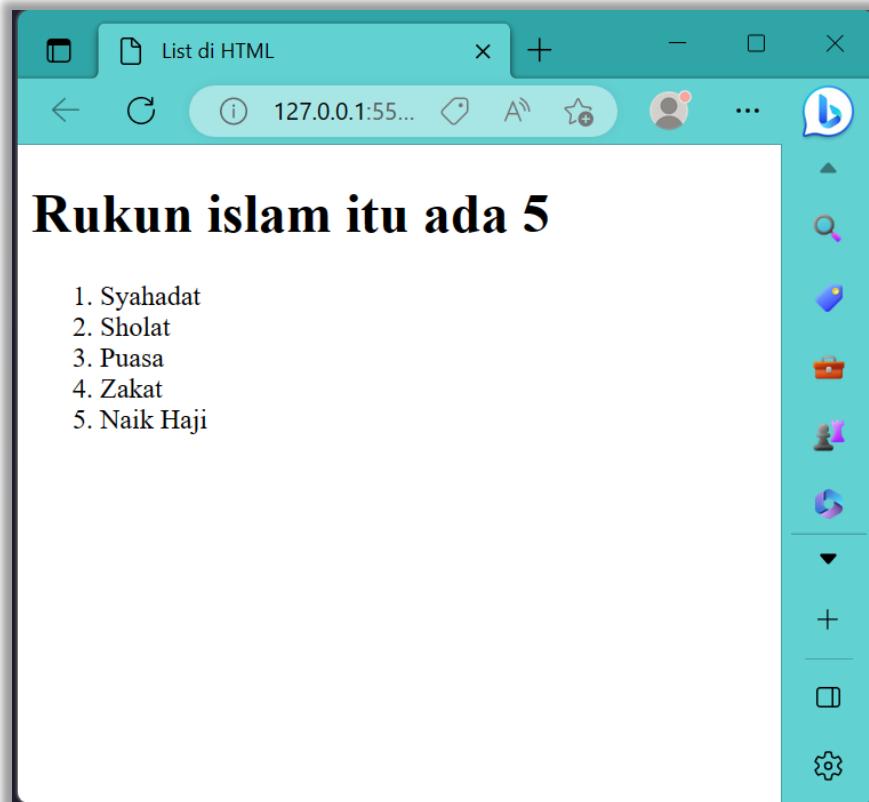
```
<body>
  <h1>Rukun islam itu ada 5</h1>
  <ol>
```

```

<li>Syahadat</li>
<li>Sholat</li>
<li>Puasa</li>
<li>Zakat</li>
<li>Naik Haji</li>
</ol>
</body>

```

Hasilnya adalah :



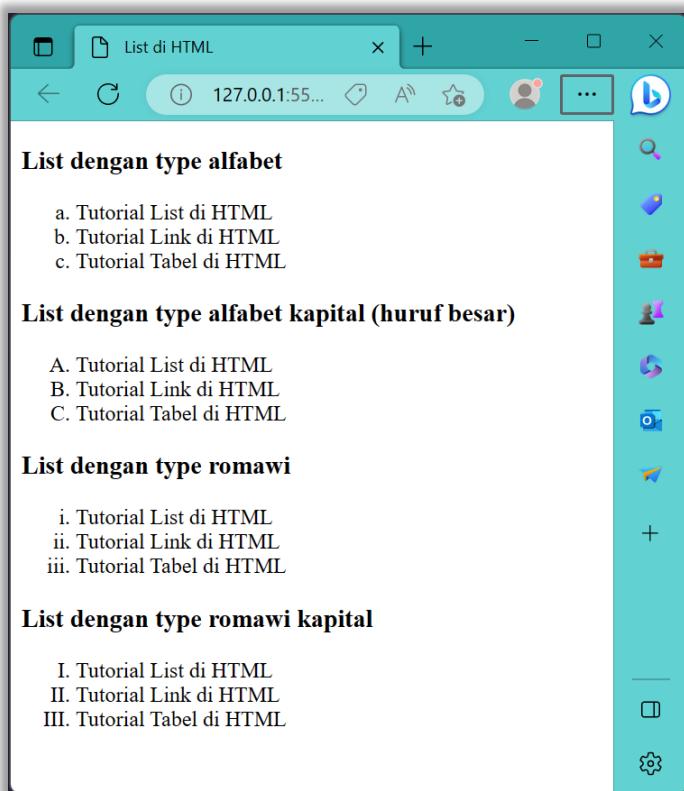
List diatas diurutkan berdasarkan angka dari 1, 2, 3, sampai 5. Untuk membuat angka romawi atau huruf, kita bisa menggunakan atribut ***type*** pada tag **** dan berikut ini nilai yang bisa diberikan:

Type	Kegunaan
a	Untuk alfabet a, b, c, dan seterusnya
A	Untuk alfabet A, B, C, dan seterusnya
i	Untuk angka romawi i, ii, iii, dan seterusnya
I	Untuk angka romawi I, II, III, dan seterusnya.

Contoh kode :

```
<body>
    <h3>List dengan type alfabet</h3>
    <ol type='a'>
        <li>Tutorial List di HTML</li>
        <li>Tutorial Link di HTML</li>
        <li>Tutorial Tabel di HTML</li>
    </ol>
    <h3>List dengan type alfabet kapital (huruf besar)</h3>
    <ol type='A'>
        <li>Tutorial List di HTML</li>
        <li>Tutorial Link di HTML</li>
        <li>Tutorial Tabel di HTML</li>
    </ol>
    <h3>List dengan type romawi</h3>
    <ol type='i'>
        <li>Tutorial List di HTML</li>
        <li>Tutorial Link di HTML</li>
        <li>Tutorial Tabel di HTML</li>
    </ol>
    <h3>List dengan type romawi kapital</h3>
    <ol type='I'>
        <li>Tutorial List di HTML</li>
        <li>Tutorial Link di HTML</li>
        <li>Tutorial Tabel di HTML</li>
    </ol>
</body>
```

Hasilnya adalah :



2. Unordered List di HTML

Unordered list adalah list yang tak terurut yang menggunakan simbol-simbol pada item-nya. Unordered list dibuat dengan tag `` dan untuk item-nya dibuat juga dengan tag ``. Contohnya adalah kode berikut :

```
<body>
  <h1>Bahasa Pemrograman untuk dipelajari:</h1>
  <ul>
    <li>Javascript</li>
    <li>PHP</li>
    <li>Java</li>
    <li>Python</li>
    <li>Kotlin</li>
  </ul>
</body>
```

Hasilnya adalah :



Secara default simbol yang digunakan oleh unordered list adalah lingkaran kecil (*disc*). Kita juga bisa menggantinya dengan atribut **type**. Berikut ini nilai yang bisa diberikan untuk atribut **type**:

- **square** untuk simbol persegi;
- **disc** (default) untuk simbol lingkaran disc;
- **none** tidak memakai simbol;
- **circle** untuk simbol lingkaran;

3. Description List di HTML

Description List adalah list yang berisi deksripsi atau penjelasan dari sesuatu.

Ada tiga tag yang digunakan untuk membuat description list :

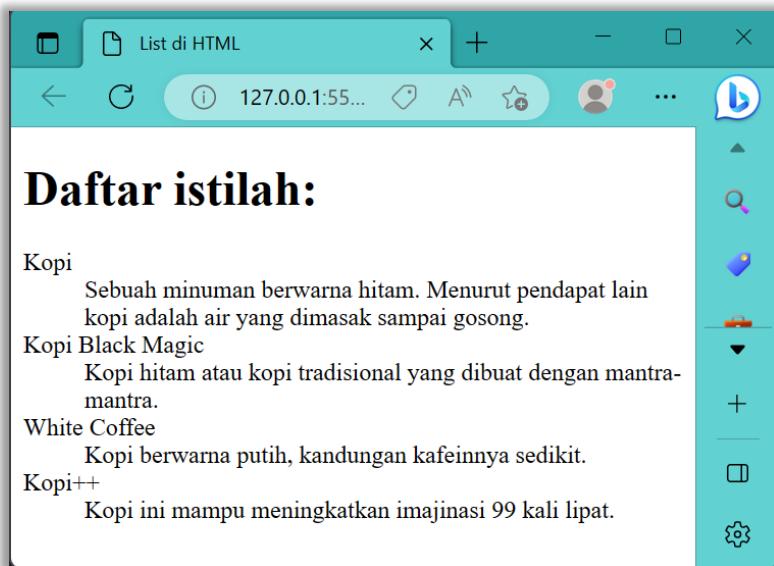
- **<dl> (description list)**, tag untuk memulai description list;
- **<dt> (description term)**, tag untuk membuat kata yang akan dideskripsikan;
- **<dd> (description description)**, tag untuk membuat penjelasan dari kata.

Contohnya adalah kode berikut :

```
<body>
    <h1>Daftar istilah:</h1>
    <dl>
        <dt>Kopi</dt>
            <dd>Sebuah minuman berwarna hitam. Menurut pendapat lain
                kopi adalah air yang dimasak sampai gosong.</dd>
        <dt>Kopi Black Magic</dt>
            <dd>Kopi hitam atau kopi tradisional yang dibuat dengan
                mantra-mantra.</dd>
        <dt>White Coffee</dt>
            <dd>Kopi berwarna putih, kandungan kafeinnya sedikit.</dd>
        <dt>Kopi++</dt>
```

```
        <dd>Kopi ini mampu meningkatkan imajinasi 99 kali lipat.  
    </dd>  
  </dl>  
</body>
```

Hasilnya adalah :



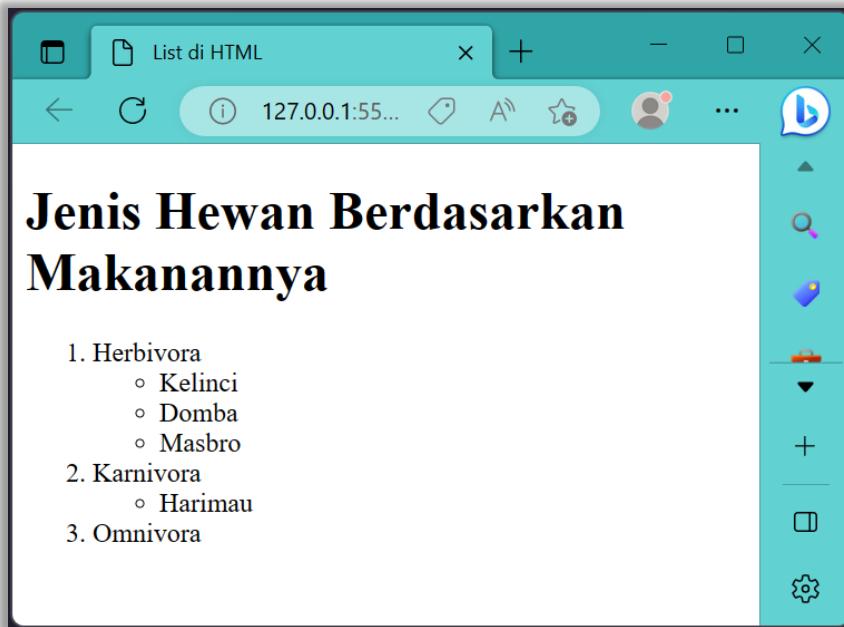
4. List di dalam List (Nested List)

List juga dapat dibuat di dalam list, misalkan kita ingin menggabungkan ordered list dengan unordered list. Caranya, list yang di dalam ditulis di dalam tag ``. Contohnya adalah kode berikut :

```
<body>  
  <h1>Jenis Hewan Berdasarkan Makanannya</h1>  
  <ol>  
    <li>Herbivora  
      <ul>  
        <li>Kelinci</li>  
        <li>Domba</li>  
        <li>Masbro</li>  
      </ul>  
    </li>  
    <li>Karnivora  
      <ul>  
        <li>Harimau</li>  
      </ul>  
    </li>  
    <li>Omnivora</li>
```

```
</ol>
</body>
```

Hasilnya :



E. KOMENTAR

Komentar adalah elemen yang akan diabaikan oleh browser. Ia tidak akan ditampilkan di dalam web. Komentar biasanya digunakan untuk memberikan informasi tambahan pada kode HTML dan kadang juga digunakan untuk menon-aktifkan beberapa kode HTML. Cara membuatnya adalah menggunakan simbol `<-- -->` atau dengan menekan `ctrl + /`

Contohnya :

```
<body>
    <!-- ini adalah komentar -->
    <p>Hello World!</p>
    <!-- ini juga komentar
        dan ditulis dalam dua baris -->
</body>
```

F. ELEMEN SEMANTIK

Elemen semantik adalah elemen-elemen yang menyatakan makna atau tujuan dari elemen itu sendiri. HTML semantik mulai hadir sejak HTML 5. Sebelumnya para developer menggunakan tag `<div>` sebagai pembungkus dari beberapa tag, namun semakin banyak

konten yang dimuat dalam sebuah web maka semakin banyak tag `<div>` yang digunakan. Hal ini dapat membingungkan para developer dalam membangun aplikasi web, maka dari itu penggunaan tag semantik sangat diperlukan.



Misalnya tag `<footer>`, tag ini digunakan untuk membuat elemen footer atau bagian kaki dari web. Jangan gunakan tag ini di bagian paling atas, karena maknanya sudah jelas untuk footer. Jadi tidak akan ada lagi yang namanya penyalahgunaan tag. Karena setiap tag sudah punya tujuan masing-masing.

Berikut ini adalah daftar elemen semantik

Nama Tag	Kegunaan
<code><article></code>	untuk membuat elemen artikel
<code><aside></code>	untuk membuat elemen bagian samping
<code><details></code>	untuk membuat elemen detail atau spoiler
<code><figcaption></code>	untuk membuat teks caption pada figur
<code><figure></code>	untuk membuat figur atau gambar pada picture

<footer>	untuk membuat elemen bagian kaki dari web
<header>	untuk membuat kepala kop web
<main>	untuk membuat elemen utama
<nav>	untuk membuat navigasi
<section>	untuk membuat bagian artikel
<summary>	untuk membuat ringkasan artikel atau isi spoiler
<time>	untuk membuat elemen yang menyatakan waktu

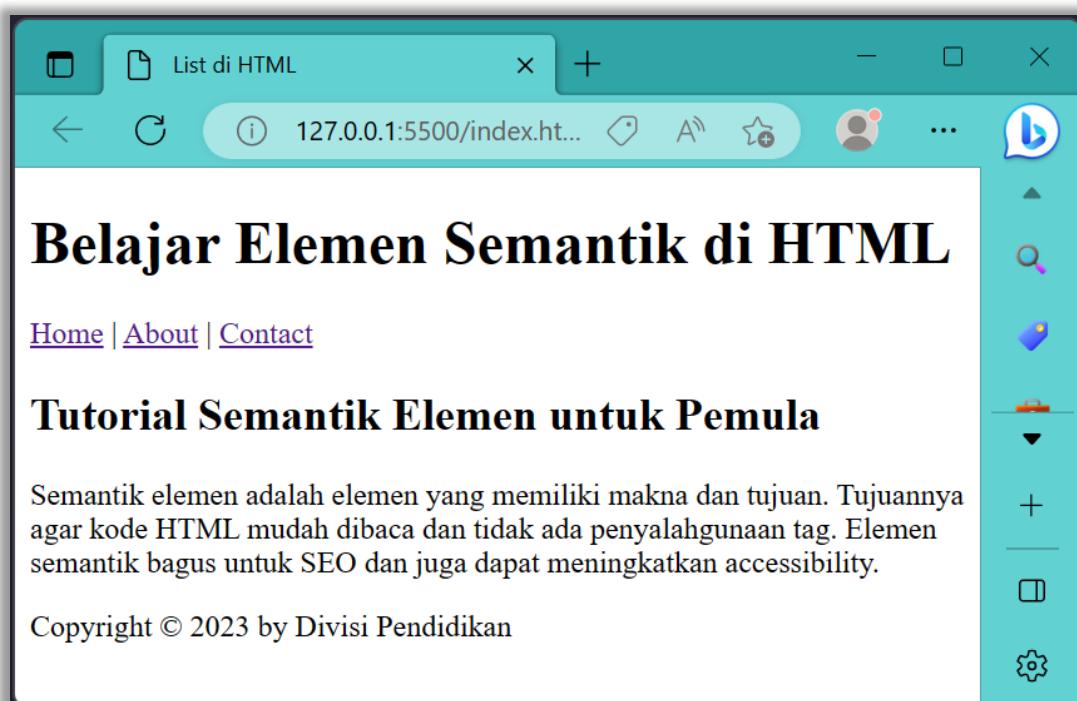
Salah satu keuntungan menggunakan elemen semantik adalah dokumen HTML kita akan mudah dibaca, baik itu oleh manusia maupun mesin. Contoh kodennya adalah sebagai berikut :

```

<body>
    <header>
        <h1>Belajar Elemen Semantik di HTML</h1>
    </header>
    <nav>
        <a href="#">Home</a> |
        <a href="#">About</a> |
        <a href="#">Contact</a>
    </nav>
    <article>
        <h1>Tutorial Semantik Elemen untuk Pemula</h1>
        <p>Semantik elemen adalah elemen yang memiliki makna dan tujuan. Tujuannya agar kode HTML mudah dibaca dan tidak ada Penyalahgunaan tag. Elemen semantik bagus untuk SEO dan juga dapat meningkatkan accessibility.
        </p>
    </article>
    <footer>
        Copyright &copy; 2023 by Divisi Pendidikan
    </footer>
</body>

```

Maka hasilnya adalah :



BAB V

TAUTAN HTML

A. APA ITU LINK

Link atau Hyperlink adalah elemen HTML yang berfungsi menghubungkan suatu halaman web ke halaman web yang lain. Elemen ini bisa di klik dan nanti akan membuka ke halaman lain sesuai alamat URL yang diberikan.



Tidak hanya untuk menghubungkan halaman, link juga punya fungsi lain seperti scroll top, download file, menjalankan fungsi javascript, dll.

Link pada HTML dapat dibuat dengan tag `<a>`, kemudian tag ini harus memiliki atribut `href` untuk menentukan alamat URL tujuan dari link. Berikut adalah contohnya :

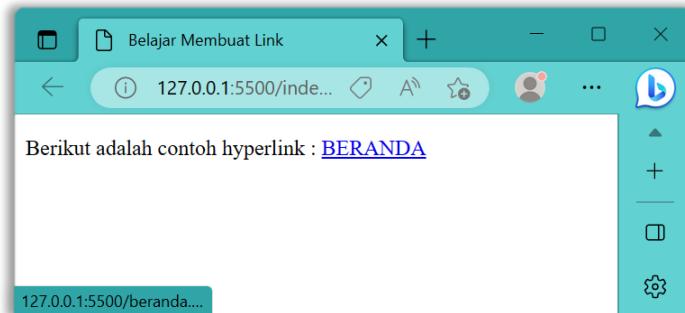
```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Belajar Membuat Link</title>
</head>

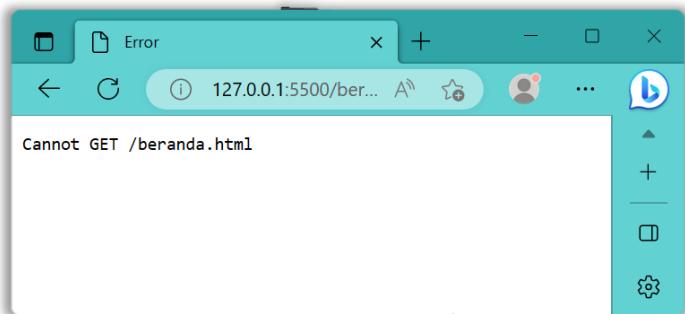
<body>
    <p>
        Berikut adalah contoh hyperlink : <a href="beranda.html">BERANDA</a>
    </p>
</body>
```

```
</html>
```

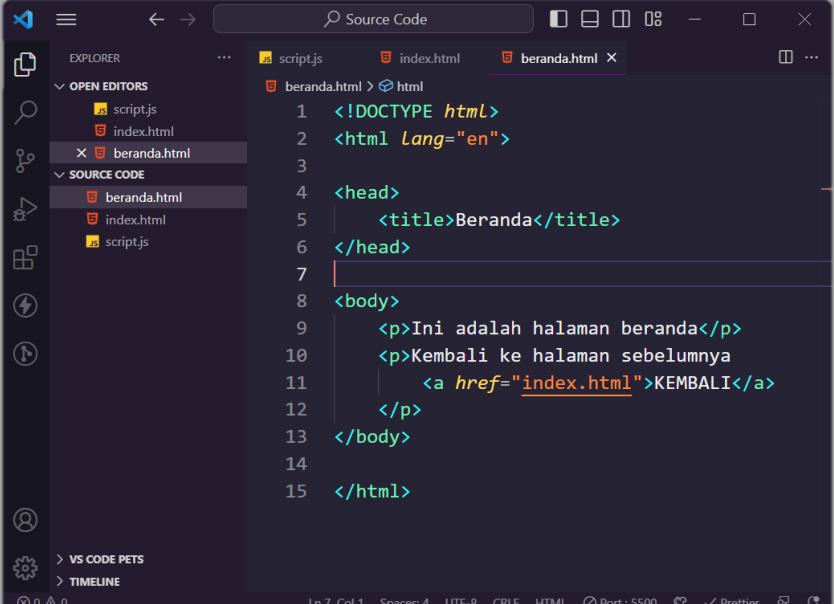
Maka outputnya adalah :



Jika kita mengarahkan cursor kita kepada teks berwarna biru, maka cursornya akan berubah menjadi berbentuk tangan yang sedang menunjuk dan menandakan bahwa linknya sudah aktif. Kemudian jika kita meng-klik link tersebut maka outputnya akan seperti ini :

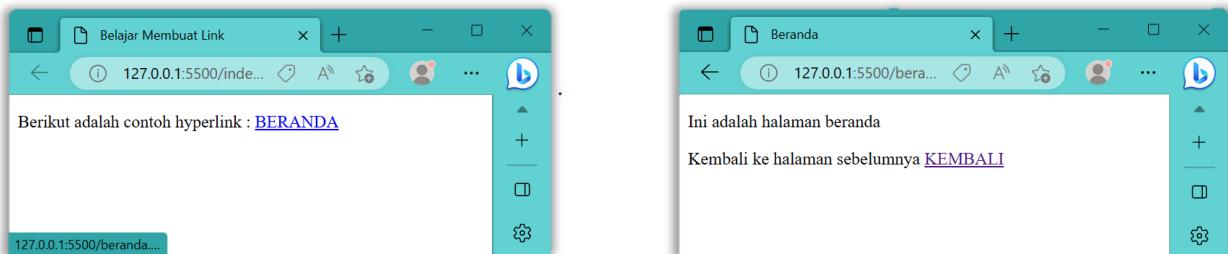


Ketika link tersebut di klik maka tampilannya akan seperti diatas / error (setiap browser berbeda beda), karena link yang kita buat akan membuka halaman '**'beranda.html'**' tapi file ini belum dibuat. Untuk mengatasi masalah ini maka harus dibuat file '**'beranda.html'**' terlebih dahulu kemudian isi dengan kode berikut.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Beranda</title>
</head>
<body>
    <p>Ini adalah halaman beranda</p>
    <p>Kembali ke halaman sebelumnya<br/><a href="index.html">KEMBALI</a></p>
</body>
</html>
```

Kemudian jika kita mengeklik link yang ada di file '**indeks.html**' maka akan langsung terhubung ke file '**beranda.html**'



B. JENIS – JENIS LINK

Atribut href (hyperlink reference) tidak harus selalu merujuk kepada file HTML, ada berbagai macam tautan yang bisa dihubungkan. Berikut adalah jenis-jenis link pada HTML

1. Internal Link

Adalah link yang menuju ke domain atau halaman web itu sendiri. Biasanya digunakan untuk menghubungkan halaman yang satu dengan yang lainnya dalam satu website atau domain. Misalnya menghubungkan halaman **Beranda** dengan halaman **About Us** pada sebuah website.

Buat kode berikut di file **indeks.html**

```
<body>
    <h1>Selamat Datang</h1>
    <p>
        <a href="index.html">Home</a> |<br/>
        <a href="contact.html">Contact</a> |<br/>
        <a href="about.html">About</a>
```

```
</p>
<hr>
<p>
    Selamat datang di websiteku. Coba klik menu di atas,
    maka kamu akan membuka halaman yang berbeda. Semua
    link di atas adalah internal link.
</p>
</body>
```

Buat kode berikut di file **contact.html**

```
<body>
    <h1>Contact</h1>
    <p>
        <a href="index.html">Home</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About</a>
    </p>
    <hr>
    <p>
        Kamu bisa menghubungi saya melalui nomer WA: 0871111111
        atau juga alamat email: mail@contoh.com
    </p>
</body>
```

Terakhir buat kode berikut di file **about.html**

```
<body>
    <h1>About</h1>
    <p>
        <a href="index.html">Home</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About</a>
    </p>
    <hr>
    <p>
        Ini adalah halaman about dari website saya.
        Jadi ini adalah contoh cara membuat link internal di HTML.
    </p>
</body>
```

Pastikan semua file tersebut ada di dalam folder yang sama, kemudian cobalah untuk mengeklik setiap link yang ada dan rasakan keajaibannya.

2. Eksternal Link

Adalah link yang menuju domain lain. Biasanya digunakan untuk menghubungkan web domain lain seperti facebook, whatsapp, dll. Perbedaan antara eksternal link dengan internal link adalah selama link itu membuka web lain, maka ia adalah external. Tapi kalau tetap membuka web itu sendiri maka itu internal. Misalnya kita tambahkan link baru di file **contact.html**

```
<body>
  <h1>Contact</h1>
  <p>
    <a href="index.html">Home</a> | 
    <a href="contact.html">Contact</a> | 
    <a href="about.html">About</a>
  </p> <hr>
  <p>
    Kamu bisa menghubungi saya melalui nomer WA: 0871111111
    atau juga alamat email: mail@contoh.com
    Jangan lupa subscribe channel
    <a href="https://www.youtube.com/channel/UCaIbbu5Xg3DpHsn_3Zw2m9w">
      Youtube
      </a>kami ya....
  </p>
</body>
```

Selain atribut **href**, masih ada atribut lain yang sering ditambahkan pada link, seperti **target**, **title**, **rel**, **style**, dll.

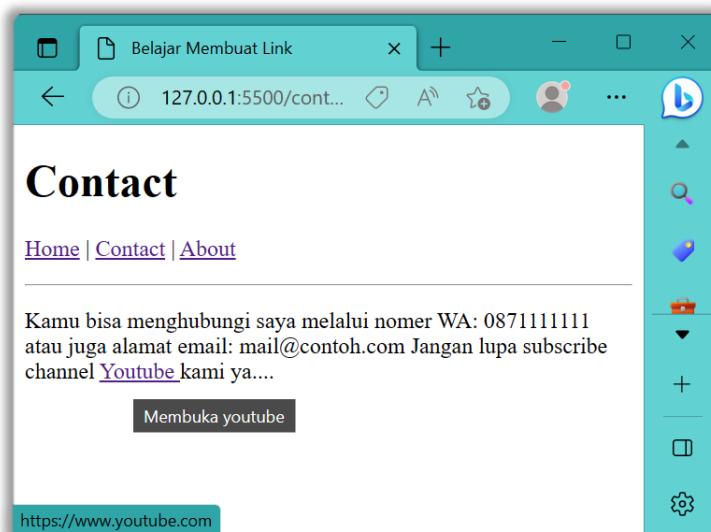
Atribut **target** berfungsi untuk menentukan target dari pembukaan link. Ada beberapa target yang biasanya digunakan :

Nama Atribut	Kegunaan
_blank	Akan membuka link pada jendela baru
_self	Akan membuka link pada halaman itu sendiri (default target)
_top	Menuju pada bagian paling atas halaman
_parent	Membuka link pada frame induk
nama-frame	Akan membuka link pada <iframe> dengan nama tertentu

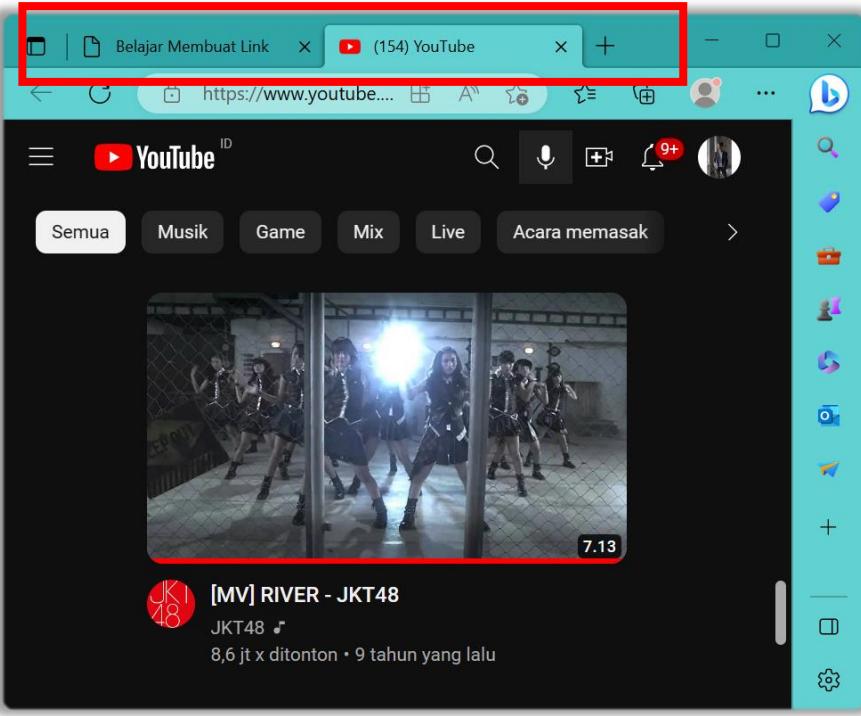
Contohnya

```
<p>
    Kamu bisa menghubungi saya melalui nomer WA: 0871111111
    atau juga alamat email: mail@contoh.com
    Jangan lupa subscribe channel
    <a href="https://www.youtube.com/" title="Membuka youtube" target="_blank">
        Youtube
    </a> kami ya....
</p>
```

Jika cursor diarahkan kepada link tersebut akan muncul title link (judul link)



Kemudian jika linknya di klik maka akan muncul tab baru (new tab)



Link tidak hanya digunakan untuk menghubungkan halaman web saja. Ia juga bisa digunakan untuk beberapa fungsi seperti :

- Membuat **link buntu**, yaitu link yang tidak akan membuka apa pun. Biasanya digunakan sebagai sampel saja. Caranya adalah memberi tanda pagar (#) pada atribut **href**. Tanda pagar sendiri artinya adalah **anchor** (jangkar). Anchor ini nantinya akan membawa kita ke lokasi tertentu di dalam dokumen HTML

```
<p>
    Selamat datang di websiteku. Coba klik menu di atas,
    maka kamu akan membuka halaman yang berbeda. Semua
    link di atas adalah internal link.
    <a href="#">BUNTU</a>
</p>
```

- Membuat **link anchor**, adalah link yang menuju ke suatu elemen tertentu. Cara kerjanya seperti saat kita mengaitkan jangkar, lalu ditarik ke sana. Karena itu, dia dinama anchor. Cara membuat link anchor adalah dengan mengisi alamat URL dengan tanda pagar (#), lalu diisi dengan nama **id** dari elemen yang akan dituju. Untuk lebih jelasnya coba salin link berikut ini dan rasakan keajaibannya.

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
    <title>Tutorial Link di HTML</title>
</head>

<body>
    <h1>Contoh Link Anchor</h1>
    <p>Ini adalah contoh link anchor. Coba klik link ini:
        <a href="#penutup">Meluncur ke Penutup</a> maka kamu
        akan dibawa ke bagian penutup dari dokumen ini.
    </p>
    <h2>Apa itu Jangkar?</h2>
    <p>
        Anchor dalam bahasa indonesia artinya jangkar. Saya yakin
        kamu pasti pernah melihat jangkar. Fungsi jangkar untuk
        menahan kapal agar tidak hanyut. Tapi dalam HTML, fungsinya
        untuk membuat link yang bisa membawa kita meluncur ke tujuan
        jangkar itu.
    </p>
    <p>
        Jangkar atau anchor di HTML tidak hanya bisa digunakan pada
        satu dokumen saja. Ia juga bisa digunakan untuk membuka dokumen
        lain, lalu mengaitkan jangkarnya.
    </p>
    <p>
        Jangkar biasanya dibuat dengan tanda pagar (#) lalu diikuti
        dengan nama id dari elemen yang ingin dituju. Contohnya #penutup,
        maka link yang menggunakan anchor tersebut akan mencari elemen
        HTML yang memiliki id `penutup` dan membawa kita ke sana.
        Kira-kira begitulah cara kerjanya.
    </p>
    <h2 id="penutup">Akhir Kata..</h2>
    <p>
        Ini adalah penutup dari artikel ini, dan coba perhatikan
        headingnya. Di sana kita menggunakan id="penutup". Maka elemen
        ini akan menjadi tujuan dari anchor link.
    </p>
    <p>
        <a href="#top">Kembali ke Atas!</a>
    </p>
</body>

</html>

```

Jika kita perhatikan terdapat **#top** sebagai URL. Sedangkan elemen yang memiliki **id="top"** tidak ada di HTML. Ini karena browser sudah paham, jika ada link anchor yang menuju ke **#top** maka ia akan dibawa ke bagian teratas dari dokumen.

- Membuat **link untuk memanggil fungsi javascript**, biasanya dilakukan dengan atribut event seperti **onclick**, **onmouseover**, **onmouseout**, dan sebagainya.

Contohnya :

```
<body>
  <p>
    <a href="#" onclick="alert('Hello World!')">Jalankan JS</a> <br>
    <a href="#" onmouseover="alert('link sudah kamu sentuh!')">
      Coba Sentuh Link ini
    </a>
  </p>
</body>
```

- Membuat **link dengan gambar**, Membuat gambar sebagai link kadang sering dilakukan dalam web. Cara membuatnya sangat mudah, kita hanya perlu membungkus tag **** dengan tag **<a>**.

```
<a href="#"></a>
```

- Membuat **link untuk mendownload file**, cara membuatnya sangat mudah, kita hanya perlu memasukkan alamat URL dari file yang akan di download. Contohnya adalah kode berikut :

```
<a href="MODUL-WEB">Download modul pelatihan web</a>
```

Link diatas hanya bisa mendownload file yang ada di satu folder dengan file HTML yang sedang dijalankan.

- Membuat **link untuk mengirim pesan email**, caranya adalah dengan mengubah URL tujuan dari link dengan **mailto**, lalu diikuti dengan alamat email yang akan menerima email.

```
<a href="mailto:divdik.pub@gmail.com" title="kirim email ke Admin">
  Hubungi Kami
</a>
```

Jika kamu membukanya di Google Chrome, maka ia akan membuka Gmail. Tapi jika dibuka di Firefox, ia akan memilih aplikasi email yang tersedia di komputer.

BAB VI

GAMBAR DI HTML

A. MENAMBAHKAN GAMBAR DI HTML

Gambar dapat kita tambahkan di HTML dengan menggunakan tag ``. Tag ini memiliki atribut wajib, yakni `src`. Jika kita tidak mengisi atribut `src`, maka gambar tidak akan ditampilkan.

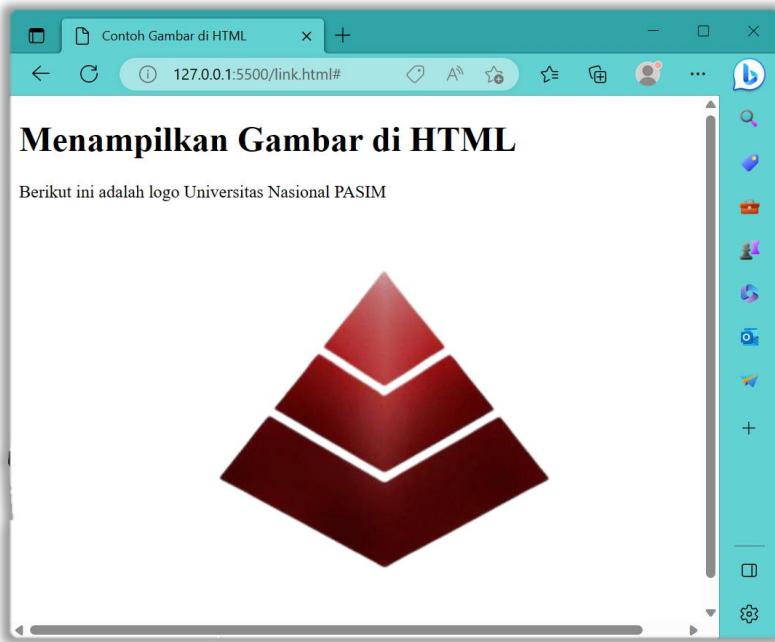
```
<!DOCTYPE html>
<html Lang="en">

<head>
    <title>Contoh Gambar di HTML</title>
</head>

<body>
    <h1>Menampilkan Gambar di HTML</h1>
    <p>Berikut ini adalah logo Universitas Nasional Pasim</p>
    <p>
        
    </p>
</body>

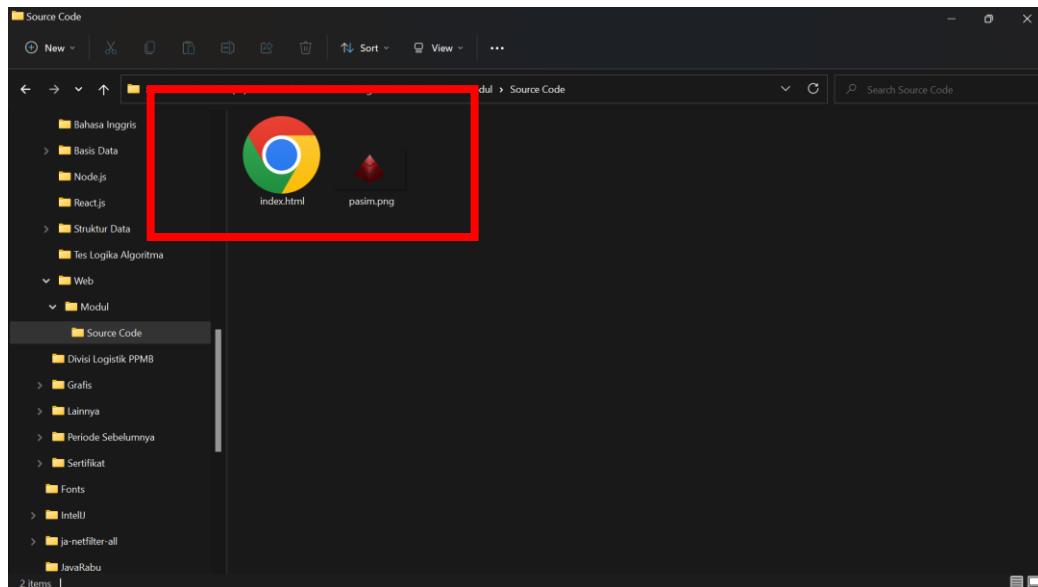
</html>
```

Alamat URL gambar pada atribut `src` dapat berupa URL maupun alamat path. Lalu tag `` harus ditutup dengan menambahkan garis miring. Dari kode di atas maka akan menghasilkan output berikut.

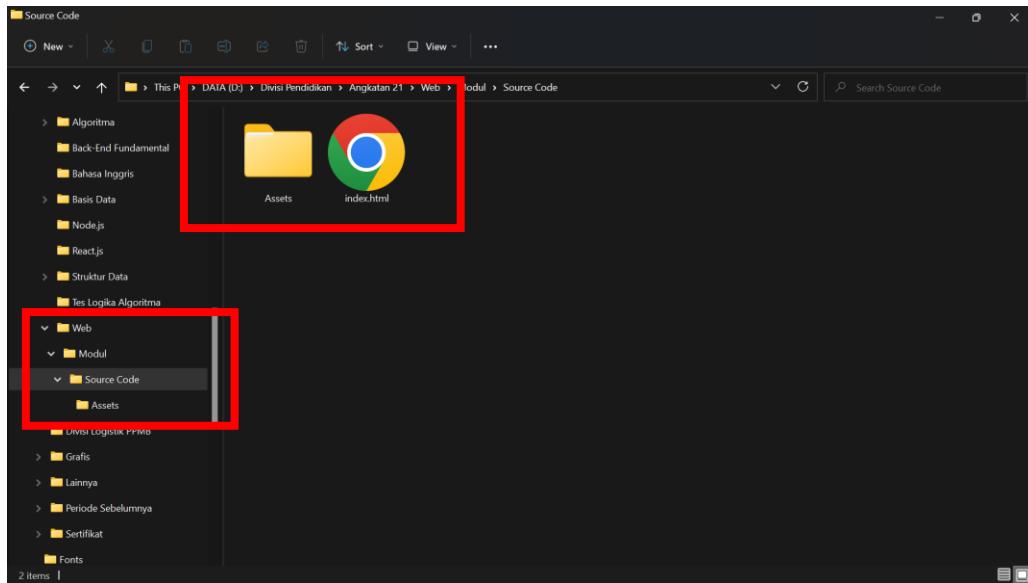


Perhatikan!

Pada contoh di atas, kita menuliskan langsung nama file dari gambar. Ini karena kita menaruh gambar di dalam folder yang sama dengan file HTML



Apabila file gambarnya tersimpan di folder yang berbeda, maka kita hanya perlu menuliskan alamat path-nya. Misalnya kita ingin membuat folder **Assets**, kemudian pindahkan gambar **pasim.png** ke dalam folder **Assets**



Maka cara untuk mengambil gambarnya adalah

```

```

Lalu bagaimana kalau gambarnya berada di internet atau website lain? Nah, jika kita menggunakan gambar dari website lain, kita harus menuliskan alamat URL lengkap dari gambar tersebut. Contohnya adalah sebagai berikut :

```

```

Maka hasilnya adalah



B. FORMAT FILE GAMBAR UNTUK HTML

Tidak semua format file gambar dapat ditampilkan di HTML. Karena tiap file gambar memiliki tujuan masing-masing. Misalnya file gambar dengan format PSD, ia adalah file gambar untuk Photoshop. Jelas ini tidak akan bisa ditampilkan di HTML. Berikut ini daftar beberapa format gambar yang sering digunakan dalam web :

Nama Format	Nama Panjang	Ekstensi
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group Image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg
WebP	Web Picture	.webp

Format file ini juga akan bergantung pada versi browser yang digunakan. Kita ambil contoh format WebP, ini adalah format file gambar baru yang dikembangkan oleh Google. Jika kita menggunakan browser versi lama, gambar ini tidak akan bisa ditampilkan.

C. ATRIBUT UNTUK TAG ``

Ada beberapa atribut yang sering digunakan pada tag `` :

Nama Atribut	Kegunaan
<code>alt</code>	Teks alternatif untuk gambar
<code>width</code>	Menentukan lebar gambar
<code>height</code>	Menentukan tinggi gambar
<code>style</code>	menentukan style css untuk gambar

1. Atribut `alt`

Atribut ***alt*** adalah atribut untuk memberikan teks alternatif pada gambar saat gambar gagal ditampilkan.

Contohnya adalah :

```
<img src="" alt="logo pasim"/>
```

Hasilnya adalah :



Pada contoh tersebut, kita sengaja mengosongkan nilai atribut ***src***. Akibatnya gambar gagal ditampilkan dan yang akan ditampilkan adalah teks alternatif.

2. Atribut ***width*** dan ***height***

Atribut ***width*** dan ***height*** adalah atribut yang digunakan untuk menentukan lebar dan tinggi dari gambar. Contohnya adalah kode berikut :

```
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>
    
    
    
  </p>
</body>
```

Hasilnya adalah :



Satuan yang digunakan untuk nilai **width** dan **height** adalah piksel (px). Jika kita memberikan nilai 200, artinya.. kita memberikan nilai 200px.

Atribut **width** dan **height** juga tidak wajib. Tapi baiknya ditambahkan agar ukuran gambar konsisten serta lebih baik digunakan di CSS.

D. TAG TAMBAHAN UNTUK GAMBAR

Pada HTML versi 5, ada beberapa tag tambahan yang bisa digunakan untuk membantu tag ****, diantaranya adalah :

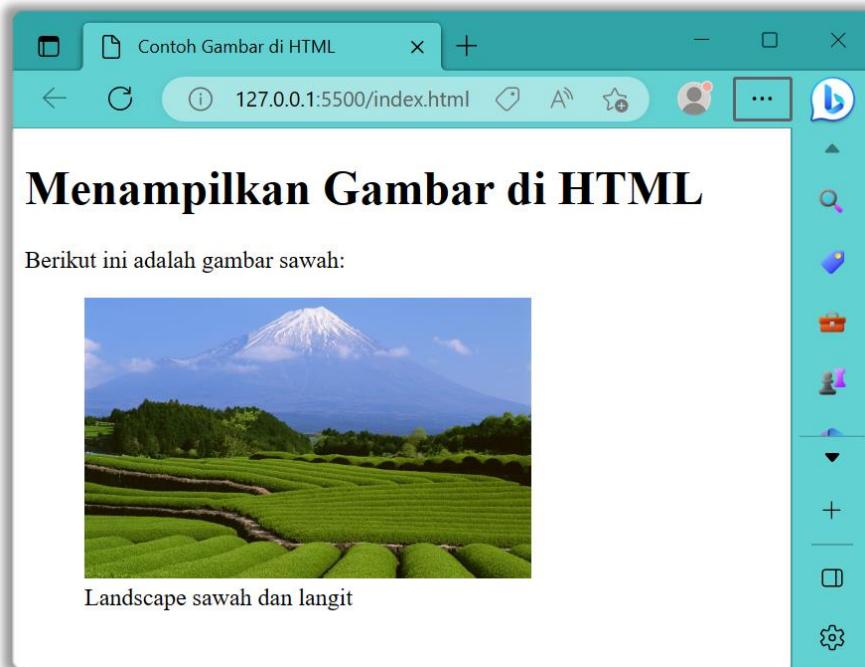
- **<figure>** untuk membungkus gambar dan teks caption-nya
- **<picture>** untuk menentukan jenis gambar pada ukuran layar yang berbeda.

1. Tag **<figure>**

Tag figure berfungsi untuk membungkus tag **** atau gambar dengan teks caption. Teks caption adalah teks yang menjelaskan tentang gambar. Teks caption bisa dibuat dengan tag **<figcaption>**. Contohnya adalah kode berikut :

```
<body>
    <h1>Menampilkan Gambar di HTML</h1>
    <p>Berikut ini adalah gambar sawah:</p>
    <p>
        <figure>
            
            <figcaption>Landscape sawah dan langit</figcaption>
        </figure>
    </p>
</body>
```

Hasilnya adalah :



2. Tag <picture>



Tag ini berfungsi untuk menentukan gambar mana yang akan ditampilkan pada ukuran layar tertentu (responsif). Berikut adalah contoh kodennya :

```
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah :</p>
  <p>
```

```
<picture>
<source media="(min-width: 670px)" srcset="Assets/sawah.jpg"
width="650px" height="500px">
<source media="(min-width: 550px)" srcset="Assets/sawah.jpg"
width="450px" height="300px">

</picture>
</p>
</body>
```

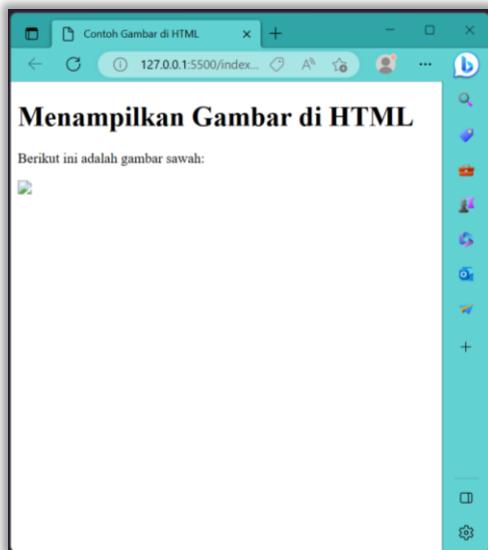
Hasilnya adalah :



Kemudian jika layarnya lebih dikecilkan sampai lebar layarnya 669px, akan menjadi seperti gambar di bawah ini :



Kemudian jika layarnya lebih dikecilkan sampai lebar layarnya 549px, akan seperti gambar di bawah ini :



Gambar terakhir tidak menampilkan apa-apa karena atribut pada kondisi terakhir kita tidak memasukkan URL gambarnya.

BAB VII

FORM HTML

A. CARA MEMBUAT FORM

Form berfungsi untuk mengambil informasi dari user / pengunjung web. Form di HTML dapat kita buat dengan tag `<form>`. Tag ini memiliki beberapa atribut yang harus diberikan, seperti:

- `action` untuk menentukan aksi yang akan dilakukan saat data dikirim;
- `method` metode pengiriman data.
- Dll.

Contoh :

```
<form action="login.js" method="GET">
    <!-- form field di sini -->
</form>
```

Untuk atribut `action`, kita dapat mengisinya dengan alaman URL dari endpoint yang akan memproses form. Secara sederhana, kita akan menyuruh file `login.js` untuk memproses data form. Kode HTML diatas tidak akan menampilkan hasil apa-apa

B. FIELD

Field adalah ruas yang dapat diisi dengan data. Salah satu contoh field yang sering digunakan adalah elemen `<input>`.

```
<input type="text" name="info" />
```

Field memiliki beberapa atribut yang harus diberikan:

- `type` merupakan type dari field.
- `name` merupakan nama dari field yang akan menjadi kunci dan variabel di dalam program.

Dalam elemen `<input>`, terdapat beberapa tipe inputan yang bisa digunakan sesuai dengan kebutuhan aplikasi web. Berikut adalah nama-nama tipe dan kegunaannya.

Tipe Inputan	Kegunaan
<code>type="text"</code>	Mendefinisikan input teks satu baris

type="password"	Mendefinisikan input kata sandi
type="checkbox"	Mendefinisikan checkbox (kotak centang), digunakan ketika user memilih 0 atau lebih dari pilihan yang diberikan
type="submit"	Mendefinisikan tombol untuk mengirimkan data formulir ke form-handler (atribut action)
type="color"	Digunakan untuk kolom input yang harus berisi warna
type="date"	Digunakan untuk kolom input yang harus berisi tanggal (DD/MM/YYYY)
type="email"	Digunakan untuk kolom input yang harus berisi alamat email
type="radio"	Mendefinisikan tombol radio, atau jika user ingin memilih salah satu pilihan
type="reset"	Mendefinisikan tombol reset yang akan mengatur ulang semua nilai formulir ke nilai defaultnya
type="time"	Untuk menginputkan data dengan format waktu (HH:MM)

C. ELEMEN FORM

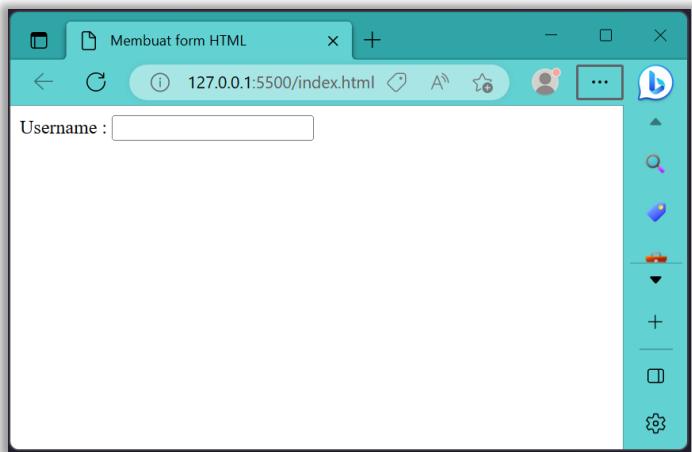
Selain elemen `<input>`, terdapat beberapa elemen pendukung lainnya dalam membuat sebuah form, diantaranya adalah sebagai berikut :

1. Elemen `<label>`

Elemen ini berguna sebagai keterangan pembantu dalam menginput suatu data.

```
<label>Username:</label>
<input type="text" /
```

Hasilnya adalah :



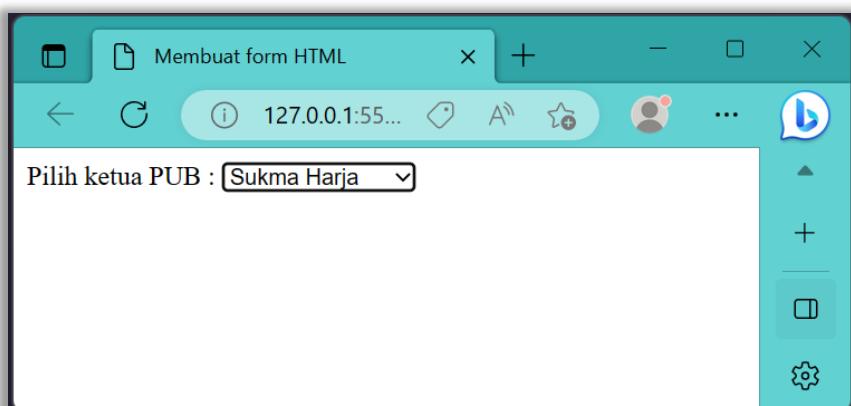
2. Elemen <select>

Elemen ini mendefinisikan daftar drop-down (drop-down list). Dalam membuat drop-down sendiri, harus ada bantuan elemen <option> untuk mendefinisikan opsi yang bisa dipilih oleh users.

Contoh :

```
<body>
    <label for="cars">Pilih ketua PUB : </label>
    <select id="cars" name="cars">
        <option value="volvo">Sukma Harja</option>
        <option value="saab">Imran Sihaloho</option>
        <option value="fiat">Fadly Ariansyah</option>
        <option value="audi">Arya Segara</option>
        <option value="audi">Teguh Ahmadi</option>
    </select>
</body>
```

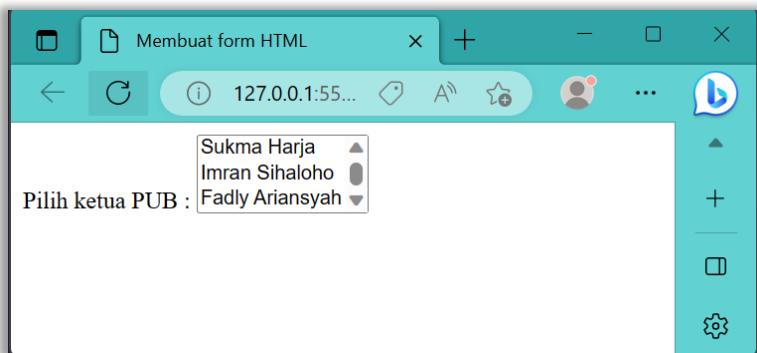
Maka hasilnya adalah :



Untuk melihat jumlah opsi yang bisa dilihat oleh user, maka gunakan atribut **size**

```
<select id="cars" name="cars" size="3">
```

Maka hasilnya adalah :



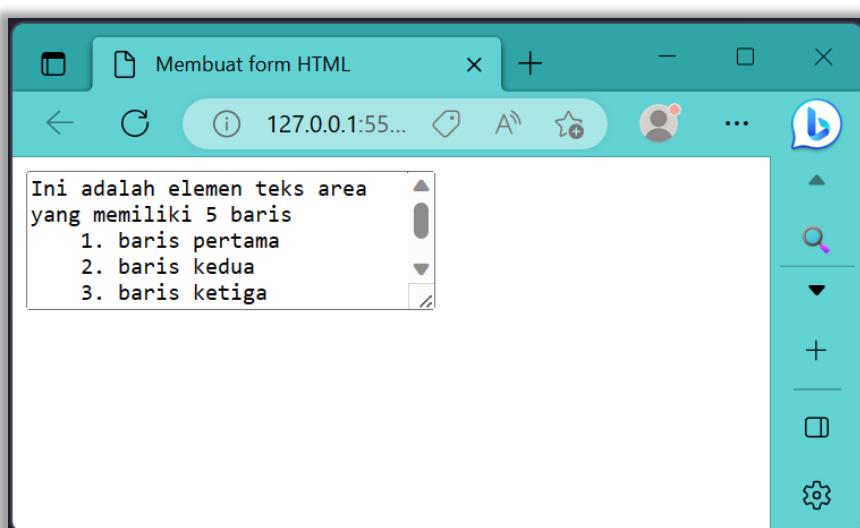
3. Elemen <textarea>

Elemen ini mendefinisikan bidang input lebih dari 1 baris. Biasanya digunakan untuk menginput data yang banyak seperti alamat, komentar, saran, dsbg.

Contoh :

```
<textarea name="message" rows="5" cols="30">Ini adalah elemen teks  
area yang memiliki 10 baris  
1. baris pertama  
2. baris kedua  
3. baris ketiga  
4. baris keempat  
</textarea>
```

Hasilnya adalah :

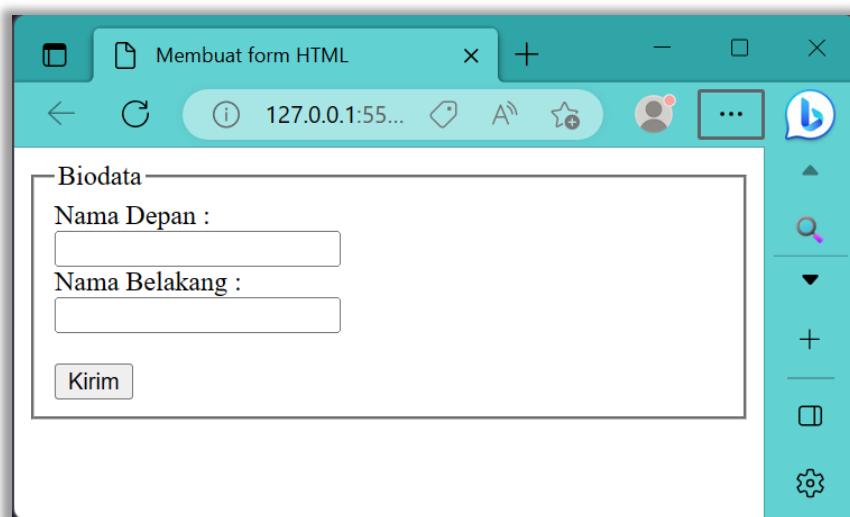


- Atribut **rows** berfungsi untuk menentukan jumlah baris yang terlihat

- Atribut **cols** berfungsi untuk menentukan lebar area teks yang terlihat
4. Elemen **<fieldset>** dan **<legend>**
- Elemen **<fieldset>** digunakan untuk mengelompokkan data yang ada didalam sebuah form sekaligus memberikan sebuah garis. Elemen **<legend>** berfungsi sebagai keterangan untuk **<fieldset>**. Contohnya adalah kode berikut :

```
<form action="proses.js">
  <fieldset>
    <legend>Biodata</legend>
    <label for="fname">Nama Depan :</label><br>
    <input type="text" id="fname" name="fname" ><br>
    <label for="lname">Nama Belakang :</label><br>
    <input type="text" id="lname" name="lname" ><br><br>
    <input type="submit" value="Kirim">
  </fieldset>
</form>
```

Hasilnya adalah :



BAB VIII

TABEL HTML

A. MEMBUAT TABEL

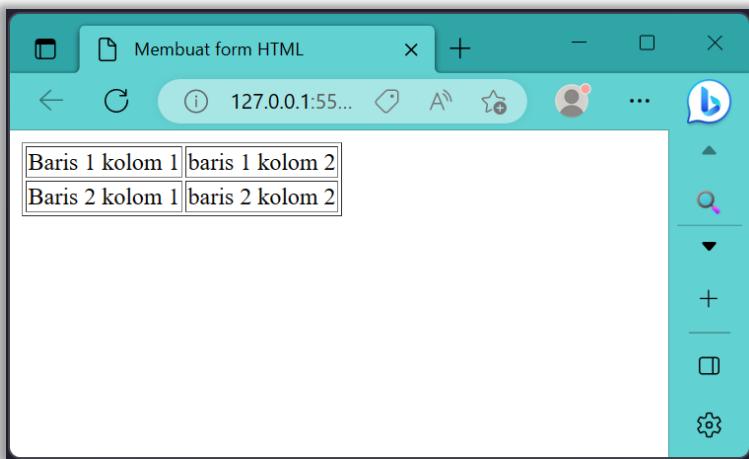
Tabel terdiri dari 3 unsur utama, yaitu baris, kolom, dan sel. Ada beberapa tag yang harus diingat untuk membuat tabel di HTML, yaitu :

Nama Elemen	Kegunaan
<table>	Untuk membungkus tabel
<thead>	Untuk membungkus kepala tabel
<tbody>	untuk membungkus bagian body dari tabel
<tr>	(table row) untuk membuat baris
<td>	(table data) untuk membuat sel
<th>	Untuk membuat judul pada tabel

Tag yang paling penting adalah `<table>`, `<tr>`, dan `<td>`. Sementara tag yang lain adalah tambahan (opsional). Perhatikan kode berikut ini :

```
<table border="1">
  <tr>
    <td>Baris 1 kolom 1</td>
    <td>baris 1 kolom 2</td>
  </tr>
  <tr>
    <td>Baris 2 kolom 1</td>
    <td>baris 2 kolom 2</td>
  </tr>
</table>
```

Hasilnya adalah :



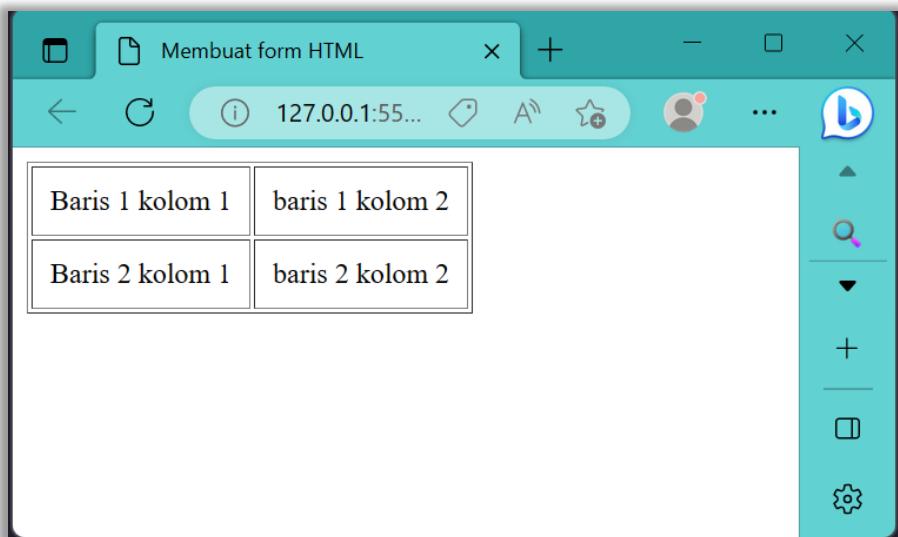
Nilai "1" pada atribut border adalah ukuran garisnya. Semakin besar ukurannya, maka akan semakin besar pula ukuran garisnya. Nilai "1" adalah ukuran garis yang paling kecil.

B. MENGATUR JARAK DENGAN CELLPADDING

Atribut ***cellpadding*** adalah atribut untuk mengatur jarak teks dengan garis di dalam sel. Atribut ini dapat kita berikan kepada tag **<table>**. Contohnya adalah :

```
<table border="1" cellpadding="10">
    <tr>
        <td>Baris 1 kolom 1</td>
        <td>baris 1 kolom 2</td>
    </tr>
    <tr>
        <td>Baris 2 kolom 1</td>
        <td>baris 2 kolom 2</td>
    </tr>
</table>
```

Nilai "10" pada atribut ***cellpadding*** adalah ukuran jarak antara teks sel dengan garis. Maka hasilnya adalah :



C. MENGGABUNGKAN SEL TABEL

Atribut yang digunakan untuk menggabungkan sel tabel adalah :

- **rowspan** untuk menggabungkan baris
- **colspan** untuk menggabungkan kolom

Atribut ini bisa digunakan untuk tag **<td>** atau **<th>**.

	colspan = "2"

Berikut adalah contoh kodennya :

```
<table border="1">
  <tr>
    <th rowspan="2" bgcolor="yellow">Bulan</th>
    <th colspan="2" bgcolor="#00ff80">Hasil Panen</th>
  </tr>
  <tr>
    <th>Padi</th>
```

```

<th>Kacang</th>
</tr>
<tr>
    <td>Januari</td>
    <td>500 Kg</td>
    <td>231 Kg</td>
</tr>
<tr>
    <td>Februari</td>
    <td>342 Kg</td>
    <td>423 Kg</td>
</tr>
<tr>
    <td>Maret</td>
    <td>432 Kg</td>
    <td>124 Kg</td>
</tr>
<tr>
    <td>April</td>
    <td>453 Kg</td>
    <td>523 Kg</td>
</tr>
</table>

```

Hasilnya adalah :

Bulan	Hasil Panen	
	Padi	Kacang
Januari	500 Kg	231 Kg
Februari	342 Kg	423 Kg
Maret	432 Kg	124 Kg
April	453 Kg	523 Kg

D. MENYISIPKAN ELEMEN LAIN KE DALAM SEL

Di dalam sel `<td>` dan `<th>`, kita bisa menyisipkan elemen HTML yang lain, seperti gambar, link, video, list, dsb. Berikut adalah contoh kodenya :

```

<table border="1">
  <tr>
    <th colspan="3" bgcolor="yellow">Keindahan Alam Kota Banjar</th>
  </tr>
  <tr>
    <td rowspan="4">
      
    </td>
  </tr>
  <tr>
    <td>Nama Tempat</td>
    <td>Gunung Fuji</td>
  </tr>
  <tr>
    <td>Harga Tiket</td>
    <td>Gratisss</td>
  </tr>
  <tr>
    <td>Fasilitas</td>
    <td>
      <ul>
        <li>Wahana bermain anak</li>
        <li>Kolam air panas</li>
        <li>Warung angkringan</li>
        <li>Toilet umum</li>
      </ul>
    </td>
  </tr>
</table>

```

Hasilnya adalah :



The screenshot shows a web browser window titled "Membuat form HTML". The address bar displays the URL "127.0.0.1:5...". The main content area contains a table with the following structure:

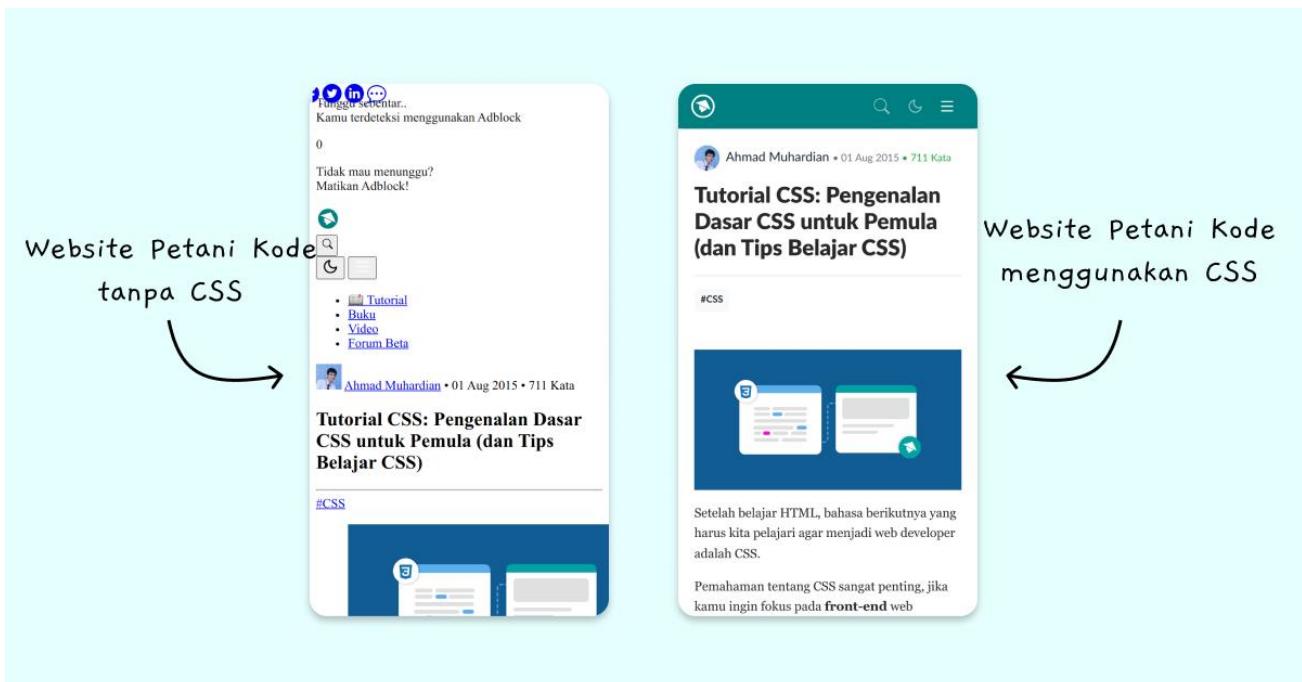
Keindahan Alam Kota Banjar		
	Nama Tempat	Gunung Fuji
	Harga Tiket	Gratisss
	Fasilitas	<ul style="list-style-type: none"> • Wahana bermain anak • Kolam air panas • Warung angkringan • Toilet umum

BAB IX

MENGENAL CSS

A. APA ITU CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.



Jika tanpa menggunakan CSS tampilan dari sebuah website akan terlihat jelek dan tidak rata. Sedangkan jika menggunakan CSS tampilannya akan jadi lebih bagus dan rapi.

B. SEJARAH CSS

Sebelum adanya CSS, tidak ada bahasa yang dipakai untuk memberikan style pada dokumen (web). Tampilan web terasa hambar dan kurang tertata. Ide awal tentang style sheet di browser bukan sesuatu yang baru. Pada tahun 1990, Tim Berners-Lee membuat web browser bernama NeXT browser/editor. Pada browser ini, kita bisa membuat style sheet. Akan tetapi, sintaks style sheet-nya tidak ia publikasikan.

Sejak awal rilis, CSS memiliki beberapa versi :

- CSS 1 (1996) : adalah versi pertama (17 Desember 1996)
- CSS 2 (1998) : adalah versi ke-2 (Mei 1998)
- CSS 2.1 (2011) : dirilis pada 7 Juni 2011
- CSS 3 (2012) : versi yang banyak digunakan saat ini.

C. CARA MENULIS CSS

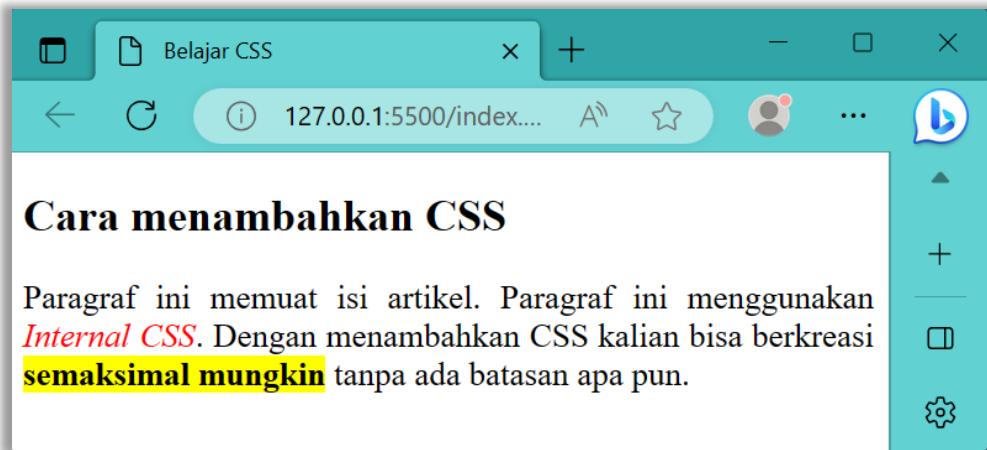
Ada 3 cara menulis CSS di HTML:

1. Internal CSS

Internal CSS adalah kode CSS yang ditulis di dalam tag `<style>`. Internal CSS juga dikenal dengan sebutan Embeded CSS. Tag `<style>` bisa ditulis di dalam tag `<head>`, bisa juga ditulis di dalam tag `<body>`. Namun kebanyakan orang menulisnya di dalam `<head>`. Cobalah untuk menulis kode berikut :

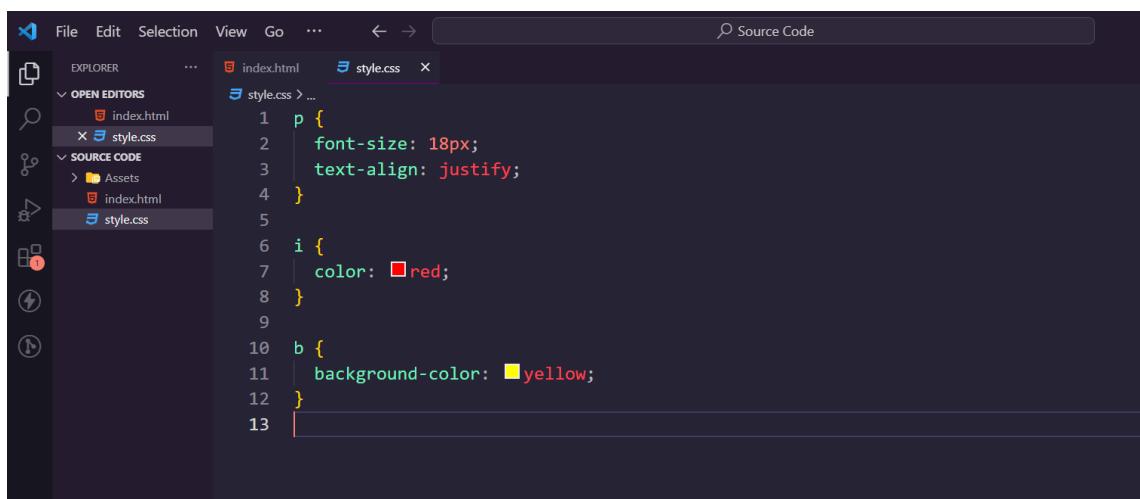
```
<!DOCTYPE html>
<html Lang="en">
<head>
    <title>Belajar CSS</title>
    <style>
        p {
            font-size: 18px;
            text-align: justify;
        }
        i {
            color: red;
        }
        b {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <h2>Cara menambahkan CSS</h2>
    <p>
        Paragraf ini memuat isi artikel. Paragraf ini menggunakan
        <i>Internal CSS</i>. Dengan menambahkan CSS kalian bisa
        berkreasi <b>seaksimal mungkin</b> tanpa ada batasan apa
        pun.
    </p>
</body>
</html>
```

Maka hasilnya adalah :



2. Eksternal CSS

Eksternal CSS adalah kode CSS yang ditulis terpisah dengan kode HTML. Eksternal CSS ditulis di dalam file khusus yang berekstensi **.css**. Pertama kita buat dulu file dengan nama **style.css** kemudian isi style yang akan kita pakai. Atau untuk lebih mudah kita pakai contoh yang tadi dengan cara memindahkan isi dari tag **<style>** ke dalam file **style.css**



Kemudian untuk menggunakan CSS ini kita harus menggunakan HTML. Ada 2 cara yang bisa dilakukan yaitu menggunakan tag **<link>** atau menggunakan **@import**.

- Jika menggunakan tag **<link>** maka buatlah kode berikut di dalam tag **<head>**, kemudian isi atribut **href** dengan file CSS yang tadi kita buat

```
<head>  
    <title>Belajar CSS</title>  
    <link rel="stylesheet" href="style.css">  
</head>
```

- Jika menggunakan `@import` maka buatlah kode berikut

```
<style>
    @import url(style.css);
</style>
```

Jika sudah melakukan salah satu cara diatas, maka output nya akan sama seperti kita menambahkan Internal CSS.

3. Inline CSS

Inline CSS adalah kode CSS yang ditulis langsung pada atribut `style` di elemen HTML. Kode CSS ditulis langsung tanpa menggunakan kurung kurawal { ... }. Kemudian properti tetap dipisah dengan titik koma. Contohnya adalah kode berikut:

```
<h2 style="color: aqua; font-family: 'Times New Roman';">
    Cara menambahkan CSS
</h2>
```

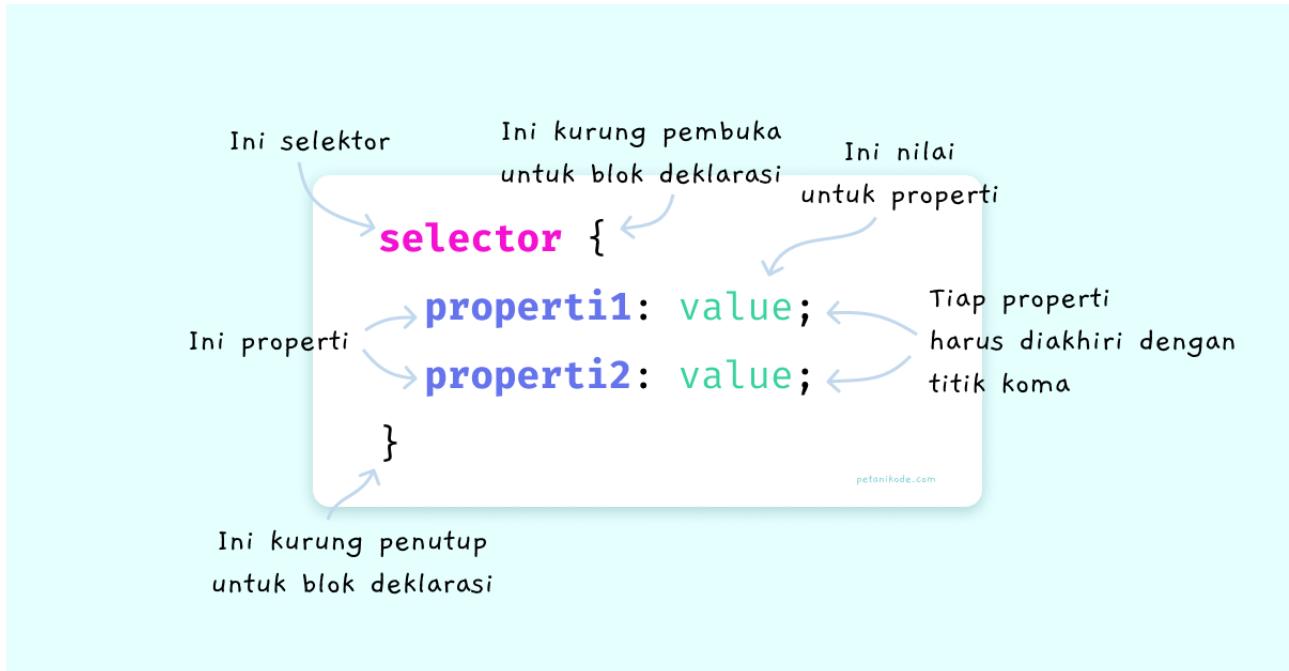
Dari 3 cara tersebut, yang paling disarankan adalah eksternal CSS, karena penggunaannya lebih fleksibel dan cocok digunakan untuk projek yang cukup kompleks.

D. SINTAKS DASAR CSS

Struktur kode CSS terdiri dari 3 bagian, yaitu :

- Selektor
- Blok dekorasi
- Properti dan nilainya (value)

Format penulisannya adalah sebagai berikut :



1. Selektor

Selektor adalah kata kunci untuk memilih elemen HTML yang akan diberi style. Contohnya

```

h1 {
    background-color: yellow;
}

```

Artinya kita memilih semua elemen `<h1>`, lalu diberikan warna teks kuning. Selektor dapat berupa nama tag, class, id, dan atribut.

```

/* selektor dengan nama tag */
h2 {
    color: aqua;
}

/* selektor dengan nama class */
.pr-1 {
    background-color: red;
}

/* selektor dengan nama id */
#contoh-id {
    text-decoration: underline;
}

```

```
/* selektor dengan atribut */  
ol[type="1"]{  
    background-color: violet;  
}
```

2. Blok Dekorasi

Blok deklarasi adalah tempat kita menuliskan atribut atau properti CSS yang akan diberikan ke pada elemen yang telah diseleksi. Perhatikan contoh berikut:

```
h1 {  
    background-color: yellow;  
}
```

Artinya kita mengatur warna latar belakang dari tag **<h2>** dengan warna kuning. Blok dekorasi dimulai atau dibuka dengan tanda kurung **{** lalu ditutup dengan **}**. Satu blok dekorasi, bisa dipakai oleh lebih dari satu selektor. Contohnya

```
h1,  
h2,  
h3,  
h4,  
h5,  
h6 {  
    color: teal;  
}
```

Ini artinya kita memberikan warna yang sama untuk tag **<h1>** sampai **<h6>**. Selektor lain juga bisa menggunakan format yang sama, contohnya adalah sebagai berikut

```
h2,  
.pr-1,  
#contoh-id,  
ol[type="1"] {  
    color: aqua;  
}
```

3. Properti dan Nilainya (Value)

Properti adalah sekumpulan aturan yang akan diberikan kepada elemen yang dipilih.

Format penulisan properti seperti ini:

```
properti: value;
```

Setiap properti harus diakhiri dengan titik koma (;). Apabila hanya terdapat satu properti, boleh tidak menggunakan titik koma. Properti harus ditulis di dalam blok deklarasi. Nilai untuk properti kadang diapit tanda petik, kadang juga tidak. Biasanya yang diapit dengan tanda petik adalah nilai yang berupa teks. Contohnya adalah :

```
font-family: 'Times New Roman', Georgia, Roboto;
```

Value dari contoh diatas menggunakan tanda petik karena ada spasi di nama **Times new Roman**. Nilai properti juga bisa diisi lebih dari satu, ini biasanya diberikan untuk value opsional. Misalnya font **Times New Roman** tidak ada, maka font yang akan dipakai adalah **Georgia**, begitu juga seterusnya.

E. MEMAHAMI 6 JENIS SELEKTOR

1. Selektor Tag (Type Selector)

Selektor ini akan memilih elemen berdasarkan nama tag. Contoh

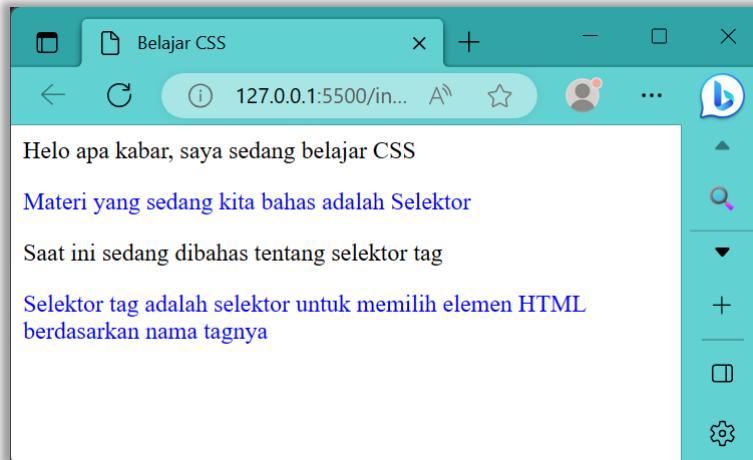
```
p {  
    color: blue;  
}
```

Artinya : Pilih semua elemen **<p>** lalu atur warna teksnya menjadi biru. Misalnya kita memiliki struktur HTML seperti di bawah ini.

```
<div>Helo apa kabar, saya sedang belajar CSS</div>  
<p>Materi yang sedang kita bahas adalah Selektor</p>  
<div>Saat ini sedang dibahas tentang selektor tag</div>  
<p>Selektor tag adalah selektor untuk memilih elemen HTML
```

```
berdasarkan nama tagnya  
</p>
```

Maka akan menghasilkan output seperti di bawah ini.



Semua elemen `<p>` berwarna biru, sedangkan elemen `<div>` tidak

2. Selektor Class

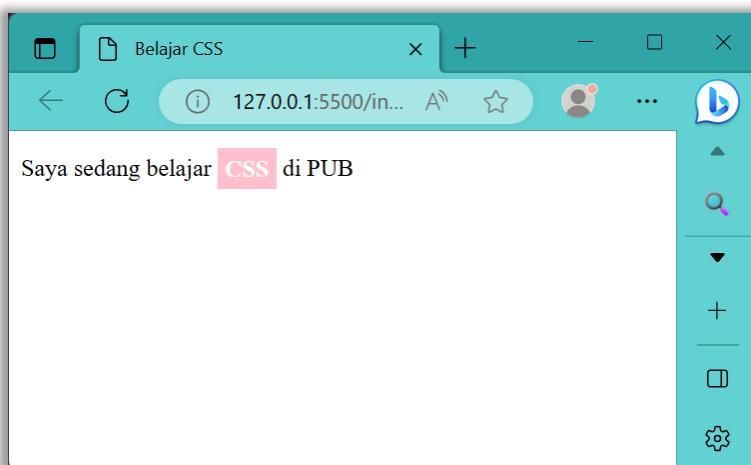
Selektor class adalah selektor yang memilih elemen berdasarkan nama class yang diberikan. Selektor class dibuat dengan tanda titik di depannya. Contoh :

```
.container {  
    color: white;  
    background: pink;  
    padding: 5px;  
}
```

Kita memiliki selektor class bernama `.container`. Nah cara menggunakan selektor ini di HTML adalah dengan menambahkan atribut class di dalamnya. Contoh

```
<p>Saya sedang belajar <b class="container">CSS</b> di PUB</p>
```

Maka hasilnya adalah :



3. Selektor ID

Selektor ID hampir sama dengan class. Bedanya, ID bersifat unik. Hanya boleh digunakan oleh satu elemen saja. Selektor ID ditandai dengan tanda pagar '#' di depannya. Contoh

```
#header {  
    background: teal;  
    color: white;  
    height: 100px;  
    padding: 50px;  
}
```

Lalu pada konten HTML:

```
<header id="header">  
    <h1>Selamat Datang di Website Saya</h1>  
</header>
```



4. Selektor Atribut

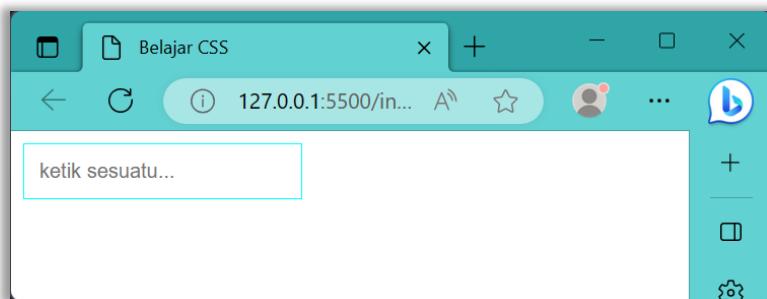
Selektor atribut adalah selektor yang memiliki elemen berdasarkan atribut. Selektor ini hampir sama seperti selektor Tag. Contoh

```
input[type=text] {  
    background: none;  
    color: cyan;  
    padding: 10px;  
    border: 1px solid cyan;  
}
```

Artinya kita akan memilih semua elemen `<input>` yang memiliki atribut `type='text'`. Contoh kode HTML:

```
<input type="text" placeholder="ketik sesuatu..." />
```

Maka hasilnya adalah :



5. Pseudo Selektor

Pseudo selektor adalah selektor untuk memilih elemen semu seperti state pada elemen, elemen before dan after, elemen ganjil, dan sebagainya. Ada 2 macam pseudo selektor, yaitu :

a. Pseudo Class

Pseudo-class adalah selektor untuk memilih state pada elemen. Contohnya seperti elemen saat diklik, saat fokus, saat disentuh, dan lain sebagainya. Dengan selektor ini kita bisa menentukan tampilan dari elemen pada state tersebut. Format penulisan pseduoclass menggunakan titik dua setelah elemen. Contoh :

```
a:hover {  
    color: green;  
}
```

Kemudian kita buat kode HTML seperti berikut dan rasakan keajaibannya:

```
<a href="#">Ini adalah contoh Pseudo Class</a>
```

Selain :hover ada beberapa selektor pseudo-class lainnya :

- **:link** : untuk memilih link yang belum dikunjungi
- **:visited** : untuk memilih link yang sudah dikunjungi
- **:active** : untuk memilih elemen yang sedang aktif, seperti saat di klik
- **:focus** : untuk memilih elemen yang sedang dalam keadaan fokus, misalnya pada saat teks sedang di input
- **:checked** : untuk memilih elemen yang dicentang pada checkbox

Masih banyak lagi, list lengkapnya bisa kalian kunjungi di <https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

b. Pseudo-element

Pseudo-element adalah selektor untuk memilih elemen semu. Elemen semu yang dimaksud di sini adalah elemen yang seolah-olah kita tambahkan di HTML. Perhatikan contoh berikut !

```
<p>ini adalah paragraf yang cukup panjang. Sengaja dibuat  
panjang agar bisa dipakai sebagai contoh untuk selector  
pseudo-elemen. Pseudo-element adalah selektor untuk memilih  
elemen semu. Elemen semu yang saya maksud di sini adalah  
elemen yang seolah-olah kita tambahkan di HTML.  
</p>
```

Lalu kita ingin memberikan style hanya pada baris pertama saja. Gimana caranya? Tanpa pseudo-element, kita bisa tambahkan elemen di baris pertama menjadi seperti ini.

```
<p><span>ini adalah paragraf yang cukup panjang. <span>  
Sengaja dibuat panjang agar bisa dipakai sebagai contoh
```

```
untuk selector pseudo-elemen. Pseudo-element adalah  
selektor untuk memilih elemen semu. Elemen semu yang saya  
maksud di sini adalah  
elemen yang seolah-olah kita tambahkan di HTML.  
</p>
```

Kemudian memilih elemen span tersebut di CSS seperti ini:

```
p span {  
    color: magenta;  
}
```

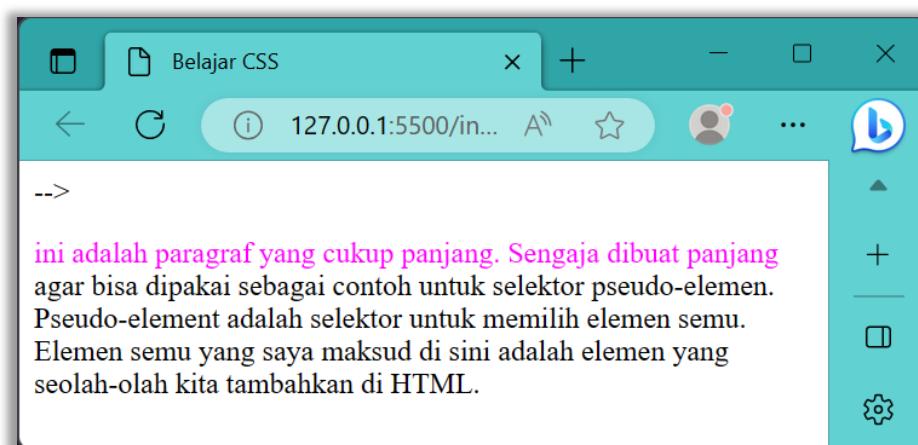
Cara ini sebenarnya tidak disarankan, karena kita harus membuat elemen baru di dalam paragraf. Belum lagi, jika paragrafnya cukup panjang baris pertama akan tambah panjang pula.

Maka di sinilah kita bisa pakai selektor pseudo-element `::first-line`

Sehingga tanpa menambahkan elemen ``, kode CSS-nya akan menjadi seperti ini:

```
p::first-line {  
    color: magenta;  
}
```

Maka hasilnya akan seperti berikut



Perbedaannya dengan pseudo-class adalah Selektor pseudo-element menggunakan tanda titik dua ganda (`::`) sedangkan pseudo-class pakai satu titik dua (`:`).

BAB X

PROPERTI YANG SERING DIPAKAI

A. COLOR

Properti **color** merupakan properti untuk memberikan warna pada teks. Nilai warna pada properti color dapat kita berikan dengan berbagai cara. Salah satunya dengan menuliskan nama warna dalam bahasa Inggris. Misalnya, warna teks untuk paragraf adalah merah, maka kita bisa berikan **color:red**. Selain menggunakan nama warna, terdapat juga cara lain yang valid seperti fungsi **rgb()**, kode heksa, dsb.

1. Nama Warna

Pastikan kamu menulis nama warnanya dengan bahasa Inggris. Jika pakai bahasa Indonesia, maka tidak akan bisa. Contoh :

```
header {  
    color: pink;  
}
```

Ini adalah sebagian dari nama-nama warna yang ada di CSS. Kalau ingin melihat daftar lengkap nama-nama warna di CSS, kamu bisa kunjungi : <https://www.w3.org/wiki/CSS/Properties/color/keywords>

Nama-nama Warna di CSS

petanikode.com



2. Kode Heksadesimal

Kode heksadesimal merupakan kode yang menggunakan bilangan heksadesimal, memiliki nilai dari **0** sampai **f** (16 dalam desimal). Kode heksadesimal diawali dengan

tanda pagar (#), kemudian diikuti kode heksanya. Kode ini dapat terdiri dari 3 digit atau 6 digit bilangan. Contoh :

```
header {  
    color: #ff2200;  
}
```

Setiap angka dalam kode heksadesimal menentukan nilai warna dalam RGB (#RRGGBB atau #RGB). Misalnya untuk yang enam digit, dua angka di depan adalah untuk nilai warna merah, dua angka berikutnya untuk warna hijau, dan dua angka terakhir untuk warna biru. Nilai yang diberikan pada parameter tersebut dimulai dari angka 0~255. Semakin besar nilai yang diberikan, semakin kuat nilai warna tersebut

3. RGB dan RGBA

Fungsi **rgb()** adalah fungsi yang menghasilkan warna berdasarkan konsep warna RGB pada komputer. RGB merupakan singkatan dari Red, Green, Blue.



Contoh penggunaan fungsi **rgb()**

```
h1 {  
    color: rgb(255, 128, 255)  
}
```

Pada contoh diatas kita akan mendapatkan warna pink karena kita memberikan komposisi warna :

- Hijau : 255
- Merah : 128

- Biru : 255

Nilai parameter ini diberikan dari rentang angka 0 sampai 255. Semakin besar angkanya, semakin banyak kandungan komposisi dari warna tersebut.

Selain menggunakan fungsi `rgb()` ada juga fungsi `rgba()`. Fungsi ini sama seperti fungsi RGB, tapi ada penambahan parameter *alpha* di belakangnya.



Nilai *alpha* dapat diberikan dari 0% sampai 100% atau 0.0 ~ 1.0. Semakin mendekati angka satu, warnanya akan semakin terang dan semakin mendekati angka nol, warnanya akan semakin transparan. Contoh penggunaan fungsi RGBA.

```

p {
  color: rgba(40, 23, 11, 0.7);
}
  
```

4. HSL dan HSLA

Fungsi `hsl()` merupakan fungsi yang menghasilkan warna berdasarkan konsep warna HSL. HSL merupakan singkatan dari Hue, Saturation, Lightness.



*harusnya `hsl(hue, saturation, lightness);`

Nilai hue yang diberikan pada fungsi ini dimulai dari `0 ~ 360`, karena model warna HSL bentuknya seperti tabung, sehingga yang digunakan adalah nilai derajat dari lingkaran tutup tabungnya. Sedangkan untuk nilai saturation dan lightness menggunakan persen, dari `0% ~ 100%`. Contoh penggunaannya adalah :

```
p {  
    color: hsl(324, 70%, 40%);  
}
```

Fungsi `hsl()` juga punya fungsi `hsla()` untuk memberikan nilai *alpha* atau transparan pada warna HSL. Contohnya adalah :

```
h1 {  
    color: hsla(267, 80%, 45%, 0.6);  
}
```

5. Gradasi

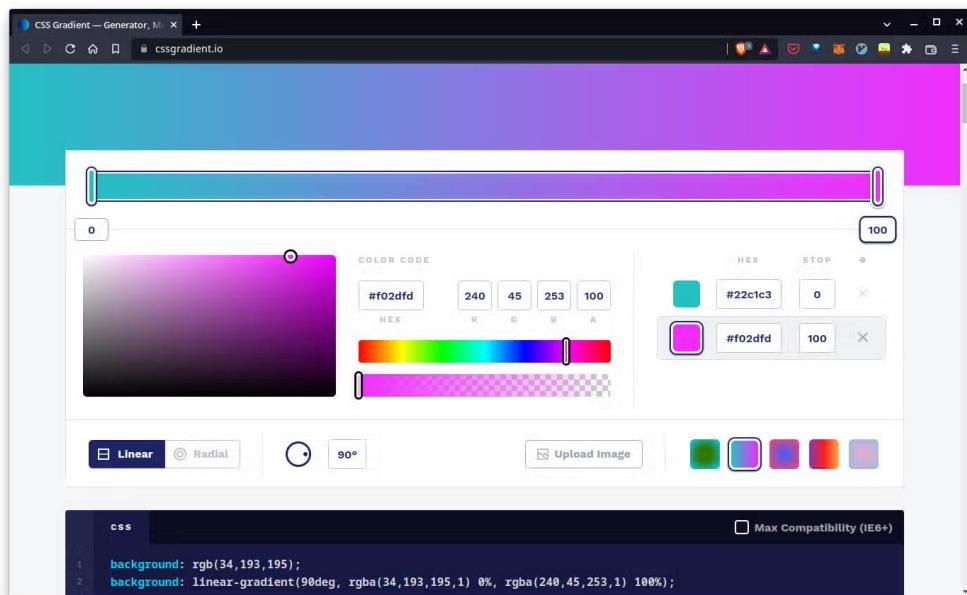
Warna gradasi adalah kumpulan dari beberapa warna yang digabungkan dengan membentuk transisi gradasi warna. Warna gradasi di CSS biasanya dibuat dengan fungsi:

- `linear-gradient()` untuk gradasi linear;
- `radial-gradient()` untuk gradasi lingkaran;
- `conic-gradient()` untuk gradasi memutar;

Tiap-tiap fungsi punya parameter yang hampir sama. Bedanya pada parameter awal saja. Fungsi `linear-gradient()` akan menggunakan parameter arah di awalnya, lalu diikuti dengan warna-warna yang akan dicampurkan. Parameter arah pada `linear-gradient()` bisa berbentuk nama arah seperti `left, right, top, bottom`, dan juga derajat seperti `90deg`.



Fungsi `radial-gradient()`, parameter awalnya berupa ukuran lingkaran dan fungsi `conic-gradient()` parameter awalnya berupa derajat lingkaran. Nah, untuk membuat warna gradasi dengan mudah.. kita bisa memanfaatkan website cssgradient.io.



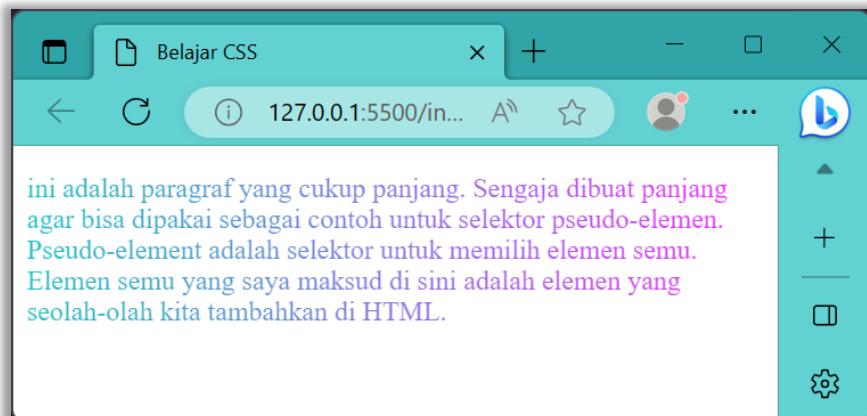
Setelah itu, kita akan dapat kode CSS untuk Gradasinya. Tinggal di-copas dan dipakai aja. Contoh :

```

p {
    background: linear-gradient(90deg, rgba(34, 193, 195, 1) 0%, 
    rgba(240, 45, 253, 1) 100%);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}

```

Properti color saat ini belum mendukung untuk menggunakan warna gradasi. Tapi kita bisa pakai properti background seperti di atas untuk menggunakan warna gradasi pada teks sehingga hasilnya akan seperti ini



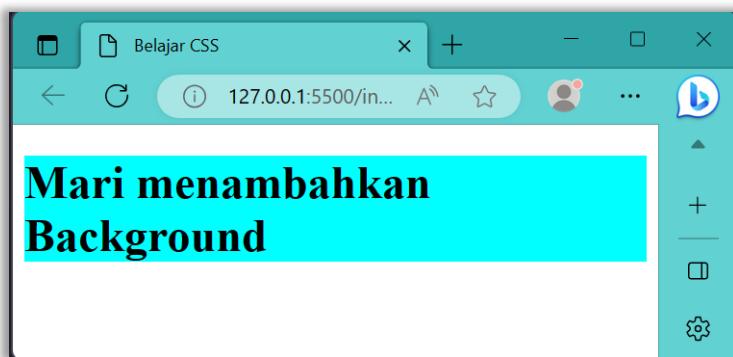
B. BACKGROUND

1. Background Warna

Untuk memberikan background warna, kita bisa memberikan nilai warna pada properti background seperti ranam warna, kode heksa warna, fungsi `rgb()`, `rgba()`, `hsl()`, dan sebagainya. Contoh :

```
header {  
    background-color: aqua;  
}
```

Maka hasilnya adalah :

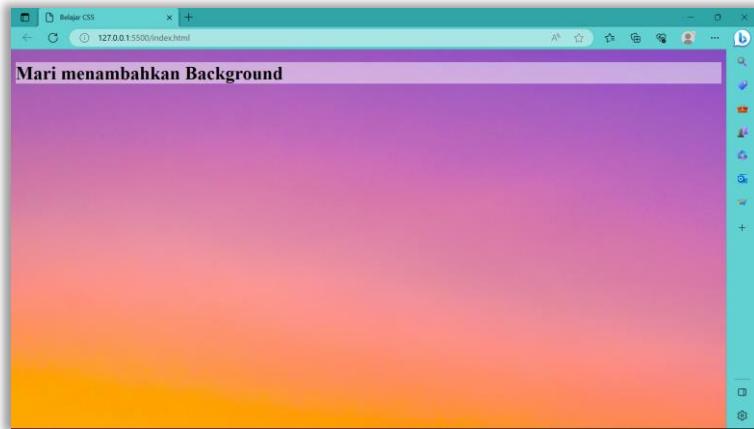


2. Background Gambar

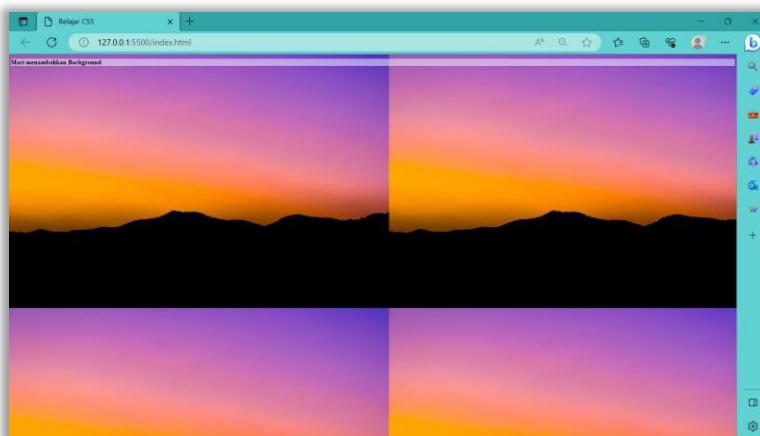
Jika ingin menggunakan warna latar gambar, maka kita bisa menggunakan properti `background-image` atau `background` saja. Format gambar yang didukung bisa: `jpg`, `png`, `gif`, `webp`, `avif`, `svg`, dll. Sebagai contoh salinlah kode berikut ini.

```
header {  
    background: rgba(255, 255, 255, 0.5);  
}  
  
body {  
    background-image: url(Assets/senja.jpg);  
}
```

Pada kode ini, kita memberikan background gambar untuk elemen **<body>** dan memberikan background putih transparan (50%) untuk elemen **<header>**. Maka hasilnya adalah :



Coba perbesar dan perkecil ukuran jendelanya. atau coba juga melakukan zoom out. Jika kita melakukan zoom out dari gambar tersebut maka akan menghasilkan gambar dibawah ini.



Gambar background-nya akan diulang-ulang. Jika kamu pakai ukuran gambar yang kecil, maka akan ditampilkan seperti itu. Jika kita tidak ingin gambar tersebut

berulang-ulang maka kita bisa gunakan properti **background-repeat**. Properti ini punya beberapa nilai:

- **repeat-x** artinya mengulang background pada sumbu **x** saja;
- **repeat-y** artinya mengulang background pada sumbu **y** saja;
- **no-repeat** artinya tidak mengulang background.

kita bisa pakai **no-repeat** bila ingin background-nya tidak diulang. Maka kode CSS-nya akan menjadi seperti ini:

```
body {  
    background-image: url(Assets/senja.jpg);  
    background-repeat: no-repeat;  
}
```

a. Menentukan Ukuran Background

Kita bisa mengatur ukuran background dengan properti **background-size**, properti ini punya beberapa nilai yang valid:

- **px, pt, em, rem** (fixed) ukuran dalam angka dan satuan, contoh **200px, 20em**.
- **vw, vh** (dynamic) mengikuti ukuran tinggi dan lebar layar (view port);
- **contain** (dynamic) mengikuti ukuran lebar elemen;
- **cover** (dynamic) mengikuti ukuran lebar dan tinggi elemen;

Mari kita gunakan contoh sebelumnya. Ubahlah kode CSS background pada contoh sebelumnya menjadi seperti ini dan rasakan keajaibannya.

```
body {  
    background-image: url(Assets/senja.jpg);  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

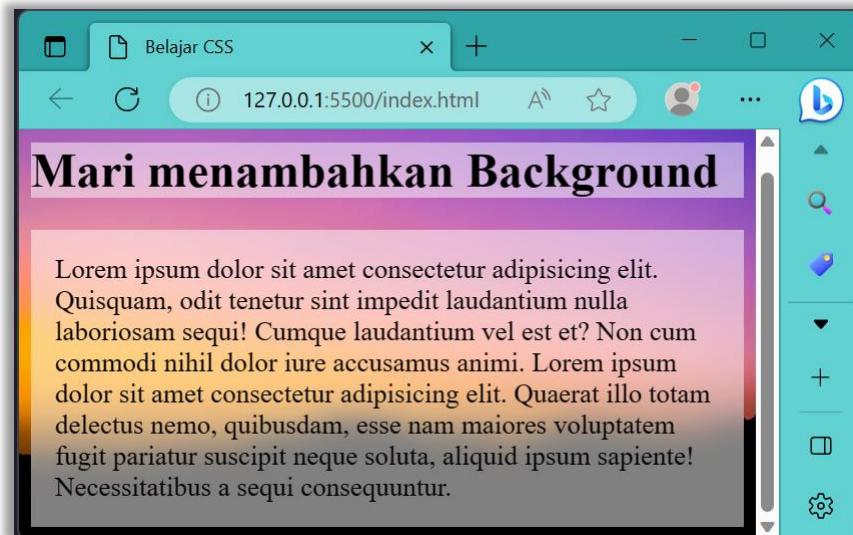
b. Menggunakan Efek Blur

Konsep blur dikenal dengan *glass morphology design*, karena bentuknya mirip seperti glass. Nah untuk membuat yang seperti ini, ada properti css baru bernama **backdrop-filter** untuk memberikan filter pada background. Jika

ingin memberikan filter blur, maka kita harus memberikan nilai dengan fungsi `blur()`. Kita gunakan contoh kode yang tadi kemudian tambahkan elemen `<article>` dan isi kontennya serta tambahkan CSS seperti kode berikut.

```
article {  
    background-color: rgba(255, 255, 255, 0.5);  
    backdrop-filter: blur(10px);  
    padding: 1rem;  
}
```

Pada contoh ini, kita memberikan efek filter blur (10px) untuk background elemen `<article>`, maka hasilnya adalah :



Selain efek `blur()`, masih ada lagi efek lainnya seperti:

- `brightness()` untuk efek kecerahan;
- `contrast()` untuk efek kontras;
- `drop-shadow()` untuk efek bayangan;
- `grayscale()` untuk efek hitam putih;
- `hue-rotate()` untuk efek warna hue;
- `invert()` untuk efek warna negatif;
- `opacity()` untuk efek transparan;
- `saturate()` untuk efek sturasi warna;
- `sepia()` untuk efek warna sepia.

C. BORDER

Properti yang digunakan untuk membuat garis pada CSS adalah **border**. Properti ini memiliki 3 nilai yang harus diberikan:

- Ukuran Garis
- Jenis Garis
- Warna Garis

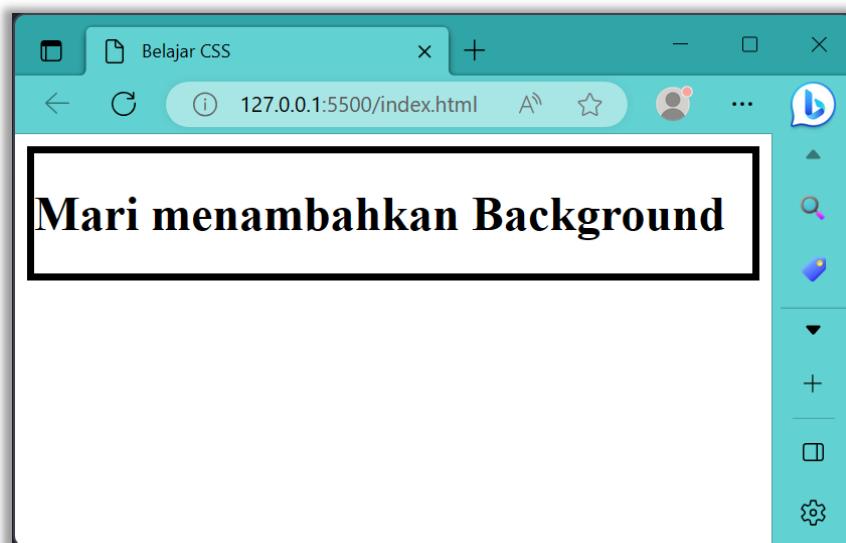
petanikode.com



Contoh :

```
header {  
    border: 5px solid black;  
}
```

Elemen `<header>` yang kita seleksi akan diberikan garis dengan ketebalan `5px` dan warna hitam. Maka hasilnya adalah :



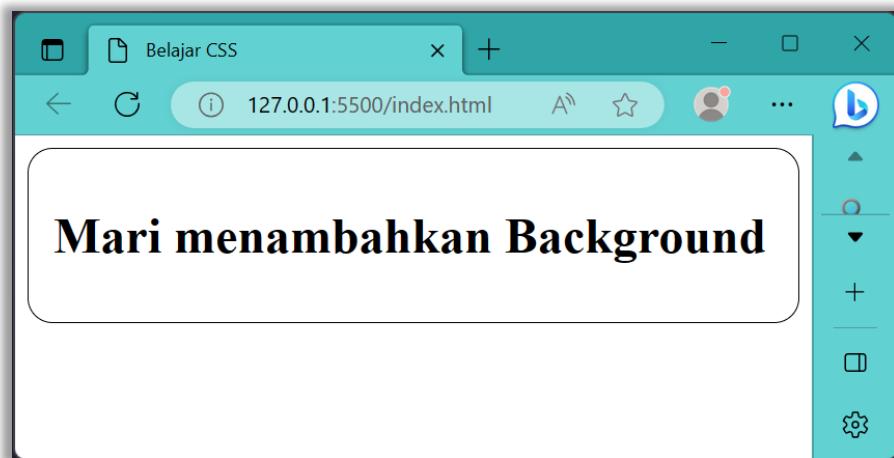
Jenis garis di CSS bisa kita ganti. Kita bisa pakai properti **border-style** untuk mengganti jenis garis. Berikut ini jenis garis yang biasanya digunakan:

- **solid** garis biasa
- **dotted** garis dengan titik putus-putus
- **dashed** garis putus-putus dengan dash
- **double** garis double
- **none** tanpa garis
- **hidden** garis tersembunyi

Jika kita ingin membuat garis tumpul (Rounded Corner) kita bisa menggunakan properti **border-radius**. Kita bisa memberikan nilai border-radius dengan satuan **px**, **%**, **em**, dll. Contoh :

```
header {  
    border: 1px solid black;  
    border-radius: 16px;  
    padding: 1rem;  
}
```

Maka hasilnya adalah sebagai berikut.



BAB XI

MARGIN & PADDING

A. MARGIN

Margin digunakan untuk membuat ruang di sekitar elemen, di luar batas yang ditentukan. Ada properti untuk mengatur margin untuk setiap sisi elemen (atas, kanan, bawah, dan kiri) yaitu :

- **margin-top** untuk mengatur bagian margin atas
- **margin-right** untuk mengatur margin bagian kanan
- **margin-bottom** untuk mengatur margin bagian bawah
- **margin-left** untuk mengatur margin bagian kiri

Semua properti margin dapat memiliki nilai berikut:

- **auto** - browser menghitung margin
- **length** - menentukan margin dalam **px**, **pt**, **cm**, dll.
- **%** - menentukan margin dalam % dari lebar elemen yang memuatnya
- **inherit** - menentukan bahwa margin harus diwarisi dari elemen induk

Sebagai contoh buatlah kode HTML seperti di bawah ini:

```
<body>
  <h1>Contoh Penggunaan Margin</h1>
  <p>
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Rem
    recusandae atque modi aliquam cumque. Quasi enim
    exercitationem officiis, quod ab quia itaque excepturi. Itaque aut
    molestiae praesentium quam ipsam nulla!
  </p>
</body>
```

Kemudian buat kode CSS berikut :

```
h1 {
  background-color: lightblue;
}

p {
  background-color: green;
  color: white;
  margin-top: 20px;
```

```
margin-bottom: 10px;  
margin-right: 15px;  
margin-left: 30px;  
}
```

Maka hasilnya adalah :



Properti margin dapat ditentukan menggunakan satu, dua, tiga, atau empat nilai. Setiap nilai adalah <panjang>, <persentase>, atau kata kunci otomatis. Jika kita memberi nilai negatif, maka menarik elemen lebih dekat ke tetangganya daripada secara default.

- Jika hanya memberi satu nilai, maka akan menerapkan margin yang sama di setiap sisinya. Contohnya

```
margin: 100px;
```

- Jika memberi dua nilai, maka nilai yang pertama akan diterapkan pada **margin-top** dan **margin-bottom**, sedangkan nilai yang kedua akan diterapkan pada **margin-left** dan **margin-right**. Contoh :

```
margin: 100px 150px;
```

- Jika memberi 3 nilai, maka nilai yang pertama akan diterapkan pada **margin-top**, nilai yang kedua akan diterapkan pada **margin-left** dan **margin-right**, sedangkan margin yang ketiga akan diterapkan pada **margin-bottom**. Contoh :

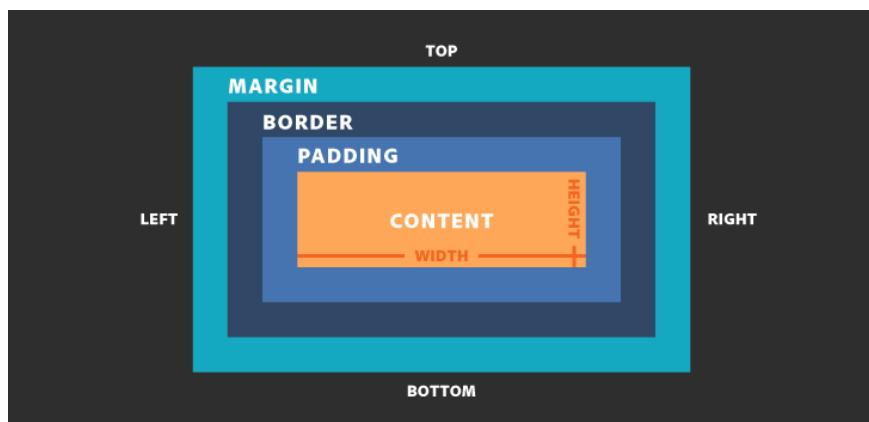
```
margin: 100px 150px 100px;
```

- Jika memberi 4 nilai, maka setiap margin diberi nilai yang ditentukan sesuai arah jarum jam (top, right, bottom, left). Contoh :

```
margin: 100px 150px 100px 200px;
```

B. PADDING

Penggunaan **padding** hampir sama dengan kita menggunakan **margin**. Namun perbedaanya adalah jika **margin** memberi ruang di luar elemen, sedangkan **padding** memberikan ruang di dalam elemen. Perhatikan ilustrasi berikut :



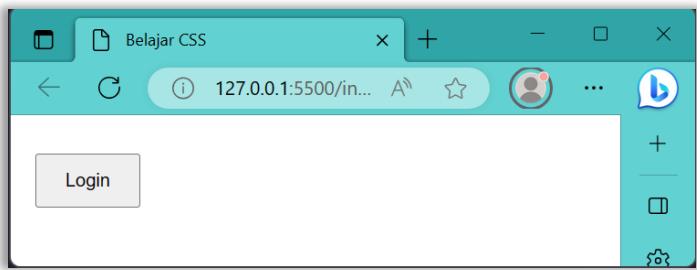
Untuk mengetahui lebih dalam, mari kita coba buat kode HTML berikut.

```
<body>
  <button>Login</button>
</body>
```

Kemudian beri style dengan menambahkan CSS berikut.

```
button {
  padding: 10px 20px 10px 20px;
  margin: 20px 10px;
}
```

Maka hasilnya adalah :



Sama seperti **margin**, **padding** juga memiliki properti khusus pada setiap sisinya. Yaitu **padding-top**, **padding-right**, **padding-bottom**, dan **padding-left**.

BAB XII

DEKORASI TEKS

A. MENGATUR PERATAAN TEKS (TEXT ALIGN)

Properti ini punya nilai yang valid:

- **left** untuk rata kiri;
- **right** untuk rata kanan;
- **center** untuk rata tengah;
- **justify** untuk rata kiri dan kanan;
- **start** rata pada awal (mengikuti **direction**);
- **end** rata pada akhir (mengikuti **direction**);

Contoh penggunaan :

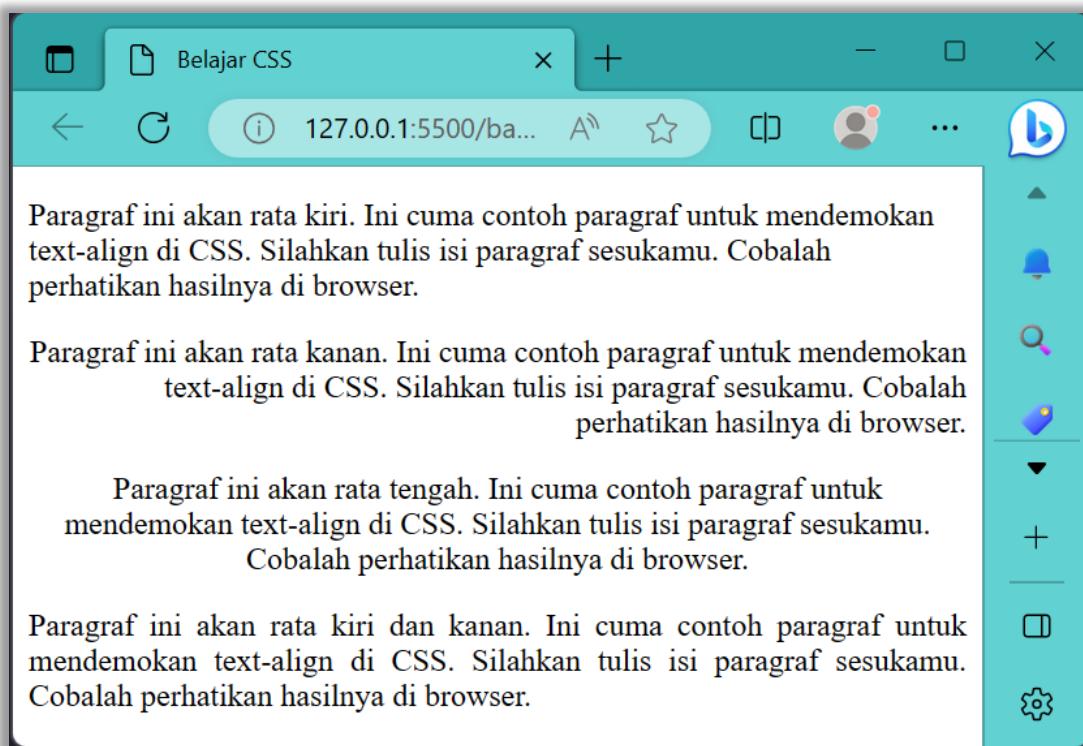
```
p {  
    text-align: justify;  
}
```

Agar lebih jelas, buatlah kode HTML berikut ini.

```
<body>  
    <article>  
        <p style="text-align: left;">  
            Paragraf ini akan rata kiri. Ini cuma contoh paragraf untuk  
            mendemokan text-align di CSS. Silahkan tulis isi paragraf  
            sesukamu. Cobalah perhatikan hasilnya di browser.  
        </p>  
        <p style="text-align: right;">  
            Paragraf ini akan rata kanan. Ini cuma contoh paragraf untuk  
            mendemokan text-align di CSS. Silahkan tulis isi paragraf  
            sesukamu. Cobalah perhatikan hasilnya di browser.  
        </p>  
        <p style="text-align: center;">  
            Paragraf ini akan rata tengah. Ini cuma contoh paragraf untuk  
            mendemokan text-align di CSS. Silahkan tulis isi paragraf  
            sesukamu. Cobalah perhatikan hasilnya di browser.  
        </p>  
        <p style="text-align: justify;">
```

```
        Paragraf ini akan rata kiri dan kanan. Ini cuma contoh  
        paragraf untuk mendemokan text-align di CSS. Silahkan tulis  
        isi paragraf sesukamu. Cobalah perhatikan hasilnya di  
        browser.  
    </p>  
  </article>  
</body>
```

Maka outputnya adalah :



B. TEXT DIRECTION

CSS punya properti **direction** untuk menentukan arah baca teks. Misalnya dari kiri ke kanan (**ltr**) atau dari kanan ke kiri (**rtl**). Secara umum teks biasanya dibaca dari kiri ke kanan. Tapi ada juga yang dibaca dari kanan ke kiri, contohnya teks bahasa arab.

Berikut ini kita akan mencoba menggunakan **text-direction** menggunakan teks berbahasa arab. Salinlah kode HTML berikut ini.

```
<body>  
  <article>  
    <p>  
      وعند موافقه العميل المبدئيه على التصميم يتم ازالة هذا النص  
      من التصميم ويتم وضع النصوص النهائية المطلوبة للتصميم ويقول البعض ان
```

وضع النصوص التجريبية بالتصميم قد تشغل المشاهد عن وضع الكثير من الملاحظات او الانتقادات للتصميم الاساسي.

```
</p>
```

```
<p>
```

وعند موافقه العميل المبدئيه على التصميم يتم ازالة هذا النص من التصميم ويتم وضع النصوص النهائية المطلوبة للتصميم ويقول البعض ان وضع النصوص التجريبية بالتصميم قد تشغل المشاهد عن وضع الكثير من الملاحظات او الانتقادات للتصميم الاساسي.

```
</p>
```

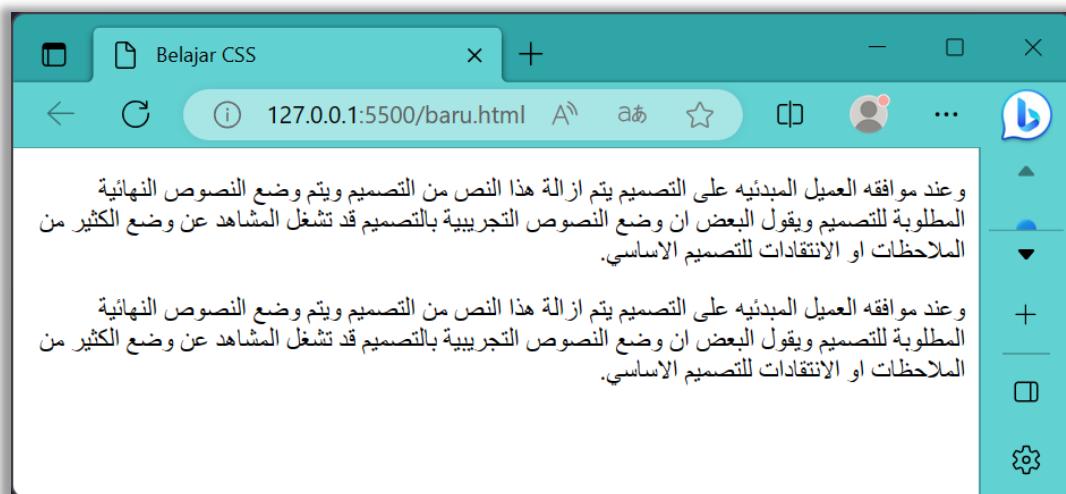
```
</article>
```

```
</body>
```

Kemudian tambahkan kode CSS berikut ini.

```
body {  
    direction: rtl;  
}
```

Maka hasilnya akan seperti ini.



C. TEXT DECORATION

Properti **text-decoration** adalah properti untuk memberikan dekorasi garis pada teks. Contohnya seperti garis bawah, garis atas, dan garis di tengah (teks tercoret). Properti ini punya nilai berupa:

- **none** artinya tidak menggunakan dekorasi apa pun;
- **underline** artinya kita akan memberikan dekorasi garis bawah;
- **overline** dekorasi dengan garis di atas teks;

- **line-through** dekorasi garis di tengah atau teks tercoret.

Contoh penggunaannya adalah sebagai berikut:

```
<body>
  <h1>Contoh Text Decoration</h1>
  <p>
    <span class="underline">Ini Teks dengan dekorasi garis bawah
    </span>. Ini teks normal, tanpa dekorasi apapun. Ini teks
    tanpa dekorasi. Ini teks <span class="strike">yang
    dicoret</span>. <span class="upperline">Ini teks dengan
    dekorasi garis di atas.</span>
  </p>
</body>
```

Kemudian beri style dengan menambahkan kode berikut :

```
h1 {
  text-decoration: underline;
}

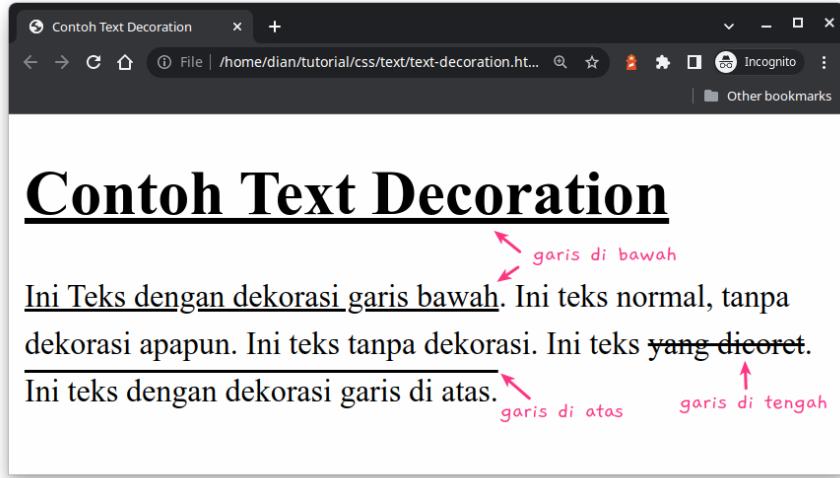
p {
  line-height: 1.5;
}

.upperline {
  text-decoration: overline;
}

.underline {
  text-decoration: underline;
}

.strike {
  text-decoration: line-through;
}
```

Pada contoh ini, kita menggunakan elemen **** dengan class **text-decoration** yang sudah kita buat di CSS. Kita sengaja menggunakan **line-height** pada paragraf agar dekorasi **upperline** tidak terlihat seperti garis bawah. Maka hasilnya adalah :



Selain dekorasi dengan garis lurus, kita juga bisa menentukan style garis dan warnanya. Contohnya adalah :

```
.underline {  
    text-decoration: underline wavy green;  
}
```

Artinya: kita menambahkan garis dibawah teks dengan gaya garis berombak (wavy) dan warnanya hijau.

D. TEXT TRANSFORM

Properti **text-transform** berfungsi untuk mengubah bentuk teks. Misalnya dari huruf kecil menjadi huruf besar. Berikut ini nilai-nilai yang valid untuk **text-transform**.

- **lowercase** untuk mengubah semua teks menjadi huruf kecil;
- **uppercase** untuk mengubah semua teks menjadi huruf kapital (besar);
- **capitalize** untuk mengubah teks pada huruf pertama kata menjadi kapital.
- **none** untuk tidak melakukan transform.

Contoh penggunaannya adalah sebagai berikut:

```
p {  
    text-transform: capitalize;  
}
```

E. TEXT SPACING

Text Spacing adalah ruang kosong pada teks. CSS menyediakan beberapa properti untuk mengatur spacing pada teks, yaitu :

- **text-indent** untuk mengatur indentasi;
- **letter-spacing** untuk mengatur jarak spasi antar huruf;
- **line-height** untuk mengatur jarak antar baris;
- **word-spacing** untuk mengatur jarak antar kata;
- **white-space** untuk mengatur white space pada teks.

Untuk lebih memahami setiap properti di atas, mari kita coba berlatih dengan menulis kode HTML berikut ini.

```
<body>
  <h1>Contoh Text Spacing</h1>
  <p class="small">Contoh teks dengan ukuran kecil</p>

  <p class="indent">
    Ini adalah contoh paragraf dengan teks indent. Artinya, teks
    pada baris pertama akan masuk ke dalam sepanjang 2em. Panjang
    2em sama dengan 2x16px, maka 2em akan bernilai 32px.
  </p>
  <p class="line-height">
    Ini adalah contoh paragraf tanpa indent. Paragraf normal
    dengan line height sebesar 1.6. Normalnya line height tiap
    teks adalah 1. Tapi line height 1 sulit dibaca jika ada banyak
    teks. Line height 1.6 akan membuat teks jadi mudah dibaca,
    karena ada spasi atau jarak di antara baris.
  </p>
  <p class="nowrap">
    Ini conoth paragraf dengan white-space: break-spaces.
    Artinya wthie space pada paragraf ini, akan mengikuti
    spasi seperti yang ditulis di dalam HTML.
  </p>
</body>
```

Kemudian tambahkan kode CSS berikut ini :

```
h1 {
  letter-spacing: -2px;
  word-spacing: 4px;
}

.small {
  font-size: 8px;
  letter-spacing: 2px;
```

```

    text-transform: uppercase;
}

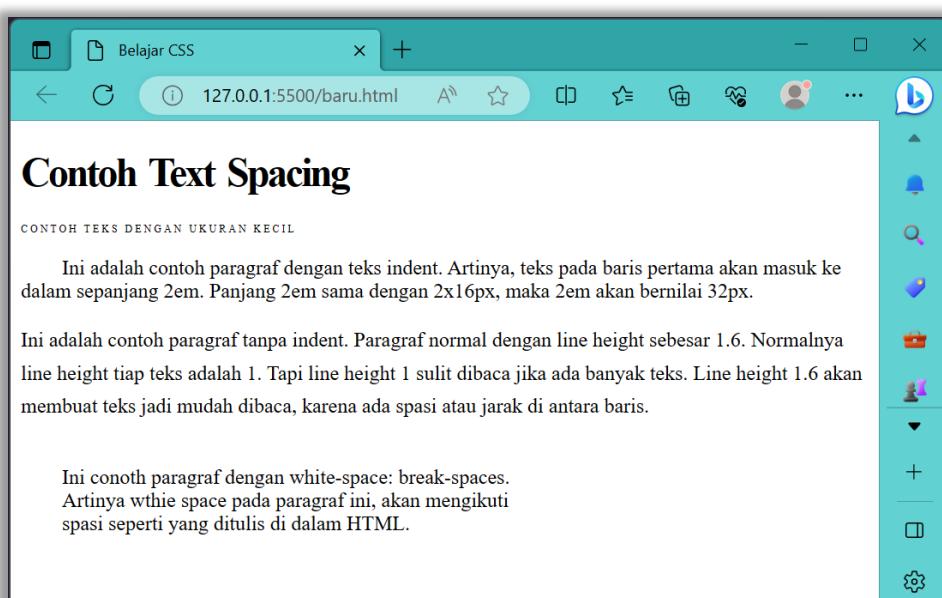
.line-height {
    line-height: 1.6;
}

.indent {
    text-indent: 2em;
}

.nowrap {
    white-space: break-spaces;
}

```

Maka hasilnya adalah :



Pada contoh ini, kita memberikan **letter-spacing: -2px** dan **word-spacing: 4px** pada elemen **<h1>**. Ini akan membuat hurufnya merapat dan jarak antar katanya **4px**.

Lalu di paragraf berikutnya, kita memberikan jarak huruf **2px** dengan transformasi teks **uppercase**.

Pada paragraf berikutnya kita menerapkan **text-indent**, properti ini akan membuat teks masuk ke dalam (menjorok) pada baris pertama paragraf.

Kemudian di paragraf berikutnya, kita menggunakan **line-height** sebesar **1.6**. Ini berfungsi untuk mengatur jarak antar baris. Secara default, nilai **line-height** adalah **1**. Jika

kita memberikan nilai di atas **1**, maka jarak barisnya akan semakin menjauh. Begitu juga sebaliknya, jika diberikan nilai di bawah **1** maka jarak barisnya akan semakin dekat.

Pada paragraf terakhir, kita menggunakan properti **white-space** untuk mengatur metode white space yang digunakan pada paragraf.

F. TEXT SHADOW

Properti **text-shadow** merupakan properti untuk memberikan bayangan pada teks. Properti ini mulai ada pada CSS3. Format nilai untuk **text-shadow** adalah sebagai berikut:

```
text-shadow: x y blur warna;
```

- **x** adalah jarak bayangan berdasarkan sumbu x (horizontal);
- **y** adalah jarak bayangan bedasarkan sumbu y (vertikal);
- **blur** adalah ukuran kepuddaran dari bayangan;
- **warna** adalah warna bayangan, bisa diberikan dalam nama warna, fungsi **rgb()**, **hsl()**, dsb.

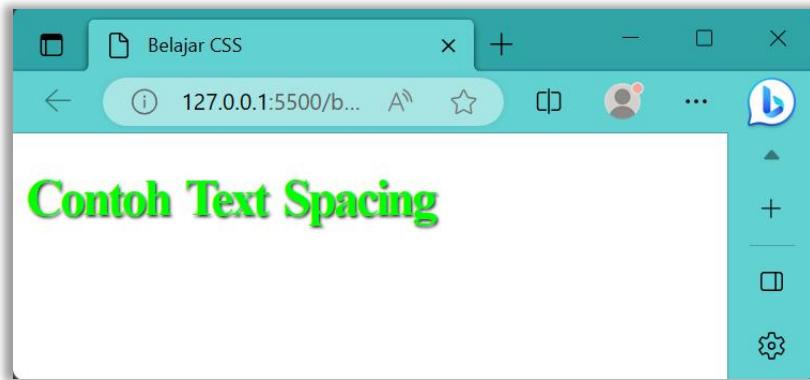
Agar lebih memahami tentang text-shadow, cobalah buat kode di bawah ini.

```
<body>
  <h1>Contoh Text Spacing</h1>
</body>
```

Kemudian tambahkan kode CSS berikut ini.

```
h1 {
  color: lime;
  text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.8);
}
```

Maka hasilnya adalah



G. MENGATUR UKURAN TEXT (FONT SIZE)

Untuk mengatur ukuran font, kita bisa menggunakan properti `font-size`. Properti ini dapat kita beri nilai dengan satuan `px` (piksel), `%` (persentase), `em`, `rem`, dan lain-lain. Contoh :

```
<p style="font-size: 18px;">Contoh Font Siza</p>
```

Artinya: Kita akan mengatur ukuran font untuk elemen `<p>` sebesar `18px`. Selain menggunakan angka, ada juga nilai bawaan yang sudah tersedia seperti :

- `normal` untuk ukuran normal (16px);
- `small` untuk ukuran kecil;
- `x-small` untuk teks yang lebih kecil dari `small`;
- `xx-small` untuk teks yang dua kali lebih kecil dari `small`;
- `large` untuk ukuran besar;
- `x-large` untuk ekstra besar;
- `xx-large` untuk ekstra-ekstra besar;

H. MENGATUR KETEBALAN (FONT WEIGHT)

Untuk mengatur ketebalan font, kita bisa menggunakan properti `font-weight`. Properti ini dapat kita berikan nilai berupa angka ukuran ketebalan atau nilai konstan yang sudah ada. Contoh nilai-nilai yang valid:

- `lighter` untuk ketebalan yang tipis;
- `normal` untuk ketebalan normal;
- `bold` untuk membuat font lebih tabal;
- `bolder` untuk membuat font lebih tebal lagi;
- `100` sampai `900` rentang angka ketebalan.

Misalnya kita punya kode HTML berikut ini.

```
<body>
    <h1>Contoh Font Weight</h1>
    <p class="light">
        Font Weight berfungsi untuk Menentukan Ketebalan Font</p>
    <p class="bold">Ini adalah contoh paragraf dengan font-weight bold</p>
</body>
```

Kemudian kita tambahkan kode CSS berikut ini.

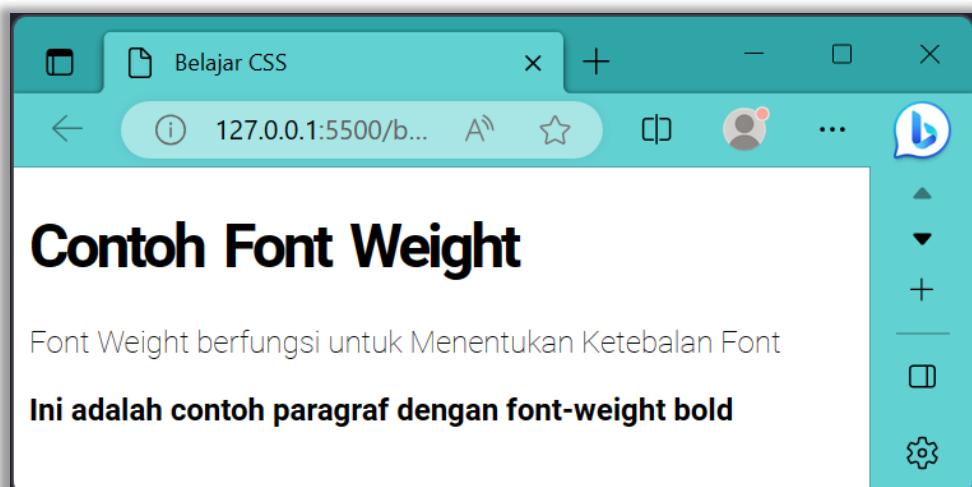
```
body {
    font-family: Roboto;
}

h1 {
    font-weight: bold;
}

.light {
    font-weight: lighter;
}

.bold {
    font-weight: bold;
}
```

Maka hasilnya adalah :



I. TEKS MIRING (FONT STYLE)

Properti **font-style** digunakan untuk mengatur agar teks miring. Nilai yang valid untuk properti ini adalah:

- **normal** – teks normal;
- **italic** – teks miring (italic);
- **oblique** – teks miring seperti italic;

Contoh penggunaannya adalah seperti kode berikut.

```
p {  
    font-style: italic;  
}
```

J. MENGATUR JENIS FONT (FONT FAMILY)

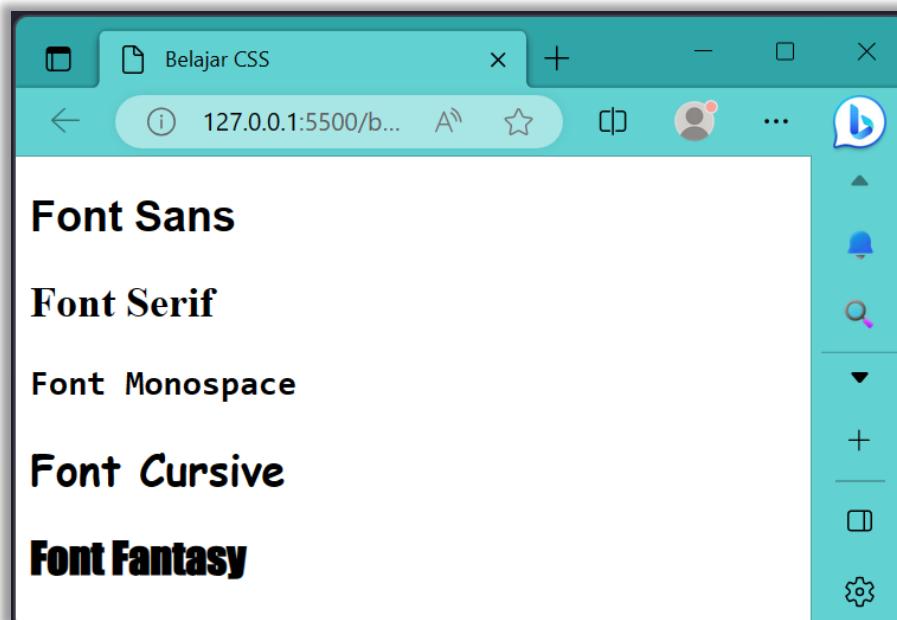
Untuk mengatur jenis font, kita bisa gunakan properti **font-family**. Properti ini bisa kita isi dengan nama font dan juga nama family dari font. Secara umum ada lima jenis family font:

- **Sans Serif** – Font tanpa tanduk;
- **Serif** – Font yang ujungnya punya tanduk;
- **Monospace** – Font dengan ukuran space yang sama;
- **Cursive** – Font seperti tulisan tangan;
- **Fantasy** – Font dengan bentuk yang menarik.

Contoh:

```
<body>  
    <h2 style="font-family: sans-serif">Font Sans</h2>  
    <h2 style="font-family: serif">Font Serif</h2>  
    <h2 style="font-family: monospace">Font Monospace</h2>  
    <h2 style="font-family: cursive">Font Cursive</h2>  
    <h2 style="font-family: fantasy">Font Fantasy</h2>  
</body>
```

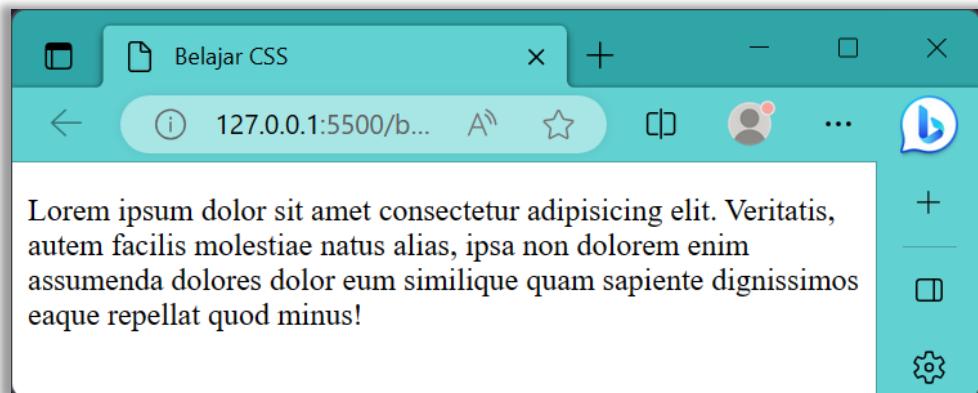
Maka hasilnya adalah :



Font yang dipakai akan menggunakan default font yang dipakai pada sistem. Selain menggunakan nama family, kita juga bisa menggunakan nama font-nya langsung. Berikut contohnya :

```
<body>
    <p style="font-family: Geneva, Neue, 'Times New Roman';">
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Veritatis, autem facilis molestiae natus alias, ipsa
        non dolorem enim assumenda dolores dolor eum similique quam
        sapiente dignissimos eaque repellat quod minus!
    </p>
</body>
```

Maka hasilnya adalah :



Pada kode CSS ini, kita menggunakan tiga font, yakni **Geneva**, **Neue**, dan '**Times New Roman**'. Jika di komputer kita tidak ada font **Geneva**, maka font **Neue** yang akan dipakai. Begitu juga jika tidak ada font **Neue**, maka font '**Times New Roman**' yang akan dipakai.

K. MENGGUNAKAN FONT DARI GOOGLE

Untuk menggunakan font dari Google Font, kita harus mengimpor font-nya terlebih dahulu dengan tag **<link>** di HTML atau kata kunci **@import** di CSS.

Contoh: dengan tag **<link>** di HTML.

```
<link  
href="https://fonts.googleapis.com/css2?family=Caveat:wght@500&display=sw  
ap" rel="stylesheet">
```

Contoh: dengan kata kunci **@import** di CSS.

```
@import  
url('https://fonts.googleapis.com/css2?family=Caveat:wght@500&display=swa  
p');
```

Misalnya kita punya kode HTML seperti di bawah ini.

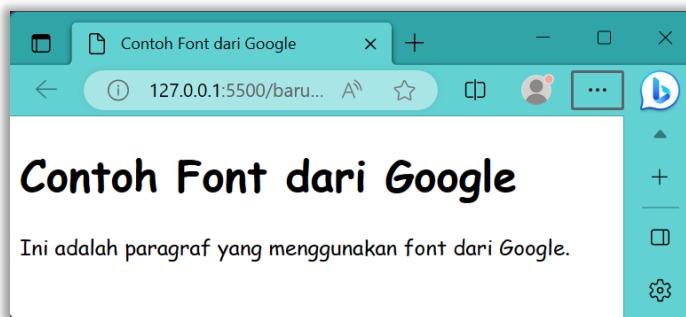
```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <title>Contoh Font dari Google</title>  
    <link rel="stylesheet" href="style.css">  
</head>  
  
<body>  
    <h1>Contoh Font dari Google</h1>  
    <p>Ini adalah paragraf yang menggunakan font dari Google.</p>  
</body>  
  
</html>
```

Kemudian tambahkan kode CSS berikut :

```
h1 {
```

```
    font-family: 'Permanent Marker', cursive;  
}  
  
p {  
    font-family: 'Caveat', cursive;  
}
```

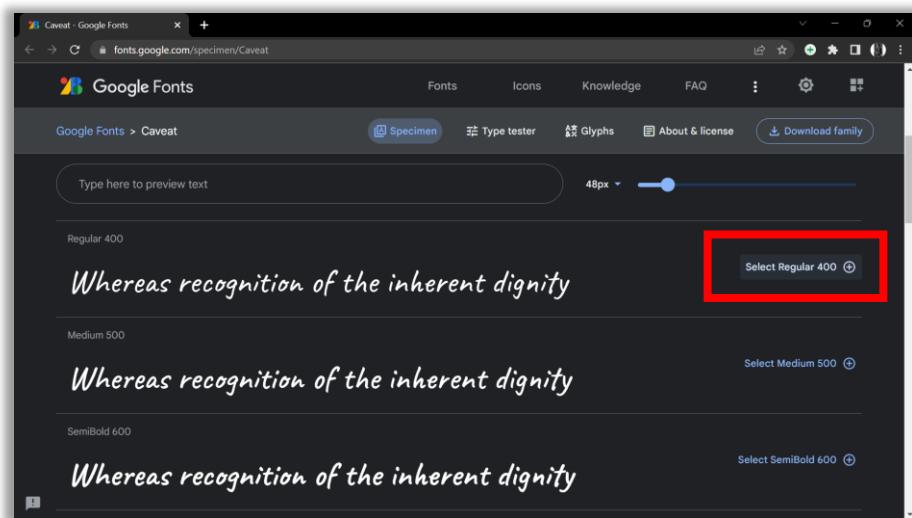
Maka hasilnya adalah :



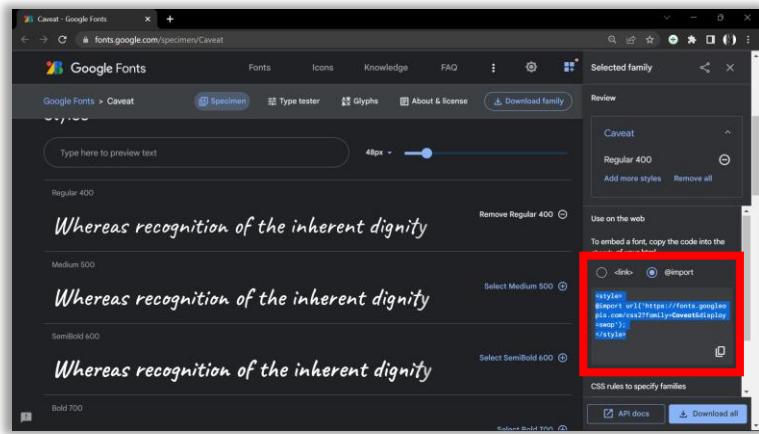
Font yang digunakan adalah font **cursive** yang ada di komputer. Sekarang mari kita coba tambahkan font dari Google. Silakan buka font berikut:

- <https://fonts.google.com/specimen/Permanent+Marker>
- <https://fonts.google.com/specimen/Caveat>

Setelah itu, pilih ukuran font yang kita inginkan.



Setelah itu, kita akan mendapatkan kode untuk mengimpor font ke HTML.



Silakan salin kode berikut, boleh menggunakan `<link>` atau `@import`. Lakukan juga hal yang sama pada font yang satunya.

```
@import
url('https://fonts.googleapis.com/css2?family=Caveat&display=swap');
@import
url('https://fonts.googleapis.com/css2?family=Permanent+Marker&display=swap');
h1 {
    font-family: 'Permanent Marker', cursive;
}
p {
    font-family: 'Caveat', cursive;
}
```

Maka hasilnya adalah:



BAB XIII

DEKORASI TABEL

A. BATAS TABEL

Batas tabel dapat kita buat menggunakan properti **border**, dimana properti border sudah kita bahas di pertemuan sebelumnya. Untuk menambahkan baris pada tabel, biasanya diterapkan pada selektor **table**, **th**, dan **td**.

Misalnya kita memiliki kode HTML berikut ini. (kode berikut akan digunakan di contoh-contoh selanjutnya).

```
<body>
  <table>
    <tr>
      <th>No.</th>
      <th>Nama</th>
      <th>Bahasa Pemrograman</th>
      <th>Instruktur</th>
      <th>Rata-rata Nilai</th>
    </tr>
    <tr>
      <td>1.</td>
      <td>Dimas Firmansyah</td>
      <td>Java</td>
      <td>Dian Nurul</td>
      <td>70</td>
    </tr>
    <tr>
      <td>2.</td>
      <td>Kurniawan</td>
      <td>Golang</td>
      <td>Habib Jannata</td>
      <td>70</td>
    </tr>
    <tr>
      <td>3.</td>
      <td>Sawaludin</td>
      <td>C#</td>
      <td>Afi'idatul</td>
      <td>70</td>
    </tr>
    <tr>
```

```

        <td>4.</td>
        <td>Kharisma</td>
        <td>Node.Js</td>
        <td>Romi Kusuma</td>
        <td>70</td>
    </tr>
</table>
</body>

```

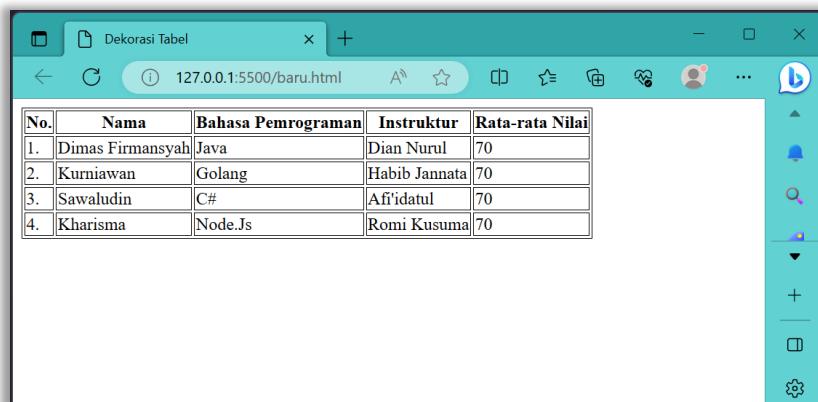
Kemudian kita tambahkan kode CSS berikut ini.

```

table, td, th {
    border: 1px solid black;
}

```

Maka hasilnya adalah :



A screenshot of a web browser window titled "Dekorasi Tabel". The address bar shows the URL "127.0.0.1:5500/baru.html". The main content area displays a table with the following data:

No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Aff'idatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

Dalam contoh di atas, setiap **cell** memiliki border sendiri-sendiri, kemudian dibungkus keseluruhan pada selektor **table**, sehingga menghasilkan 2 garis yang bersebelahan. Bagaimana jika kita hanya ingin menambahkan 1 border untuk keseuruan isi tabel?

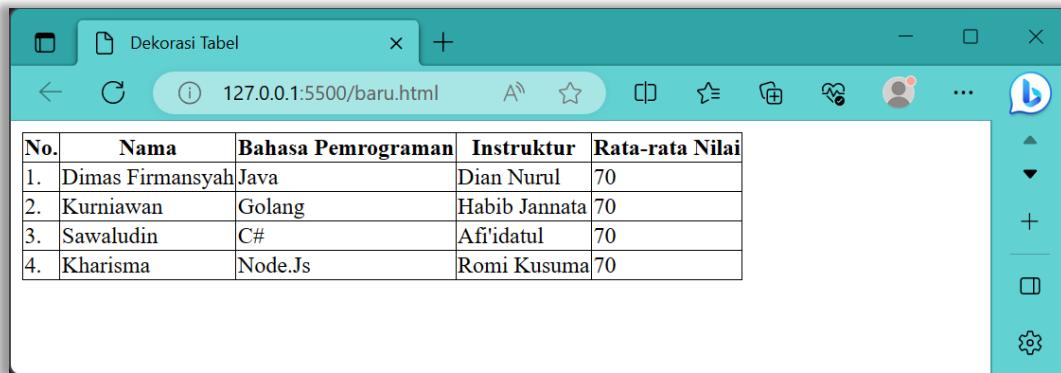
Maka kita bisa menggunakan properti **border-collapse** dimana nantinya border akan diciutkan. Cara menggunakannya yaitu sebagai berikut.

```

table {
    border-collapse: collapse;
}

```

Maka hasilnya adalah :



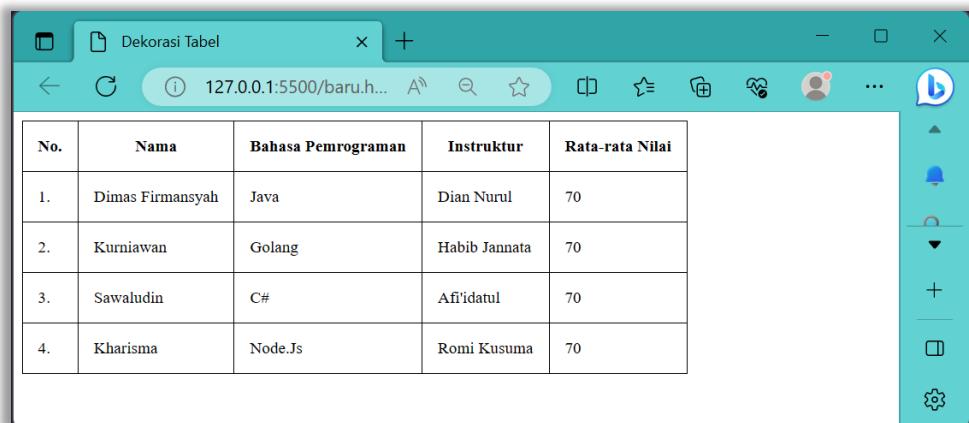
No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Afi' idatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

B. PADDING TABEL

Penambahan **padding** pada tabel pada dasarnya sama seperti kita menambahkan padding pada elemen lain. Penggunaan **padding** ini biasanya digunakan pada selektor **<th>** dan **<td>**. Maka penggunaannya yaitu :

```
th, td {  
    padding: 15px;  
}
```

Maka hasilnya adalah :



No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Afi' idatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

C. PENYELARASAN TABEL (ALIGNMENT)

Ada dua cara menerapkan Alignment pada tabel, yaitu

1. Horizontal Alignment

Properti ini mengatur perataan kiri, kanan, tengah, dan rata kiri-kanan. Penggunaannya biasanya diterapkan di selektor **td**. Caranya adalah sebagai berikut:

```
td {  
    text-align: center;  
}
```

Maka hasilnya adalah :



A screenshot of a web browser window titled "Dekorasi Tabel". The address bar shows "127.0.0.1:5500/baru.html". The main content is a table with the following data:

No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Afi'iddatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

2. Vertical Alignment

Properti ini hampir sama dengan Horizontal Alignment, bedanya properti ini hanya mengatur perataan atas, tengah (middle), dan bawah. Contoh penggunaannya adalah :

```
td {  
    height: 50px;  
    vertical-align: bottom;  
}
```

Misalnya dalam contoh diatas, kita menambahkan properti **height** untuk menambahkan tinggi suatu sel, tujuannya agar kita mudah dalam memahami properti **vertical-align**. Kemudian hasilnya adalah sebagai berikut:

A screenshot of a web browser window titled "Dekorasi Tabel". The address bar shows the URL "127.0.0.1:5500/baru.h...". The main content area displays a table with five columns: No., Nama, Bahasa Pemrograman, Instruktur, and Rata-rata Nilai. There are four rows of data. The first and third rows have a light gray background color, while the second and fourth rows have a white background color. The table is styled with horizontal and vertical borders.

No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Afi' idatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

D. STRIPED TABLE

Striped table adalah tampilan tabel bergaris. Untuk membuat striped table maka kita harus menggunakan pseudo-class `nth-child()`. Biasanya diterapkan pada selektor `tr`. Berikut adalah contoh penggunaannya.

```
tr:nth-child(even) {
    background-color: #ddd;
}
```

Maka hasilnya adalah :

A screenshot of a web browser window titled "Dekorasi Tabel". The address bar shows the URL "127.0.0.1:5500/baru.h...". The main content area displays a table with five columns: No., Nama, Bahasa Pemrograman, Instruktur, and Rata-rata Nilai. There are four rows of data. The first and third rows have a light gray background color, while the second and fourth rows have a white background color. The table is styled with horizontal and vertical borders.

No.	Nama	Bahasa Pemrograman	Instruktur	Rata-rata Nilai
1.	Dimas Firmansyah	Java	Dian Nurul	70
2.	Kurniawan	Golang	Habib Jannata	70
3.	Sawaludin	C#	Afi' idatul	70
4.	Kharisma	Node.Js	Romi Kusuma	70

Dalam contoh diatas kita menambahkan parameter `even` pada pseudo-class `nth-child()`, artinya kita mengubah warna background semua baris yang ganjil. Jika kita ingin mengubah baris yang genap, maka parameternya diganti menjadi `odd`.