



BELAJAR **HTML & CSS**

Hirzi Widyan Putra



Daftar Isi

BAB I	3
1.1 Aplikasi Pendukung	3
1.2 Pengertian HTML	3
1.3 Pengertian CSS.....	3
BAB II	4
2.1 HTML	4
2.1.1 Penulisan Kode Html	4
2.1.2 Membuat Heading	5
2.1.3 Paragraf	6
2.1.4 Macam – macam Format Text.....	7
2.1.5 Membuat Link	8
2.1.6 Menampilkan Gambar.....	9
2.1.7 Menampilkan Video	10
2.1.8 Tabel	12
2.1.9 List pada HTML.....	16
2.1.10 Form.....	18
2.1.11 Tag HTML & Kegunaannya.....	22
2.1.12 Latihan HTML	26
2.2 CSS.....	27
2.2.1 Aturan penulisan CSS.....	27
2.2.2 Menghubungkan HTML dan CSS	27
2.2.3 Tag Span dan Div pada HTML	29
2.2.4 Mengenal Class dan ID pada HTML	29
2.2.5 CSS Inline, Internal dan External	33
2.2.6 Mengubah Background HTML.....	36
2.2.7 Mengatur Jarak Luar dengan Margin	38
2.2.8 Mengatur Jarak Dalam dengan Padding	40
2.2.9 Mengatur Link	42
2.2.10 Text Style dan Text Layout	44



2.2.11 Macam – Macam Border	46
2.2.12 Properti pada Page Layout	48
2.2.13 Belajar Layout HTML dengan CSS	50
2.2.14 Bonus Desain Web HTML & CSS	53



BAB I

1.1 Aplikasi Pendukung

Sebelum kita mempelajari HTML dan CSS, apa saja sih yang perlu dilakukan? Yaitu Menginstall Aplikasi pendukung kita untuk mempelajari HTML & CSS.

Diantaranya :

- **Web Browser**
 - o Adalah Aplikasi atau perangkat yang digunakan untuk mencari atau menjelajahi informasi di dalam jaringan internet dari sebuah media seperti situs, blog, jejaring sosial, dan lain sebagainya yang tersimpan di dalam internet. (Chrome, Mozilla, dll)
 - o Nantinya Web Browser akan **menampilkan** hasil Html & CSS yang sudah kita buat.
- **Text Editor**
 - o Adalah Aplikasi yang biasa digunakan Programmer untuk membuat kode program.
 - o Contohnya Sublime Text, Atom, Visual Studio Code, Notepad++

1.2 Pengertian HTML

HTML singkatan dari Hyper Text Markup Language, adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web. HTML inilah yang menyusun sebuah halaman web menjadi sebagaimana yang kita lihat melalui browser.

1.3 Pengertian CSS

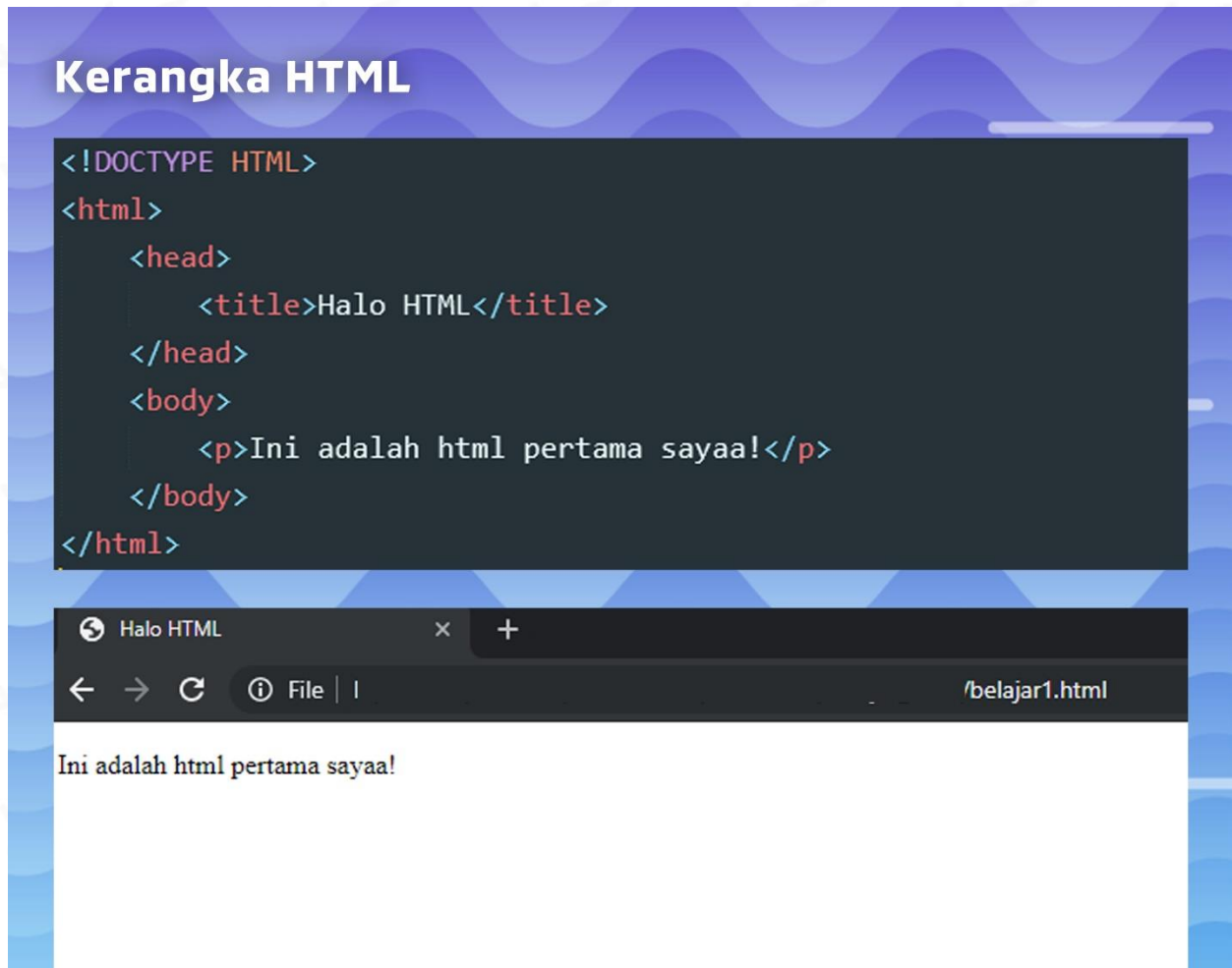
CSS adalah kependekan dari Cascading Style Sheet. CSS merupakan kumpulan kode-kode yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web kita supaya lebih elegan dan menarik.



BAB II

2.1 HTML

2.1.1 Penulisan Kode Html



Silahkan kalian buat dulu Folder **Belajar Html** lalu ketikkan Kode Html diatas pada Text Editor kalian dan simpan nama file **belajar1.html**

Penjelasan :

- Kode `<!DOCTYPE HTML>` itu berfungsi sebagai pengenalan Versi HTML5
- Setiap Menulis kode Html kita harus memulainya dengan kode `<html>` dan diakhiri dengan `</html>`
- Kode diantara `<head>` dan `</head>` digunakan untuk memberikan informasi tambahan kepada browser, contohnya ada `<title>` yang artinya memberikan informasi ke browser bahwa halaman web yang dibuat ini memiliki judul *Halo HTML*



- Semua yang ingin kita tampilkan pada browser terletak di antara kode `<body>` dan `</body>`. Yuup, itulah inti dari kode kita. Disitulah kode-kode yang akan menyusun halaman web ditulis.
- `<p> </p>` Digunakan untuk membuat sebuah Paragraf.

2.1.2 Membuat Heading

The image shows a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Halo HTML</title>
</head>
<body>
  <h1>Ini Heading 1</h1>
  <h2>Ini Heading 2</h2>
  <h3>Ini Heading 3</h3>
  <h4>Ini Heading 4</h4>
  <h5>Ini Heading 5</h5>
  <h6>Ini Heading 6</h6>
</body>
</html>
```

Below the code editor is a browser preview window titled "Halo HTML" showing the rendered output:

- Ini Heading 1 (Large font)
- Ini Heading 2 (Medium-large font)
- Ini Heading 3 (Medium font)
- Ini Heading 4 (Medium-small font)
- Ini Heading 5 (Small font)
- Ini Heading 6 (Smallest font)

Tulis Kode tersebut dan beri nama **heading.html**, yang membedakan H1 s/d H6 hanya ukuran Font Saja.

Heading biasa digunakan untuk judul dari bagian – bagian tertentu, misal Judul dari Artikel kita, atau judul iklan produk kita pada Web.




2.1.3 Paragraf

Contoh Paragraf benar & salah

```

<!DOCTYPE html>
<html>
<head>
  <title>Halo HTML</title>
</head>
<body>
  <p>Ini Contoh Paragraf Kesatu</p>
  <p>Ini Contoh Paragraf Kedua</p>
</body>
</html>


```



```

<!DOCTYPE html>
<html>
<head>
  <title>Halo HTML</title>
</head>
<body>
  <p>Ini Contoh Paragraf Kesatu
  <p>Ini Contoh Paragraf Kedua
</body>
</html>

```



Halo HTML
×
+

← → ↻ ⓘ File |
Paragraf.html

Ini Contoh Paragraf Kesatu

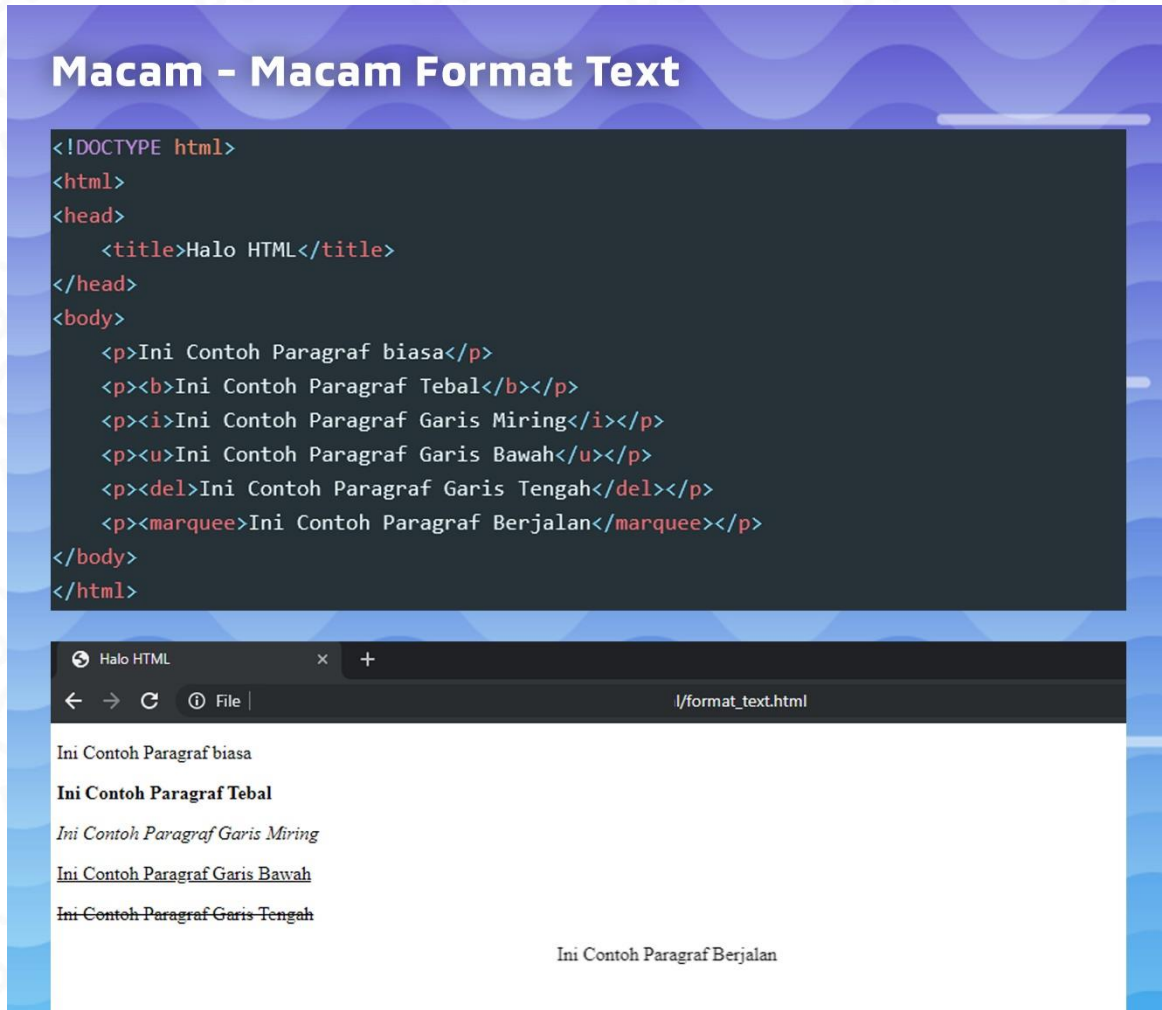
Ini Contoh Paragraf Kedua

Beri nama **paragraf.html**, Paragraf ini biasa digunakan untuk isi detail dari deskripsi produk kita atau artikel – artikel yang biasa kita baca pada Website. Itu terdiri dari beberapa Paragraf dan Heading.

Ingat, Setiap tag HTML menggunakan penutup kecuali beberapa tag special seperti **
** ****, dll, Jadi pada contoh diatas ada paragraf yang benar dan salah dimana keduanya tetap bisa dijalankan tetapi akan berantakan jika kita mempunyai paragraf yang banyak dan panjang.

Sedangkan tag **
** digunakan untuk **membuat baris baru** pada HTML.

2.1.4 Macam – macam Format Text



The screenshot displays an HTML editor with a dark theme. The top section shows the HTML code for a document titled 'Halo HTML'. The code includes basic HTML structure and several text formatting tags: `<p>` for a normal paragraph, `` for bold, `<i>` for italic, `<u>` for underlined, `` for strikethrough, and `<marquee>` for a scrolling text. The bottom section shows the rendered output in a browser window. The text is displayed with the corresponding formatting: normal, bold, italic, underlined, and strikethrough. The scrolling text is also visible.

```
<!DOCTYPE html>
<html>
<head>
  <title>Halo HTML</title>
</head>
<body>
  <p>Ini Contoh Paragraf biasa</p>
  <p><b>Ini Contoh Paragraf Tebal</b></p>
  <p><i>Ini Contoh Paragraf Garis Miring</i></p>
  <p><u>Ini Contoh Paragraf Garis Bawah</u></p>
  <p><del>Ini Contoh Paragraf Garis Tengah</del></p>
  <p><marquee>Ini Contoh Paragraf Berjalan</marquee></p>
</body>
</html>
```

Halo HTML

Ini Contoh Paragraf biasa

Ini Contoh Paragraf Tebal

Ini Contoh Paragraf Garis Miring

Ini Contoh Paragraf Garis Bawah

Ini Contoh Paragraf Garis Tengah

Ini Contoh Paragraf Berjalan

Beri nama File **format_text.html** dan silahkan lihat hasilnya.

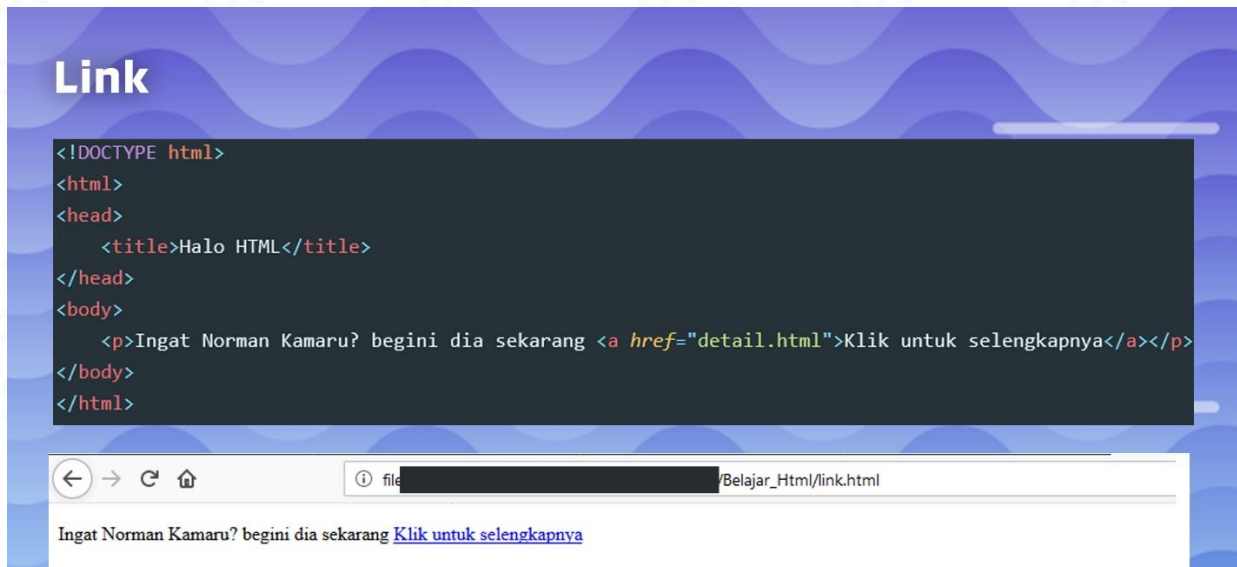
Penjelasan :

- ` ` Membuat text menjadi bold atau tebal
- `<i> </i>` Membuat text garis miring
- `<u> </u>` Membuat text jadi garis bawah
- ` ` Membuat text jadi garis tengah
- `<marquee> </marquee>` Membuat text yang ada dalam tag ini menjadi berjalan

Masih ada beberapa lagi seperti `` untuk membuat text menjadi Pangkat dibawah dan `` membuat text jadi pangkat diatas, dll Silahkan dibaca fungsi nya pada kumpulan Tag HTML yang akan saya tuliskan di Modul ini juga.



2.1.5 Membuat Link



Beri nama **link.html**

Penjelasan :

- Yang bertulisan biru tersebut ialah yang dimaksud dengan Link
- `<a > ` Link yaitu untuk mengarahkan kita ke halaman tertentu. Misal pada contoh diatas merupakan paragraf yang bertujuan untuk mengklik selengkapnya agar bisa melihat detail berita tentang norman kamaru hehe
- Didalam Tag `<a>` terdapat `href=""` , didalam href inilah kita sisipkan halaman tertentu kita. Misal detail-norman.html, maka kita akan diarahkan ke halaman detail norman tersebut.
- Lalu, pernah lihat ketika kita klik suatu link kita diarahkan ke tab baru? Nah itu menggunakan atribut **target="_blank"** , jadi kita tidak menimpa tab yang ada, tetapi akan membuat tab baru. Silahkan kalian coba langsung yaa



2.1.6 Menampilkan Gambar

Nah sekarang kita akan menampilkan gambar pada Web, pasti sudah sering lihat hehe tapi mau tau gimana cara menampilkannya?



Beri nama **gambar.html**

Penjelasan :






- `` tidak memerlukan tag penutup ``
- Atribut `src=""` pada `img` merupakan alamat / lokasi gambar yang akan ditampilkan
- Tag `img` juga memiliki atribut penting lainnya, yaitu `alt=""` digunakan untuk keterangan dari gambar yang ditampilkan. Contoh ``
- Atribut `width` dan `height` memaksa gambar untuk tampil dengan nilai yang kita tetapkan, contoh diatas panjangnya 450 dan tingginya 250.
- **Tetapi** untuk mengatur width height gambar sebaiknya langsung pada CSS saja agar lebih mudah kita modifikasi sesuai kreasi kita, bisa kita buat `rounded`, `border` dan lainnya. Banyak deh yang bisa kita lakukan dengan CSS biar gambarnya lebih menarik hehe



2.1.7 Menampilkan Video

HTML `<video>` element digunakan untuk merujuk sumber file video, sehingga dapat diputar (dimainkan) pada halaman web.

Tapi perlu diketahui, hanya ada beberapa Format video yang support pada Web Browser. Diantaranya :

Browser	MP4	Ogg	WebM
 Chrome	✓	✓	✓
 Firefox	✓	✓	✓
 Internet Explorer	✓	✗	✗
 Safari	✓	✗	✗
 Opera	✓	✓	✓

Lalu, ada beberapa atribut yang bisa kita pakai pada `<video>` yaitu :

autoplay

Menginstruksikan browser untuk mulai memutar video secara otomatis sesegera mungkin tanpa berhenti. Value : autoplay atau (dikosongkan)

controls

Menginstruksikan browser untuk menampilkan antarmuka (user interface) untuk video control (seperti menampilkan tombol putar (play), kontrol volume dan lain sebagainya). Value: controls atau (dikosongkan)

height

Menentukan (ukuran) tinggi pemutar video (video player). Value: *Pixels*

loop

Menginstruksikan browser untuk memutar ulang video ketika selesai (mencapai waktu akhir putar). Value: loop atau (dikosongkan)

muted

Menginstruksikan browser untuk membisukan (tanpa suara) video. Value: muted atau (dikosongkan)

**src**

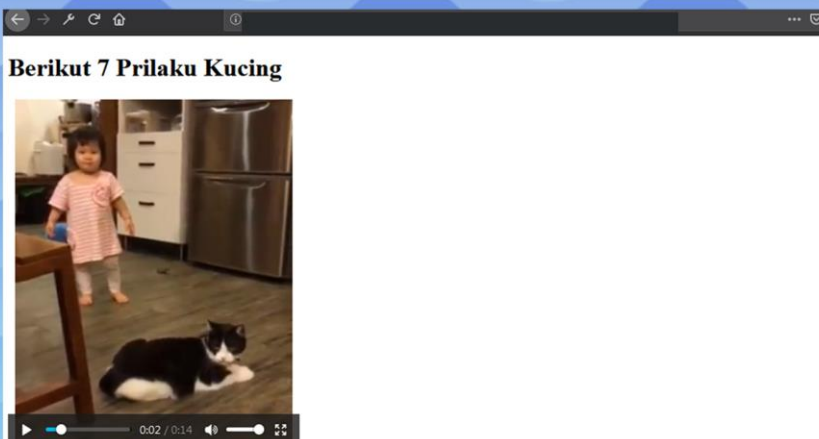
Menentukan URL yang mengarah pada sumber video file berada (disimpan). Value: *URL*

width

Menentukan (ukuran) lebar pemutar video (video player). Value: *Pixels*

Video

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Video</title>
</head>
<body>
  <h1>Berikut 7 Prilaku Kucing</h1>
  <video width="370" height="440" src="contoh_video.mp4" controls>
</video>
</body>
</html>
```



Buat file lagi dengan nama **video.html**

Note :

- Pastikan nama file videonya sama dengan yang ada pada `src=""`



2.1.8 Tabel

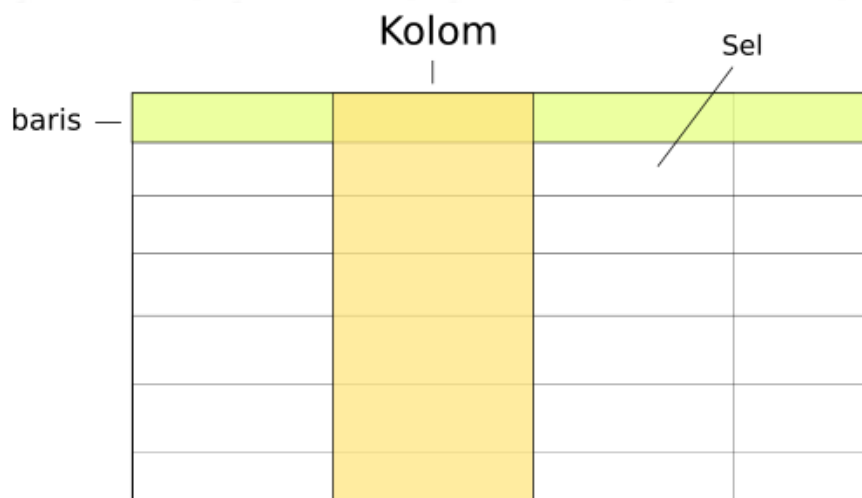
Tabel ini sangat berguna banget untuk menampilkan data atau informasi dalam bentuk **Baris** dan **Kolom** dan tabel memiliki beberapa atribut dan tag.

Atribut pada Tabel :

Atribut	Nilai	Keterangan
Align	left, right, center	Digunakan untuk perataan keseluruhan teks didalam tabel
bgcolor	rgb(x,x,x), #xxxxxx, namawarna	Digunakan untuk pemberian warna latar pada tabel
Border	Angka (0, 1, 2)	Untuk menampilkan garis tabel.
Cellpadding	Pixels	Digunakan untuk memberikan jarak antara sel.
Cellspacing	Pixels	Digunakan untuk memberikan jarak antara sel dan tabel.
Width	Pixels, %	Untuk menentukan lebar tabel.
Colspan	Angka	Untuk menggabungkan / Merge Cell Kolom
Rowspan	Angka	Untuk menggabungkan / Merge Cell Baris

Tag yang digunakan pada Tabel :

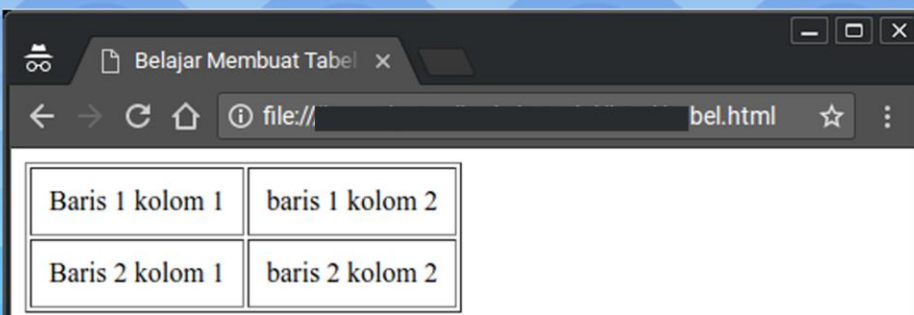
<table> </table>	Untuk membungkus Tabel
<thead> </thead>	Membungkus bagian kepala tabel
<tbody> </tbody>	Membungkus bagian body dari tabel
<tfoot> </tfoot>	Membungkus bagian footer / kaki / bagian bawah dari tabel
<tr> </tr>	Membuat baris tabel
<th> </th>	Biasa digunakan untuk judul Kolom / Baris tabel
<td> </td>	Untuk Cell / Sel dari tabel





Belajar Tabel

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Membuat Tabel HTML</title>
</head>
<body>
  <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>
</body>
</html>
```



Buat dengan nama file **tabel1.html**

Penjelasan :

- `border="1"` artinya memberi garis border 1 kepada tabel, kalau tidak ingin pakai border bisa kok cukup dikasih `border="0"` atau tidak usah dimasukkan atribut border pada tag tabelnya.
- `cellpadding="1"` artinya memberikan jarak teks dengan garis di dalam sel.
- Jika ingin mengatur lebar tabel, tambahkan atribut `width=""` pada tag `<table width="500">` `</table>`

Lalu kita juga bisa memberikan warna baris maupun kolom pada tabel dengan memasukkan atribut `bgcolor="warnadisini"`, contohnya kalau mau baris yang berwarna `<tr bgcolor="#00ff80">` `</tr>` atau kalau mau cuman kolomnya `<td bgcolor="#00ff80">` `</td>`

Nilai atribut `bgcolor` bisa kita isi dengan kode warna dalam heksadesimal atau nama warna dalam bahasa inggris.

Silahkan dicontohkan langsung

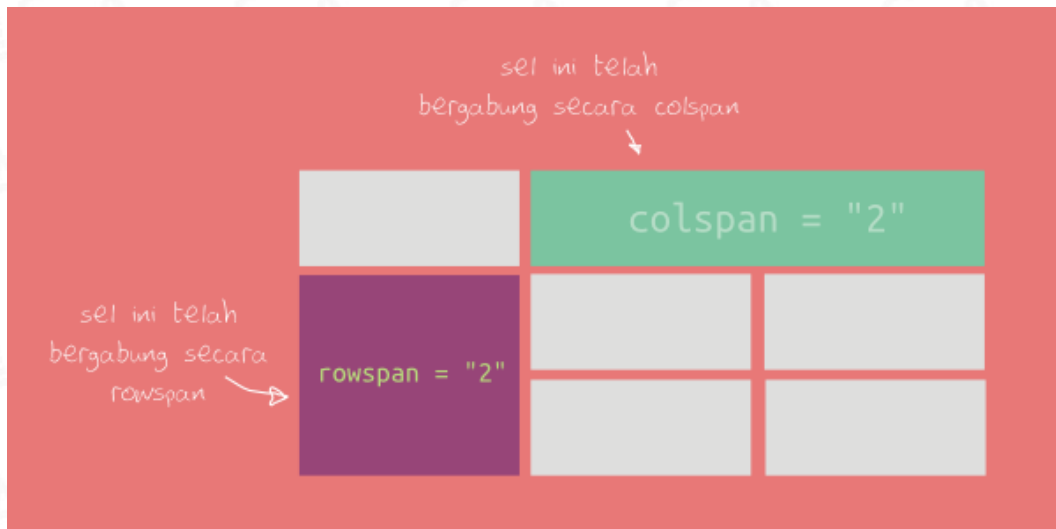


Menggabungkan Sel Tabel

Atribut yang digunakan untuk menggabungkan sel tabel adalah **rowspan** dan **colspan**

- **Rowspan** untuk menggabungkan baris
- **Colspan** untuk menggabungkan kolom.

Atribut ini bisa kita berikan kepada tag `<td>` atau `<th>`



Contoh : Buat Seperti dibawah ini dan beri nama **tabel2.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar Membuat Tabel HTML</title>
  </head>
  <body>
    <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>
    </table>
  </body>
</html>
```



Hasilnya :

Bulan	Hasil Panen	
	Padi	Kacang
Januari	500 Kg	231 Kg
Februari	342 Kg	423 Kg

Yang kuning itu hasil dari Rowspan 2 baris, dan yang hijau Colspan 2 kolom.

Kalian juga bisa memodifikasi tampilan tabel dengan CSS sesuai kreasi, dan bisa juga memasukkan gambar pada tabel sesuai keperluan kita. Misalnya membuat biodata dengan tabel.

Dan bagian tersulit dari tabel ini ialah rowspan dan colspan karena harus teliti berapa sel yang akan digabungkan dengan **colspan** maupun **rowspan**. Sering – sering saja latihan dengan beberapa kasus tertentu yang biasa kita buat pada tabel Excel lalu kita coba praktekan di tabel HTML yang akan kita tampilkan pada web.

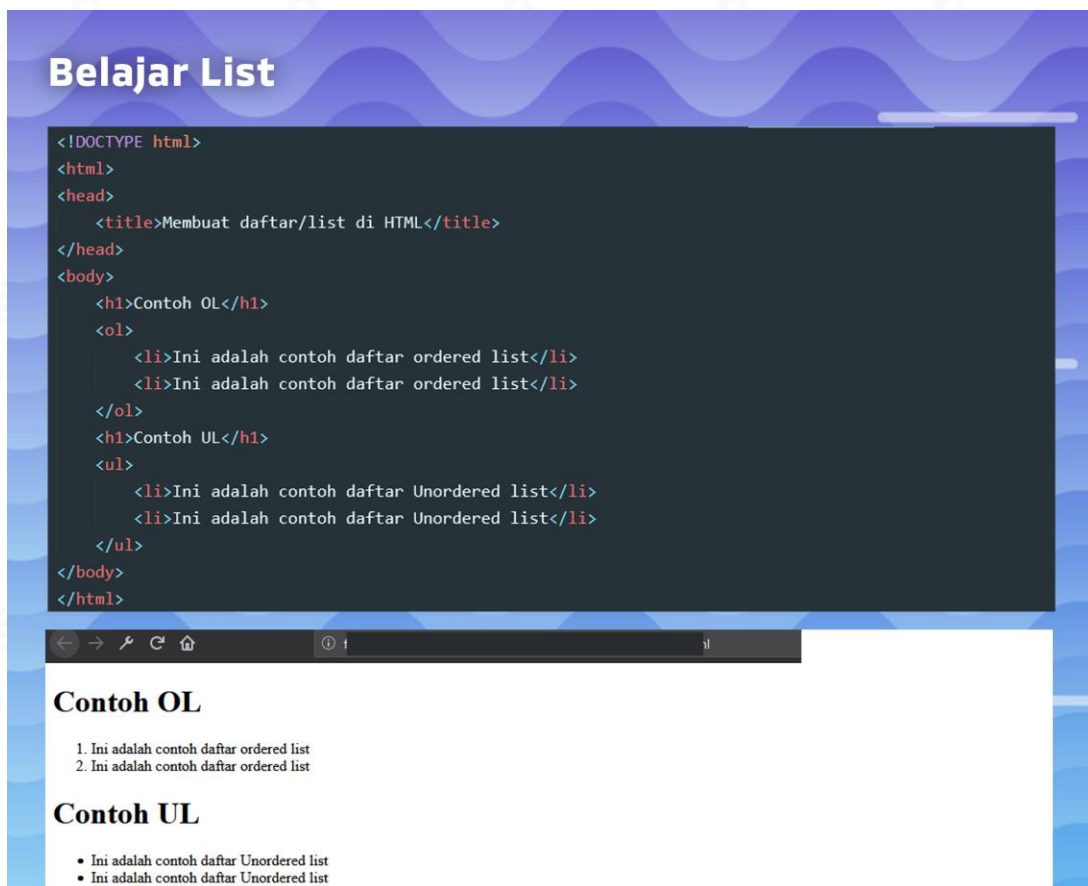


2.1.9 List pada HTML

List adalah elemen yang digunakan untuk menampilkan informasi dalam bentuk list. Biasanya digunakan untuk membuat menu.

Dua jenis list yang sering digunakan :

1. **Ordered List** adalah list yang terurut, menggunakan tag ``
2. **Unordered List** adalah list yang tak terurut, menggunakan tag ``



Beri Nama **list.html**

Penjelasan :

Sebenarnya pada **ordered list** selain angka juga bisa menampilkan tipe pengurutan lainnya, seperti romawi dan lain-lain, caranya hanya dengan menambahkan atribut `type=""` tanpa tanda kutip, contoh penulis akan menampilkan pengurutan daftar alfabet dengan huruf besar.

Contoh :

```
<ol type="A">
  ...
</ol>
```



Berikut ini adalah tabel tipe-tipe pengurutan yang bisa di gunakan untuk ordered list.

Tipe	Keterangan
type="1"	Pengurutan menggunakan angka (default)
type="A"	Pengurutan berdasarkan huruf besar
type="a"	Pengurutan berdasarkan huruf kecil
type="I"	Pengurutan menggunakan angka romawi huruf besar
type="i"	Pengurutan menggunakan angka romawi huruf kecil

Lalu pada contoh **unordered list** menggunakan simbol, juga cara penulisan tipe berbeda, apabila ordered list langsung menggunakan tipe maka unordered list menggunakan atribut style, atribut ini memungkinkan penulisan kode css langsung ditulis di dalam tag yang bersangkutan tanpa membuat file css lagi, berikut ini contohnya.

```
<ul style="list-style-type: circle">
  ....
</ul>
```

Tipe	Keterangan
list-style-type:disc	Pengurutan menggunakan simbol bullets (default)
list-style-type:circle	Pengurutan menggunakan simbol lingkaran
list-style-type:square	Pengurutan menggunakan simbol kotak
list-style-type:none	Menghilangkan simbol pengurutan



2.1.10 Form

Form biasanya digunakan untuk mengumpulkan data dari pengunjung web kita. Mulai dari form untuk login, form kontak, form untuk pendaftaran user, bahkan untuk mengirimkan data antar halaman web.

Perlu diketahui Web tidak hanya digunakan untuk menampilkan informasi saja. Web juga digunakan untuk mengambil informasi atau data dari pengunjung.

Form dalam web bisa disamakan dengan formulir di dunia nyata. Form dapat diisi, kemudian diproses dengan bahasa pemrograman tertentu misalnya **PHP**, tapi kali ini kita akan belajar cara membuat form di HTML saja, belum sampai proses menggunakan PHP ya.

Tag dasar yang akan kita gunakan untuk membuat form di HTML adalah **tag form, input, textarea, select** dan **option**.

Pengertian tag **<form>**

Sebuah form dalam HTML harus berada di dalam tag form, yang diawali dengan **<form>** dan diakhiri dengan **</form>**. Tag form akan membutuhkan beberapa atribut untuk dapat berfungsi dengan seharusnya.

Atribut pertama adalah **action=""**, yang berfungsi untuk menjelaskan kemana data form akan dikirimkan. Biasanya nilai dari atribut action ini adalah alamat dari sebuah halaman PHP yang digunakan untuk memproses isi data form.

Atribut kedua adalah **method=""**, yang berfungsi untuk menjelaskan bagaimana data isian form akan dikirim oleh web browser. Nilai dari atribut method ini bisa berupa **get** atau **post**.

Perbedaan method get dan method post adalah, jika kita mengisi atribut method dengan get maka isian form akan terlihat pada url browser. Method get ini biasanya digunakan untuk query pencarian. Method post biasanya digunakan untuk data yang lebih sensitif seperti yang berisi password, atau registrasi user. Data hasil form tidak akan terlihat pada browser.



Selengkapnya nanti kalian akan pelajari langsung pada saat belajar PHP, tapi cukup mengetahui saja dulu oke hehe

Struktur dasar form akan terlihat sebagai berikut:

```
<form action="proses.php" method="post">
...isi form...
</form>
```

Mengenal tag <input>

Tag input merupakan tag paling banyak digunakan di dalam form dan memiliki banyak bentuk, mulai dari isian text biasa, text password, checkbox, radio, sampai dengan tombol submit, semuanya dalam bentuk tag <input>.

Bentuk-bentuk dari keluarga tag input ini dibedakan berdasarkan atribut type:

- `<input type="text" />` atau bisa juga `<input />` adalah textbox inputan biasa yang menerima input berupa text, contohnya digunakan untuk inputan nama, username, dan inputan yang berupa text pendek. Input type text ini juga bisa memiliki atribut value yang bisa diisi nilai tampilan awal dari text
- `<input type="password" />` dalam tampilannya sama dengan type text, namun teks yang diinput tidak akan terlihat, akan berupa bintang atau bulatan. Biasanya hanya digunakan untuk inputan yang sensitif seperti password.
- `<input type="checkbox" />` adalah inputan berupa checkbox yang dapat diceklist atau dicentang oleh user. User dapat memilih atau tidak memilih checkbox ini. Type checkbox memiliki atribut checked yang jika ditulis atau diisi dengan nilai checked, akan membuat checkbox langsung terpilih pada saat pertama kali halaman ditampilkan. Contoh inputan checkbox berupa hobi, yang oleh user dapat dipilih beberapa hobi.
- `<input type="radio" />` mirip dengan checkbox, namun user hanya bisa memilih satu diantara pilihan group radio. Type radio ini berada dalam suatu grup dan user hanya bisa memilih salah satunya. Contoh inputan type radio adalah jenis kelamin.
- `<input type="submit" />` akan menampilkan tombol untuk memproses form. Biasanya diletakkan pada baris terakhir dari form. Atribut value jika diisi akan membuat text tombol submit berubah sesuai inputan nilai value.
- `<input type="date" />` menyatakan jenis inputnya tipe nya berupa tanggal yang akan tampil adalah kalender



Mengenal tag <textarea>

Tag textarea pada dasarnya sama dengan input type text, namun lebih besar dan dapat berisi banyak baris. Panjang dan banyak baris untuk text area di atur melalui atribut rows dan cols, atau melalui CSS.

Contoh penggunaan textarea adalah sebagai berikut:

```
<textarea rows="5" cols="20">
  Text yang diisi dapat mencapai banyak baris
</textarea>
```

Mengenal tag <select>

Tag select digunakan untuk inputan yang telah tersedia nilainya, dan user hanya dapat memilih dari nilai yang ada. Tag select digunakan bersama-sama dengan tag option untuk membuat box pilihan.

Contoh penggunaan tag select adalah sebagai berikut:

```
<select>
  <option value="pilihan kesatu" selected>Pilihan 1</option>
  <option value="pilihan kedua">Pilihan 2</option>
  <option value="pilihan ketiga">Pilihan 3</option>
</select>
```

Atribut value berisi nilai, maka nilai value-lah yang akan dikirim. Ada atau tidaknya atribut value ini tidak akan tampak dalam tampilan form.

Tag select yang memiliki atribut **selected** akan ditampilkan paling utama ketika form dibuka.

Mengenal Atribut: Name

Setiap tag inputan di dalam form harus ditambahkan atribut **name** agar dapat diproses oleh web server nantinya. Di dalam halaman proses (yang biasanya berupa bahasa PHP atau ASP), nilai dari atribut name inilah yang akan menjadi variabel form.

```
<input type="text" name="username">
<input type="text" name="email">
```

Kedua input diatas akan tampak sama persis, namun pada saat pemrosesan data, masing-masing akan dibedakan menurut atribut name.



Contoh Utuh Form

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Membuat Form</title>
</head>
<body>
  <form action="proses_form.php" method="get">
    Nama: <input type="text" name="nama" />
    <br />
    Password: <input type="password" name="password" />
    <br />
    Jenis Kelamin :
    <input type="radio" name="jenis_kelamin" value="laki-laki" checked />
    Laki - Laki
    <input type="radio" name="jenis_kelamin" value="perempuan" />
    Perempuan
    <br />
    Hobi: <input type="checkbox" name="hobi_ngoding" /> Ngoding
         <input type="checkbox" name="hobi_nulis" checked /> Menulis
         <input type="checkbox" name="hobi_mancing" /> Memancing
    <br />
    Asal Kota:
    <select name="asal_kota" >
      <option value="Kota Jakarta"> Jakarta</option>
      <option value="Kota Bandung">Bandung</option>
      <option value="Kota Semarang" selected>Banjarmasin</option>
    </select>
    <br />
    Komentar Anda:
    <textarea name="komentar" rows="5" cols="20">
    Isi text
    </textarea>
    <br />
    <input type="submit" value="Mulai Proses!" >
  </form>
</body>
</html>
```

Copy dan buat file **form.html**,



Nama:
 Password:
 Jenis Kelamin : ☒ Laki - Laki ☐ Perempuan
 Hobi: ☐ Ngoding ☒ Menulis ☐ Memancing
 Asal Kota:
 Isi text
 Komentar Anda:
 Mulai Proses!

Ini adalah hasil dari form.html

2.1.11 Tag HTML & Kegunaannya

Nama Tag	Keterangan / Kegunaan
<!DOCTYPE>	Tag untuk menentukan tipe dokumen
<html>	Tag untuk membuat sebuah dokumen HTML
<title>	Tag untuk membuat judul dari sebuah halaman
<body>	Tag untuk membuat tubuh dari sebuah halaman
<h1> to <h6>	Tag untuk membuat heading
<p>	Tag untuk membuat paragraf

	Memasukan satu baris putus
<hr>	Tag untuk membuat perubahan dasar kata didalam isi
<!--...-->	Tag untuk membuat komentar
Formatting	
<acronym>	Tag untuk membuat sebuah akronim (tidak disupport lagi di HTML5)
<abbr>	Tag untuk membuat sebuah singkatan
<address>	Tag untuk membuat kontak alamat
	Tag untuk membuat huruf bercetak tebal
<bdi>	Mengisolasi bagian dari teks yang dapat diformat dalam arah yang berbeda dari teks lain di luarnya (tag baru HTML5)
<bdo>	Mengganti arah teks
<big>	Tag untuk membuat text berhuruf besar (tidak disupport lagi di HTML5)
<blockquote>	Tag untuk membuat sebuah bagian text yang dikutip dari sumber lain
<center>	Tag untuk membuat jajaran teks menjadi ditengah (tidak disupport lagi di HTML5)
<cite>	Tag untuk membuat judul karya
<code>	Tag untuk membuat potongan kode komputer di antara text
	Tag untuk membuat teks yang telah dihapus dari dokumen
<dfn>	Tag untuk membuat sebuah istilah definisi
	Tag untuk membuat penekanan teks (tidak disupport lagi di HTML5)

	Tag untuk membuat font, warna, dan ukuran untuk teks (tidak disupport lagi di HTML5)
<i>	Tag untuk membuat sebuah bagian dari teks yang disesuaikan dengan mood
<ins>	Tag untuk membuat teks yang telah dimasukkan ke dalam dokumen
<kbd>	Tag untuk membuat input keyboard
<mark>	Tag untuk membuat teks yang disorot / ditandai (tag baru HTML5)
<meter>	Tag untuk membuat pengukuran skalar
<pre>	Tag untuk membuat teks terformat
<progress>	Memperlihatkan kemajuan tugas (tag baru HTML5)
<q>	Tag untuk membuat kutipan pendek
<rp>	Tag untuk membuat apa yang harus ditampilkan di browser yang tidak mendukung penjelasan ruby (tag baru HTML5)
<rt>	Tag untuk membuat sebuah anotasi / pengucapan karakter (untuk tipografi Asia Timur)
<ruby>	Tag untuk membuat sebuah anotasi ruby (untuk tipografi Asia Timur) (tag baru HTML5)
<s>	Tag untuk membuat teks yang tidak lagi benar
<samp>	Tag untuk membuat contoh keluaran dari program komputer
<small>	Tag untuk membuat teks kecil
<strike>	Tag untuk membuat teks yang di coret tengah (tidak disupport lagi di HTML5)
	Tag untuk membuat teks penting
<sub>	Tag untuk membuat teks subskrip (seperti dalam penulisan Jat Kimia)
<sup>	Tag untuk membuat teks superscripted (seperti dalam penulisan akar kuadrat)
<time>	Tag untuk membuat tanggal / waktu (tag baru HTML5)
<tt>	Tag untuk membuat teks teletype (tidak disupport lagi di HTML5)
<u>	Tag untuk membuat teks yang memiliki Gaya yang berbeda dari teks biasa lainnya
<var>	Tag untuk membuat sebuah variabel
<wbr>	Tag untuk membuat kemungkinan garis-putus
Forms	
<form>	Tag untuk membuat sebuah form HTML untuk input pengguna
<input>	Tag untuk membuat sebuah kontrol input
<textarea>	Tag untuk membuat sebuah kontrol input multibaris (text area)
<button>	Tag untuk membuat sebuah tombol yang dapat diklik
<select>	Tag untuk membuat sebuah daftar drop-down
<optgroup>	Tag untuk membuat sebuah kelompok pilihan yang terkait dalam daftar drop-down
<option>	Tag untuk membuat pilihan dalam daftar drop-down
<label>	Tag untuk membuat sebuah label untuk sebuah elemen <input>
<fieldset>	Grup unsur terkait dalam bentuk
<legend>	Tag untuk membuat sebuah caption untuk sebuah elemen <fieldset>, <figure>, atau <details>
<datalist>	Menentukan daftar pilihan yang telah ditetapkan untuk kontrol input (tag baru HTML 5)



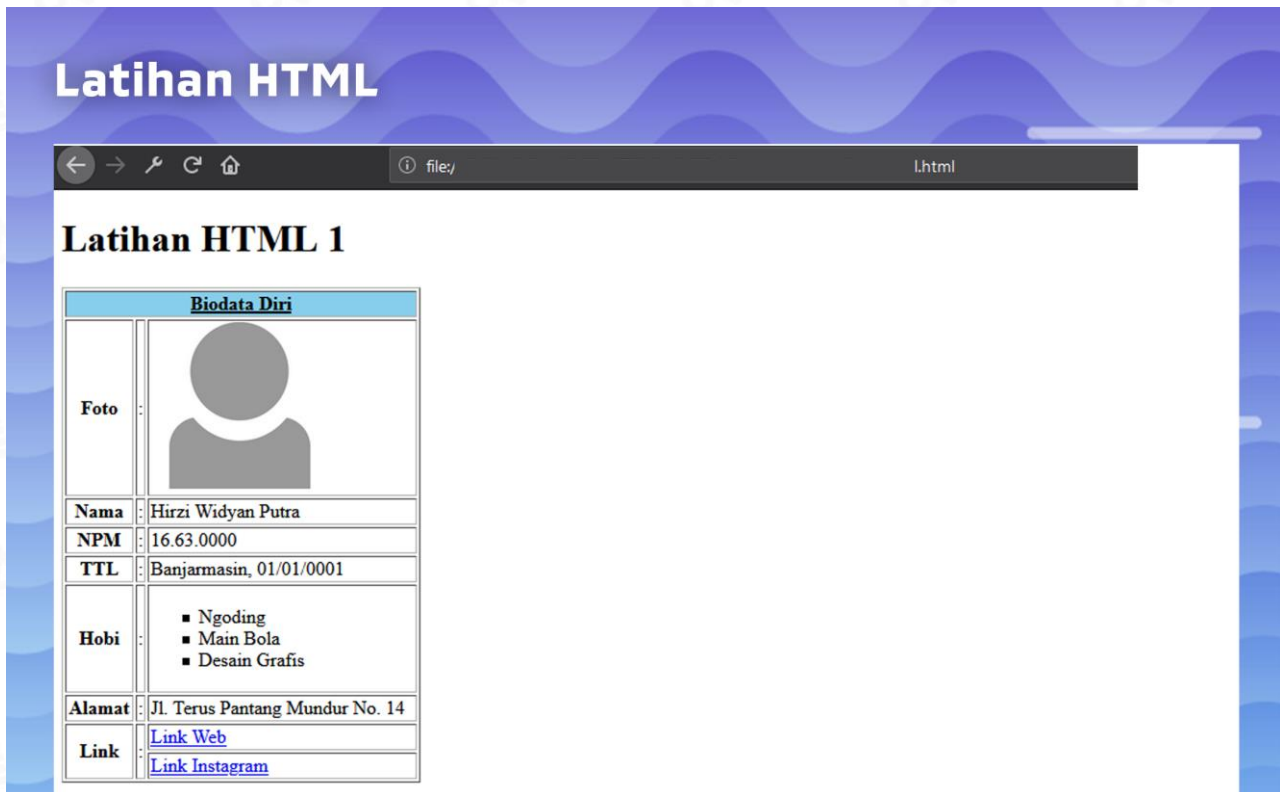
<keygen>	Tag untuk membuat key-pair generator kolom input (tag baru HTML5)
<output>	Tag untuk membuat hasil penghitungan (tag baru HTML5)
Frames	
<frame>	Tag untuk membuat sebuah window (bingkai) dalam sebuah frameset (tidak disupport lagi di HTML5)
<frameset>	Tag untuk membuat satu set bingkai (tidak disupport lagi di HTML5)
<noframes>	Tag untuk membuat sebuah konten alternatif untuk pengguna yang tidak mendukung frame (tidak disupport lagi di HTML5)
<iframe>	Tag untuk membuat sebuah bingkai
Images	
	Tag untuk membuat gambar
<map>	Tag untuk membuat gambar-peta
<area>	Tag untuk membuat area dalam gambar-peta
<canvas>	Digunakan untuk menggambar grafik, melalui scripting (JavaScript) (tag baru HTML5)
<figcaption>	Tag untuk membuat sebuah caption untuk elemen <figure> (tag baru HTML5)
<figure>	Menentukan konten mandiri (tag baru HTML5)
Audio/Video	
<audio>	Tag untuk membuat isi suara (tag baru HTML5)
<source>	Tag untuk membuat sumber beberapa media untuk elemen media (<video> dan <audio>) (tag baru HTML5)
<track>	Tag untuk membuat trek teks untuk elemen media (<video> dan <audio>) (tag baru HTML5)
<video>	Tag untuk membuat sebuah video atau film (tag baru HTML5)
Links	
<a>	Tag untuk membuat hyperlink
<link>	Tag untuk membuat hubungan antara dokumen dan sumber daya eksternal (paling sering digunakan untuk link ke style sheet)
<nav>	Tag untuk membuat navigasi link (tag baru HTML5)
Lists	
	Tag untuk membuat daftar dengan selain nomor
	Tag untuk membuat daftar dengan nomor
	Tag untuk membuat sebuah item daftar
<dir>	Tag untuk membuat sebuah daftar direktori (tidak disupport lagi di HTML5)
<dl>	Tag untuk membuat sebuah daftar definisi
<dt>	Tag untuk membuat istilah (item) dalam daftar definisi
<dd>	Defines a description of an item in a definition list
<menu>	Tag untuk membuat deskripsi dari item dalam daftar definisi
<command>	Tag untuk membuat sebuah tombol perintah bahwa seorang pengguna dapat meminta (tag baru HTML5)
Tables	
<table>	Tag untuk membuat tabel
<caption>	Tag untuk membuat sebuah caption tabel
<th>	Tag untuk membuat sebuah sel header tabel
<tr>	Tag untuk membuat baris dalam sebuah tabel
<td>	Tag untuk membuat sel dalam sebuah tabel
<thead>	Mengelompokan isi header dalam sebuah tabel
<tbody>	Mengelompokan isi tubuh dalam sebuah tabel
<tfoot>	Mengelompokan isi footer dalam sebuah tabel



<col>	Menentukan properti kolom untuk setiap kolom dalam elemen <colgroup>
<colgroup>	Menentukan kelompok dari satu atau lebih kolom dalam sebuah tabel untuk diformat
Style/Sections	
<style>	Tag untuk membuat informasi style untuk dokumen
<div>	Tag untuk membuat sebuah bagian dalam dokumen
	Tag untuk membuat sebuah bagian dalam dokumen
<header>	Tag untuk membuat sebuah header untuk dokumen atau bagian (tag baru HTML5)
<footer>	Tag untuk membuat footer untuk dokumen atau bagian (tag baru HTML5)
<hgroup>	Pengelompokan elemen heading (<h1> sampai <h6>) (tag baru HTML5)
<section>	Tag untuk membuat bagian dalam dokumen (tag baru HTML5)
<article>	Tag untuk membuat sebuah artikel (tag baru HTML5)
<aside>	Tag untuk membuat konten lain selain dari konten halaman (tag baru HTML5)
<details>	Tag untuk membuat rincian tambahan yang pengguna dapat lihat atau sembunyikan (tag baru HTML5)
<dialog>	Tag untuk membuat sebuah kotak dialog atau jendela (tag baru HTML5)
<summary>	Tag untuk membuat sebuah judul terlihat untuk elemen <detail> (tag baru HTML5)
Meta Info	
<head>	Tag untuk membuat informasi tentang dokumen
<meta>	Tag untuk membuat metadata tentang dokumen HTML
<base>	Menentukan URL dasar / target untuk semua URL relatif dalam dokumen
<basefont>	Menentukan standar warna, ukuran, dan font untuk semua teks dalam dokumen (tidak disupport lagi di HTML5)
Programming	
<script>	Tag untuk membuat script di sisi klien
<noscript>	Tag untuk membuat sebuah konten alternatif bagi pengguna yang tidak mendukung script di sisi klien
<applet>	Tag untuk membuat sebuah java applet yang ditanam (tidak disupport lagi di HTML5)
<embed>	Tag untuk membuat sebuah wadah untuk aplikasi eksternal (non-HTML) (tag baru HTML5)
<object>	Tag untuk membuat sebuah objek yang ditanam
<param>	Tag untuk membuat sebuah parameter untuk objek



2.1.12 Latihan HTML



Note :

- Beri nama dengan **npm.html**
- Pastikan pada latihan ini terdapat beberapa tag untuk [tabel](#), [rowspan](#), [colspan](#), [list](#), [gambar](#), [link](#), [heading](#), [garis bawah](#).



2.2 CSS

2.2.1 Aturan penulisan CSS

Dalam penulisan CSS ada aturan yang digunakan, pada CSS ada komponen utama dalam penulisannya yaitu **Selektor** dan **Deklarator**. Perhatikan contoh dasar penulisan CSS dibawah ini.

Selektor { Deklarator };

Selektor merupakan Tag dari penulisan HTML semisal `H1,H2,p,b,div, span` dll **Deklarator** adalah yang memerintahkan browser untuk membuat tampilan pada selector sesuai dengan perintah yang ada pada deklarator. Sedangkan deklarator terdiri dari **Property** dan **Value**.

Contoh kecil : Biasanya pada penulisan artikel banyak digunakan Tag `<p>` atau paragraph, jika anda ingin merubah penulisan font dari artikel anda pada paragraph, pada CSS penulisannya adalah sebagai berikut :

```
P {font-family: verdana};
```

P Merupakan selektor dari paragraph dan yang berada dalam kurung kurawal adalah deklaratornya yang memerintahkan browser untuk menampilkan artikel paragraph dengan font verdana. Sedangkan **font-family** adalah **Property** dan **verdana** adalah **Value** Detailnya rumusnya sebagai berikut ini :

SELECTOR		PROPERTY		VALUE
↓		↓		↓
p	{	color	:	blue; }

Rumus diatas merupakan kunci kamu untuk menguasai CSS, dan terus di ingat. Sebenarnya CSS itu sangat sederhana sekali dan simple loh hehe

2.2.2 Menghubungkan HTML dan CSS

Untuk Cara menghubungkan HTML dengan CSS caranya sangat mudah. Gunakan tag `<link>` untuk menghubungkan html dengan css. tag `link` di letakkan pada bagian element head pada struktur HTML.

Buat sebuah file html dengan nama **index.html** kemudian buat sebuah file css lagi dengan nama **tampilan.css** , lalu **simpan pada Folder latihan1** jadi ada dua buah file yang dibuat yaitu file `index.html` dan file `tampilan.css`



index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Menghubungkan HTML dengan CSS</title>
  <link rel="stylesheet" type="text/css" href="tampilan.css" />
</head>
<body>
  <h1>Belajar menghubungkan HTML dengan CSS</h1>
</body>
</html>
```

tampilan.css

```
h1 {
  color: #0288d1; /*ini untuk memberi warna*/
  font-family: "Segoe UI"; /*untuk ganti font*/
  text-align: center; /*membuat text ketengah*/
}
```

Penjelasan :

- **Color** : untuk memberi warna font
- **Font-family** : mengganti jenis font
- **Text-align** : (left, right, center) membuat text berada di kiri, kanan atau tengah.
- **/*ini merupakan komentar saja untuk CSS*/**

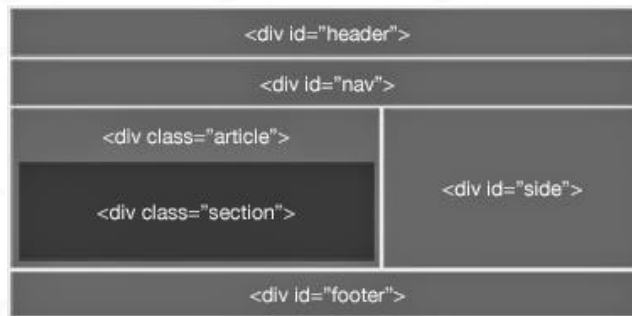
atribut **rel="stylesheet"** dan **type="text/css"** merupakan deklarasi untuk menjelaskan bahwa file yang kita panggil atau hubungkan adalah file stylesheet atau css.

Pada atribut **href** digunakan untuk meletakkan lokasi dan nama file css yang ingin kita hubungkan. Contoh ini saya menghubungkan file html saya dengan file css yang sudah saya berinama dengan **tampilan.css**



2.2.3 Tag Span dan Div pada HTML

Tag `` `` dan tag `<div>` `</div>` adalah tag yang tidak memiliki makna apa-apa. Tapi Tag `` dan tag `<div>` menjadi salah satu tag yang paling sering digunakan untuk membuat struktur web.



Sebagai contohnya, saat kita membuka sebuah halaman website atau blog. Maka kita akan dipertemukan dengan tampilan yang biasanya berisi logo, slogan, nama website, menu dan lainnya pada bagian atas. Nah, elemen tersebut biasanya dikelompokkan dalam sebuah tag bernama Division (Div).

Div biasanya digunakan untuk membuat layer yang akan memudahkan kita untuk membuat layout sesuai dengan design yang di inginkan. Untuk membuat layer kita menggunakan tag Div dan diberi atribut **ID** ataupun **Class**.

2.2.4 Mengenal Class dan ID pada HTML

Class dan ID bisa di bilang sebagai penanda pada html, maksud penanda di sini adalah element-element html dapat di beri tanda dengan class atau id. untuk apa di beri tanda? Agar dapat di manipulasi menggunakan css atau javascript.

Pada CSS perbedaan class dan ID yaitu :

- Class menggunakan **.kotak** (titik didepan)
- ID menggunakan **#kotak** (tanda # didepan)

Contoh sederhananya jika kamu memiliki lima buah kotak, semua kotak yang kamu miliki berwarna biru, kemudian kamu ingin mengubah warna kotak yang ketiga, nah di sinilah letak kegunaan class dan id, untuk memberikan tanda atau nama pada kotak kamu agar bisa di ubah dan kotak yang lain tidak akan berubah.



ID : Atribut ini digunakan untuk penamaan elemen pada HTML yang memiliki karakteristik yang unik atau berbeda. Tidak diperkenankan ada dua elemen yang memiliki ID yang sama. Salah penggunaan ID, maka akan berdampak pada hasil tampilan web itu sendiri.

Contoh Penggunaan ID Salah :

```
<div id="menuku">
  <ul id="menuku">
    <li>Home</li>
    <li>About Us</li>
    <li>Contact Us</li>
  </ul>
</div>
```

Contoh benar :

```
<div id="menuku">
  <ul id="menu-list">
    <li>Home</li>
    <li>About Us</li>
    <li>Contact Us</li>
  </ul>
</div>
```

Class : Digunakan untuk memberikan penamaan elemen yang memiliki karakteristik atau struktur yang dapat digunakan secara berulang-ulang dalam tag HTML. Sebagai contohnya, silahkan perhatikan atribut Class pada script HTML berikut ini :

```
<div class="menu">
  <ul>
    <li class="menu-merah">Home</li>
    <li>About Us</li>
    <li class="menu-merah">Contact Us</li>
  </ul>
</div>
```



Contoh Penggunaan ID dan Class :

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh ID Dan Class</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <h1>Contoh ID Dan Class</h1>
  <div class="kotak">Kotak 1</div>
  <div class="kotak">Kotak 2</div>
  <div class="kotak">Kotak 3</div>

  <div id="kotak">Kotak 4</div>
</body>
</html>
```

Beri nama **index.html** pada folder **latihan2**, lalu buat lagi **style.css** di folder yang sama, berikut isi dari **style.css**

```
h1{
  color: orange;
  font-family: "Segoe UI";
}
.kotak{
  padding: 50px; /*memberi jarak dalam pada kotak*/
  width: 100px; /*membuat lebar kotak 100px*/
  height: 50px; /*membuat tinggi kotak 100px*/
  color: #fff; /*memberi warna pada text / font */
  margin: 10px; /* memberi jarak luar kotak */
  background: #0288d1; /*memberi warna kotak */
  float: left; /*membuat kotak jadi kesamping kiri*/
}
#kotak {
  padding: 50px;
  width: 100px;
  color: #666666;
  height: 50px;
  margin: 10px;
  background: #f8bbd0;
  float: left;
}
```

Note :

- Untuk komentar tidak perlu diketik lagi, cukup di pahami maksud nya saja.
- Tanda **#**(pagar) untuk **ID**, tanda **.**(titik) untuk **Class**



Hasilnya :

Contoh ID Dan Class

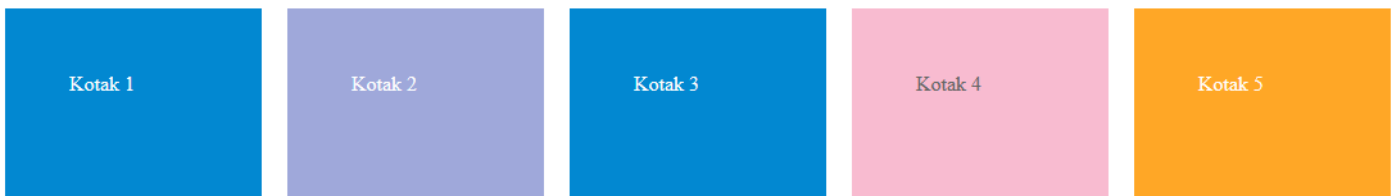


Lalu jika sudah silahkan latihan buat file **latihan.html & style.css**, simpan pada folder **latihan3**

Note :

- Warna tidak harus sama persis, tapi perbedaan warna harus seperti contoh ini.
- Posisi : **biru, ungu, biru, pink, orange.**

Latihan ID Dan Class





2.2.5 CSS Inline, Internal dan External

Ada tiga teknik metode penulisan CSS, yaitu :

- **Inline CSS Style**
- **Internal CSS Style**
- **External CSS Style**

Tapi perlu diketahui, **Css Inline**, **Internal** maupun **External** bisa dikolaborasikan menjadi sebuah halaman web sesuai dengan keperluan. Kelebihan External yaitu Style website kita menjadi terpusat dan mudah untuk diedit.

Inline CSS

Penulisan Inline style adalah teknik cara penulisan syntax CSS yang tidak memerlukan selector(id dan class).

Sehingga syntax CSS di letakkan atau langsung disisipkan pada element HTML. syntax CSS di letakkan di dalam atribut style=""

Buat file baru **contohinline.html** simpan di folder **latihan4**

```
<!DOCTYPE html>
<html>
<head>
  <title>Penulisan CSS dengan Inline Style</title>
</head>
<body>
  <h1 style="color:blue;text-align: center;">Ini merupakan contoh Inline</h1>
</body>
</html>
```

Penjelasan :

Pada contoh di atas syntax css di letakkan dalam element h1 menggunakan atribut **style=""**

Perintah color adalah perintah css yang berfungsi untuk mengatur warna font. Jadi pada contoh inline css style ini kita membuat heading h1 dengan warna font yang kita atur menjadi warna biru (Blue).



Internal CSS

Teknik penulisan syntax css yang di letakkan satu file dengan file html atau file php. Syntax CSS di letakkan di dalam tag `<style>` dan di akhiri dengan tag `</style>`. Biasa nya tag `<style>` .. `</style>` di letakkan pada bagian antara tag `<head>` dan sebelum tag `</head>` pada HTML.

Contoh Internal CSS :

Masih di folder **latihan4**, beri nama file **contohinternal.html**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Penulisan CSS dengan Internal Style</title>
    <style type="text/css">
      .kotak {
        background: #0288d1;
        padding: 10px;
      }
      .tulisan{
        color: #f8bbd0;
      }
    </style>
  </head>
  <body>
    <div class="kotak">
      <h1 class="tulisan">
        Ini merupakan contoh CSS Internal.
      </h1>
    </div>
  </body>
</html>
```

Penjelasan :

Perhatikan contoh penulisan internal css style di atas. Syntax css di letakkan satu file dengan file html. Syntax css di letakkan di dalam tag `<style>` diantara tag `<head>`

Pada contoh diatas kita membuat kotak dan text h1 menjadi berwarna.

Hasilnya :

Ini merupakan contoh CSS Internal.



External CSS

Teknik penulisan External CSS yaitu dengan memisahkan file css dan html. penggunaan css yang baik adalah menggunakan teknik penulisan CSS ini karena kode program kita tidak akan berantakan karena syntax css di simpan pada file css. File css dan html di hubungkan menggunakan :

```
<link rel="stylesheet" type="text/css" href="file css anda">
```

Sebelumnya kita sudah membuat latihan1, 2 dan 3. Itu semua menggunakan CSS External.

Buat File **contohexternal.html** dan **style.css** pada folder **latihan4**

contohexternal.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Penulisan CSS dengan External Style</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <div class="kotak">
      <h1 class="tulisan">
        Ini merupakan contoh CSS External.
      </h1>
    </div>
  </body>
</html>
```

style.css

```
.kotak {
  background: #039be5;
  padding: 10px;
}
.tulisan {
  color: #ffea00;
}
```

Hasilnya :

Ini merupakan contoh CSS External.