**Hyper Text Markup Language**

### Sejarah Singkat HTML

  Pada tahun 1980 ahli fisika Tim Berners-Lee, dan juga seorang kontraktor di CERN (Organisasi Eropa untuk Riset Nuklir) mengusulkan dan menyusun ENQUIRE, sebuah sistem untuk ilmuwan CERN dalam membagi dokumen.

Sembilan tahun kemudian Berners-Lee mengusulkan adanya sistem markah berbasis internet. Dan pertama kalinya pada tahun 1991 sebuah dokumen dengan nama “Tanda HTML”, disebutkan di Internet oleh Tim Berners-Lee.

HTML*(Hyper Text Markup Language)* merupakan sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan disebut dengan SGML *(Standard Generalized Markup Language),* HTML merupakan sebuah standar yang digunakan secara luas untuk menampilkan halaman web. Saat ini HTML merupakan standar Internet yang didefinisikan dan dikendalikan oleh *World Wide Web Consortium* (W3C). Berikut adalah rincian perkembangan HTML :

1. Tahun 1980 mengembangkan sebuah bahasa pemograman yang dimana teks dan format dokumen dijadikan jadi satu yang sekarang disebut dengan HTML, tetapi saat itu IBM memberikan sebuah nama GML (*Generalized Markup Language).*
2. Tahun 1986 GML diganti namanya oleh ISO dengan SGML (Standard Generalized Markup Language).
3. Tahun 1989 Caillau Tim dengan Banners Lee Robert yang bekerja di CERN memberi nama HTML, dan mempopulerkanya pertama kali dengan browser Mosaic.
4. Tahun 1990 HTML sangat berkembang sangat cepat hingga mencapai HTML versi 5.0 yang digarap pada 4 Maret 2010 oleh W3C.
5. Tahun 1996 tanggal 14 Januari, HTML versi 2.0 versi ini menjadikan sebuah pioneer dalam perkembangan homepage interaktif.
6. Tahun 1997 tanggal 18 Desember, HTML versi 3.0, sering disebut sebut sebagai HTML+ yang mempunyai penambahan fitur table dalam paragraph, akan tetapi versi ini tidak bertahan lama.
7. Tahun 1996 bulan Mei , dikeluarkan HTML versi 3.2 digunakan oleh pengembang browser yang dapat diterima, dan dikatakan bahwa versi 3.2 ini merupakan versi 3.0 yang dikembangkan oleh beberapa pengembang browser seperti Netscape dan Microsoft.
8. Tahun 1999 tanggal 24 Desember yaitu HTML versi 4.0 sebagai penyempurnaan versi 3.2.
9. Tahun 2010 tanggal 4 Maret, HTML versi 5.0 yang dikembangkan oleh W3C dan IETF (*Internet Engineering Task Force*) yaitu sebuah organisasi yang menangani HTML sejak versi 2.0.

Tanda ini menggambarkan 18 elemen awal mula, versi sederhana dari HTML, yang masih ada didalam HTML4.

1. Sejarah Versi HTML:

·         HTML 2.0 dipublikasikan pada tanggal 24 November 1995 sebagai IETF  
·         HTML 3.2 dipublikasikan pada tanggal 14 Januari 1976 sebagai Konsorium World Wide Web  
·         HTML 4.0 dipublikasikan pada tanggal 18 Desember 1997 sebagai W3C Recommendation  
·         HTML 4.01 dipublikasikan pada tanggal 05 Mei 2000, yang distandarkan dengan ISO HTML  
·         HTML 5 (stabil) dipublikasikan pada tanggal 28 Oktober 2014, dan menjadi standard HTML tertinggi saat ini

1. Didalam HTML 5 terbaru, banyak perubahan yang terjadi dari versi sebelumnya. Fitur baru yang sudah diterapkan antara lain: penanganan eror lebih banyak, Scripting diganti dengan lebih banyak markup, HTML5 menjadi device-dependent, proses pengembangnya bisa dilihat oleh public.
2. Dan fitur elemen HTML5 baru yang lain:

·         Elemen <canvas> untuk menggambar 2D.  
·         Elemen <video> dan <audio> untuk memutar multimedia.  
·         Elemen seperti <article>, <header>, <nav>, <section>, <footer>.  
·         Kontrol di dalam Form seperti:  tanggal, kalender, nomor, email.

### Tag-tag HTML

Tag adalah suatu penanda untuk menandai elemen-elemen dalam suatu dokumen HTML dan Fungsi Tag adalah untuk memberikan instruksi atau memberitahu kepada browser bagaimana suatu objek di tampilkan berdasarkan Tag yang di gunakan, objek disini bisa berupa teks, video, audio dan gambar.

**1.2.1 Elemen**

Pengertian elemen pada HTML adalah satu blok code yang dimulai dari tag pembuka hingga tag penutup, namun ada juga tag yang dalam penulisannya berdiri sendiri seperti <nama\_tag/> tag tersebut merupakan contoh dari elemen HTML yang tidak memiliki tag penutup atau disebut dengan Void Element. **Contoh penggunaan *Void Element***

<img src="url\_gambar.png"/>

**1.2.2 Atribut**

lemen, Tag dan atribut pada HTML sebenarnya saling berkaitan, hanya saja disini atribut memiliki tugas khusus untuk memberikan informasi atau sifat tambahan yang akan diberikan kepada tag dan elemen yang mengandung atribut tersebut.  Penulisan Atribut selalu diletakan dalam <tag\_pembuka> setelah nama tag, sebagai contoh: <tag\_pembuka name="value">.

**1.2.3 Heading**

Heading adalah sekumpulan kata yang menjadi judul atau sub judul dalam dokumen HTML, dengan berbagai ukuran yang berbeda. Untuk menyusun Heading dan sub-heading dengan HTML, kita akan menggunakan format heading yang telah disediakan.

Berikut contoh Tag Heading :

<html>  
<head>  
<title> **Pengertian Dan Fungsi Tag Heading** </title>  
</head>  
<body>  
<h1> Ukuran Heading 1</h1>  
<h2> Ukuran Heading 2</h2>  
<h3> Ukuran Heading 3</h3>  
<h4> Ukuran Heading 4</h4>  
<h5> Ukuran Heading 5</h5>  
<h6> Ukuran Heading 6</h6>  
</body>  
</html>

**1.2.4 PARAGRAPHS**

*Tag Paragraf adalah tag yang berfungsi untuk membuat paragraf pada dokumen HTML.* Tag ini dituliskan dengan <p> dan ditutup dengan </p>.

*<p>…</p>*

**1.2.5 HTML STYLE**

**HTML**<style>**element** digunakan untuk menyisipkan kode style atau CSS ke dalam sebuah dokumen web (HTML).Contoh

<style media="screen and (max-width: 768px)">

p {color: blue;}

</style>

**1.2.6 HTML FORMATTING**

Text Formatting yaitu melakukan peraturan / konfigurasi pada teks yang akan dijadikan isi dari sebuah halaman web. Pengaturan tersebut akan terlihatsecara real, sehingga apa yang diinput pada document HTML akan diperlihatkan oleh web browser. Pengaturan ini format dari teks meliputi pengaturan heading ( penjudulan pada sebuah teks / paragraf ) aligment ( pengaturan atau perataan pada suatu teks / paragraf ), dan juga pengaturan font ( menagtur jenus, ukuran dan warna font pada suatu teks / paragraph ).

**1.2.7 COMMENTS**

Meletakan komentar pada halaman web tidak akan Nampak pada browser.

**1.2.8 COLORS**

Warna Font dapat diatur pada BODY dan juga tag FONT. Untuk warna latar belakang teks dapat ditentukan dengan menggunakan atribut BGCOLOR pada tag BODY. Dan untuk warna latar depan teks diatur melalui atribut COLOR pada tag FONT.  
  
Sebagai contoh perhatikan potongan script berikut :

|  |
| --- |
| <HTML><HEAD><TITLE=”Halaman web dengan HTML”></TITLE></HEAD><BODY bgcolor="#00FF00"><font color="blue">Mari belajar HTML..gampang kan?</font></BODY></HTML> |

**1.2.9 LINKS**

**HTML**<link>**element** menunjukkan sebuah hubungan (link) antara dokumen (HTML) yang bersangkutan dengan sumber file dari luar (eksternal).

**1.2.10 IMAGES**

**HTML**<img>**element** memiliki beragam attribute yang dapat ditulis sesuai kebutuhan. Akan tetapi, hanya dua attribute yang wajib ditulis pada setiap <img /> element, yaitu src dan altattribute. src attribute menujukkan sumber file tersebut berada, sedangkan alt menujukkan alternatif text yang akan muncul sebagai pengganti apabila gambar tersebut tidak dapat ditampilkan.

**1.2.11 TABLES**  
Untuk membuat tabel kaya di MS.excel tapi cara membuat tabelnya

contoh :

<table border="1">  
 **<tr>**  
         **<th>**No**</th>**  
         **<th>**nama Siswa**</th>**  
  **</tr>**  
 **<tr>**  
         **<td>**1.**</td>**  
         **<td>**mu'alif lihawa**</td>**  
**</tr>**  
</table>

**1.2.12 LISTS**

Bagian teks yang didalam dokumen yang berisi daftar item dari suatu kelompok atau grup atau bisa juga di artikan sebagai prosedur(urutan langkah-langkah) dari suatu pekerjaan tertentu yang sifatnya harus dilakukan secara berurutan dari langkah pertama hingga terahir.

**1.2.13 UNORDERED LIST**

Seperti namanya, memungkinkan kita untuk membuat daftar dari hal-hal yang tidak memiliki urutan tertentu. Contohnya : **<ul>**

**<li>**Windows 8**</li>**

**<li>**Microsoft Office 2013**</li>**

**<li>**Visual Studio 2012**</li>**

**</ul>**

**1.2.14 ORDERED LIST**

Ordered list membuat daftar yang terurut. Elemen untuk pembuatan ordered list yaitu ol (**o**rdered **l**list), dan isi dari list sendiri dibuat dengan menggunakan elemen li, sama seperti pada unordered lsit. Secara standar ordered list akan menggunakan angka sebagai penanda daftar:

**<ol>**

Ordered List

**<li>**Windows 8**</li>**

**<li>**Microsoft Office 2013**</li>**

**<li>**Visual Studio 2012**</li>**

**</ol>**

**1.2.15 DEFINITION LIST**

Daftar yang berisi kumpulan definisi dari suatu istilah tertentu, ataupun membuat daftar definisi mirip seperti di kamus. Tiga buah pasang tag yang terikat dengan definiton list. Perhatikan contoh pembuatan sebuah definition list berikut:

**<dl>**

**<dt>**study**</dt>**

**<dd>**

the devotion of time and attention to acquiring knowledge

on an academic subject, esp. by means of books **</dd>**

**<dt>**design**</dt>**

**<dd>**

a plan or drawing produced to show the look and function

or workings of a building, garment, or other object before

it is built or made

**</dd>**

**<dd>**

purpose, planning, or intention that exists or is thought

to exist behind an action, fact, or material object

**</dd>**

**<dt>**business**</dt>**

**<dt>**work**</dt>**

**<dd>**a person's regular occupation, profession, or trade**</dd>**

**</dl>**

**1.2.16 NESTED LIST**

Salah satu fitur list yang paling menarik dari HTML ialah kemampuan untuk membangun list di dalam list. Sebuah ordered list dapat dituliskan kembali di dalam unordered list, yang keduanya berada di dalam definiton list, ataupun sebaliknya. Sama sekali tidak ada batasan dalam menuliskan list di dalam lsit, selain permasalahan makna semantik tentunya.

Menuliskan kode untuk list di dalam list juga sangat sederhana, dengan langsung memasukkan list yang ingin ditambahkan ke dalam elemen li, seperti berikut:

**<ol>**

**<li>** HTML dan CSS Dasar

**<ul>**

**<li>**

Elemen, Tag, dan Atribut

**<ol>**

**<li>**Elemen**</li>**

**<li>**Tag**</li>**

**<li>**Atribut**</li>**

**</ol>**

**</li>**

**<li>**Struktur Dokumen HTML**</li>**

**<li>**

Dasar CSS

**<ol>**

**<li>**Selector**</li>**

**<li>**Property**</li>**

**<li>**Value**</li>**

**<li>**Sintaks CSS**</li>**

**</ol>**

**</li>**

**<li>**Mengimplementasikan CSS pada HTML**</li>**

**</ul>**

**</li>**

**</ol>**

**1.2.17 IFRAMES**

Sebuah iframe berfungsi untuk menampilkan web page(halaman web) pada sebuah web page(halaman web). Maksudnya kita akan menampilkan suatu halaman web tertentu pada web kita. Tag yang digunakan untuk menampilkan iframe adalah <iframe> dan penutupnya </iframe>.

Cara penulisannya :

<iframe src="URL"></iframe>

**1.2.18 JAVASCRIPT**

**Javascript** adalah program dalam bentuk script, yang akan dijalankan oleh interpreter yang telah ditanamkan ke dalam browser web, sehingga browser web dapat mengeksekusi program **Javascript**. Program **Javascript** disisipkan ke dalam dokumen HTML dengan ditandai dengan tag yang diawali dengan <Script ...> dan diakhiri dengan </script>.. Contoh Program sederhana **Javascript:**

<html>

<head>

<tittle> Belajar Javascript </title>

</head>

<body bgcolor=#FFFFAA>

<h1> Contoh Program JavaScript </h1>

<script type=”text/javascript>

Document.write(“Hallo, nama saya Javascript”);

</script>

</body>

</html>

**1.2.19 FILE PATHS**

HTML File Paths menggambarkan lokasi file dalam struktur folder situs web. HTML File Paths digunakan saat menghubungkan ke file eksternal seperti:  
     1. Halaman web  
     2. Gambar  
     3. Style sheets  
     4. JavaScripts

Contohnya :

*<!DOCTYPE html>  
<html>  
<body>  
<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain" style="width:300px">  
</body>  
</html>*

**1.2.20 HEAD**

* Tag <head> digunakan untuk bagian header, tempat untuk menyisipkan link file CSS atau Javascript external, dan harus di akhiri dengan </head>

**1.2.21 LAYOUT**

Layout atau susunan kerangka website adalah salah satu unsur pembuatan web yang harus paling diperhatikan, karena tampilan website akan menggambarkan tingkat professional seorang web desainer.

**1.2.22 RESPONSIVE**

Responsive Web Design adalah suatu keadaan sebuah halaman web dimana tampilannya akan cocok, rapi dan tetap enak dilihat jika diakses dari perangkat apapun dengan resolusi layar yang berbeda.

**1.2.23 COMPUTERCODE**

**HTML**<code>**tag** digunakan untuk menunjukkan sebuah bagian dari kode komputer (Computer code). Kode komputer mencakup sebuah nama element XML, nama sebuah file, program komputer, atau kode-kode lainnya yang dapat dikenali oleh komputer. Contohnya, jika menuliskan kode CSS, maka dapat menggunakan nama class seperti: "language-css", sehingga dapat ditulis: <code class="language-css">...</code>.

**1.2.24 ENTITIES**

Entities (entitas) pada HTML digunakan untuk membuat karakter yang tidak mungkin digunakan pada teks dalam HTML, seperti kurang dari (<), lebih dari (>), dan yang lainnya. karakter entitas akan terlihat seperti contoh dibawah ini:

&entity\_name, atau &#entity\_number

**1.2.25 SYMBOL**

Contoh untuk menyajikan rumus matematika maupun rumus fisika tentunya kita memerlukan kode atau karakter khusus tersebut. Kode dan simbol rumus memang tidak disediakan dalam susunan tombol keyboard, sehingga kita tidak bisa mengetiknya secara langsung. Alternatifnya yaitu dengan menambahkan manual kode penggantinya dalam dokumen html.

Contoh simbol yang tidak disediakan pada keyboard :

≤ ⅛ ¼ ∑ dan banyak lagi lainya seperti pada tabel berikut.

Contohnya : Kode berikut digunakan sebagai referensi pustaka simbol-simbol diatas.

<meta http-equiv="content-type" content="text/html;charset=utf-8" />

**1.2.26 CHARSET**

**Meta tag** adalah ‘data tentang data‘, dimana tag ini ditujukan bukan kepada user, tetapi kepada web browser atau kepada ‘robot program‘ seperti mesin pencari (search engine). Charset ***UTF-8*** merupakan meta tag yang paling sering digunakan dalam HTML5, penulisannya adalah sebagai berikut:

|  |  |
| --- | --- |
| 1 | <meta charset="UTF-8"> |

**1.2.27 Character Set ASCII**

ASCII merupakan kependekan dari American Standard Code for Information Interchange, sehingga dapat ditebak bahwa ASCII ini dibuat oleh Amerika, yang memang ditujukan untuk mengakomodasi karakter yang digunakan pada bahasa mereka dan oleh sebab itu ASCII generasi pertama ini sering disebut US-ASCII.

Character Set ASCII hanya terdiri dari 128 karakter yang terdiri dari karakter nyata (huruf, angka, simbol dan tanda baca) dan karakter tidak nyata (tab, enter, alt, dsb), contoh karakter ASCII adalah seperti yang ada pada tombol keyboard yang kita gunakan sekarang ini,

Setiap karakter yang ada pada ASCII cukup ditampung kedalam 7 bit  binary digit, sehingga dapat dikatakan ASCII menggunakan 7 bit sistem encoding, contoh:

|  |  |  |  |
| --- | --- | --- | --- |
| **Karakter** | **Binary** | **Karakter** | **Binary** |
| A | 1100001 | @ | 1000000 |
| B | 1100010 | # | 0100011 |
| C | 1100011 | % | 0100101 |
| 1 | 0110001 | ! | 0100001 |
| 2 | 0110010 | $ | 0100100 |
| 3 | 0110011 | & | 0100110 |

**1.2.28 Character Set Latin-1 atau ISO/IEC 8859-1**

Karakter yang ada pada ISO-8859-1 hanya terdiri dari karakter ASCII ditambah dengan beberapa karakter yang khusus digunakan di bahasa negara-negara Eropa Barat, penambahan ini dimulai dari karakter ke 128 s.d 255. Penggunaan 1 byte ini disebut juga **single byte encoding**. Berikut ini contoh karakter Latin1 dan nilai binary nya:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEXT** | **ASCII** | **BINARY** | **TEXT** | **ASCII** | **BINARY** |
| Ä | chr(228) | 11100100 | ø | chr(248) | 11111000 |
| Å | chr(229) | 11100101 | ù | chr(249) | 11111001 |
| Æ | chr(230) | 11100110 | ú | chr(250) | 11111010 |
| Ç | chr(231) | 11100111 | ý | chr(253) | 11111101 |
| È | chr(232) | 11101000 | þ | chr(254) | 11111110 |
| É | chr(233) | 11101001 | ÿ | chr(255) | 11111111 |

**1.2.29 Character Set UTF-16 dan UCS-2**

UTF-16 merupakan sistem variable length encoding yang berarti setiap code point (kode yang mencerminkan suatu karakter) di encode menggunakan satu atau dua kali 16-bit (1 code unit = 16 bit).

**1.2.30 Character Set UTF-32 dan UCS-4**

UTF-32 merupakan alternatif dari UTF-16, jika pada UTF-16 terdapat penggunaan sistem doubel 16 bit (untuk karakter ke 65.535 keatas), UTF-32 menggunakan fixed – width encoding, dengan base 32 bit (1 code unit = 32 bit), dimana  semua karakter disimpan menggunakan sistem 32 bit encoding. Contoh karakter dan nilai binary nya pada UTF-32:

|  |  |  |
| --- | --- | --- |
| **TEXT** | **Chr Ke** | **BINARY** |
| ! | 32 | 00000000 00000000 00000000 00100001 |
| Љ | 1025 | 00000000 00000000 11111001 01100101 |
|  | 65.537 | 00000000 00000010 00100111 01001000 |

**1.2.31 Character Set dan Character Encoding UTF-8**

UTF-8 menggunakan sistem variable length  encoding dengan basis 8 bit (1 code unit = 8bit), angka ini tercermin pada angka 8 yang berada di belakang UTF.

**1.2.32 URL Encoding**

[**URL Encoding**](http://fhirman-ilham.blogspot.com/2013/06/pengertian-url-encoding.html) adalah proses konversi URL (Uniform Resource Locator) saat kita melakukan proses request suatu halaman situs ke web server.

**1.2.33 Form**

Form merupakan salah satu elemen dalam HTML yang digunakan untuk menerima input dari pengunjung. Berikut perintah dasar Form :

<form>

Input elements

</form>

**1.2.34 Form Elemen**

**1.2.35 Input Type**

**1.2.36 Input Atribut**

**HTML**<input>**element** digunakan untuk menunjukkan sebuah inputan (masukkan) dalam bentuk kotak dan sejenisnya yang dapat diedit/diketik untuk diisi data tertentu (seperti memasukkan data diri nama, email, tanggal dan lain sebagainya).

**1.2.37 Atribut Nilai**

Atribut Value adalah Nilai dari sebuah informasi atau nilai sifat apa yang akan di berikan kepada tag dan elemen yang mengandung atribut tersebut.

**1.2.38 Atribut Read-Only**

Atribut readonly digunakan untuk menentukkan bahwa elemen <input> hanya dapat dibaca (read only) tanpa bisa diedit. Berikut contoh penulisan dan penggunaan atribut readonly pada form HTML :

<!DOCTYPE html>

<html>

<head>

<title>Penggunaan Atribut Readonly pada Form HTML</title>

</head>

<body>

<form action="proses\_input.php" method="get">

Kode Barang :

<input type="text" name="kode" value ="BRG01" readonly> <br><br>

Jumlah Barang :

<input type="text" name="jumlah"><br><br>

Harga per satuan :

<input type="text" name="harga"><br><br>

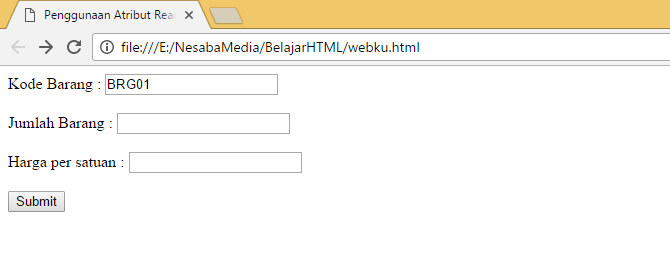
<input type="submit" value="Submit">

</form>

</body>

</html>

Setelah kode diatas dijalankan pada browser maka akan tampil seperti gambar dibawah ini :



Pada kolom **Kode Barang**, anda tidak dapat menginput data karena kolom tersebut sudah diberi atribut readonly, jadi kolom hanya bisa dibaca saja dan tidak bisa diedit atau diubah.

**1.2.39 Atribut Disable**

Atribut disabled digunakan untuk menonaktifkan sebuah elemen (contoh elemen input). Sama halnya dengan atribut readonly, atribut ini tidak perlu ditulis nilainya cukup ditulis disabled saja. Berikut contoh penulisan dan penggunaan atribut disabled pada form HTML :

<!DOCTYPE html>

<html>

<head>

<title>Penggunaan Atribut Disabled pada Form HTML</title>

</head>

<body>

<form action="proses\_input.php" method="get">

Kode Barang :

<input type="text" name="kode" value ="BRG01" disabled> <br><br>

Jumlah Barang :

<input type="text" name="jumlah"><br><br>

Harga per satuan :

<input type="text" name="harga"><br><br>

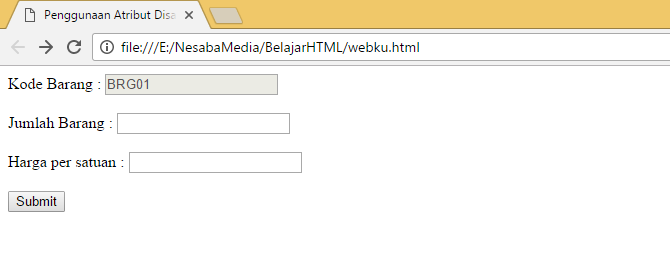
<input type="submit" value="Submit">

</form>

</body>

</html>

Setelah kode diatas dijalankan pada browser maka akan tampil seperti gambar dibawah ini :



**Perbedaan Atribut Readonly dan Disabled**

Dari kedua kontoh code diatas, Anda dapat melihat perbedaan dari penggunaan atribut readonly dan disabled. Atribut readonly membuat kotak teks tetap seperti biasanya sedangkan atribut disabled membuat kotak teks bewarna abu – abu. Meskipun begitu, ada perbedaan yang perlu anda ketahui yakni ketika Anda klik tombol Submit, nilai pada atribut readonly tetap dikirim ke server sedangkan atribut disabled tidak.

**1.2.40 Atribut Ukuran**

Atribut HTML size ini biasa ditempatkan pada tag <img>. Berfungsi untuk menentukan size/ukuran gambar dalam ukuran tertentu. Dibahasakan dengan **width** untuk ukuran lebar dan **height** untuk ukuran tinggi. Sedangkan *source* gambarnya alias alamat tempat gambar tersebut berada ditulis dengan **src**.

Contohnya :

<img src="https://cewek-cakep.jpg" alt="Cewek Cakep" width="104" height="142"

**1.2.41 Atribut Max-Length**

Atribut maxlength kebalikan dari minlength, memberi batasan maksimal karakter yang bisa dimasukkan pengguna. Berlaku untuk atribut: text, email, search, password, tel dan url. Selain atribut ini, fungsi maxlength diabaikan.  
  
Atribut minlength dan maxlength merupakan atribut baru pada elemen input di HTML5.   
  
**Penulisan sintak**  
<input type="atribut" minlength="angka" maxlength="angka" />

* 1. **HTML 5**

**1.3.1 Tentang HTML 5**

HTML5 adalah kepanjangan dari HyperText Markup Language versi 5, merupakan HTML baru penerus dari HTML 4, XHTML1, dan DOM Level 2 HTML. HTML5 merupakan pengembangan bahasa HTML yang lebih baik, lebih berarti atau semantik (semantic meaning) yang sebelumnya adalah bahasa markup sederhana menjadi sebuah flatform canggih, penuh fitur dan kaya akan antarmuka pemrograman aplikasi yang disebut API (Application Programming Interface). Banyak orang yang menganggap bahwa HTML 5, HTML yang sangat support dengan mobile.

**1.3.2 New Elements dan Semantic Tag**

1. **Elemen <article>**

Untuk menampilkan block artikel-artikel yang ada di website, tag article juga digunakan dengan element header  dan footer . Karena tag article memiliki type block , artinya element ini memiliki lebar sesuai lebar browser.

Contoh:

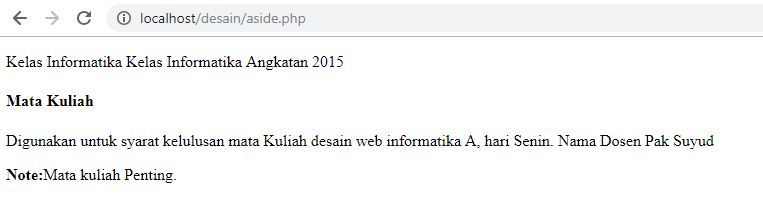


Kode Elemen <article>

|  |
| --- |
| <html>  <body>  <article>  <h1>Desain Web</h1>  <p>Digunakan untuk syarat kelulusan mata Kuliah desain web informatika A, hari  Senin. Nama Dosen Pak Suyud  </p>  </article>  <p><strong>Note:</strong> Mata kuliah Penting.</p>  </body>  </html> |

1. **Elemen <aside>**

Untuk mendefinisikan beberapa konten selain konten tempat konten ditempatkan.

Contoh: :

Kode elemen <aside>

|  |
| --- |
| <html>  <body>  <p>Kelas Informatika Kelas Informatika Angkatan 2015</p>  <aside>  <h4>Mata Kuliah</h4>  <p>Digunakan untuk syarat kelulusan mata Kuliah desain web informatika A,  Hari Senin. Nama Dosen Pak Suyud</p>  </aside>  <p><strong>Note:</strong>Mata kuliah Penting.</p>  </body>  </html> |

1. **Elemen <bdi>**

Untuk mengisolasi bagian teks yang mungkin diformat dalam arah yang berbeda dari teks lain di luarnya.

Contoh :



Kode elemen <bdi>:

|  |
| --- |
| <html>  <body>  <p>Digunakan untuk syarat kelulusan mata Kuliah desain web informatika A,  Hari Senin. Nama Dosen Pak Suyud</p>  <ul>  <li>Nilai <bdi>UAS</bdi>: 25 %</li>  <li>Nilai <bdi>UTS</bdi>: 25 %</li>  <li>Nilai <bdi>TUGAS</bdi>: 50 %s</li>  </ul>  </body>  </html> |

1. **Elemen <detail>**

Menentukan detail tambahan yang dapat dilihat atau disembunyikan oleh pengguna.

Contoh:



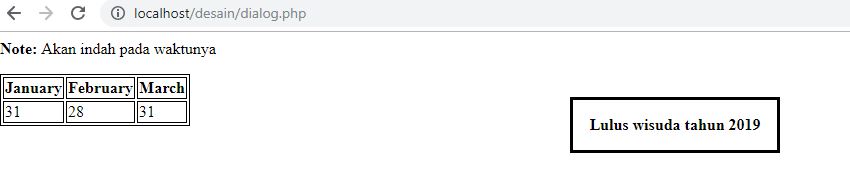
Kode elemen <detail>

|  |
| --- |
| <html>  <body>  <details>  <summary>Copyright 2015-2018.</summary>  <p> - by Refsnes Data. All Rights Reserved.</p>  <p>All content and graphics on this web site are the property of the company Refsnes Data.</p>  </details>  <p><b>Note:</b> Sumber dari Internet Explorer.</p>  </body>  </html> |

1. **Elemen <dialog>**

Menentukan kotak dialog atau jendela, memudahkan untuk membuat dialog popup dan modals pada halaman web.

Contoh:



Kode elemen <dialog>

|  |
| --- |
| <html>  <head>  <style>  table, th, td {  border: 1px solid black;  }  </style>  </head>  <body>  <p><b>Note:</b> Akan indah pada waktunya</p>  <table>  <tr>  <th>January <dialog open>Lulus wisuda tahun 2019</dialog></th>  <th>February</th>  <th>March</th>  </tr>  <tr>  <td>31</td>  <td>28</td>  <td>31</td>  </tr>  </table>  </body>  </html> |

1. **ELEMEN <figcaption>**

Untuk Menentukan keterangan di elemen <figure>

Contoh:



Kode elemen <figcaption>

|  |
| --- |
| <html>  <body>  <h2>Tempat untuk Dikunjungi</h2>  <p>Pemandangan yang indah .</p>  <figure>  <img src="html/w.jpg">  <alt="web" style="width:200%">  <figcaption>Fig.1 - Udara, Indah,Yogyakarta.</figcaption>  </figure>  </body>  </html> |

1. **ELEMEN <figure>**

Digunakan untuk membuat element yang berdiri sendiri, misal sebuah layout dengan gambar di dalamnya. Element figure memiliki type block

Contoh:



Kode elemen <figure>

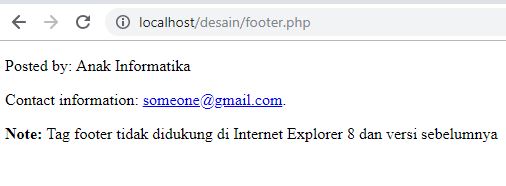
|  |
| --- |
| <html>  <body>  <p>Mahasiswa informatika Universitas Teknologi Yogyakarta.</p>  <figure>  <img src="html/we.png">  <alt="Trulli" style="width:100%">  <figcaption>Fig.1 - Web, IT, Komputer.</figcaption>  </figure>  <p><strong>Note:</strong> Tahun 2019 Lulus Wisuda.</p>  </body>  </html> |

1. **ELEMEN <footer>**

Mendefinisikan footer untuk dokumen atau bagian. Elemen <footer> harus berisi informasi tentang elemen yang mengandung yaitu:

* Informasi kepenulisan
* Informasi hak cipta
* Kontak informasi
* Sitemap
* Kembali ke tautan teratas

Contoh:



Kode elemen <footer>

|  |
| --- |
| <html>  <body>  <footer>  <p>Posted by: Anak Informatika </p>  <p>Contact information: <a href="mailto:someone@gmail.com">someone@gmail.com</a>.</p>  </footer>  <p><strong>Note:</strong> Tag footer tidak didukung di Internet Explorer 8 dan versi sebelumnya</p>  </body>  </html> |

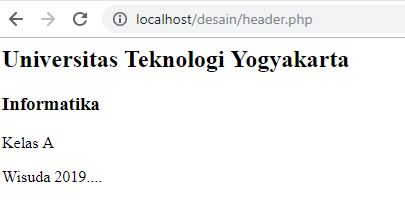
1. **ELEMEN <header>**

Menentukan header untuk dokumen atau bagian. Elemen <header> biasanya berisi:

* Satu atau lebih elemen heading (<h1> - <h6>)
* Logo atau ikon
* Informasi kepenulisan
* Dapat memiliki beberapa elemen <header> dalam satu dokumen

Catatan: Tag <header> tidak dapat ditempatkan di dalam <footer>, <address> atau elemen <header> lainnya.

Contoh:



Kode elemen <header>

|  |
| --- |
| <html>  <body>  <article>  <header>  <h1>Universitas Teknologi Yogyakarta</h1>  <h3>Informatika</h3>  <p>Kelas A</p>  </header>  <p>Wisuda 2019....</p>  </article>  </body>  </html> |

1. **ELEMEN <main>**

Digukan untuk menentukan konten utama dokumen.Konten di dalam elemen <main> harus unik untuk dokumen, tidak boleh berisi konten apa pun yang diulang di seluruh dokumen seperti sidebar, tautan navigasi, informasi hak cipta, logo situs, dan formulir pencarian.

Catatan:

* Tidak boleh ada lebih dari satu elemen <utama> dalam dokumen.
* Elemen <utama> TIDAK harus merupakan turunan dari elemen <article>, <aside>, <footer>, <header>, atau <nav>.

Contoh:



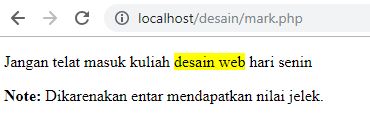
Kode eleman <main>

|  |
| --- |
| <html>  <body>  <main>  <h1>Yogyakarta</h1>  <p>Bantul, Sleman</p>  <article>  <h1>Bantul</h1>  <p>Banyak tempat wisata.</p>  </article>  <article>  <h1>Sleman</h1>  <p>Banyak kampus yang berada di daerah sleman.</p>  </article>  </body>  </html> |

1. **ELEMEN <mark>**

Mark memiliki type/sifat ***inline***, artinya panjang atau lebar element ini hanya sebatas text yang ada di dalamnya. Element mark biasa digunakan untuk membuat text berwarna, secara default warnanya kuning.

Contoh:



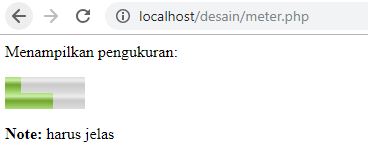
Kode elemen <mark>

|  |
| --- |
| <html>  <body>  <p>Jangan telat masuk kuliah <mark>desain web</mark> hari senin</p>  <p><strong>Note:</strong> Dikarenakan entar mendapatkan nilai jelek.</p>  </body>  </html> |

1. **ELEMEN <meter>**

**Digunakan** untuk mengukur data berdasarkan tingkat tertentu. Memeteri data dengan bilangan yang menunjukkan tingkat meteran atau dikenal dengan istilah gauge. Contohnya, digunakan untuk menunjukkan kapasitas hardisk / memory disk dan hasil pencarian (query result).

Contoh:



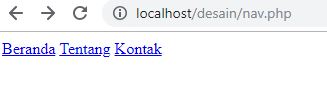
Kode elemen <meter>

|  |
| --- |
| <html>  <body>  <p>Menampilkan pengukuran:</p>  <meter value="2" min="0" max="10">2 out of 10</meter><br>  <meter value="0.6">60%</meter>  <p><strong>Note:</strong> harus jelas</p>  </body>  </html> |

1. **ELEMEN <nav>**

Digunakan  untuk membuat component navbar. Tag nav secara default memiliki type block biasa digunakan untuk membuat link navigasi sebuah website.

Contoh:



Kode elemen <nav>

|  |
| --- |
| <html>  <body>  <nav>  <a href="/beranda">Beranda</a>  <a href="/about">Tentang</a>  <a href="/contact">Kontak</a>  </nav>  </body>  </html> |

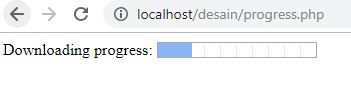
1. **ELEMEN <progress>**

Menggambarkan kemajuan tugas

Tips dan Catatan:

* Gunakan tag <progress> bersama dengan JavaScript untuk menampilkan kemajuan tugas.
* Tidak cocok untuk mewakili pengukur (misalnya penggunaan ruang disk atau relevansi hasil kueri). Untuk mewakili pengukur, gunakan tag <meter> sebagai gantinya.

Contoh:



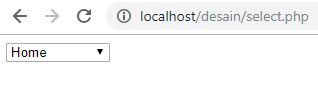
Kode elemen <progress>

|  |
| --- |
| <html>  <body>  Downloading progress:  <progress value="22" max="100">  </progress>  </body>  </html> |

1. **ELEMEN <selection>**

Digunakan untuk membuat daftar drop-down. Tag <option> di dalam elemen <select> menentukan opsi yang tersedia dalam daftar.

Contoh:



Kode Elemen <Selection>

|  |
| --- |
| <!DOCTYPE html>  <html>  <body>  <select>  <option value="Home">Home</option>  <option value="Profil">Profil</option>  <option value="Tentang Kami">Tentang kami</option>  <option value="kontak">Kontak</option>  </select>  </body>  </html> |

1. **ELEMEN <time>**

Menentukan tanggal / waktu yang dapat dibaca manusia.

* Elemen ini juga dapat digunakan untuk mengkodekan tanggal dan waktu dengan cara yang dapat dibaca oleh mesin sehingga agen pengguna dapat menawarkan untuk menambahkan pengingat ulang tahun atau acara terjadwal ke kalender pengguna, dan mesin pencari dapat menghasilkan hasil pencarian yang lebih cerdas.

Contoh:



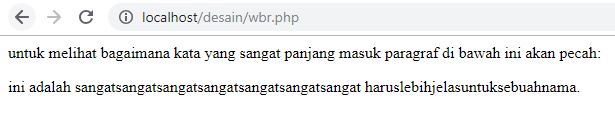
Kode elemen <time>

|  |
| --- |
| <html>  <body>  <p>kita buka pada <time>10:00</time> setiap pagi.</p>  <p>Saya memiliki tanggal pada <time datetime = "2008-02-14 20:00"> Hari kasih sayang</time>.</p>  </body>  </html> |

1. **ELEMEN <wbr>**

Word Break Opportunity (wbr ) di gunakan utuk menentukan di mana di dalam teks tidak ada masalah untuk menambahkan baris-break.

Contoh:



Kode elemen<wbr>

|  |
| --- |
| <html>  <body>  <p>untuk melihat bagaimana kata yang sangat panjang masuk  paragraf di bawah ini akan pecah:</p>  <p>ini adalah sangatsangatsangatsangatsangatsangatsangat  <wbr>haruslebihjelas<wbr>untuksebuahnama.</p>  </body>  </html> |

1. **ELEMEN <track>**

Digunakan untuk Mendefinisikan lagu <video> dan <audio>

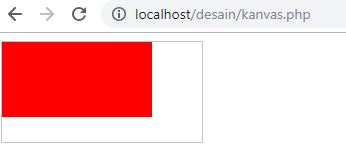
|  |
| --- |
| <video width="320" height="240" controls>  <source src='halwa.mp4' type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>  <source src='halwa.ogv' type='video/ogg; codecs="theora, vorbis"'>  <source src="halwa.webm" type='video/webm; codecs="vp8, vorbis"'>  <track label="Indonesian" kind="subtitles" srclang="id" src="subtitles-id.vtt" default>  Browser Anda tidak mendukung HTML video tag.  </video> |

* + 1. **Graphics dan Canvas**

1. **Tentang Canvas**

Canvas adalah element yang digunakan untuk menggambar grafik, object di website yg umumnya melalui script javascript. Canvas hanya merupakan wadah untuk menggambar, dan untuk menggambarnya harus menggunakan script, misal javascript.

1. **Kanvas Gambar**



Kode Kanvas Gambar:

|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="200" height="100"  style="border:1px solid #c3c3c3;">  Your browser does not support the canvas element.  </canvas>  <script>  var canvas = document.getElementById("myCanvas");  var ctx = canvas.getContext("2d");  ctx.fillStyle = "#FF0000";  ctx.fillRect(0,0,150,75);  </script>  </body>  </html> |

1. **Koordinat kanvas**

Koordinat Canvas HTML adalah kisi dua dimensi. Sudut kiri atas kanvas memiliki koordinat (0,0)



koordinat x dan y

1. **Draw Line**

Untuk menggambar garis lurus pada kanvas, gunakan metode berikut:

* moveTo (x, y) - mendefinisikan titik awal dari garis
* lineTo (x, y) - mendefinisikan titik akhir dari garis
* Untuk benar-benar menarik garis, Anda harus menggunakan salah satu metode "tinta", seperti stroke ()

Contoh Draw Line:



Menentukan titik awal dalam posisi (0,0), dan titik akhir pada posisi (200.100). Kemudian gunakan metode stroke () untuk benar-benar menarik garis:

|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="200" height="100"  style="border:1px solid #d3d3d3;">  Your browser does not support the canvas element.  </canvas>  <script>  var canvas = document.getElementById("myCanvas");  var ctx = canvas.getContext("2d");  ctx.moveTo(0,0);  ctx.lineTo(200,100);  ctx.stroke();  </script>  </body>  </html> |

1. **Draw a Circle**

Untuk menggambar lingkaran pada canvas, gunakan metode berikut:

* beginPath () - memulai jalur
* arc (x, y, r, startangle, endangle) - membuat busur / kurva. Untuk membuat lingkaran dengan busur (): Atur mulai sudut ke 0 dan ujung ke 2 \* Math.PI. Parameter x dan y menentukan koordinat x dan y dari pusat lingkaran. Parameter r mendefinisikan radius lingkaran.

Contoh Draw a Circle



Menentukan lingkaran dengan metode arc (). Kemudian gunakan metode stroke () untuk benar-benar menggambar lingkaran:

|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="200" height="100"  style="border:1px solid #d3d3d3;">  Your browser does not support the canvas element.  </canvas>  <script>  var canvas = document.getElementById("myCanvas");  var ctx = canvas.getContext("2d");  ctx.beginPath();  ctx.arc(95,50,40,0,2\*Math.PI);  ctx.stroke();  </script>  </body>  </html> |

1. **Canvas Gradient**

Gradien dapat digunakan untuk mengisi segi empat, lingkaran, garis, teks, dll. Bentuk pada kanvas tidak terbatas pada warna solid.

1. **CreateLinearGradient**

createLinearGradient (x, y, x1, y1) - menciptakan gradien linier

Contoh gradien linear dengan persegi panjang:



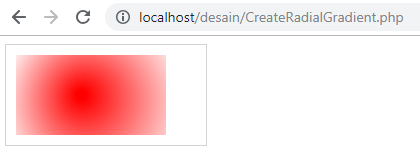
Kode CreateLinearGradient:

|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="200" height="100"  style="border:1px solid #d3d3d3;">  Your browser does not support the HTML5 canvas tag.</canvas>  <script>  var c = document.getElementById("myCanvas");  var ctx = c.getContext("2d");  // Create gradient  var grd = ctx.createLinearGradient(0,0,200,0);  grd.addColorStop(0,"red");  grd.addColorStop(1,"white");  // Fill with gradient  ctx.fillStyle = grd;  ctx.fillRect(10,10,150,80);  </script>  </body>  </html> |

1. CreateRadialGradient

CreateRadialGradient (x, y, r, x1, y1, r1) - menciptakan gradien radial / melingkar.

Contoh gradien linear dengan persegi panjang:



Kode CreateRadialGradient:

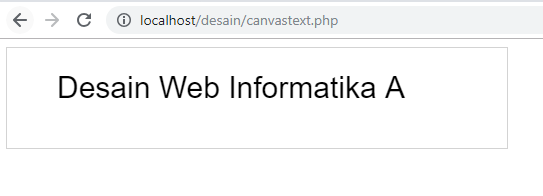
|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="200" height="100"  style="border:1px solid #d3d3d3;">  Your browser does not support the HTML5 canvas tag.</canvas>  <script>  var c = document.getElementById("myCanvas");  var ctx = c.getContext("2d");  // Buat gradien  var grd = ctx.createRadialGradient(75,50,5,90,60,100);  grd.addColorStop(0,"red");  grd.addColorStop(1,"white");  // Isi dengan gradien  ctx.fillStyle = grd;  ctx.fillRect(10,10,150,80);  </script>  </body>  </html> |

1. **Canvas Text**

Menggambar Teks di Canvas. Untuk menggambar teks pada Canvas, properti dan metode yang paling penting adalah:

* font - mendefinisikan properti font untuk teks
* fillText (teks, x, y) - menggambar teks "penuh" di kanvas
* strokeText (teks, x, y) - menggambar teks pada kanvas (tidak ada isian)

Contoh tulisan teks yang terisi di Canvas:



Kode Canvas Text:

|  |
| --- |
| <html>  <body>  <canvas id="myCanvas" width="500" height="100"  style="border:1px solid #d3d3d3;">  Your browser does not support the canvas element.  </canvas>  <script>  var canvas = document.getElementById("myCanvas");  var ctx = canvas.getContext("2d");  ctx.font = "30px Arial";  ctx.fillText("Desain Web Informatika A",50,50);  </script>  </body>  </html> |

1. **Canvas Images**

Untuk menggambar gambar menggunakan HTML5 Canvas, bisa menggunakan metode drawImage () yang membutuhkan objek gambar dan titik tujuan. Titik tujuan mendefinisikan sudut kiri atas gambar relatif ke sudut kiri atas kanvas.



Kode Canvas Images:

|  |
| --- |
| <html>  <body>  <p> Gambar untuk digunakan:</p>  <img id="scream" width="220" height="277"  src="pic\_the\_scream.jpg" alt="The Scream">  <p>Canvas:</p>  <canvas id="myCanvas" width="240" height="297"  style="border:1px solid #d3d3d3;">  </canvas>  <script>  window.onload = function() {  var canvas = document.getElementById("myCanvas");  var ctx = canvas.getContext("2d");  var img = document.getElementById("scream");  ctx.drawImage(img, 10, 10);  };  </script>  </body>  </html> |

* + 1. **SVG**

SVG adalah bahasa untuk menggambarkan grafis 2D dalam XML, SVG juga  berbasis XML yang berarti bahwa setiap elemen tersedia dalam DOM SVG. Anda dapat melampirkan pengendali event JavaScript untuk elemen.

Dalam SVG, setiap bentuk yang digambar diingat sebagai obyek. Jika atribut dari suatu obyek SVG berubah, browser secara otomatis dapat membuat bentuk kembali.

1. **Apa itu SVG?**

* SVG adalah singkatan dari Scalable Vector Graphics
* SVG digunakan untuk mendefinisikan vektor berbasis grafis untuk Web
* SVG mendefinisikan grafis dalam format XML
* Grafis SVG TIDAK kehilangan kualitas apa pun jika mereka diperbesar atau diubah ukurannya
* Setiap elemen dan atribut setiap file SVG dapat animasi
* SVG adalah rekomendasi W3C

B. **Gambar SVG**

Menggambar Lingkaran dengan SVG

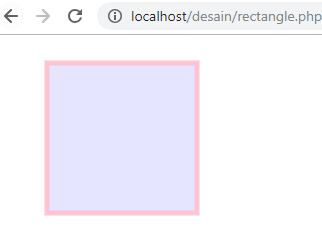
Contoh:



Kode gambar SVG :

|  |
| --- |
| <html>  <head>  <style>  #svgelem {  position: relative;  left: 50%;  -webkit-transform: translateX(-20%);  -ms-transform: translateX(-20%);  transform: translateX(-20%);  }  </style>  <title>SVG</title>    </head>  <body>    <h2 align="center">  Lingkaran dengan SVG</h2>  <svg height="200" id="svgelem" xmlns="http://www.w3.org/2000/svg">  <circle cx="50" cy="50" fill="red" id="redcircle" r="50">  </circle></svg>  </body>  </html> |

1. **SVG Rectangle**



Berikut adalah kode SVG:

|  |
| --- |
| <html>  <body>  <svg width="400" height="180">  <rect x="50" y="20" width="150" height="150" style="fill:blue;stroke:pink;stroke-width:5;fill-opacity:0.1;stroke-opacity:0.9" />  </svg>  </body>  </html> |

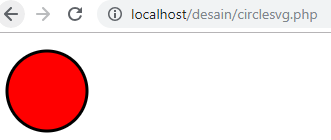
Penjelasan kode:

* Atribut x menentukan posisi kiri persegi panjang (misalnya x = "50" menempatkan persegi panjang 50 px dari margin kiri)
* Atribut y mendefinisikan posisi teratas persegi panjang (mis. Y = "20" menempatkan persegi panjang 20 px dari margin atas)
* Properti fill-opacity CSS mendefinisikan opacity dari warna isi (rentang hukum: 0 hingga 1)
* Properti CSS stroke-opacity mendefinisikan opacity dari warna stroke (jangkauan hukum: 0 hingga 1)

1. Circle

Elemen <circle> SVG adalah bentuk dasar SVG, digunakan untuk membuat lingkaran berdasarkan titik pusat dan radius.

Contoh:



Kode Circle SVG:

|  |
| --- |
| <html>  <body>  <svg height="100" width="100">  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />  </svg>  </body>  </html> |

Penjelasan Kode:

* Atribut cx dan cy menentukan koordinat x dan y dari pusat lingkaran. Jika cx dan cy dihilangkan, pusat lingkaran diatur ke (0,0)
* Atribut r mendefinisikan radius lingkaran

1. **Ellipse**

Elipse adalah lingkaran yang memiliki 2 jari-jari yang berbeda, untuk itu dibutuhkan *tag*baru yang memiliki atribut tambahan.

Contoh berikut membuat tiga elipse di atas satu sama lain:



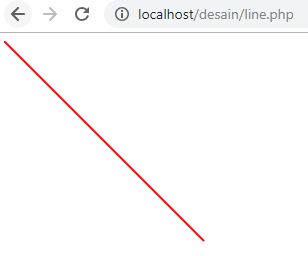
Kode Ellipse SVG :

|  |
| --- |
| <html>  <body>  <svg height="150" width="500">  <ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />  <ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />  <ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" />  </svg>  </body>  </html> |

1. **Line**

Line adalah bentuk dasar SVG yang digunakan untuk membuat garis yang menghubungkan dua titik.

Contoh:



Kode Line SVG:

|  |
| --- |
| <html>  <body>  <svg height="210" width="500">  <line x1="0" y1="0" x2="200" y2="200" style="stroke:rgb(255,0,0);stroke-width:2" />  </svg>  </body>  </html> |

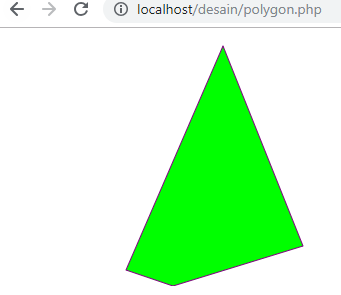
Penjelasan kode:

* Atribut x1 mendefinisikan awal garis pada sumbu x
* Atribut y1 mendefinisikan awal garis pada sumbu y
* Atribut x2 mendefinisikan akhir garis pada sumbu x
* Atribut y2 mendefinisikan akhir garis pada sumbu y

1. **Polygon**

Polygon berasal dari bahasa Yunani. "Poli" berarti "banyak" dan "gon" berarti "sudut". digunakan untuk membuat grafik yang mengandung setidaknya tiga sisi. Poligon terbuat dari garis lurus, dan bentuknya "tertutup" (semua garis terhubung).

Contoh berikut membuat poligon dengan empat sisi:



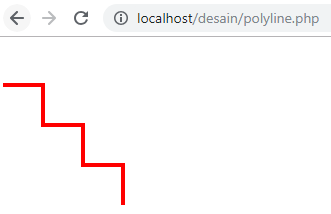
Kode Polygon SVG:

|  |
| --- |
| <html>  <body>  <svg height="250" width="500">  <polygon points="220,10 300,210 170,250 123,234" style="fill:lime;stroke:purple;stroke-width:1" />  </svg>  </body>  </html> |

1. **Polyline**

Polyline adalah bentuk dasar SVG yang menciptakan garis lurus yang menghubungkan beberapa titik. Biasanya polyline digunakan untuk membuat bentuk terbuka karena titik terakhir tidak harus terhubung ke titik pertama.

Contoh:



Kode Polyline SVG:

|  |
| --- |
| <html>  <body>  <svg height="180" width="500">  <polyline points="0,40 40,40 40,80 80,80 80,120 120,120 120,160" style="fill:white;stroke:red;stroke-width:4" />  </svg>  </body>  </html> |

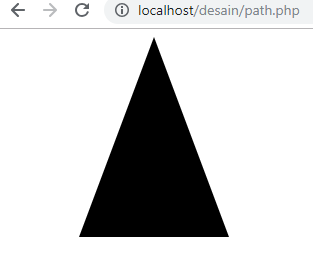
1. **Path**

Path SVG adalah elemen generik untuk menentukan bentuk. Semua bentuk dasar dapat dibuat dengan elemen jalur atau digunakan untuk menentukan jalur.Perintah berikut tersedia untuk data jalur:

* M = pindah
* L = lineto
* H = garis horisontal
* V = garis vertikal
* C = curveto
* S = kelancaran curveto
* Q = kurva kuadrat Bézier
* T = kurva Bézier curveto yang halus
* A = Arc elips
* Z = closepath

Catatan: Semua perintah di atas juga dapat diekspresikan dengan huruf yang lebih rendah. Huruf kapital berarti benar-benar diposisikan, huruf kecil berarti posisi relatif.

Contoh di bawah ini mendefinisikan jalur yang dimulai pada posisi 150,0 dengan garis ke posisi 75.200 kemudian dari sana, garis ke 225.200 dan akhirnya menutup jalan kembali ke 150,0:



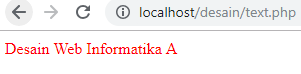
Kode Path SVG:

|  |
| --- |
| <html>  <body>  <svg height="210" width="400">  <path d="M150 0 L75 200 L225 200 Z" />  </svg>  </body>  </html> |

1. **Text**

Mendefinisikan elemen grafis yang terdiri dari teks. Dapat menerapkan gradient, pattern, clipping path, mask, atau filter ke <text>, sama seperti elemen grafis SVG lainnya.

Contoh tulis Text:

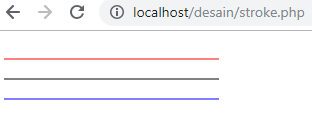


Kode Text SVG:

|  |
| --- |
| <html>  <body>  <svg height="30" width="200">  <text x="0" y="15" fill="red">Desain Web Informatika Atext>  </svg>  </body>  </html> |

1. **Stroke**

Semua properti stroke dapat diterapkan untuk semua jenis garis, teks dan garis besar elemen seperti lingkaran. Properti stroke mendefinisikan warna garis, teks atau garis besar elemen:



Kode Stroke SVG

|  |
| --- |
| <html>  <body>  <svg height="80" width="300">  <g fill="none">  <path stroke="red" d="M5 20 l215 0" />  <path stroke="black" d="M5 40 l215 0" />  <path stroke="blue" d="M5 60 l215 0" />  </g>  </svg>  </body>  </html> |

1. **Blur Effect**

Blur Efect digunakan untuk membuat efek buram:



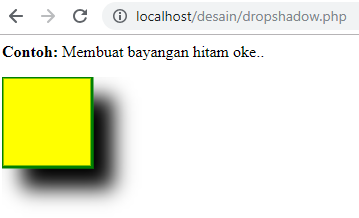
Kode Blur Effect SVG:

|  |
| --- |
| <html>  <body>  <p><strong>Note:</strong> Ini bewarna Kuning..</p>  <svg height="110" width="110">  <defs>  <filter id="f1" x="0" y="0">  <feGaussianBlur in="SourceGraphic" stdDeviation="15" />  </filter>  </defs>  <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow" filter="url(#f1)" />  </svg>  </body>  </html> |

Penjelasan Kode:

* Atribut id dari elemen <filter> menentukan nama unik untuk filter
* Efek blur didefinisikan dengan elemen <feGaussianBlur>
* Bagian = "SourceGraphic" mendefinisikan bahwa efek dibuat untuk seluruh elemen
* Atribut stdEviation mendefinisikan jumlah blur
* Atribut filter dari elemen <rect> menautkan elemen ke filter "f1"

1. **Drop Shadow**



Kode Drop Shadow:

|  |
| --- |
| <html>  <body>  <p><strong>Contoh:</strong> Membuat bayangan hitam oke..</p>  <svg height="140" width="140">  <defs>  <filter id="f1" x="0" y="0" width="200%" height="200%">  <feOffset result="offOut" in="SourceAlpha" dx="20" dy="20" />  <feGaussianBlur result="blurOut" in="offOut" stdDeviation="10" />  <feBlend in="SourceGraphic" in2="blurOut" mode="normal" />  </filter>  </defs>  <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow" filter="url(#f1)" />  </svg>  </body>  </html> |

Penjelasan Kode:

* Atribut elemen <feOffset> diubah menjadi "SourceAlpha" yang menggunakan saluran Alpha untuk blur bukan seluruh RGBA pixel

1. **Linear**

Gradien adalah transisi yang mulus dari satu warna ke warna lainnya. Selain itu, beberapa transisi warna dapat diterapkan ke elemen yang sama. Ada dua jenis utama gradien di SVG:

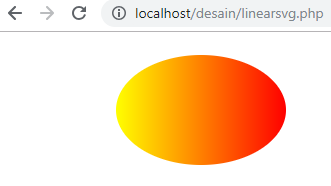
* Linear
* Radial

Elemen <linearGradient> digunakan untuk mendefinisikan gradien linier. Elemen <linearGradient> harus disarangkan dalam tag <defs>. Tag <defs> pendek untuk definisi dan berisi definisi elemen khusus (seperti gradien).

Gradien linier dapat didefinisikan sebagai gradien horizontal, vertikal atau sudut:

* Gradien horizontal dibuat ketika y1 dan y2 sama dan x1 dan x2 berbeda
* Gradien vertikal dibuat ketika x1 dan x2 sama dan y1 dan y2 berbeda
* Gradient sudut dibuat ketika x1 dan x2 berbeda dan y1 dan y2 berbeda

Contoh elips dengan gradien linier horizontal dari kuning ke merah:



Kode Linear SVG:

|  |
| --- |
| <html>  <body>  <svg height="150" width="400">  <defs>  <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">  <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />  <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />  </linearGradient>  </defs>  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />  </svg>  </body>  </html> |

Penjelasan Kode:

* Atribut id dari tag <linearGradient> mendefinisikan nama unik untuk gradien
* Atribut x1, x2, y1, y2 dari tag <linearGradient> menentukan posisi awal dan akhir gradien
* Rentang warna untuk gradien dapat terdiri dari dua atau lebih warna. Setiap warna ditentukan dengan tag <stop>. Atribut offset digunakan untuk menentukan di mana warna gradien mulai dan berakhir
* Atribut fill menghubungkan elemen elips ke gradien

1. **Radial**

Digunakan untuk menentukan gradien radial. Elemen <radialGradient> harus disarangkan dalam tag <defs>. Tag <defs> pendek untuk definisi dan berisi definisi elemen khusus (seperti gradien).

Contoh elips dengan gradien radial dari putih menjadi biru:



Kode Radial SVG:

|  |
| --- |
| <html>  <body>  <svg height="150" width="500">  <defs>  <radialGradient id="grad1" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">  <stop offset="0%" style="stop-color:rgb(255,255,255);stop-opacity:0" />  <stop offset="100%" style="stop-color:rgb(0,0,255);stop-opacity:1" />  </radialGradient>  </defs>  <ellipse cx="200" cy="70" rx="85" ry="55" fill="url(#grad1)" />  </svg>  </body>  </html> |

Penjelasan Kode:

* Atribut id dari tag <radialGradient> mendefinisikan nama unik untuk gradien
* Atribut cx, cy dan r mendefinisikan lingkaran terluar dan fx dan fy mendefinisikan lingkaran terdalam
* Rentang warna untuk gradien dapat terdiri dari dua atau lebih warna. Setiap warna ditentukan dengan tag <stop>. Atribut offset digunakan untuk menentukan di mana warna gradien mulai dan berakhir
* Atribut fill menghubungkan elemen elips ke gradien

1. **Google Maps**

Google Maps adalah layanan [pemetaan web](https://id.wikipedia.org/w/index.php?title=Pemetaan_web&action=edit&redlink=1) yang dikembangkan oleh Google. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi [beta](https://id.wikipedia.org/wiki/Daur_hidup_rilis_peranti_lunak)), atau angkutan umum.

1. Media

Media di web adalah suara, musik, video, film, dan animasi.

Format Multimedia:

* Elemen multimedia (seperti audio atau video) disimpan dalam file media.Cara paling umum untuk menemukan jenis file, adalah dengan melihat ekstensi file.File multimedia memiliki format dan ekstensi yang berbeda seperti: .swf, .wav, .mp3, .mp4, .mpg, .wmv, dan .avi.

Format Video Umum:

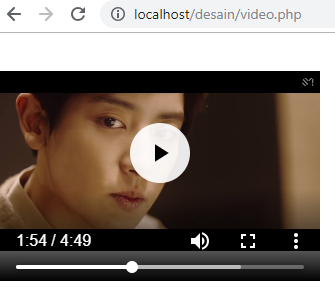


* Videoformats MP4 adalah format baru dan yang akan datang untuk video internet.
* MP4 direkomendasikan oleh YouTube.
* MP4 didukung oleh Flash Players.
* MP4 didukung oleh HTML5.

1. **Video**

Memutar Video dalam HTML Sebelum HTML5, video hanya dapat diputar di browser dengan plug-in (seperti flash). Di elemen HTML5 <video> menentukan cara standar untuk menyematkan video di halaman web.

Untuk menampilkan video dalam HTML, gunakan elemen <video>:



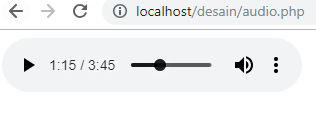
Kode video html:

|  |
| --- |
| <html>  <body>  <video width="320" height="240" controls>  <source src="movie.mp4" type="video/mp4">  <source src="movie.ogg" type="video/ogg">  </video>  </body>  </html> |

1. **Audio**

Audio di Web Sebelum HTML5, file audio hanya dapat diputar di browser dengan plug-in (seperti flash). Di Elemen HTML5 <audio> untuk menentukan cara standar menambahkan audio di halaman web.

Contoh audio Heaven.mp3:



kode Audio

|  |
| --- |
| <html>  <body>  <audio controls>  <source src="Heaven..mp3" type="audio/mpeg">  </audio>  </body>  </html> |

HTML Audio - Cara Kerjanya:

* Atribut kontrol menambahkan kontrol audio, seperti putar, jeda, dan volume.
* Elemen <source>untuk menentukan file audio alternatif yang dapat dipilih oleh browser.Browser akan menggunakan format yang dikenal pertama.
* Teks antara tag <audio> dan </ audio> hanya akan ditampilkan di browser yang tidak mendukung elemen <audio>

1. **Plug-ins**

Tujuan dari sebuah plug-in adalah untuk memperluas fungsionalitas dari suatu browser web

Pembantu HTML (Plug-in):

* Aplikasi pembantu (plug-in) adalah program komputer yang memperluas fungsi standar browser web.
* Contoh dari plug-in yang terkenal adalah applet Java.Plug-in dapat ditambahkan ke halaman web dengan tag <object> atau tag <embed>.
* Plug-in dapat digunakan untuk banyak tujuan: menampilkan peta, memindai virus, memverifikasi ID bank Anda, dll.
* Untuk menampilkan video dan audio: Gunakan tag <video> dan <audio>.

Elemen <object> :

* Elemen <object> didukung oleh semua browser.
* Elemen <object> mendefinisikan objek tertanam dalam dokumen HTML.
* Ini digunakan untuk menanamkan plug-in (seperti applet Java, pembaca PDF, Flash Players) di halaman web.

Contoh kode Plug-ins

|  |
| --- |
| <html>  <body>  <object width="400" height="50" data="bookmark.swf"></object>  </body>  </html> |

## 1.4. Latihan Html

### 1.4.1 HTML5

HTML5 adalah singkatan Hyper Text Markup Language Versi 5, HTML5 merupakan sebuah teknologi penerus dari generasi sebelumnya yang dikeluarkan oleh W3C (World Wide Web Consortium) dan WHATWG (Web Hypertext Application Technology Working Group). dengan tambahan feature- feature baru untuk memperbaiki dan melengkapi dari HTML versi sebelumnya.

### 1.4.2 New Elements dan Semantic Tag

1. Elemen <article> bisa digunakan untuk menampilkan *block* artikel-artikel yang ada di website. Tag <article> dapat digunakan dengan element header dan footer, karena tag <article> memiliki type *block* artinya element ini memiliki lebar sesuai lebar *browser* jika tidak ditentukan lebarnya. Contoh :

<article>

<header>Judul Artikel</header>

<p>

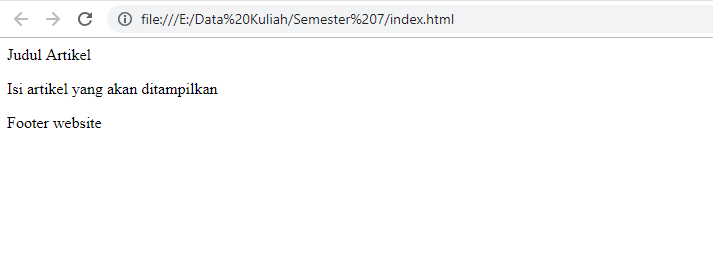
Isi artikel yang akan ditampilkan

</p>

<footer>Footer website</footer>

</article>

Hasilnya sebagai berikut



1. Elemen <aside> bisa digunakan untuk melengkapi tampilan utama. Misalnya memiliki halaman blog maka kita bisa menambahkan <aside> dibagian bawah postingan untuk menjelaskan identitas penulis. Karena tag <article> memiliki type *block* artinya element ini memiliki lebar sesuai lebar *browser* jika tidak ditentukan lebarnya. Contoh :

<main>

<h1>Judul Blog</h1>

<p>

Blog ini berisi tugas desain web

yang akan membahas tentang html5,

mulai dari elemen sampai selesai.

</p>

</main>

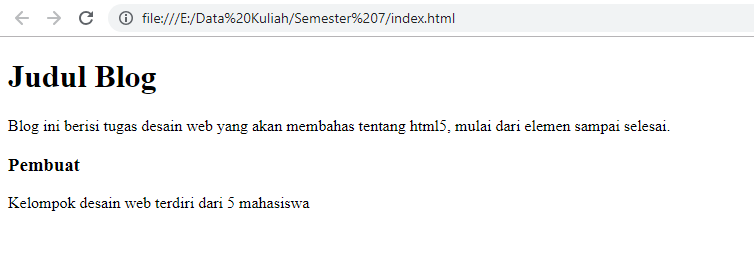
<aside>

<h3>Pembuat</h3>

<p>Kelompok desain web terdiri dari 5 mahasiswa</p>

</aside>

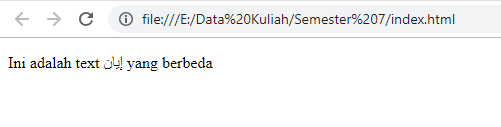
Hasilnya sebagai berikut



1. Elemen <bdi> bisa digunakan untuk mengisolasi bagian dari text yang memiliki ejaan yang berbeda dari text normal.

<p>Ini adalah text <bdi>إيان</bdi> yang berbeda</p>

Hasilnya sebagai berikut :



1. Elemen <details> bisa digunakan sebagai alat singkat. Apabila terdapat banyak penjelasan dari suatu halaman maka disembunyikan menggunakan elemen <details>. Contoh :

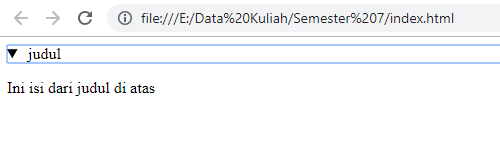
<details>

<summary> judul </summary>

<p>Ini isi dari judul di atas</p>

</details>

Hasilnya sebagai berikut:



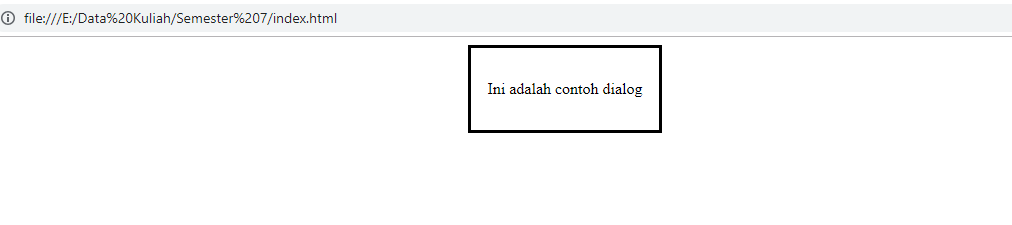
1. Elemen <dialog> digunakan untuk menampilkan sebuah kotak dialog atau komponen interaktif lainnya seperti halnya inspector atau jendela. Contoh :

<dialog open>

<p>Ini adalah contoh dialog</p>

</dialog>

Hasilnya sebagai berikut :



1. Elemen <figcaption> dapat digunakan untuk memarkup judul atau keterangan dari sebuah gambar. Contoh :

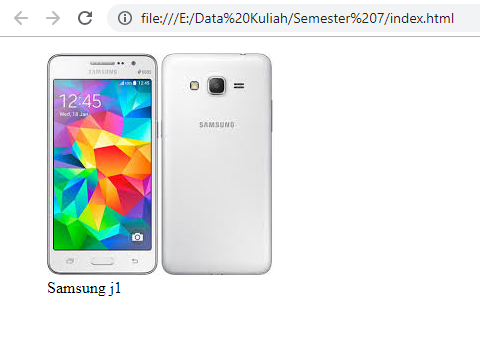
<figure>

<img src="1.jpg">

<figcaption>Samsung j1</figcaption>

</figure>

Hasilnya sebagai berikut :



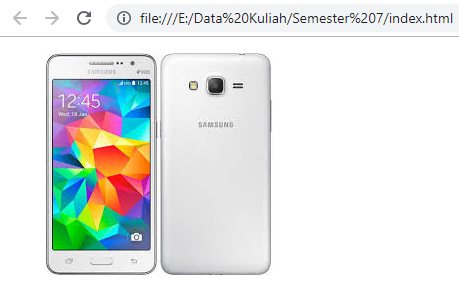
1. Elemen <figure> digunakan untuk membuat elemen yang berdiri sendiri, misalnya sebuah layout dengan gambar didalamya. Elemen <figure> memiliki type *block.* Contoh :

<figure>

<img src="1.jpg">

</figure>

Hasilnya sebagai berikut :



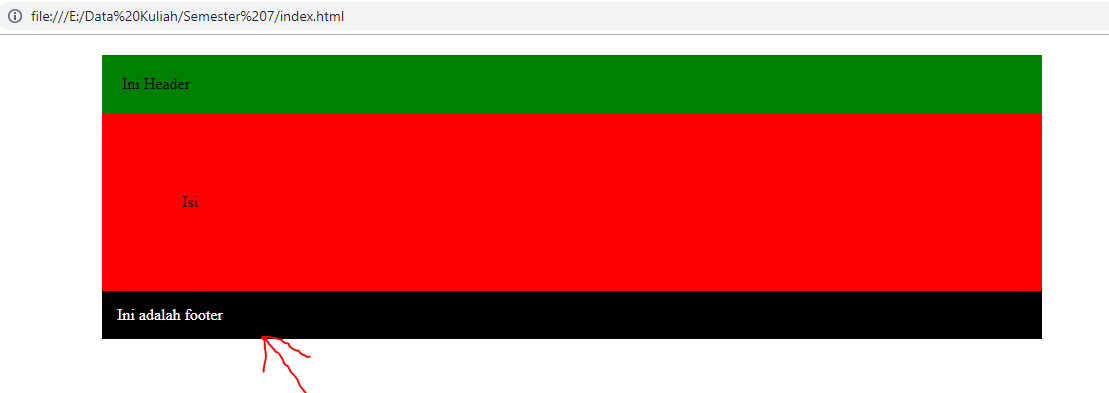
1. Elemen <footer> digunakan untuk membuat footer sebuah website. <footer> memiliki type *block*. Contoh :

<footer>

Ini adalah footer

</footer>

Hasilnya seperti yang terlihat ditanda panah merah :



1. Elemen <header> digunakan untuk membuat header sebuah website. Di dalam tag header dapat di tambah elemen lain. Contoh :

<header>

<h1>Desain Web</h1>

<nav>

<a href="">Home</a>

<a href="">About</a>

<a href="">Contact</a>

</nav>

</header>

Hasilnya seperti yang terlihat ditanda panah merah :



1. Elemen <main> digunakan untuk membungkus content yang ada di sebuah website. Contoh :

<main>

<h1>Web Design</h1>

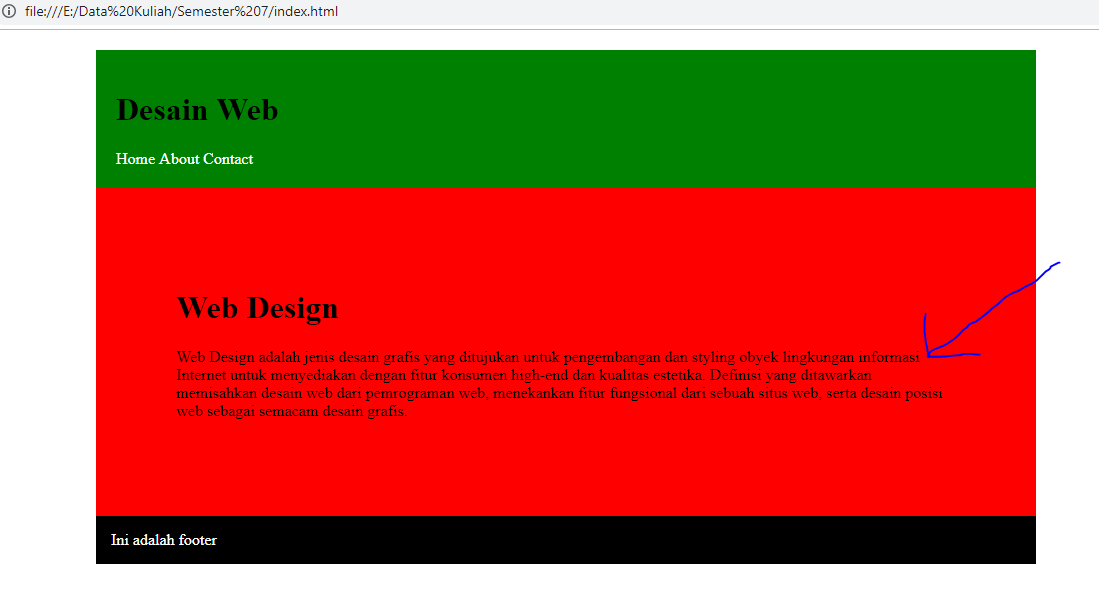
<p>

Web Design adalah jenis desain grafis yang ditujukan untuk pengembangan dan styling obyek lingkungan informasi Internet untuk menyediakan dengan fitur konsumen high-end dan kualitas estetika. Definisi yang ditawarkan memisahkan desain web dari pemrograman web, menekankan fitur fungsional dari sebuah situs web, serta desain posisi web sebagai semacam desain grafis.

</p>

</main>

Hasilnya seperti yang terlihat ditanda panah biru :



1. Elemen <mark> digunakan untuk membuat text berwarna, secara default warnya kuning. Contoh:

Tutorial <mark>Desain Web</mark> utnuk pemula

Hasilnya seperti yang terlihat ditanda panah biru :

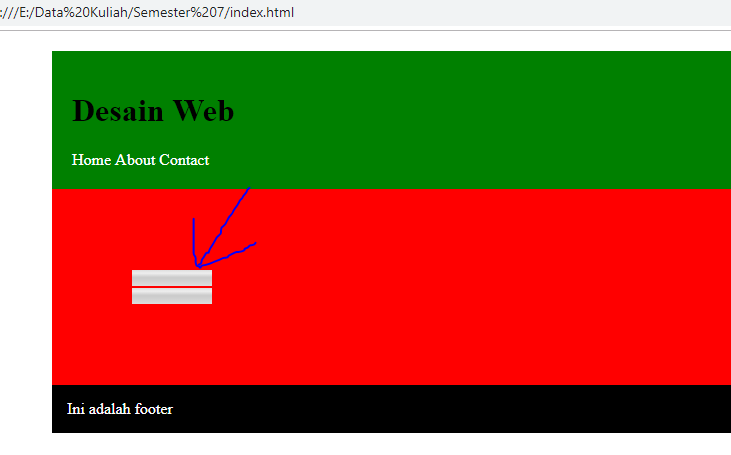


1. Elemen <meter> mendefinisikan pengukuran skalar dalam rentang diketahui, atau nilai pecahan. Contoh :

<meter value=”2″ min=”0″ max=”10″>2 dari 10</meter><br />

<meter value=”0.6″>60%</meter>

Hasilnya seperti yang terlihat ditanda panah biru :



1. Elemen <nav> digunakan untuk membuat link navigasi sebuah website. Contoh :

<nav>

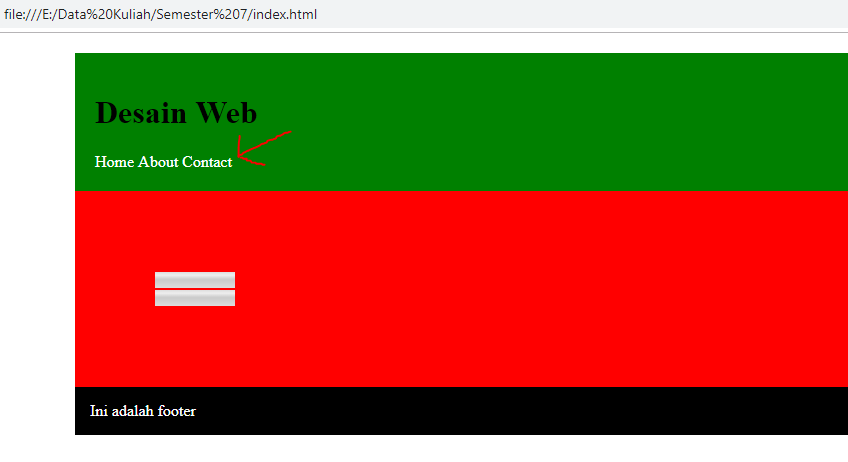
<a href="">Home</a>

<a href="">About</a>

<a href="">Contact</a>

</nav>

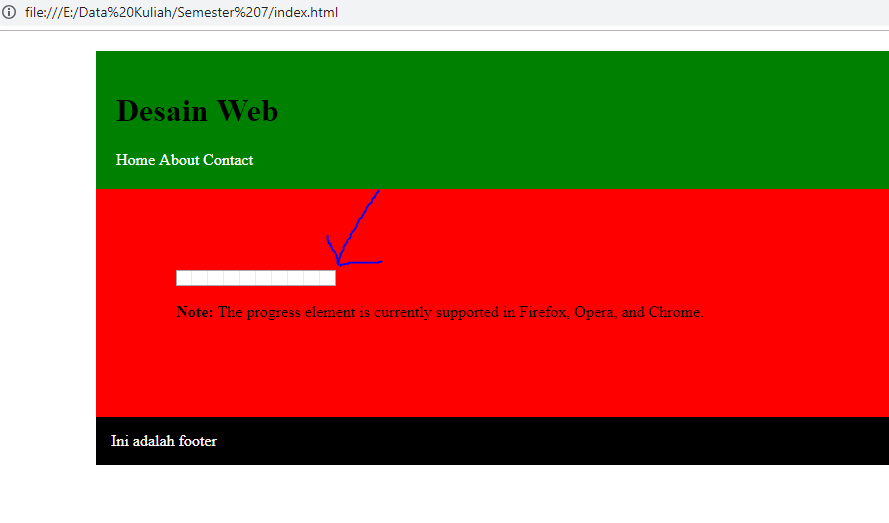
Hasilnya seperti yang terlihat ditanda panah merah :



1. Elemen <progress> digunakan untuk menampilkan kemajuan tugas. Contoh :

<progress value=”22″ max=”100″></progress>

Hasilnya seperti yang terlihat ditanda panah biru :



1. Elemen <section> digunakan untuk membuat bagian dari sebuah halaman website. Contoh :

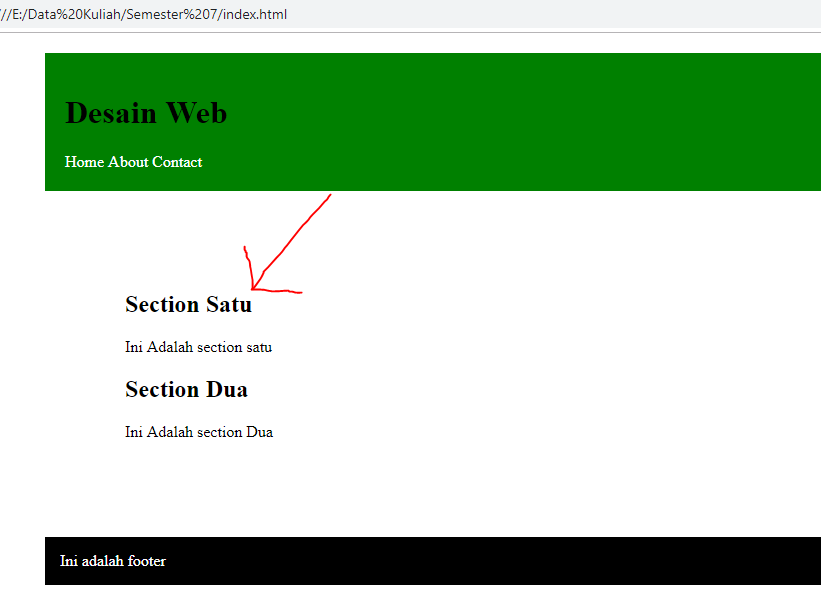
<section><h2>Section Satu</h2>

<p>Ini Adalah section satu</p></section>

<section><h2>Section Dua</h2>

<p>Ini Adalah section Dua</p></section>

Hasilnya seperti yang terlihat ditanda panah merah :



1. Elemen <time> digunakan untuk mendefinisikan waktu. Elemen ini memiliki tipe *inline*. Contoh :

Sekarang Jam <time>19:00</time>.

Hasilnya seperti yang terlihat ditanda panah merah :



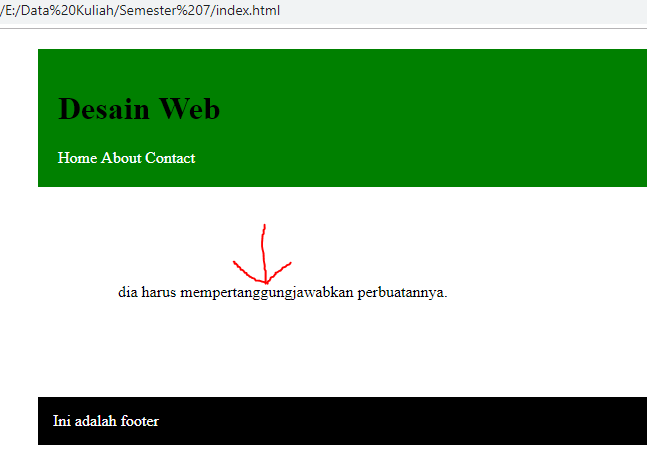
1. Elemen <wbr> digunakan untuk merepresentasikan posisi sebuah kata atau frase yang dapat dipecah (dipindahkan) ke baris baru (line break). Contoh :

<p>

dia harus mem<wbr>pertanggung<wbr>jawabkan perbuatannya.

</p>

Hasilnya seperti yang terlihat ditanda panah merah :



1. Elemen <track> digunakan untuk menampilkan teks berdasarkan waktu putar media (audio maupun video) atau tepatnya adalah timed text tracks. Contoh :

<video width="320" height="240" controls>

<source src="oke.mp4" type="video/mp4">

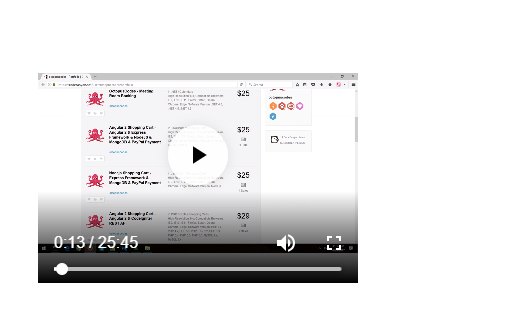
<source src="oke.ogg" type="video/ogg">

<source src="oke.webm" type="video/webm">

<track src="iya.en.vtt" kind="subtitles" srclang="en" label="English" default="">

</video>

Hasilnya sebagai berikut :



### 1.4.3 Graphics and Canvas

digunakan untuk mempermudah user dalam melihat perbandangin hasil sekarang, kemarin dan hari ini, canvas adalah elemen yang bisa digambar dengan kode. Contoh:

<div class="container">

<canvas id="grafikBatang"></canvas>

</div>

<script>

var ctx = document.getElementById("grafikBatang").getContext('2d');

var myChart = new Chart(ctx, {

type: 'bar',

data: {

labels: ["Red", "Blue", "Yellow", "Green", "Purple", "Orange"],

datasets: [{

label: '# Jumlah Survei',

data: [12, 19, 3, 5, 2, 3],

backgroundColor: [

'rgba(255, 99, 132, 0.2)',

'rgba(54, 162, 235, 0.2)',

'rgba(255, 206, 86, 0.2)',

'rgba(75, 192, 192, 0.2)',

'rgba(153, 102, 255, 0.2)',

'rgba(255, 159, 64, 0.2)'

],

borderColor: [

'rgba(255,99,132,1)',

'rgba(54, 162, 235, 1)',

'rgba(255, 206, 86, 1)',

'rgba(75, 192, 192, 1)',

'rgba(153, 102, 255, 1)',

'rgba(255, 159, 64, 1)'

],

borderWidth: 1

}]

},

options: {

scales: {

yAxes: [{

ticks: {

beginAtZero:true

}

}]

}

}

});

</script>

Hasilnya sebagai berikut :



### 1.4.4 Tenatang Canvas

Canvas awalnya diperkenalkan oleh [Apple](https://id.wikipedia.org/wiki/Apple_Inc.) untuk digunakan dalam komponen sendiri OS [Mac](https://id.wikipedia.org/wiki/Mac) X [WebKit](https://id.wikipedia.org/wiki/WebKit) pada tahun 2004,menggerakkan aplikasi seperti widget Dashboard dan browser Safari. Kemudian, pada tahun 2005 itu diadopsi dalam versi 1,8 browser Gecko,dan Opera pada tahun 2006,dan standar oleh Teknologi Hypertext Aplikasi Web Kelompok Kerja (WHATWG) pada spesifikasi yang diusulkan baru untuk teknologi web generasi berikutnya.

### Kanvas Gambar

Kanvas gambar ini digunakan untuk menampilkan gambar. Contoh :

<script>

function drawImage() {

var canvas=document.getElementById("myCanvas");

var c=canvas.getContext("2d");

var destX = 69;

var destY = 50;

var imageObj = new Image();

imageObj.onload = function() {

c.drawImage(imageObj,destX,destY);

}

imageObj.src = "1.jpg";

}

</script>

</head>

<body onload="drawImage()">

<canvas id="myCanvas" width="578" height="400"></canvas>

</body>

Hasilnya sebagai berikut :



### 1.4.6 Koordinat Kanvas

Kanvas adalah grid dua dimensi. Sudut kiri atas kanvas memiliki koordinat (0,0) Jadi, metode fillRect() di atas memiliki parameter (0,0,150,75).

Ini berarti: Mulai di pojok kiri atas (0,0) dan menggambar persegi panjang piksel 150×75.Contoh :

.Contoh :

<!DOCTYPE HTML>

<html>

<head>

<style>

body {

margin: 0px;

padding: 0px;

}

#myCanvas {

border: 1px solid #9C9898;

}

</style>

<script>

function writeMessage(canvas, message) {

var context = canvas.getContext('2d');

context.clearRect(0, 0, canvas.width, canvas.height);

context.font = '18pt Calibri';

context.fillStyle = 'black';

context.fillText(message, 10, 25);

}

function getMousePos(canvas, evt) {

var rect = canvas.getBoundingClientRect(), root = document.documentElement;

// return relative mouse position

var mouseX = evt.clientX - rect.top - root.scrollTop;

var mouseY = evt.clientY - rect.left - root.scrollLeft;

return {

x: mouseX,

y: mouseY

};

}

window.onload = function() {

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

canvas.addEventListener('mousemove', function(evt) {

var mousePos = getMousePos(canvas, evt);

var message = "Mouse position: " + mousePos.x + "," + mousePos.y;

writeMessage(canvas, message);

}, false);

};

</script>

</head>

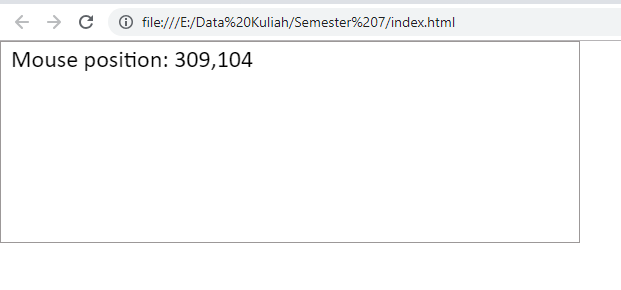
<body>

<canvas id="myCanvas" width="578" height="200"></canvas>

</body>

</html>

Hasilnya sebagai berikut :



### 1.4.7 Draw Line

Digunakan untuk membuat kanvas garis lurus. Contoh :

<!DOCTYPE HTML>

<html>

<head>

<style>

body {

margin: 0px;

padding: 0px;

}

</style>

</head>

<body>

<canvas id="myCanvas" width="578" height="200"></canvas>

<script>

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

context.beginPath();

context.moveTo(100, 150);

context.lineTo(450, 50);

context.lineWidth = 15;

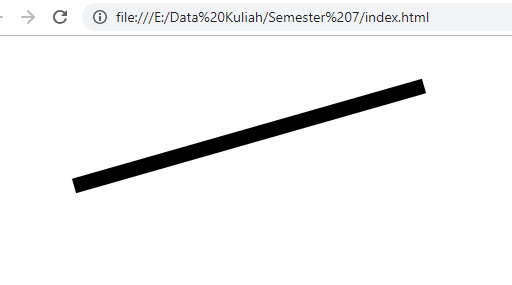
context.stroke();

</script>

</body>

</html>

Hasilnya sebagai berikut :



### 1.4.8 Draw a Circle

Digunakan untuk menggambar lingkaran di dalam kanvas. Contoh :

<canvas id="myCanvas" width="300" height="150" style="border:1px solid #d3d3d3;"></canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

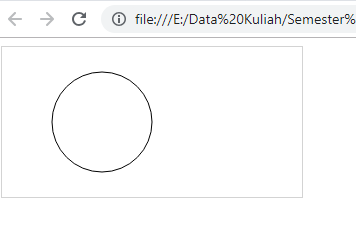
ctx.beginPath();

ctx.arc(100, 75, 50, 0, 2 \* Math.PI);

ctx.stroke();

</script>

Hasilnya sebagai berikut :



### 1.4.9 Canvas Gradient

Digunakan untuk membuat gradasi warna di canvas. Contoh :

<canvas id="myCanvas" width="200" height="100"

style="border:1px solid #d3d3d3;"></canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var grd = ctx.createLinearGradient(0,0,200,0);

grd.addColorStop(0,"red");

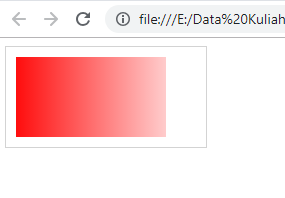
grd.addColorStop(1,"white");

ctx.fillStyle = grd;

ctx.fillRect(10,10,150,80);

</script>

Hasilnya sebagai berikut :



### 1.4.10 SVG

adalah singkatan dari Scalable Vector Graphics, yang digunakan untuk mendefinisikan vektor berbasis grafis untuk Web. Adapun grafik yang didefinisikan berformat XML.

### 1.4.11 SVG Rectangle

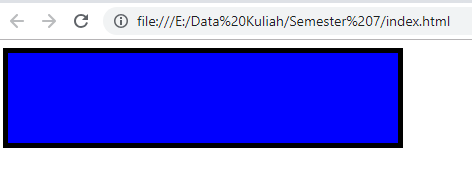
Contoh :

<svg width="400" height="100">

<rect width="400" height="100" style="fill:rgb(0,0,255);stroke-width:10;stroke:rgb(0,0,0)" />

</svg>

Hasilnya sebagai berikut :



### 1.4.11 Path

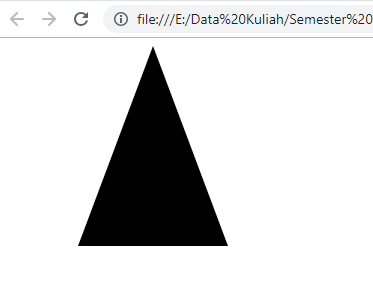
Contoh :

<svg height="210" width="400">

<path d="M150 0 L75 200 L225 200 Z" />

</svg>

Hasilnya sebagai berikut :



### 1.4.12 Plug-ins

Contoh :

<object data="1.jpg"></object>

Hasilnya sebagai berikut :



### Latihan HTML5

{semua dari sub-bab 1.2. Tag-tag HTML sampai selesai di coba, hasil percobaan berupa kode sumber / source code di taruh disini termasuk capture hasil dari browser, dirapikan}

1.4.1 Elemen

1. Sebuah elemen HTML biasanya terdiri dari **start** tag dan **end** tag, dengan konten yang dimasukkan di antara:
2. < h1 > Paragraf < / h1 >

1.4.2 Atribut

1. Atribut memberikan **informasi tambahan** tentang elemen
2. <a href="https://www.wikipedia.com">Pilih Paragaraf</a>
3. 

1.4.3 Heading

1. Mendefinisikan Judul yang dibuat dan arti dari judul tersebut
2. <h1>KUCING </h1>

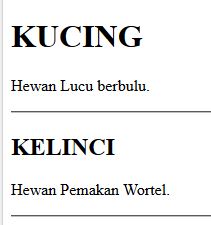
<p>Hewan Lucu berbulu.</p>

<hr>

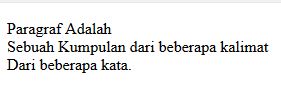
<h2>KELINCI</h2>

<p>Hewan Pemakan Wortel.</p>

<hr>

1. 

1.4.4 PARAGRAPHS

1. element yang mendefinisikan ayat atau sebuah penjelasan-penjelasan.
2. <p>Paragraf Adalah<br>Sebuah Kumpulan dari beberapa kalimat<br>Dari beberapa kata.</p>
3. 

1.4.6 HTML FORMATTING

1. Mendefinisikan sebuah tulisan menjadi tebal,miring, dll
2. <b>DESAIN WEB</b>
3. 

1.5.7 COMMENTS

1. digunakan untuk menyisipkan komentar dalam kode sumber HTML.
2. <!-- This is a comment -->

<p>Paragraph.</p>

<!-- Comments are not displayed in the browser -->

1. 

1.5.8 COLORS

a. digunakan untuk memberikan warna

b. <h1 style="background-color:DodgerBlue;">First Paragraf</h1>

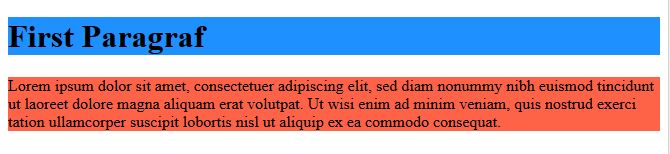
<p style="background-color:Tomato;">

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

</p>

c.



1.5.9 LINKS

1. Memberikan sebuah alamat yang bisa diakses ketika kursor diarahkan pada tulisan
2. <h2>HTML Links</h2>

<p><a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a></p>

1. 

**1.5.10 IMAGES**

a. Melampirkan atau menyisipkan sebuah gambar pada htlm

b. <h2>HTML Image</h2>

<img src="img\_girl.jpg" alt="Girl in a jacket" width="200" height="400">

c.

**1.5.11 TABLES**

a. Meberikan sebuah table pada html

b. <h2>Cell that spans two columns</h2>

<p>To make a cell span more than one column, use the colspan attribute.</p>

<table style="width:100%">

<tr>

<th>Name</th>

<th colspan="2">Telephone</th>

</tr>

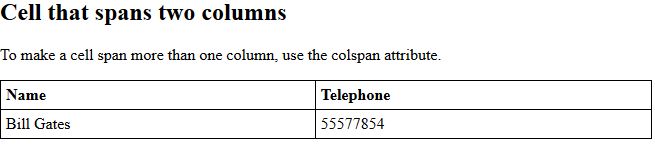
<tr>

<td>Bill Gates</td>

<td>55577854</td>

</tr>

</table>

c. 

**1.5.12 LISTS**

a. Memberikan List berupa urutan daftar

b. <h2>Daftar Menu</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

c. 

**1.5.13 UNORDERED LIST**

a. Memberikan list berupa symbol titik hitam

b. <h2>Daftar Menu</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

c.

**1.5.14 ORDERED LIST**

a. Memberikan sebuah list pada menu dengan mengunakan nomer

b. <h2>An Ordered HTML List</h2>

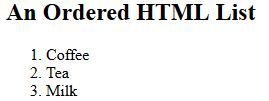
<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

c. 

**1.5.15 DEFINITION LIST**

a. Sebuah daftar deskripsi adalah daftar istilah, dengan deskripsi setiap istilah.

b. <h2>A Description List</h2>

<dl>

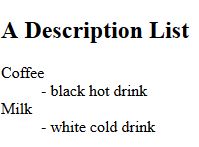
<dt>Coffee</dt>

<dd>- black hot drink</dd>

<dt>Milk</dt>

<dd>- white cold drink</dd>

</dl>

1. 

**1.5.16 NESTED LIST**

a. sebuah daftar yang bersarang yaitu daftar didalam daftar

b. <h2>A Nested List</h2>

<p>List can be nested (lists inside lists):</p>

<ul>

<li>Coffee</li>

<li>Tea

<ul>

<li>Black tea</li>

<li>Green tea</li>

</ul>

</li>

<li>Milk</li>

</ul>

c. 

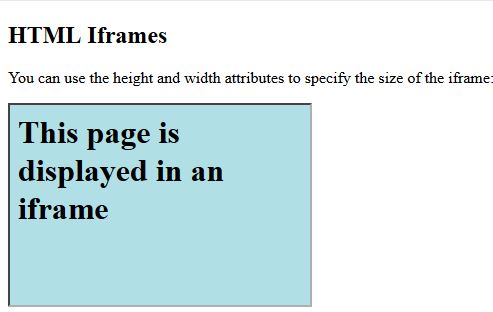
**1.5.17 IFRAMES**

a. meberikan bolder atau bingkai pada tulisan sehingga tulisan dalam sebuah bingkai

b. <h2>HTML Iframes</h2>

<p>You can use the height and width attributes to specify the size of the iframe:</p>

<iframe src="demo\_iframe.htm" height="200" width="300"></iframe>

c.

**1.5.18 JAVASCRIPT**

a. Membuat halaman HTML lebih menarik dan dinamis

b. <h1>My First JavaScript</h1>

<button type="button"

onclick="document.getElementById('demo').innerHTML = Date()">

Click me to display Date and Time.</button>

<p id="demo"></p>

c.

**1.5.19 FILE PATHS**

a. sebuah path file menjelaskan lokasi file dalam struktur folder situs Web.

path file yang digunakan saat menghubungkan ke file eksternal seperti:

* Halaman web
* Images
* style sheet
* javascripts

b. <h2>Using a Full URL File Path</h2>

<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain" style="width:300px">

1. 

**1.5.20 HEAD**

* 1. wadah untuk metadata (data tentang data) dan ditempatkan di antara <html>tag dan <body>tag.
  2. <head>

<title>Page Title</title>

</head>

<body>

<p>The content of the body element is displayed in the browser window.</p>

<p>The content of the title element is displayed in the browser tab, in favorites and in search engine results.</p>

C.

**1.5.21 LAYOUT**

a. Website sering menampilkan konten dalam beberapa kolom (seperti majalah atau koran).

b. <html lang="en">

<head>

<title>CSS Template</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {

box-sizing: border-box;

}

body {

font-family: Arial, Helvetica, sans-serif;

}

/\* Style the header \*/

header {

background-color: #666;

padding: 30px;

text-align: center;

font-size: 35px;

color: white;

}

/\* Create two columns/boxes that floats next to each other \*/

nav {

float: left;

width: 30%;

height: 300px; /\* only for demonstration, should be removed \*/

background: #ccc;

padding: 20px;

}

/\* Style the list inside the menu \*/

nav ul {

list-style-type: none;

padding: 0;

}

article {

float: left;

padding: 20px;

width: 70%;

background-color: #f1f1f1;

height: 300px; /\* only for demonstration, should be removed \*/

}

/\* Clear floats after the columns \*/

section:after {

content: "";

display: table;

clear: both;

}

/\* Style the footer \*/

footer {

background-color: #777;

padding: 10px;

text-align: center;

color: white;

}

/\* Responsive layout - makes the two columns/boxes stack on top of each other instead of next to each other, on small screens \*/

@media (max-width: 600px) {

nav, article {

width: 100%;

height: auto;

}

}

</style>

</head>

<body>

<h2>CSS Layout Float</h2>

<p>In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other.</p>

<p>Resize the browser window to see the responsive effect (you will learn more about this in our next chapter - HTML Responsive.)</p>

<header>

<h2>Cities</h2>

</header>

<section>

<nav>

<ul>

<li><a href="#">London</a></li>

<li><a href="#">Paris</a></li>

<li><a href="#">Tokyo</a></li>

</ul>

</nav>

<article>

<h1>London</h1>

<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>

<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>

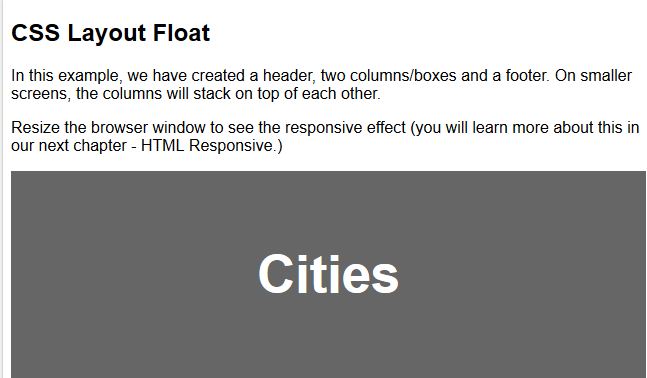
</article>

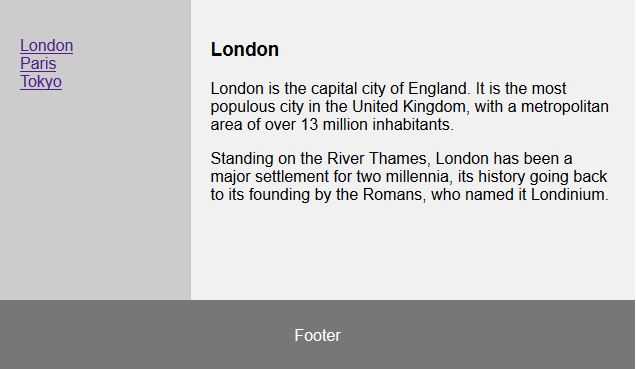
</section>

<footer>

<p>Footer</p>

</footer>





1.5.22 RESPONSIVE

a. Mengatur responsive terhadap website di buka pada browser apa

b. <head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

\* {

box-sizing: border-box;

}

.menu {

float: left;

width: 20%;

}

.menuitem {

padding: 8px;

margin-top: 7px;

border-bottom: 1px solid #f1f1f1;

}

.main {

float: left;

width: 60%;

padding: 0 20px;

overflow: hidden;

}

.right {

background-color: lightblue;

float: left;

width: 20%;

padding: 10px 15px;

margin-top: 7px;

}

@media only screen and (max-width:800px) {

/\* For tablets: \*/

.main {

width: 80%;

padding: 0;

}

.right {

width: 100%;

}

}

@media only screen and (max-width:500px) {

/\* For mobile phones: \*/

.menu, .main, .right {

width: 100%;

}

}

</style>

</head>

<body style="font-family:Verdana;">

<div style="background-color:#f1f1f1;padding:15px;">

<h1>Cinque Terre</h1>

<h3>Resize the browser window</h3>

</div>

<div style="overflow:auto">

<div class="menu">

<div class="menuitem">The Walk</div>

<div class="menuitem">Transport</div>

<div class="menuitem">History</div>

<div class="menuitem">Gallery</div>

</div>

<div class="main">

<h2>The Walk</h2>

<p>The walk from Monterosso to Riomaggiore will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.</p>

<img src="img\_5terre.jpg" style="width:100%">

</div>

<div class="right">

<h2>What?</h2>

<p>Cinque Terre comprises five villages: Monterosso, Vernazza, Corniglia, Manarola, and Riomaggiore.</p>

<h2>Where?</h2>

<p>On the northwest cost of the Italian Riviera, north of the city La Spezia.</p>

<h2>Price?</h2>

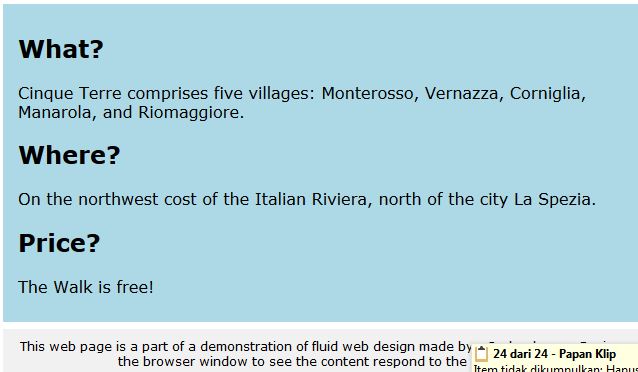
<p>The Walk is free!</p>

</div>

</div>

<div style="background-color:#f1f1f1;text-align:center;padding:10px;margin-top:7px;font-size:12px;"> This web page is a part of a demonstration of fluid web design made by w3schools.com. Resize the browser window to see the content respond to the resizing.</div>

</body>

c.

1.5.23 COMPUTERCODE

a. HTML <kbd>elemen mewakili input pengguna, seperti input keyboard atau perintah suara.Teks dikelilingi oleh <kbd>tag biasanya ditampilkan dalam font default monospace browser:

b. <h2>Computer Code</h2>

<p>Some programming code:</p>

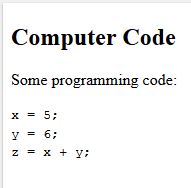
<code>

x = 5;<br>

y = 6;<br>

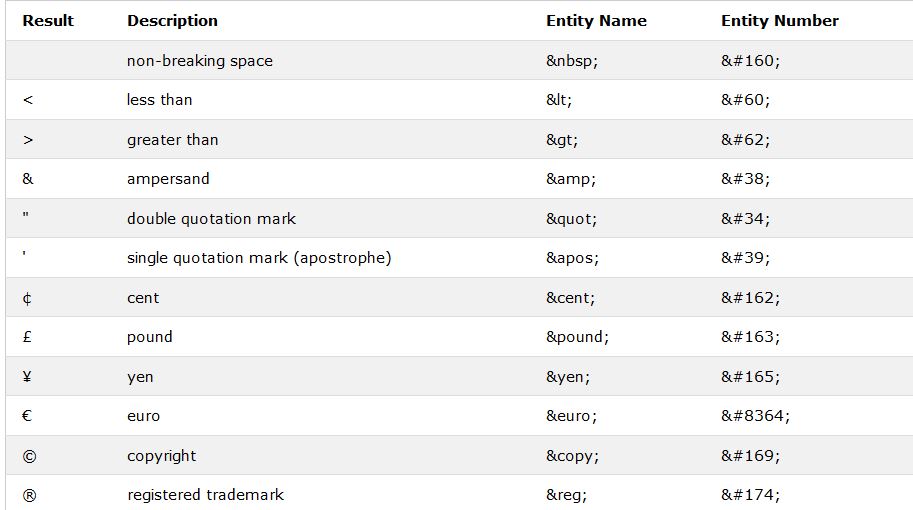
z = x + y;

</code>

* 1. 

**1.5.24 ENTITIES**

a. Reserved karakter dalam HTML harus diganti dengan entitas karakter. Karakter yang tidak hadir pada keyboard Anda juga dapat digantikan oleh entitas.

b. 

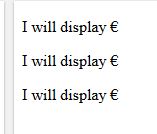
**1.5.25 SYMBOL**

a. Merupakan symbol-simbol yang digunakan dalam html, entitas HTML digambarkan dalam bab sebelumnya.Banyak matematika, teknis, dan simbol mata uang, yang tidak hadir pada keyboard biasa.Untuk menambahkan simbol seperti ke halaman HTML, Anda dapat menggunakan HTML nama entitas.Jika tidak ada nama entitas ada, Anda dapat menggunakan nomor entitas, desimal, atau referensi heksadesimal.

b. <p>I will display &euro;</p>

<p>I will display &#8364;</p>

<p>I will display &#x20AC;</p>

* 1. 

**1.5.26 CHARSET**

a. Untuk menampilkan halaman HTML dengan benar, web browser harus mengetahui set karakter yang digunakan di halaman.

b. <meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">

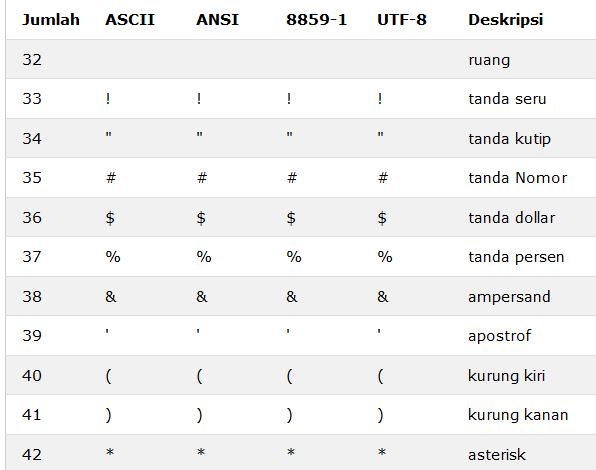
<meta charset="UTF-8">

**1.5.27 Character Set ASCII**

a. ANSI adalah yang pertama karakter default resmi ditetapkan di Windows.

Karakter default diatur dalam HTML 4 adalah 8859-1.

Karakter default diatur dalam HTML 5 adalah UTF-8.

b. 

1.5.28 Character Set Latin-1 atau ISO/IEC 8859-1

1.5.29 Character Set UTF-16 dan UCS-2

1.5.30 Character Set UTF-32 dan UCS-4

1.5.31 Character Set dan Character Encoding UTF-8

1.5.32 URL Encoding

a. URL hanya dapat dikirim melalui Internet menggunakan [ASCII karakter-set](https://www.w3schools.com/charsets/ref_html_ascii.asp) .

Sejak URL sering mengandung karakter luar ASCII set, URL tersebut harus dikonversi ke dalam format ASCII valid.

encoding URL menggantikan karakter ASCII yang tidak aman dengan "%" diikuti oleh dua digit heksadesimal.

URL tidak dapat berisi spasi. URL encoding biasanya menggantikan spasi dengan tanda tambah (+) tanda atau dengan% 20.

b.

**1.5.33 Form**

a. element untuk proses inputan

b. <h2>HTML Forms</h2>

<form action="/action\_page.php">

First name:<br>

<input type="text" name="firstname" value="Mickey">

<br>

Last name:<br>

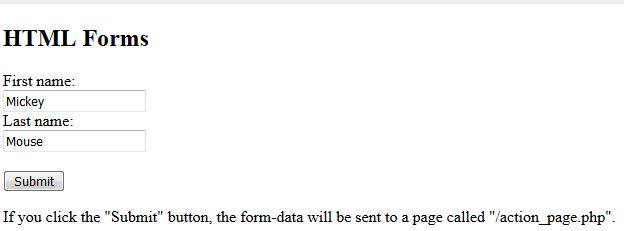
<input type="text" name="lastname" value="Mouse">

<br><br>

<input type="submit" value="Submit">

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".</p>

c. 

1.5.34 Form Elemen

a. Bentuk elemen yang paling penting adalah <input>elemen.

The <input>elemen dapat ditampilkan dalam beberapa cara, tergantung pada type atribut.

b. <h2>The input Element</h2>

<form action="/action\_page.php">

Enter your name:

<input name="firstname" type="text">

<br><br>

<input type="submit">

</form>

C. 

1.5.35 Input Type

a. membuat proses Inputan dalam html

b. <h2>Text field</h2>

<p>The <strong>input type="text"</strong> defines a one-line text input field:</p>

<form action="/action\_page.php">

First name:<br>

<input type="text" name="firstname">

<br>

Last name:<br>

<input type="text" name="lastname">

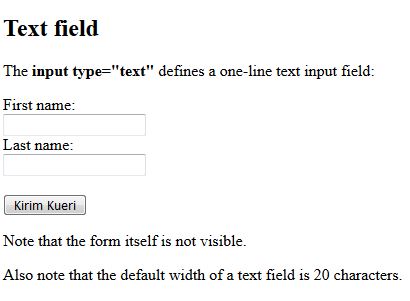
<br><br>

<input type="submit">

</form>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of a text field is 20 characters.</p>

c. 

1.5.36 Input Atribut

a. Menentukan nilai awal inputan

b. <h2>The value Attribute</h2>

<p>The value attribute specifies the initial value for an input field:</p>

<form action="">

First name:<br>

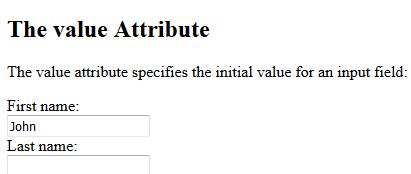
<input type="text" name="firstname" value="John">

<br>

Last name:<br>

<input type="text" name="lastname">

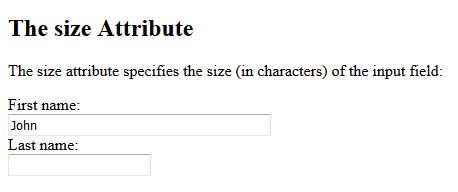
</form>

c.

1.5.37 Atribut Nilai

a. Memberikan atribut dengan ukuran angka

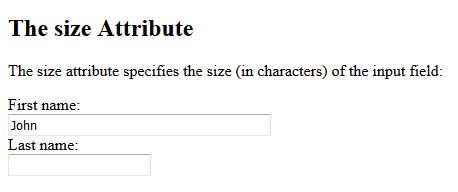
b. <form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John">  
</form>

c.

1.5.38 Atribut Read-Only

a. hanya memberikan atribut untuk dibaca saja

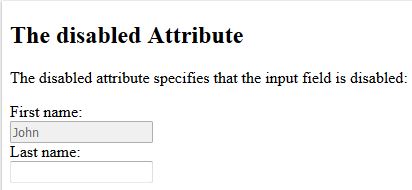
b. <form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John" readonly>  
</form>

c.

1.5.39 Atribut Disable

a. menonaktifkan atribut

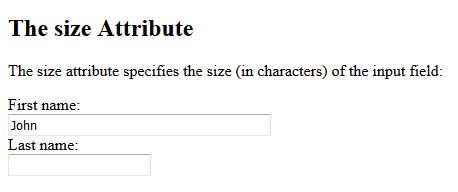
b. <form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John" disabled>  
</form>

c. 

1.5.40 Atribut Ukuran

a. memberikan nilai ukuran pada atribut

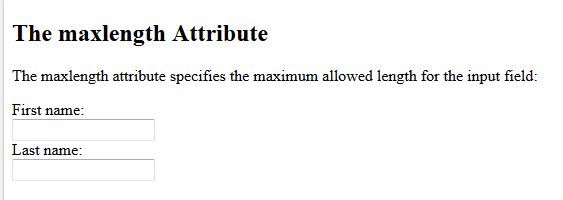
b. <form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John" size="40">  
</form>

c.

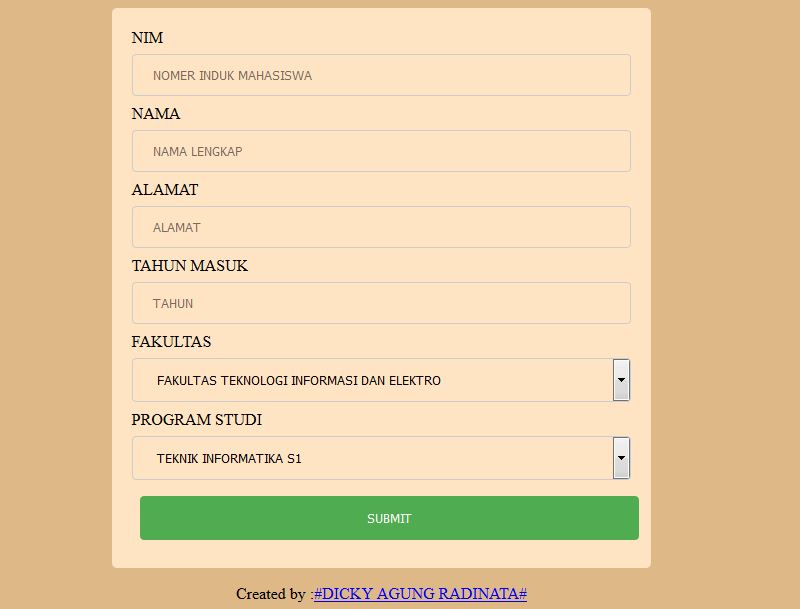
1.5.41 Atribut Max-Length

a. memeberikan panjang maxsimum atribut yang diperbolehkan

b. <form action="">  
  First name:<br>  
  <input type="text" name="firstname" maxlength="10">  
 </form>

c.

**CONTOH PEMBUATAN FORM MENGGUNAKAN HTML DAN CSS**

****

Contoh :

<html>

<head>

<title>UTY MAHASISWA</title>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

input[type=text], select{

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #CCC;

border-radius: 4px;

box-sizing: border-box;

background-color: bisque;

}

input[type=submit]{

width: 100%;

background-color: #4FAC50;

color: white;

padding: 14px 20px;

margin: 8px;

border: none;

border-radius: 4px;

box-sizing: pointer;

}

input[type=submit]:hover{background-color: #45A049;}

div{

border-radius: 5px;

background-color: bisque;

margin-left: auto;

margin-right: auto;

alignment-adjust: central;

padding: 20px;

}

footer{

text-align: center;

}

body{

background-color: burlywood;

margin-top: 15%;

margin-right: 30%;

margin-left: 30%;

}

header,h1{

text-align: center;

color: black;

}

</style>

</head>

<body>

<header role="banner"><h1>UTY</h1>

<h1>MAHASISWA</h1></header>

<article>

<section>

<div>

<form action="#">

<label for="nim">NIM</label>

<input type="text" id="nim" name="NIM" placeholder="NOMER INDUK MAHASISWA">

<label for="nama">NAMA</label>

<input type="text" id="nama" name="NAMA" placeholder="NAMA LENGKAP">

<label for="alamat">ALAMAT</label>

<input type="text" id="alamat" name="ALAMAT" placeholder="ALAMAT">

<label for="tahun">TAHUN MASUK</label>

<input type="text" id="tahun" name="TAHUN MASUK" placeholder="TAHUN">

<label for="fak">FAKULTAS</label>

<select id="fak" name="FAKULTAS">

<option value="FITE">FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO</option>

<option value="FEB">FAKULTAS EKONOMI DAN BISNIS</option>

<option value="FST">FAKULTAS SAINT DAN TEKNIK</option>

<option value="HUMANIORA">FAKULTAS HUMANIORA</option>

<option value="FIP">FAKULTAS ILMU PENDIDIKAN</option>

</select>

<label for="country">PROGRAM STUDI</label>

<select id="country" name="country">

<option value="TI S1">TEKNIK INFORMATIKA S1</option>

<option value="SI S1">SISTEM INFORMASI S1</option>

<option value="MI S1">MANAGEMENT INFORMATIKA D3</option>

<option value="TI D3">TEKNIK INFORMATIKA D3</option>

</select>

<input type="submit" value="SUBMIT">

</form>

</div>

</section>

<footer><P>Created by :<a href="#">#DICKY AGUNG RADINATA#</a></P></footer>

<footer><P>Id Number :<a href="#">#5150411019#</a></P></footer>

</article>

</body>

</html>