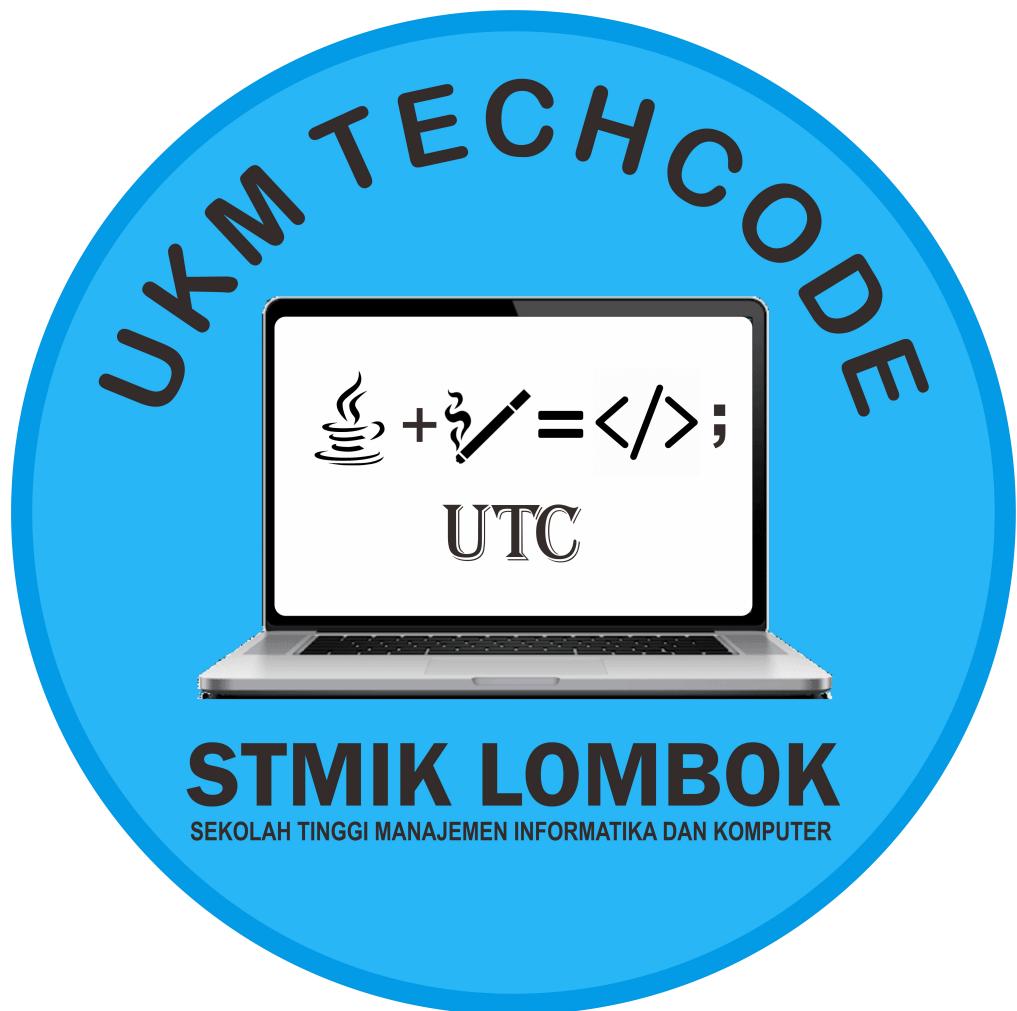


MODUL HTML UKM TECHCODE



**UNIT KEGIATAN MAHASISWA TECHNOLOGY CODE
STMIK LOMBOK TAHUN 2022**

Apa itu HTML?

Mari kita lihat pengertian HTML menurut wikipedia:

HTML atau HyperText Markup Language merupakan sebuah bahasa **markah** untuk membuat halaman web.¹

Paham kan maksudnya?

Kalau belum paham, sini saya jelaskan...

Jadi, HTML itu adalah **sebuah bahasa** yang menggunakan **markup** atau penanda untuk **membuat halaman web**.

Penanda atau markup ini, nanti akan kita sebut dengan **Tag**.

HTML berperan untuk menentukan struktur konten dan tampilan dari sebuah web.

Kalau kita ibaratkan nih..

HTML itu seperti batu bata untuk membangun rumah. Batu bata ini dapat disusun, hingga menjadi fondasi dasar.



Dalam membuat halaman web, HTML tidak sendirian. Ada bahasa lain lagi yang menjadi pelengkapnya, yakni CSS dan Javascript.

CSS adalah bahasa khusus yang digunakan untuk memperindah tampilan web.

Lalu Javascript bertugas untuk membuat halaman web menjadi hidup. Karena dengan Javascript, kita dapat menentukan fungsi-fungsi maupun efek yang akan diterapkan di website.

Peralatan untuk Belajar HTML

Nah, sekarang.. Tiba saatnya kita praktik!

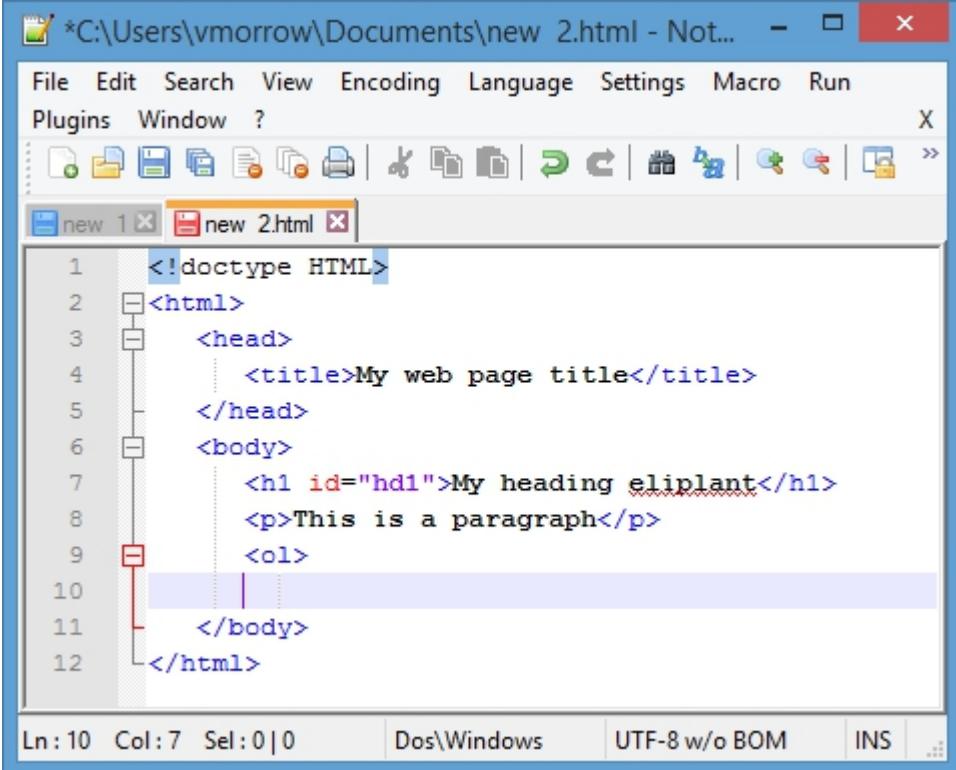
Tapi sebelum itu, kamu harus siapkan beberapa alat yang akan digunakan untuk praktik.

Adapun peralatan yang harus kamu persiapkan adalah:

1. Teks Editor untuk Menulis HTML

Teks editor akan kita gunakan untuk menulis kode-kode HTML. Kamu bebas menggunakan teks editor apapun.

Notepad boleh, Notepad++ juga boleh..



```
<!doctype HTML>
<html>
<head>
    <title>My web page title</title>
</head>
<body>
    <h1 id="hd1">My heading eliplant</h1>
    <p>This is a paragraph</p>
    <ol>
        <li></li>
    </ol>
</body>
</html>
```

Apapun teks editornya, yang penting bisa digunakan untuk membuat dan menulis dokumen HTML.

Namun, pada tutorial ini.. kita akan menggunakan teks editor [Visual Studio Code](#).

2. Web Browser untuk Membuka HTML

Web browser akan kita gunakan untuk membuka HTML. Kamu juga bebas menggunakan web browser apapun.



Saran saya sih.. gunakan web browser yang terbaru, karena kita juga akan menggunakan HTML versi yang terbaru.

Firefox atau Google Chrome, saya kira sudah cukup.

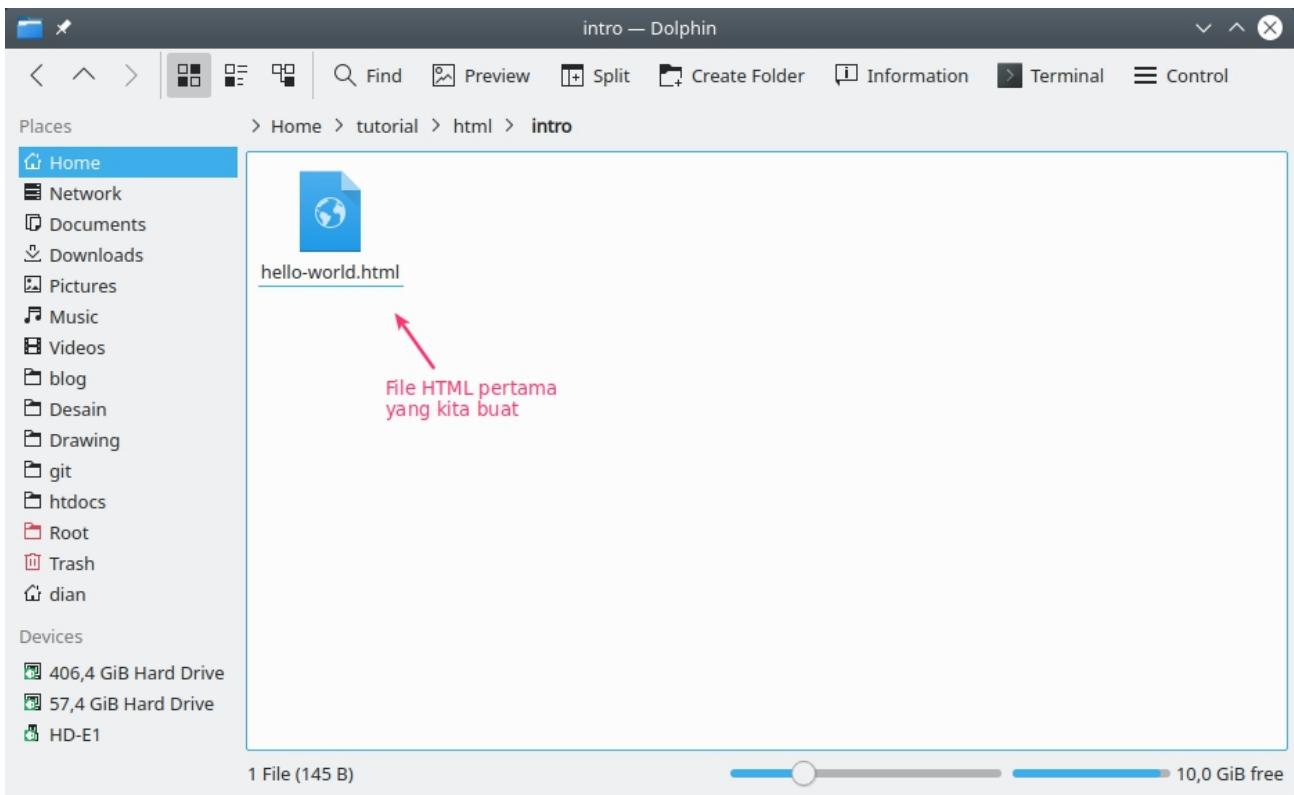
Membuat Dokumen HTML Pertamamu

Kini tiba saatnya, kamu harus mencoba sendiri membuat dokumen HTML. Caranya sangat mudah.

Mari kita mulai dengan membuka teks editor, lalu tulislah kode berikut.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Belajar HTML #01</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

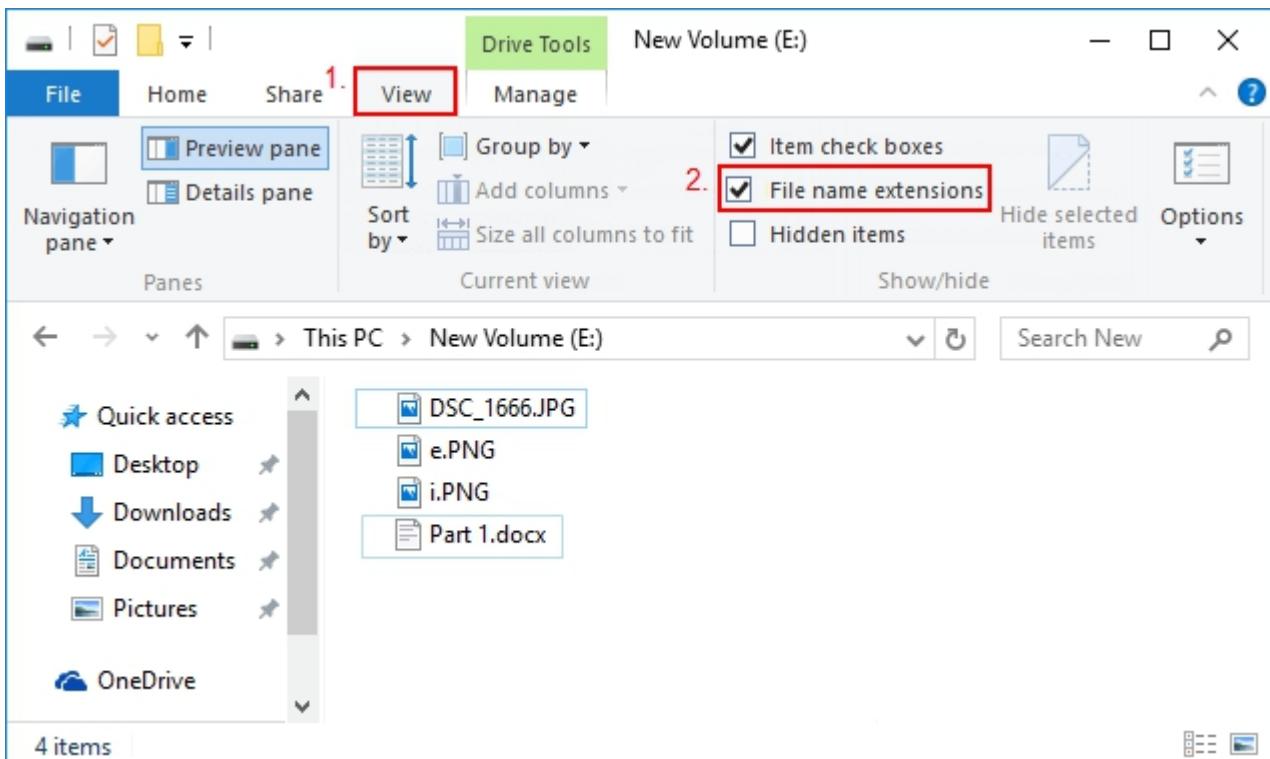
Setelah itu, simpan dengan nama **hello-world.html**.



Tips: buat kamu yang menggunakan Notepad di Windows, simpanlah nama filenya dengan menggunakan tanda petik "**hello-world.html**" agar ekstensinya **.html**, bukan **.txt**.

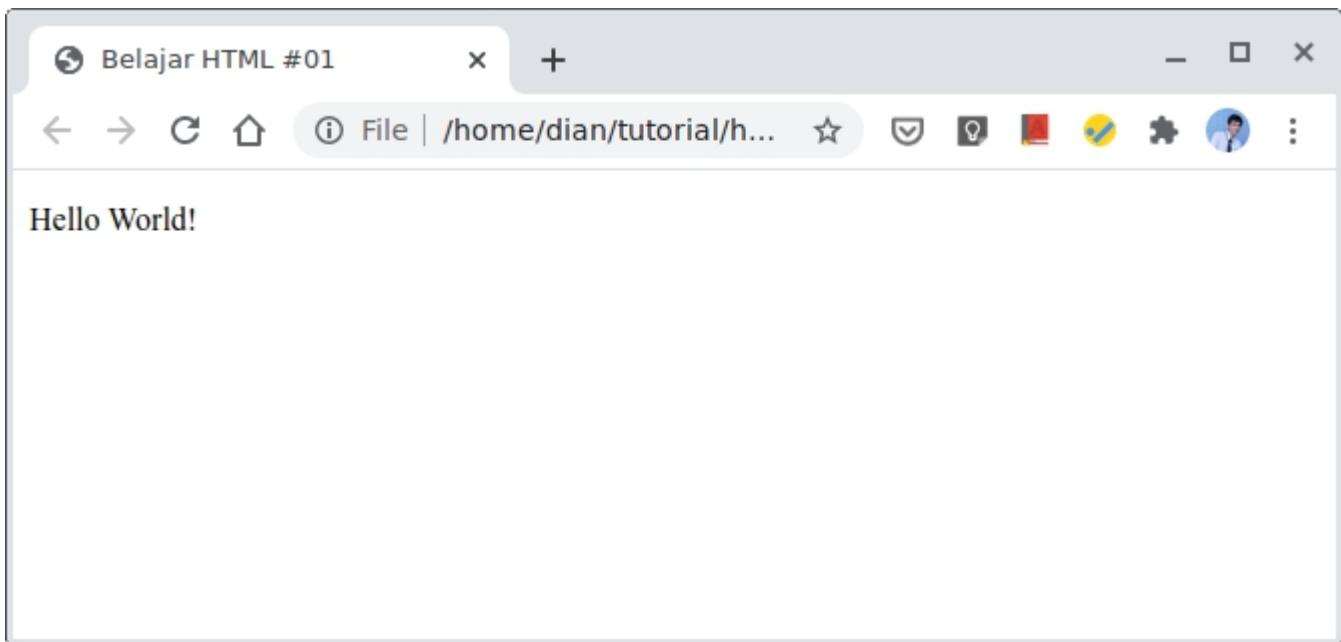
Atau kamu bisa aktifkan fitur show extension pada Windows Explorer, agar bisa tahu ekstensi filenya.

Caranya.. masuk ke menu View, lalu centang **File name extensions**.



Oke.. lanjut!

Sekarang cobalah buka file **hello-world.html** dengan web browser.. maka hasilnya:



Selamat!

Kita sudah berhasil membuat halaman web pertama dengan HTML.

Kini giliran saya menjelaskan maksud dari kode di atas, tapi sebelum itu.. saya akan jelaskan dulu tentang nama file untuk HTML.

Nama File untuk HTML

Ada beberapa hal yang perlu diperhatikan dalam membuat nama file HTML:

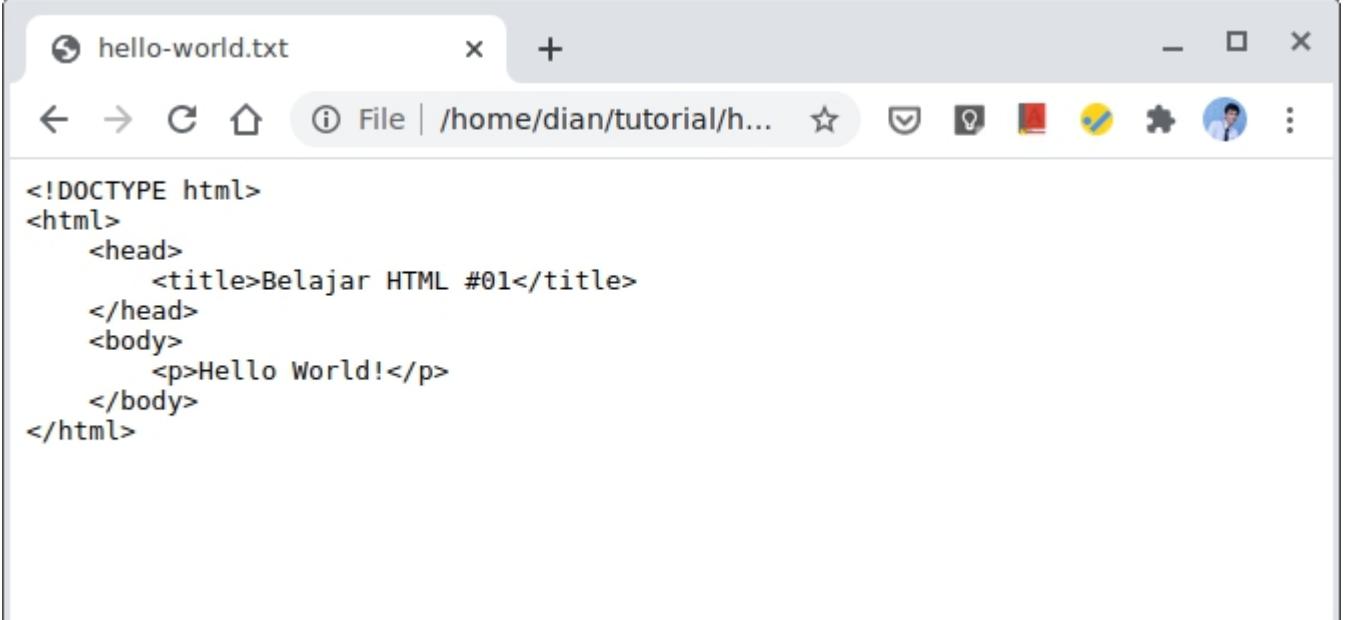
1. Extensi file HTML

Setiap file HTML harus berekstensi **.html**, **.xhtml** (untuk XHTML), dan **.htm** saja.

Jika tidak menggunakan ekstensi ini, maka ia tidak akan bisa dibaca oleh web browser.

Contohnya seperti ini:

Nama filenya adalah **hello-world.txt**.



```
hello-world.txt * +  
← → C ⌂ ⌄ File | /home/dian/tutorial/h... ☆ 🌐 ⓘ 💡 📑 🎯 🚧 🚧 🚧 🚧 🚧 🚧 :  
  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Belajar HTML #01</title>  
  </head>  
  <body>  
    <p>Hello World!</p>  
  </body>  
</html>
```

Sedah jelas, ini bukan file HTML, melainkan sebuah file teks biasa. Maka.. browser akan menampilkan isi file tersebut apa adanya.

2. Nama Khusus untuk Homepage

Jika kamu ingin membuat halaman untuk homepage, maka sebaiknya gunakan nama **index.html**. Karena ia akan otomatis dibuka saat website dikunjungi.

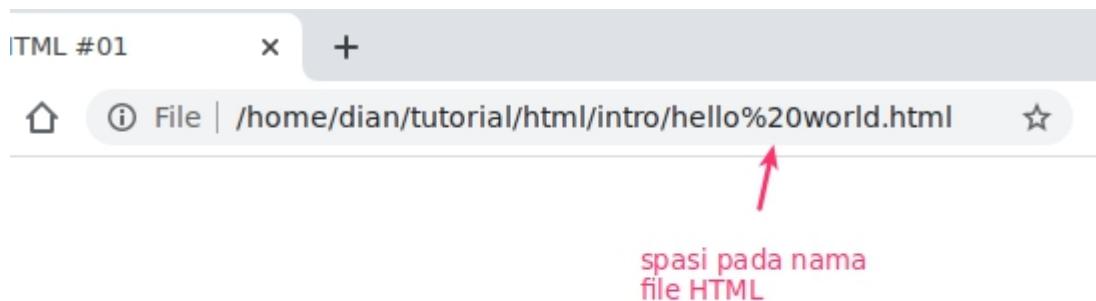
Contohnya, saat kamu membuka www.petanikode.com.. maka file HTML yang akan dibuka adalah **index.html** yang berada di server petanikode.

3. Hindari Beberapa Hal ini..

Penggunaan Spasi

Nama file HTML biasanya akan tercantum pada URL, maka sebaiknya hindari menggunakan spasi pada nama file HTML, agar URL yang dibentuk lebih bagus.

Spasi pada URL, biasanya akan otomatis diubah menjadi **%20**.



Sebagai ganti spasi, kamu bisa gunakan tanda min (-) atau underscore (_).

Jangan Alay!

Berikanlah nama file HTML dengan file yang biasa, hindari menggunakan huruf besar, campuran dari huruf besar dan kecil, dan juga menggunakan simbol.

Contoh:

HeLLoWORLD.html

da*#\$.html

Meskipun nama ini bisa valid, tapi kurang bagus untuk dibaca.. baik oleh manusia, maupun mesin.

Baiklah, berikutnya kita akan pelajari tentang kode-kode HTML.

Struktur Dasar HTML

Berikut ini adalah kode HTML yang baru saja kita buat:



Penulisan kode HTML selalu dimulai dengan deklarasi DOCTYPE, lalu menuliskan tag `<html>` dan di dalamnya terdapat tag `<head>` dan `<body>`.

Kalau kita perhatikan.. struktur dasar kode HTML terdiri dari tiga bagian penting:

1.Bagian Deklarasi

2.Bagian HEAD

3.Bagian BODY

Mari kita bahas satu-per-satu:

1. Bagian Deklarasi

Coba perhatikan kode pada baris pertama: `<!DOCTYPE html>`.

Ini adalah tag deklarasi untuk menyatakan tipe dokumen dan versinya. Pada contoh di atas, kita menyatakan dokumen ini bertipe HTML dan versinya adalah HTML 5.

Nah, untuk HTML versi 4.. beda lagi cara deklarasinya.

Contoh untuk HTML 4.01:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

Cukup panjang bukan..

Lalu apakah boleh kita tidak menulis kode `<!DOCTYPE html>` ini?

Boleh-boleh saja.. dokumen HTML-nya akan tetap bisa dibuka di web browser. Tapi, ini tentunya akan melanggar aturan standar yang dibuat W3C.

O ya, kita bisa mengecek.. apakah HTML yang ditulis sudah benar atau tidak di <https://validator.w3.org/>.

Showing results for contents of text-input area - Nu Html Checker - Mozilla Firefox

Showing results for cont... +

Showing results for contents of text-input area - Nu Html Checker - Mozilla Firefox

Message Filtering

Error Start tag seen without seeing a doctype first. Expected [<!DOCTYPE html>]. From line 1, column 1; to line 1, column 16

<html lang="en"><

Document checking completed.

Source

```
1. <html lang="en">
2.   <head>
3.     <title>Belajar HTML #01</title>
4.   </head>
5.   <body>
6.     <p>Hello World!</p>
7.   </body>
8. </html>
```

Used the HTML parser.
Total execution time 2 milliseconds.

Error disebabkan karena tidak ada deklarasi DOCTYPE

Lanjut...

Berikutnya, di bawah tag deklarasi **<!DOCTYPE html>** terdapat tag pembuka untuk HTML:

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

Tag **<html>** wajib ada di setiap dokumen HTML..

Pada tag ini, kita memberikan atribut **lang="en"** untuk menyatakan kalau konten dokumen HTML ini akan menggunakan bahasa inggris.

Nah di dalam tag **<html>** ini, terdapat dua tag penting lagi.. yakni: tag **<head>** dan tag **<body>**.

Setelah itu barulah terakhir tag HTML ditutup dengan **</html>**.

2. Bagian HEAD

Bagian HEAD adalah bagian kepala dari HTML. Dimulai dari tag **<head>** dan ditutup dengan **</head>**.

```
<!DOCTYPE html>

<html lang="en">

    <head>
        <title>Belajar HTML #01</title>
    </head>
    <body>
        <p>Hello World!</p>
    </body>
</html>
```

Pada bagian HEAD, biasanya digunakan untuk menuliskan tag-tag yang akan dibaca oleh mesin.

Seperti:

Tag meta untuk SEO;

Tag **<title>** untuk judul;

Tempat menulis kode CSS dan Javascript;

dan lain-lain.

3. Bagian BODY

Bagian BODY adalah bagian yang akan ditampilkan pada web browser. Penulisannya dimulai dari tag **<body>** dan ditutup dengan **</body>**.

```
<!DOCTYPE html>

<html lang="en">

    <head>
        <title>Belajar HTML #01</title>
    </head>
    <body>
```

```
<p>Hello World!</p>  
</body>  
</html>
```

Di sinilah nanti kita akan banyak menuliskan konten dengan berbagai macam tag. Saat ini kita baru mengisinya dengan tag `<p>`. Tag `<p>` adalah tag yang digunakan untuk membuat paragraf.

Apa itu Tag, Elemen, dan Atribut dalam HTML?

Tag, elemen, dan atribut merupakan tiga bagian penting yang ada di dalam HTML. Bagi kamu yang baru belajar HTML, wajib hukumnya untuk mengetahui ketiga Hal ini.

Apa itu Tag?

Apa itu Elemen?

Apa itu Atribut?

Serta apa perbedaan Tag, Elemen, dan Atribut?

Mari kita bahas...

Apa itu Tag?

Tag adalah sebuah **penanda awalan** dan **akhiran** dari sebuah **elemen** di HTML. Tag dibuat dengan kurung siku (`<...>`), lalu di dalamnya berisi nama tag dan kadang juga ditambahkan dengan atribut.

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

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



Setiap tag memiliki fungsi masing-masing. Ada yang digunakan untuk membuat judul, membuat link, membuat paragraf, heading, dan lain-lain.

Masih ingat sejarah HTML?

Dulu.. awalnya HTML cuma punya **18 tag**. Sekarang HTML sudah punya sekitar **250** tag.

Wah! banyak ya.

Apa semua tag ini harus kita hafal?

Jawabannya:

Tidak harus, saja juga tidak bisa menghafal semanya. Cukup ketahui tag-tag dasar saja.

Berikut ini daftar tag-tag dasar, yang menurut saya harus kamu ingat.

Tag	Fungsi
<html>	untuk memulai dokumen HTML
<head>	untuk membuat bagian head
<body>	untuk membuat bagian body
<h1> sampai <h6>	untuk membuat heading pada artikel
<p>	untuk membuat paragraf
<!-- -->	untuk membuat komentar

Beberapa tag ini sudah kita coba pada [tutorial pertama](#) dan juga ada yang belum.

Tenang saja.. Nanti juga saya akan perkenalkan tag-tag lain yang umum digunakan dalam pembuatan web.

Untuk saat ini, cukup pahami: **Apa itu tag dan cara menulisnya.**

Cara Menulis Tag HTML yang Benar!

Beberapa orang kadang sering salah dalam menuliskan tag. Ada yang lupa menutup, ada yang salah mengetik namanya, dan semacamnya.

Berikut ini beberapa saran yang perlu diikuti agar bisa menulis tag dengan benar:

1. Tag-tag wajib

Ada beberapa tag yang wajib ada di HTML. Tag ini harus kamu tulis.. kalau tidak, bisa jadi kode HTML-mu akan error menurut validator W3C.

Berikut ini daftar tag yang wajib ada di HTML:

<!DOCTYPE html> — untuk deklarasi type dokumen;

<html> — tag utama dalam HTML;

<head> — untuk bagian kepala dari dokumen;

<title> — untuk judul web;

<body> — untuk bagian body dari dokumen.

2. Gunakan Huruf Kecil

Hindari menggunakan huruf besar dalam menuliskan nama tag dan sebaiknya gunakan huruf kecil saja.

Huruf kecil lebih gampang diketik dan juga akan membuat kode HTML terlihat lebih bersih dan rapi.

Contoh: (bagus)

```
<body>
<p>Belajar tentang tag di HTML</p>
</body>
```

Contoh: (buruk)

```
<BODY>
<P>Belajar tentang tag di HTML</P>
</BODY>
```

Tapi khusus untuk tag **<!DOCTYPE html>**.. ia ditulis dengan huruf besar. Sebenarnya bisa juga dengan huruf kecil dan akan valid menurut validator W3C. Tapi untuk dokumen XHTML, menggunakan **DOCTYPE** dengan huruf kecil akan mengakibatkan error.

3. Pastikan Menutup Tag dengan Benar

Tag HTML nantinya akan ditulis bertumpuk-tumpuk. Artinya, di dalam tag ada tag lagi. Kadang kita sering salah dalam menutup tag yang bertumpuk ini. Akibatnya, kode HTML kita tidak valid.

Tapi tenang saja.. saya punya resep agar kamu mudah mengingatnya:

Jika kamu paham maksud dari jokes di atas, maka bagus.. saya tidak perlu jelaskan lagi.

Tapi kalau belum paham, berikut ini penjelasannya:

Tag yang pertama dibuat, harus ditutup terakhir. Bukan ditutup pertama.

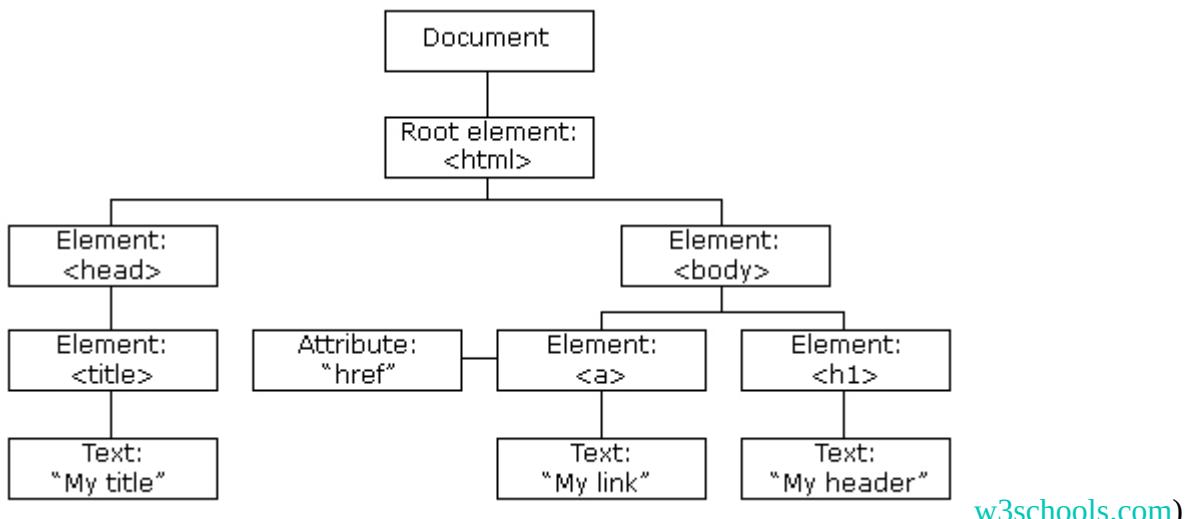
Contoh:

```
<i><b><u>memasak</u></b></i>
```

Tag **<i>** ditutup terakhir, karena ia yang ditulis pertama. Lalu tag **</u>** ditutup pertama kali karena ia berada di dalam tag **** dan **<i>**.

Apa itu Elemen?

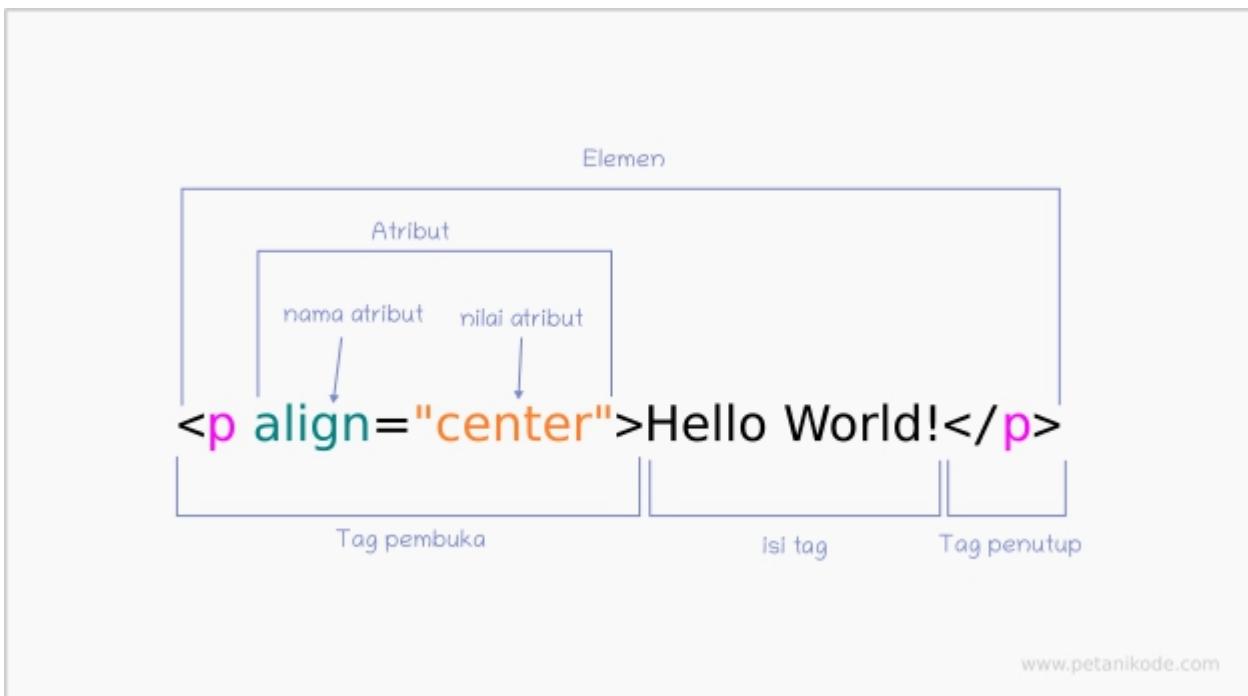
Elemen dalam HTML adalah sebuah komponen yang menyusun dokumen HTML. Elemen kadang juga disebut sebagai node, karena ia merupakan salah satu jenis node yang menyusun dokumen HTML dalam diagram HTML tree.



Pada diagram tersebut, terdapat tiga macam node.. yakni: **Node elemen**, **Node atribut**, dan **Node teks**.

Elemen dibentuk dari **tag pembuka**, **isi tag**, dan **tag penutup**. Kadang juga ditambahkan beberapa atribut.

Contoh:



Pada contoh di atas, terdapat satu elemen <p> dengan atribut **align="center"** dan memiliki isi berupa teks, yakni **Hello World!**.

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

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

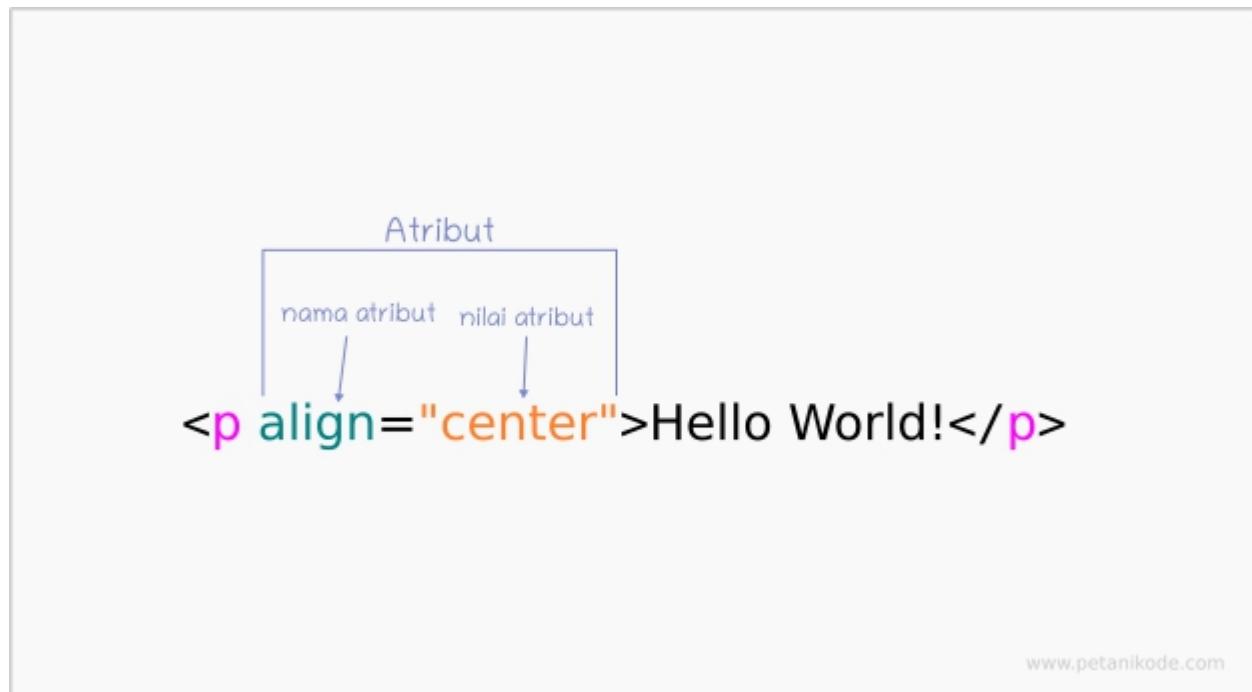
```
<html>
  <body>
    <h1> Ini heading </ h1>
    <p> Ini adalah sebuah paragraf. </ p>
    <p> Ini adalah paragraf lain. </ p>
  </ Body>
</ Html>
```

Elemen HTML ada banyak jenisnya. Ada elemen khusus untuk teks, ada elemen untuk multimedia, script, tabel, metadata, dll. Nanti kita akan pelajari ini secara bertahap. Semua jenis elemen HTML bisa kamu baca di sini: [HTML elements reference](#).

Beberapa elemen HTML kadang ditambahkan atribut sebagai pelengkap.

Apa itu Atribut?

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



Atribut dapat ditambahkan pada elemen manapun. Ada juga elemen yang mewajibkan menggunakan atribut seperti elemen `<a>`, ``, `<video>`, dll.

Contoh:

```
<a href="https://petanikode.com">Petanikode.com</a>
```

Tag `<a>` adalah tag untuk membuat link. Tag ini mewajibkan menambahkan atribut `href` untuk menyatakan halaman tujuan dari link.

Jumlah atribut pada elemen bisa lebih dari satu.

Contoh:

```

```

Atribut `src` adalah atribut khusus untuk tag `` yang fungsinya untuk menentukan gambar yang akan ditampilkan. Lalu atrubut `width` dan `height` adalah atribut yang mengatur ukurannya.

Jenis-jenis Atribut HTML

Tiap-tiap elemen kadang memiliki atribut khusus yang hanya bisa digunakan pada elemen tersebut. Ada juga atribut yang bersifat global dan bisa ditambahkan ke semua elemen.

Berikut ini jenis-jenis atribut yang harus diketahui:

1. Atribut Global

Atribut Global adalah atribut yang bisa ditambahkan pada semua elemen HTML.

Berikut ini daftar atribut global dan fungsinya:

Nama Atribut	Deskripsi atau fungsi
<code>accesskey</code>	Menentukan tombol <i>shortcut</i> untuk mengaktifkan atau fokus pada elemen
<code>class</code>	Menentukan <i>class CSS</i> yang akan digunakan
<code>contenteditable</code>	Menentukan isi elemen bisa diedit atau tidak
<code>data-*</code>	Digunakan untuk menyimpan data
<code>dir</code>	Menentukan arah teks dari elemen (kiri ke kanan atau sebaliknya)
<code>draggable</code>	Menentukan apakah elemen bisa di <i>drag</i> atau tidak
<code>hidden</code>	untuk menyembunyikan elemen
<code>id</code>	Menentukan id unik untuk elemen
<code>lang</code>	Menentukan bahasa yang digunakan untuk isi elemen
<code>spellcheck</code>	Menentukan apakah isi elemen harus dilakukan pengejaan <i>grammer</i> atau tidak
<code>style</code>	Menentukan <i>inline CSS</i> untuk elemen
<code>tabindex</code>	Menentukan urutan atau indeks tab dari elemen (saat tombol tab ditekan)
<code>title</code>	Menentukan informasi tambahan tentang elemen
<code>translate</code>	Menentukan apakah konten dari elemen bisa diterjemahkan atau tidak

2. Atribut Event

Atribut event adalah atribut yang digunakan untuk menentukan aksi yang akan dilakukan saat terjadi sesuatu pada elemen. Atribut ini nanti akan banyak digunakan pada [pemrograman Javascript](#).

Berikut ini daftar atribut event saat terjadi sesuatu pada Jendela browser:

Nama atribut	Nilai	Fungsi: Menjalankan script
<code>onafterprint</code>	kode javascript	setelah dokumen dicetak
<code>onbeforeprint</code>	kode javascript	sebelum dokumen dicetak
<code>onbeforeunload</code>	kode javascript	sebelum dokumen tidak ter-load
<code>onerror</code>	kode javascript	saat terjadi error
<code>onhashchange</code>	kode javascript	saat terjadi perubahan pada bagian <i>anchor</i> di URL
<code>onload</code>	kode javascript	setelah <i>loading</i> selesai
<code>onmessage</code>	kode javascript	saat ada pesan
<code>onoffline</code>	kode javascript	saat tiba-tiba offline
<code>ononline</code>	kode javascript	saat tiba-tiba online
<code>onpagehide</code>	kode javascript	saat user tidak sedang membuka halaman web

Nama atribut	Nilai	Fungsi: Menjalankan script
onpageshow	kode javascript saat user membuka kembali halaman web	
onpopstate	kode javascript saat history browser berubah	
onresize	kode javascript saat ukuran jendela browser berubah	
onstorage	kode javascript saat area penyimpanan (Web Storage) di-update	
onunload	kode javascript saat web browser ditutup	

Selain atribut tersebut, masih banyak lagi atribut event yang lainnya. Semuanya bisa kamu lihat di: [HTML Event Attributes](#).

3. Atribut Khusus

Atribut khusus adalah atribut yang hanya bisa dipakai pada elemen tertentu dan kadang atribut ini tidak bisa digunakan pada elemen yang lain.

Contoh:

Nama atribut Bisa dipakai di tag

src	<audio>, <embed>, <iframe>, , dll
href	<a>, <link>
action	<form>
autoplay	<audio>, <video>

Cara Menulis Atribut yang Benar!

Penulisan atribut sebenarnya gampang.. kita hanya perlu menambahkannya pada tag pembuka dengan format seperti ini:

```
bana·atribut="nilai"
```

Namun, ada beberapa hal yang perlu diperhatikan agar penulisannya benar dan valid:

1. Gunakan Huruf Kecil

Menulis atribut dengan huruf besar sah-sah saja, dan bahkan valid menurut validator W3C.

Tapi saya sarankan menggunakan huruf kecil saja. Karena lebih umum digunakan dan juga mudah terbaca.

Contoh: (bagus)

```
<p align="center">Belajar HTML di Petani Kode</p>
```

Contoh: (kurang bagus)

```
<p ALIGN="CENTER">Belajar HTML di Petani Kode</p>
```

2. Gunakan Tanda Petik

Gunakan tanda petik untuk mengisi nilai atribut yang mengandung teks.

Mengapa?

Karena jika terdapat lebih dari satu kata, ia akan menciptakan spasi dan akan dianggap sebagai atribut baru.

Contoh: (bagus)

```
<h1 title="tutorial HTML untuk pemula">Belajar HTML</h1>
```

Contoh: (buruk)

```
<h1 title=tutorial HTML untuk pemula>Belajar HTML</h1>
```

Tanda petik yang digunakan, boleh petik ganda ("") dan juga petik tunggal ('').

Nah untuk nilai angka, boleh pakai tanda petik dan juga boleh tidak.

Contoh:

```

```

Lalu, untuk atribut yang bernilai boolean (**true** dan **false**).. nilainya boleh ditulis dan boleh tidak.

Contoh:

```
<input type="text" required="true" />  
<input type="text" required />
```

3. Penggunaan Spasi

Jika ada atribut yang memiliki lebih dari satu nama, maka ia ditulis dengan tanda min (-), bukan spasi.

Contoh:

```

```

Lalu, spasis juga digunakan untuk memisahkan dua atau lebih atribut.

Contoh:

```

```

Bisa juga ditulis seperti ini:

```

```

Membuat Paragraf pada HTML

Membuat Paragraf di HTML

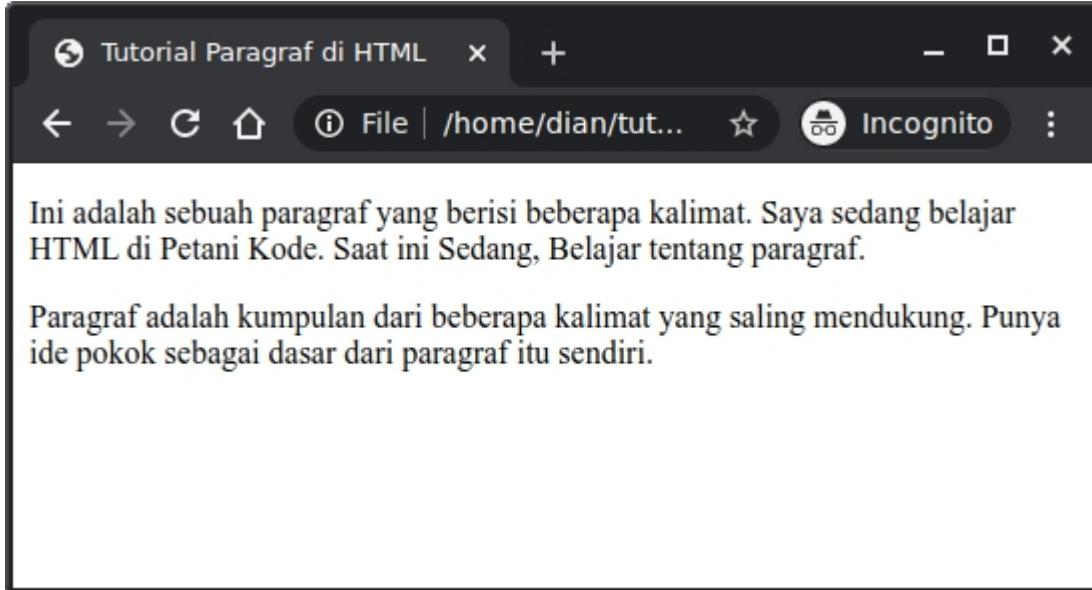
Paragraf adalah kumpulan dari beberapa kalimat. Pada web, Paragraf biasanya digunakan untuk menampilkan teks atau artikel.

Paragraf pada HTML dibuat dengan tag `<p>`. Selain tag ini, ada juga tag pendukung lain seperti `<div>`, `<hr>`, `
`, dan `<pre>`.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Paragraf di HTML</title>
  </head>
  <body>
    <p>Ini adalah sebuah paragraf yang berisi beberapa kalimat.
      Saya sedang belajar HTML di Petani Kode. Saat ini Sedang,
      Belajar tentang paragraf.</p>
    <p>Paragraf adalah kumpulan dari beberapa kalimat yang saling
      mendukung. Punya ide pokok sebagai dasar dari paragraf itu sendiri.</p>
  </body>
</html>
```

Hasilnya:



Atribut untuk Paragraf

Biasanya paragraf ditambahkan dengan beberapa atribut seperti **align**, **id**, **class**, dll.

Contoh:

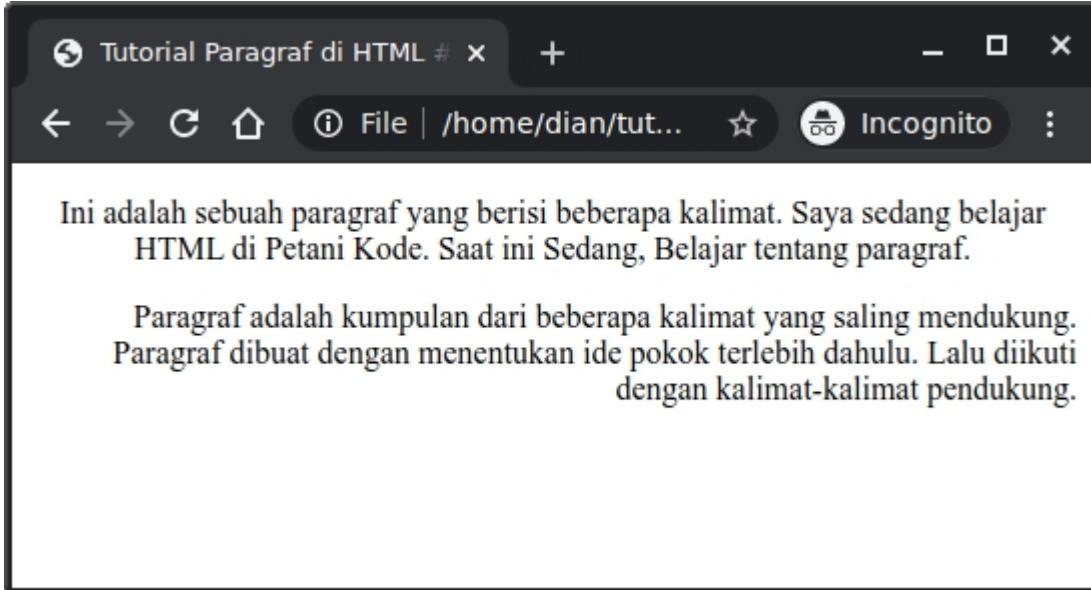
```
<!DOCTYPE html>

<html lang="en">

  <head>
    <title>Tutorial Paragraf di HTML #2</title>
  </head>

  <body>
    <p align="center">Ini adalah sebuah paragraf yang berisi beberapa
    kalimat. Saya sedang belajar HTML di Petani Kode. Saat ini Sedang,
    Belajar tentang paragraf.</p>
    <p align="right">Paragraf adalah kumpulan dari beberapa kalimat yang
    saling mendukung. Paragraf dibuat dengan menentukan ide pokok terlebih
    dahulu. Lalu diikuti dengan kalimat-kalimat pendukung.</p>
  </body>
</html>
```

Hasilnya:



Atribut **align** merupakan atribut yang digunakan untuk pertaan teks pada paragraf. Namun, menurut validator W3C.. penggunaan tag ini sebaiknya diganti dengan CSS.

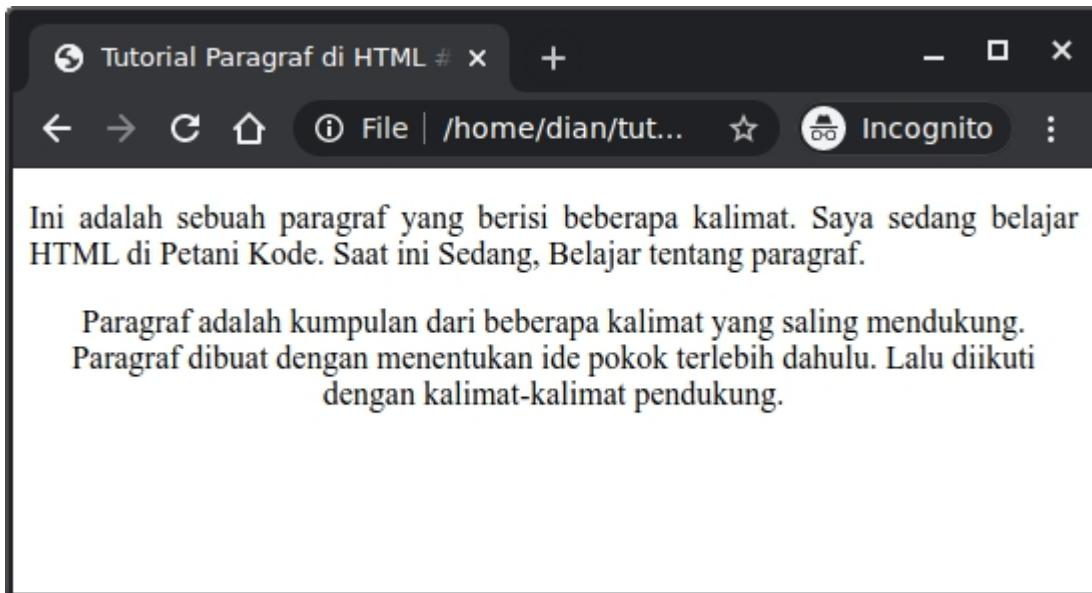
Mengapa demikian?

Karena atribut **align** dapat merubah tampilan dari web. Ini sebenarnya tigas dari CSS. Tugas utama dari HTML adalah membuat struktur dasar dari web.

Contoh perataan menggunakan CSS:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Paragraf di HTML #2</title>
  </head>
  <body>
    <p style="text-align: justify">Ini adalah sebuah paragraf yang berisi
       beberapa kalimat. Saya sedang belajar HTML di Petani Kode. Saat ini
       Sedang, Belajar tentang paragraf.</p>
    <p style="text-align: center">Paragraf adalah kumpulan dari beberapa
       kalimat yang saling mendukung. Paragraf dibuat dengan menentukan ide
       pokok terlebih dahulu. Lalu diikuti dengan kalimat-kalimat pendukung.</p>
  </body>
</html>
```

Hasilnya:



Tag **
** untuk Membuat Paragraf

Tag **
** sebenarnya bukanlah tag untuk membuat paragraf. Tapi tag ini, biasanya digunakan untuk membantu tag **<p>**.

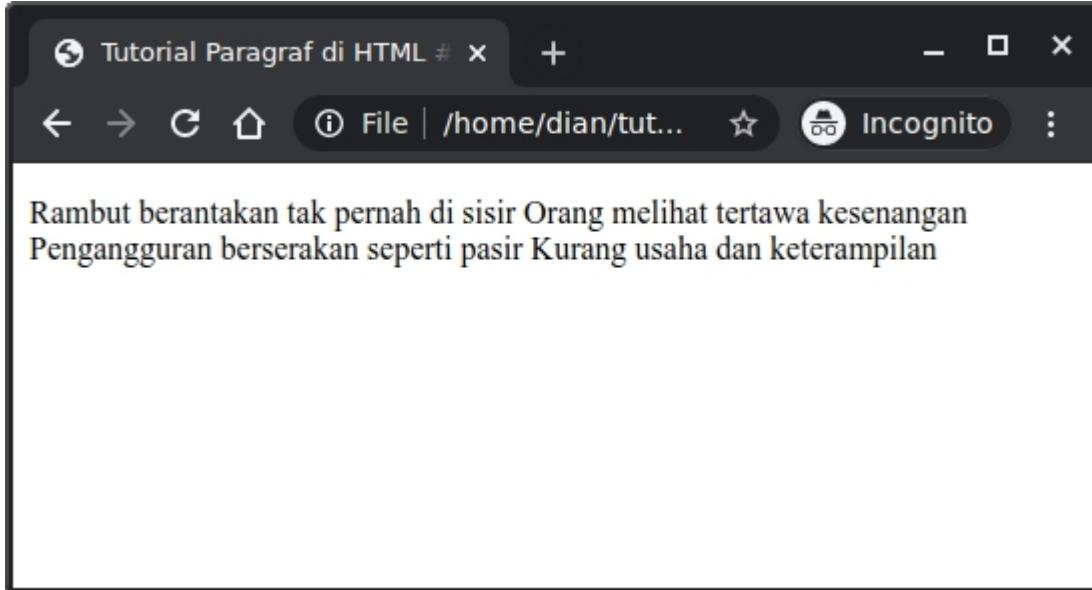
Fungsi utama tag **
** adalah untuk membuat baris baru.

Contoh:

Misalkan kita ingin menampilkan pantun, bisa saja kita buat seperti ini dengan tag **<p>**.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Paragraf di HTML #3</title>
  </head>
  <body>
    <p>Rambut berantakan tak pernah di sisir
      Orang melihat tertawa kesenangan
      Pengangguran berserakan seperti pasir
      Kurang usaha dan keterampilan</p>
  </body>
</html>
```

Hasilnya:

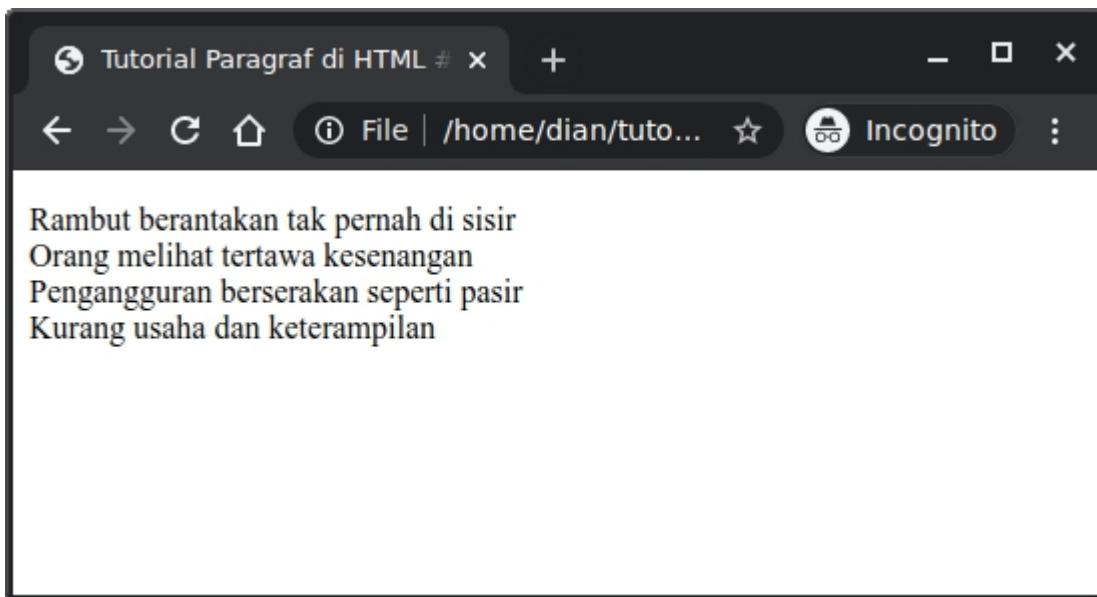


Meskipun pada kode HTML kita sudah menulis tiap bait pantun dalam baris yang baru. Tapi ia akan tetap ditampilkan seperti baris.

Di sinilah saatnya kita harus menggunakan tag **
**. Maka, kode di atas.. bisa kita perbaiki menjadi seperti ini:

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <title>Tutorial Paragraf di HTML #3</title>  
  </head>  
  <body>  
    <p>Rambut berantakan tak pernah di sisir <br />  
      Orang melihat tertawa kesenangan <br />  
      Pengangguran berserakan seperti pasir <br />  
      Kurang usaha dan keterampilan</p>  
  </body>  
</html>
```

Hasilnya:



Oh iya, tag **
** adalah tag yang tidak memiliki pasangan penutup. Cara menutupnya, tambahkan saja garis miring seperti ini **
**.

Tag **
** boleh ditutup, boleh juga tidak. Namun, sebaiknya ditutup agar valid menurut validator W3C.

Tag **<hr>** untuk Membuat Garis

Sama seperti tag **
, tag **<hr> juga bukanlah tag untuk membuat paragraf.

Tag **<hr>** merupakan tag yang digunakan untuk membuat garis lurus secara horizontal (horizontal rule). Biasanya digunakan untuk memisahkan beberapa konten atau paragraf.

Contoh:

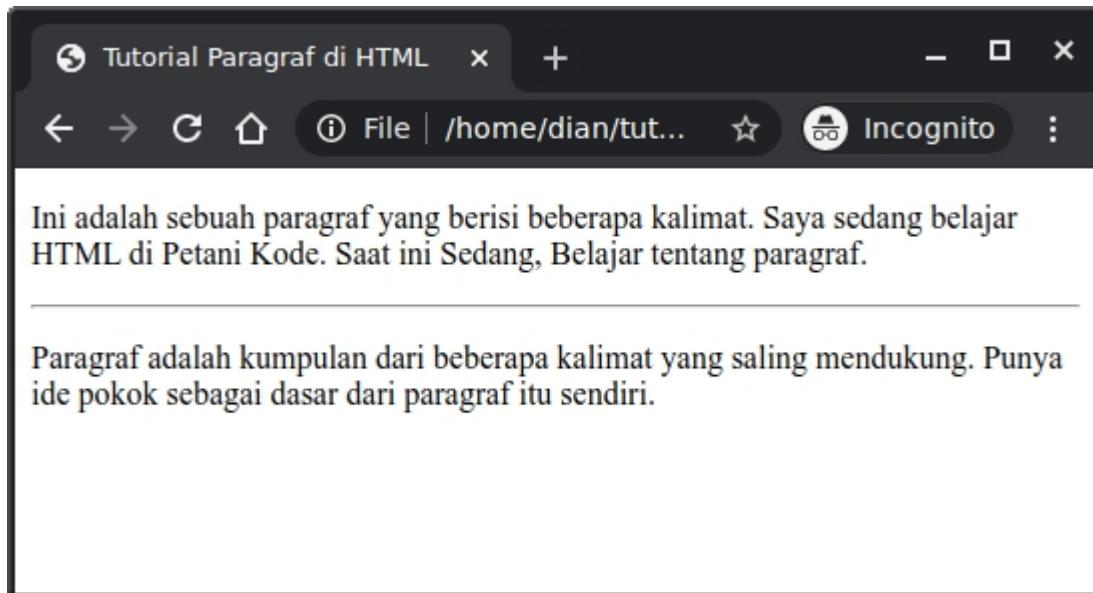
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Paragraf di HTML</title>
  </head>
  <body>
    <p>Ini adalah sebuah paragraf yang berisi beberapa kalimat.  

      Saya sedang belajar HTML di Petani Kode. Saat ini Sedang,  

      Belajar tentang paragraf.</p>
    <hr />
```

```
<p>Paragraf adalah kumpulan dari beberapa kalimat yang saling  
mendukung. Punya ide pokok sebagai dasar dari paragraf itu sendiri.</p>  
</body>  
</html>
```

Hasilnya:



Tag `<pre>`

Pada kasus tertentu, kita ingin menampilkan paragraf dengan format yang lebih spesifik. Contohnya seperti pantun dan puisi yang paragrafnnya diulis dengan garis baru dan juga indentasi.

Hal ini bisa dilakukan dengan bantuan tag `
`. Namun ada juga tag lain yang bisa jadi alternatif, yakni tag `<pre>`.

Tag `<pre>` (preformatting) merupakan tag yang digunakan untuk menampilkan teks atau paragraf dalam format yang sudah kita tentukan di HTML.

Contoh:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <title>Tutorial Paragraf di HTML</title>  
</head>  
<body>  
<h1>Hujan Bulan Juni</h1>
```

<**pp**

<**pre**>

tak ada yang lebih tabah
dari hujan bulan Juni
dirahasiakannya rintik rindunya
kepada pohon berbunga itu

tak ada yang lebih bijak
dari hujan bulan Juni
dihapusnya jejak-jejak kakinya
yang ragu-ragu di jalan itu

tak ada yang lebih arif
dari hujan bulan Juni
dibiarkannya yang tak terucapkan
diserap akar pohon bunga itu

</**pre**>

</**body**>

</**html**>

Hasilnya:

Hujan Bulan Juni ← Tag `<h1>`

oleh Sapardi Djoko Damono ← Tag `<p>`

tak ada yang lebih tabah
dari hujan bulan Juni
dirahasiakannya rintik rindunya ← Tag `<pre>`
kepada pohon berbunga itu

 tak ada yang lebih bijak
 dari hujan bulan Juni
 dihapusnya jejak-jejak kakinya
 yang ragu-ragu di jalan itu

 tak ada yang lebih arif
 dari hujan bulan Juni
 dibiarkannya yang tak terucapkan
 diserap akar pohon bunga itu

Coba perhatikan!

Paragraf yang kita buat di dalam kode HTML akan ditampilkan dengan format yang sama seperti yang kita tulis di sana.

Biasanya tag `<pre>` sering digunakan untuk menampilkan source code. Karena, tag ini menggunakan font Monospace atau Courier New (di Windows).

Tag `<p>` vs Tag `<div>`

Tag `<p>` dan Tag `<div>`, memiliki perilaku yang hampir sama. Tapi tag `<div>` sebenarnya bukanlah tag untuk membuat paragraf, melainkan tag untuk membuat layout web.

Kadang tag `<div>` sering ‘disalahgunakan’ untuk membuat paragraf.

Contoh:

```
<!DOCTYPE html>  
<html lang="en">  
    <head>  
        <title>Tutorial Paragraf di HTML</title>
```

```

</head>

<body>

<div>Ini adalah sebuah paragraf yang berisi beberapa kalimat.  

Saya sedang belajar HTML di Petani Kode. Saat ini Sedang,  

Belajar tentang paragraf.</div>

<div>Paragraf adalah kumpulan dari beberapa kalimat yang saling  

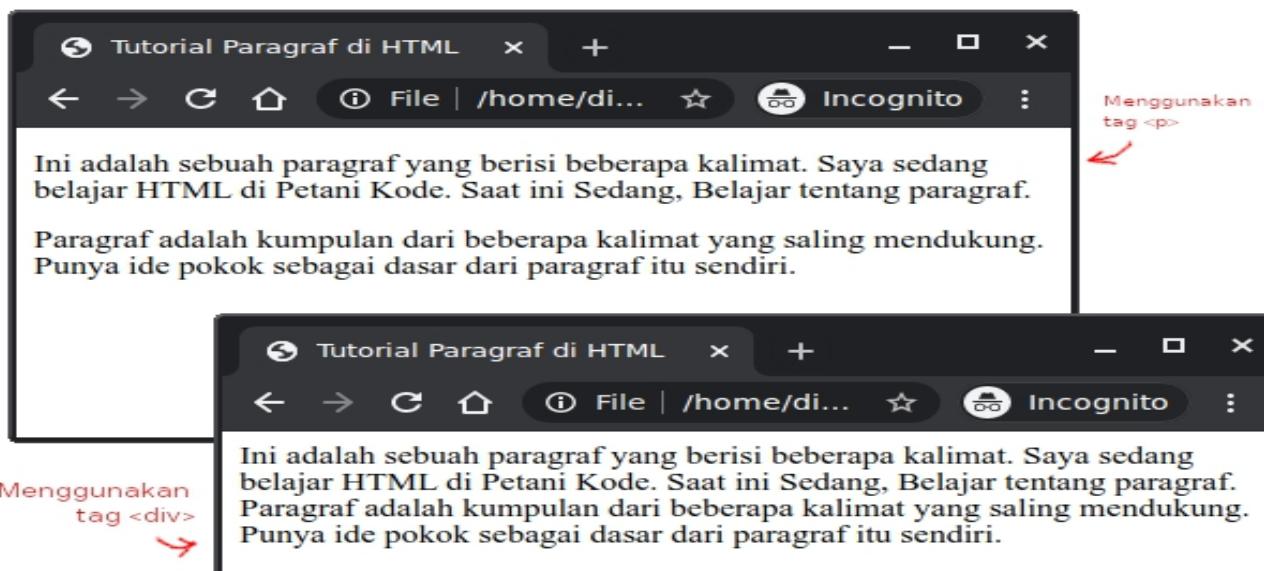
mendukung. Punya ide pokok sebagai dasar dari paragraf itu sendiri.</div>

</body>

</html>

```

Ini boleh-boleh, saja. Tapi hasilnya tidak akan sama seperti tag **<p>**.



Paragraf yang dibuat dengan tag **<div>** tidak akan memiliki jarak margin antar paragraf.

Tag **<div>** biasanya digunakan untuk membungkus teks yang ada di luar artikel. Contoh seperti teks pada footer, header, sidebar, dll.

Contoh:

```

<!DOCTYPE html>

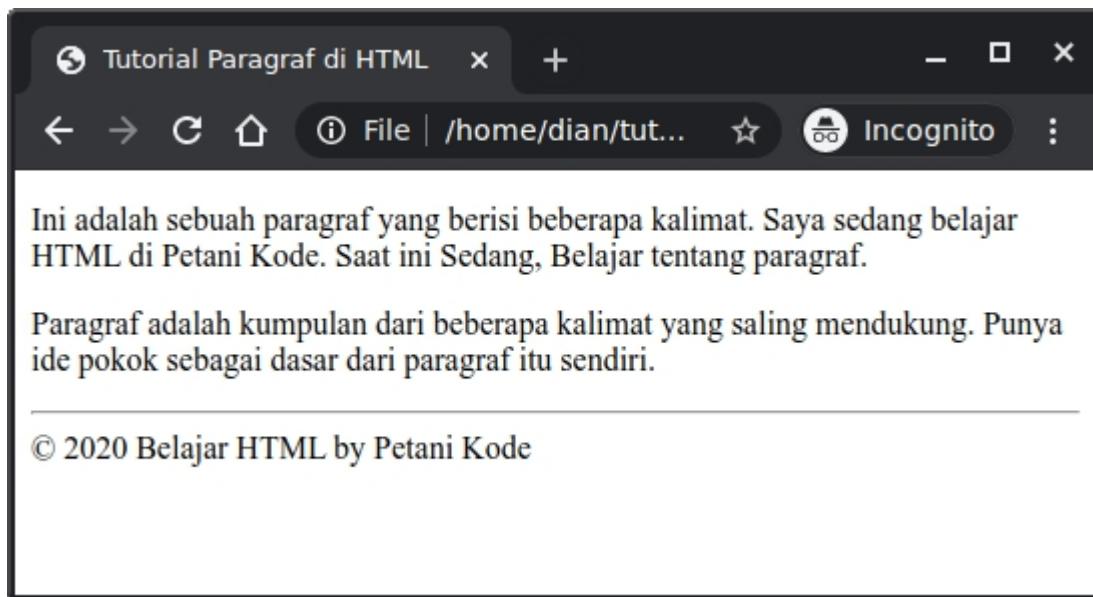
<html lang="en">

<head>
    <title>Tutorial Paragraf di HTML</title>
</head>
<body>
    <p>Ini adalah sebuah paragraf yang berisi beberapa kalimat.

```

```
Saya sedang belajar HTML di Petani Kode. Saat ini Sedang,  
Belajar tentang paragraf.</p>  
<p>Paragraf adalah kumpulan dari beberapa kalimat yang saling  
mendukung. Punya ide pokok sebagai dasar dari paragraf itu sendiri.</p>  
<hr />  
<footer>  
<div>&copy; 2020 Belajar HTML by Petani Kode</div>  
</footer>  
</body>  
</html>
```

Hasilnya:



Bonus: Paragraph Style

Sebenarnya ini ranah pembahasan tentang CSS bukan HTML. Karena itu, kamu boleh baca dan juga boleh tidak.

Paragraph Style adalah style CSS yang kita berikan kepada paragraf agar tampil lebih menarik.

Berikut ini beberapa style yang bisa diberikan pada paragraf:

1. Baris Pertama Masuk ke Dalam

Baris pertama pada paragraf kadang sering ditulis masuk ke dalam atau ini juga disebut dengan indentasi.

Contoh:

Baris pertama masuk ke dalam

Contoh ini baris pertama yang ditulis masuk ke dalam.
Baris kedua tetap ditulis seperti biasa. Dan juga baris Ke-3
ditulis seperti biasa.

Bagaimanakah cara membuat yang seperti itu di HTML?

Gampang!

Kita bisa memanfaatkan style CSS.

```
<!DOCTYPE html>

<html lang="en">

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

<body>
    <h1>Paragraf Indentasi</h1>
    <p style="text-indent: 45px;">Contoh ini baris pertama yang ditulis masuk ke dalam.
    Baris kedua tetap ditulis seperti biasa. Dan juga baris
    Ke-3 ditulis seperti biasa.</p>
</body>
</html>
```

Hasilnya:



Atribut **style** merupakan atribut untuk menambahkan style CSS. Pada contoh di atas kita menambahkan **text-indent** dengan nilai **45px**, nilai ini akan menentukan jarak (ke dalam) dari baris pertama.

2. Warna untuk Paragraf

Warna bisa kita berikan kepada teks dan background. Properti CSS yang digunakan untuk memberikan warna adalah **color** (untuk teks) dan **background-color** (untuk background).

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Tutorial Paragraf di HTML</title>
</head>
<body>
<h1>Warna untuk Paragraf</h1>

<p style="color: blue;">Contoh paragraf dengan warna teks biru.
Ini baris kedua dari paragraf. Semua teks yang ada di dalam
paragraf ini akan berwarna biru.</p>

<p style="color: white; background-color:purple">Contoh paragraf dengan
```

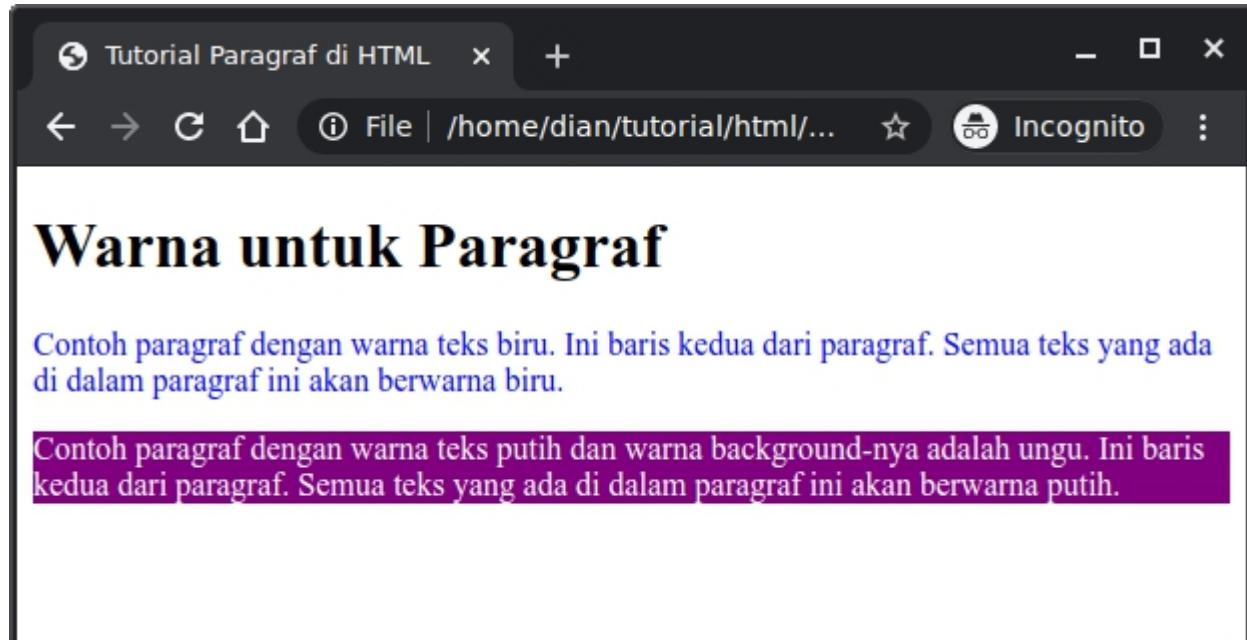
```
warna teks putih dan warna background-nya adalah ungu.
```

```
Ini baris kedua dari paragraf. Semua teks yang ada di dalam  
paragraf ini akan berwarna putih.</p>
```

```
</body>
```

```
</html>
```

Hasilnya:



3. Mengubah Jenis Font

Tidak suka dengan jenis font yang digunakan, kamu bisa menggantinya dengan style CSS.

Caranya..

Kita bisa gunakan properti **font-family** lalu diberikan nilai dengan nama font yang akan digunakan.

Contoh:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <title>Tutorial Paragraf di HTML</title>  
</head>  
<body>  
    <h1>Font untuk Paragraf</h1>
```

```
<p style="font-family: Arial;">Ini adalah contoh paragraf yang menggunakan font Arial. Semua teks pada paragraf ini akan menggunakan font Arial. Arial adalah salah satu font yang umum digunakan pada Windows.</p>
```

```
<p style="font-family: 'Times New Roman'">Ini adalah paragraf yang menggunakan font Times New Roman. Font ini juga umum digunakan dalam penulisan teks. Perhatikan penulisan Times New Roman pada style CSS, ia diapit dengan tanda petik karena mengandung lebih dari satu kata.</p>
```

```
</body>
```

```
</html>
```

Hasilnya:

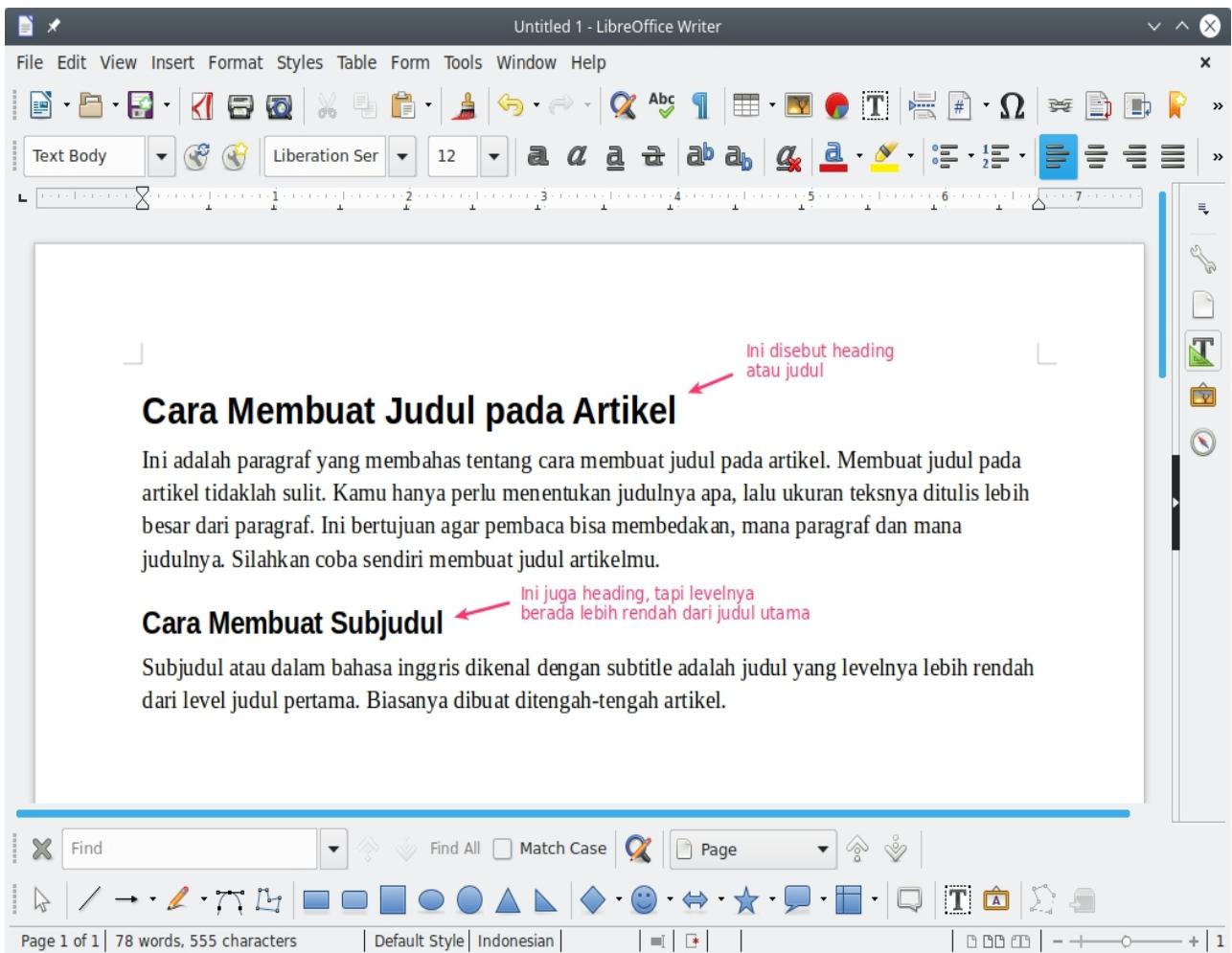


Cara Membuat Heading di HTML

Apa itu Heading?

Heading adalah sebuah judul yang biasanya diberikan pada halaman atau beberapa bagian dari artikel.

Jika kamu sering menulis artikel, pasti ini tidak asing buatmu.



Lalu, bagaimana caranya membuat judul di HTML?

Mari kita bahas:

Cara Membuat Heading di HTML

Judul pada HTML dapat kita buat dengan tag **<h1>** sampai **<h6>**.

Tag **<h1>** digunakan pada judul level pertama. Lalu tag lainnya digunakan pada sub heading atau level berikutnya.

Masing-masing judul akan ditampilkan dengan ukuran teks yang berbeda.

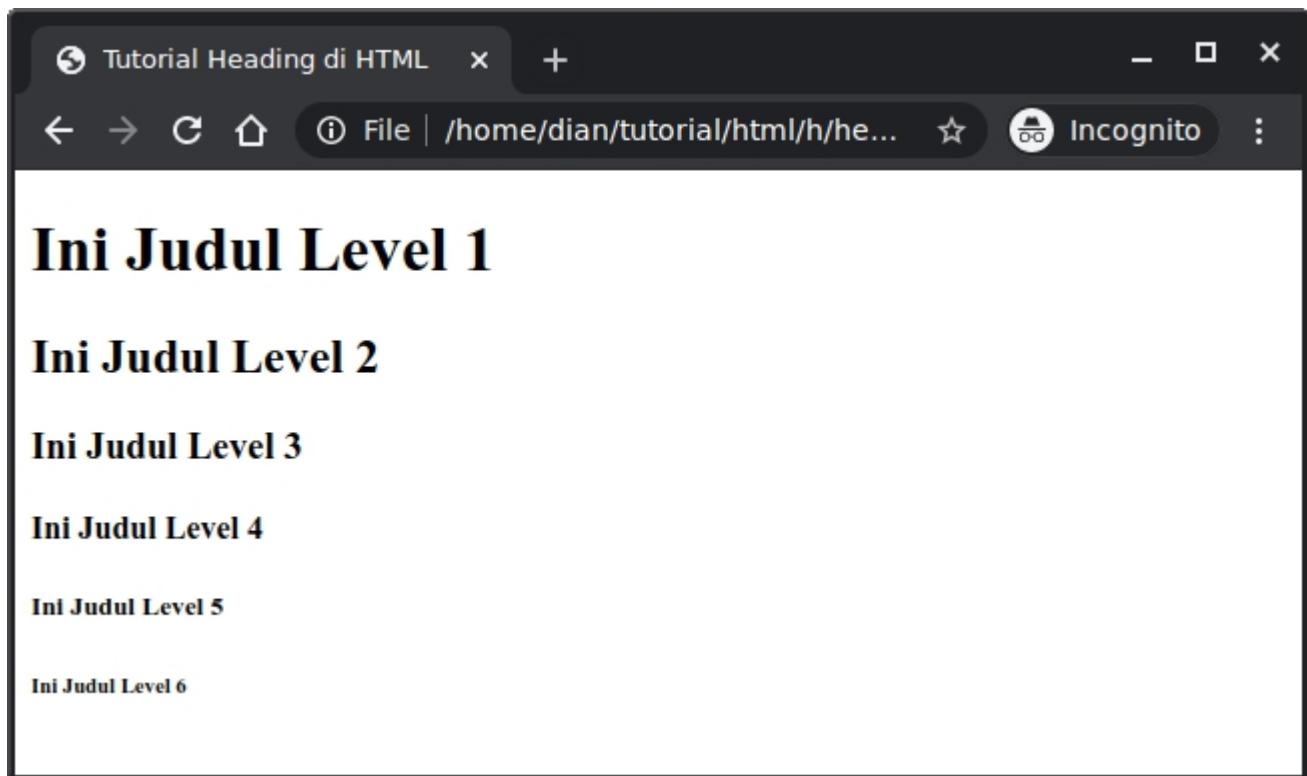
Tag **<h1>** adalah yang paling besar, dan tag **<h6>** paling kecil.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
```

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

Hasilnya:



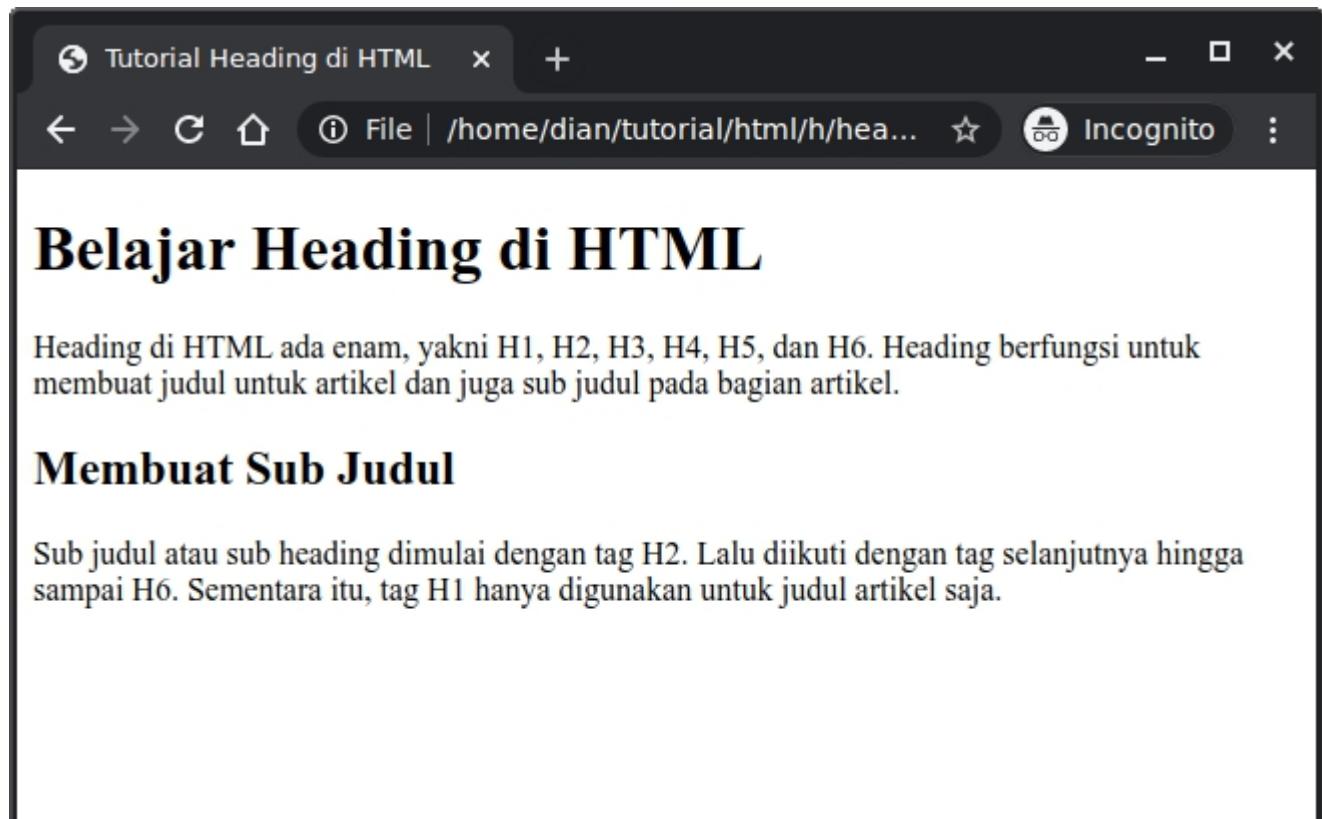
Tag **<h1>** biasanya dipakai pada judul artikel. Lalu tag **<h2>**, **<h3>**, **<h4>**, **<h5>**, dan **<h6>** dipakai pada sub judul atau sub heading.

Mari kita coba membuat sebuah artikel yang dilengkapi dengan heading. Buatlah file HTML baru kemudian isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Heading di HTML</title>
    </head>
```

```
<body>
  <h1>Belajar Heading di HTML</h1>
  <p>
    Heading di HTML ada enam, yakni H1, H2, H3, H4, H5, dan H6.
    Heading berfungsi untuk membuat judul untuk artikel dan juga
    sub judul pada bagian artikel.
  </p>
  <h2>Membuat Sub Judul</h2>
  <p>
    Sub judul atau sub heading dimulai dengan tag H2. Lalu diikuti
    dengan tag selanjutnya hingga sampai H6. Sementara itu, tag H1
    hanya digunakan untuk judul artikel saja.
  </p>
</body>
</html>
```

Hasilnya:



The screenshot shows a web browser window with the title "Tutorial Heading di HTML". The address bar indicates the file is located at "/home/dian/tutorial/html/h/he...". The main content area displays two heading elements: a large **Belajar Heading di HTML** and a smaller **Membuat Sub Judul**. Below the headings, there is explanatory text about the differences between H1 and H2 headings.

```
Belajar Heading di HTML
Heading di HTML ada enam, yakni H1, H2, H3, H4, H5, dan H6. Heading berfungsi untuk membuat judul untuk artikel dan juga sub judul pada bagian artikel.

Membuat Sub Judul

Sub judul atau sub heading dimulai dengan tag H2. Lalu diikuti dengan tag selanjutnya hingga sampai H6. Sementara itu, tag H1 hanya digunakan untuk judul artikel saja.
```

Berdasarkan pengalaman saya—dalam menulis artikel—tag **<h5>** dan **<h6>** jarang sekali dipakai, karena judul yang dibuat hanya sampai pada level 2, 3, dan 4.

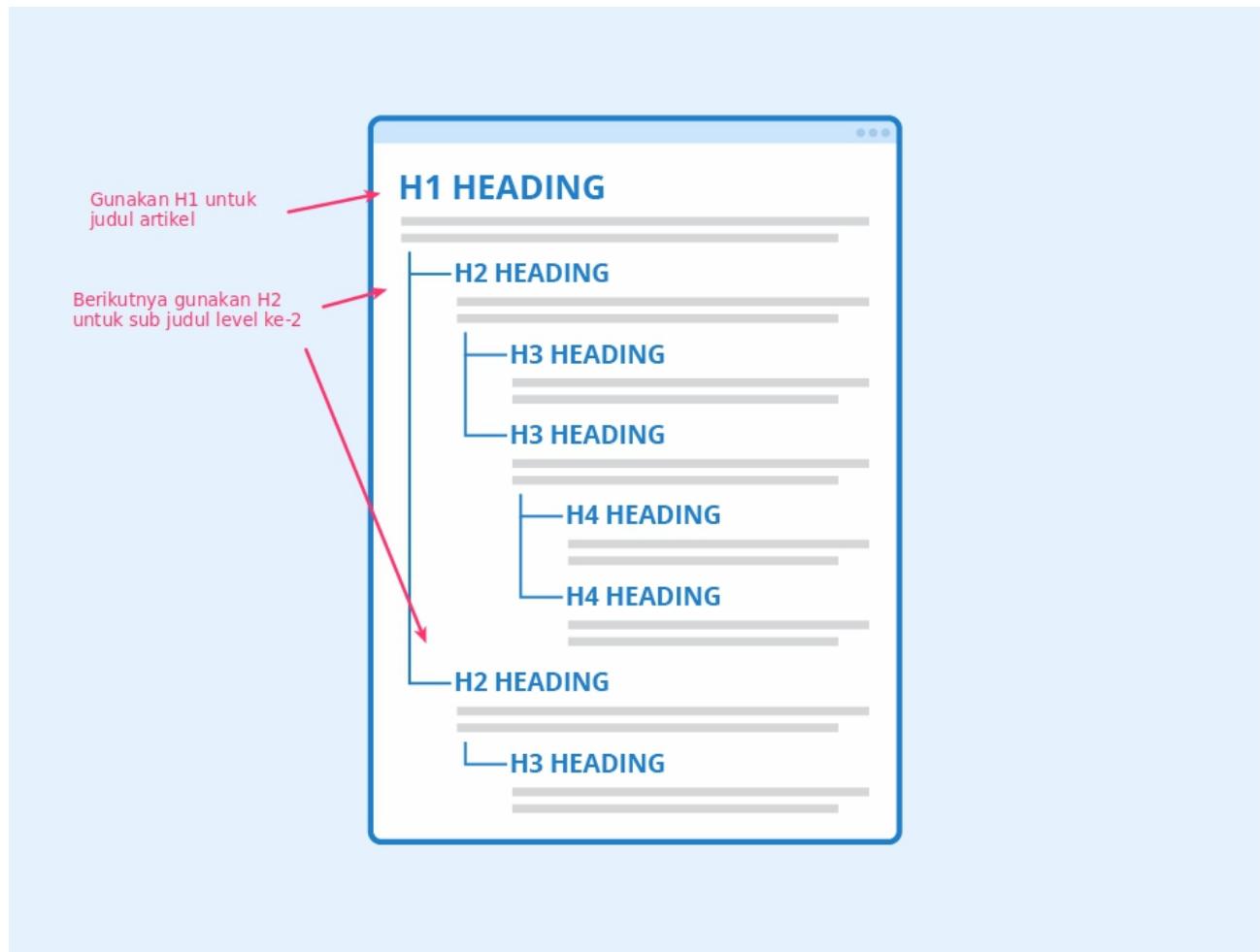
Urutan Penulisan Heading

Apakah boleh menulis judul yang tidak urut?

Maksudnya.. seperti menggunakan **<h6>** untuk judul awal, lalu berikutnya **<h4>** untuk sub judul.

Ini boleh-boleh saja, tapi kurang bagus untuk SEO. [1](#)

Penulisan judul yang bagus adalah judul yang mengikuti levelnya.



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

Atribut untuk Heading

Masih ingat tentang atribut?

Ya, dia adalah _____

Sudah lupa ya?

Coba buka lagi pembahasan tentang: [Apa itu Tag, Elemen, dan Atribut di HTML?](#)

Heading tidak memiliki atribut khusus. Ia biasanya menggunakan atribut global.

Contohnya seperti:

id untuk memberikan nama id unik. Biasanya ini akan digunakan oleh link,

CSS, dan Javascript;

class untuk memberikan nama class yang akan dipakai oleh CSS;

style untuk memberikan kode css secara langsung;

title untuk menambahkan informasi tambahan;

dll.

Daftar atribut global, bisa kamu lihat di [MDN: HTML Global Attributes](#).

Perbedaan Heading dengan Title dan Header

Meskipun sama-sama digunakan untuk urusan judul, tapi tiga elemen ini berbeda.

Heading adalah judul untuk **artikel** dan **bagian artikel** yang dibuat dengan

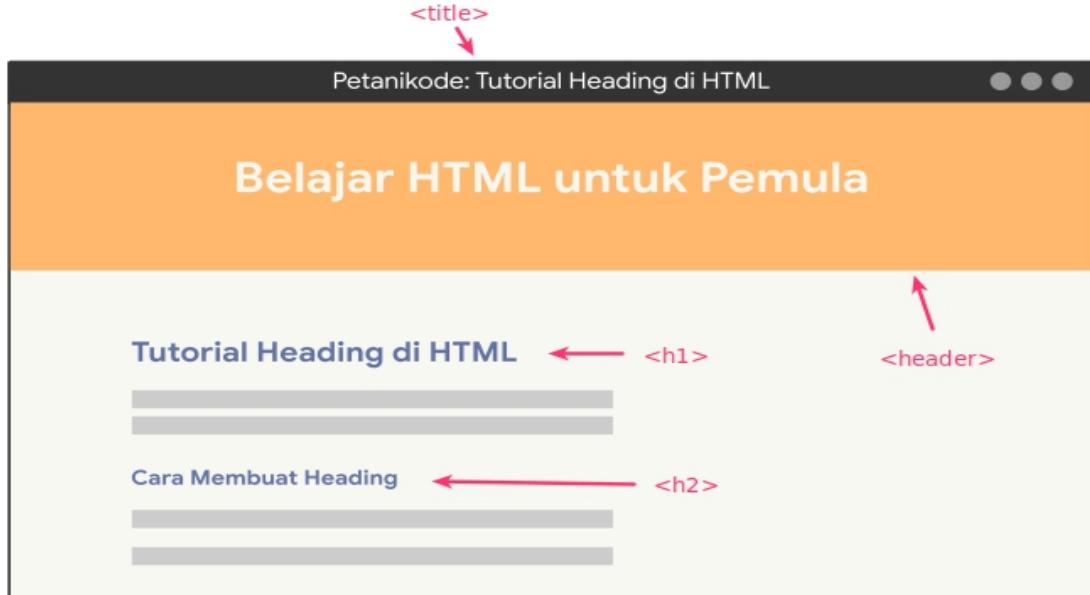
tag **<h1>** sampai **<h6>**

Title adalah **judul dari web** yang dibuat dengan tag **<title>**

Header adalah **bagian kepala** (kop) pada web yang dibuat dengan

tag **<header>**

Coba perhatikan gambar berikut:



Sudah jelas kan bedanya?

Tag `<title>` untuk judul yang ditampilkan di web browser. Tag `<header>` sama seperti kop surat. Lalu heading `<h1>` sampai `<h6>` menjadi judul untuk artikel.

Bonus: Heading Style

Sebenarnya ini termasuk dalam pembahasan materi tentang CSS. Karena itu, saya menjadikannya bonus yang boleh kamu baca dan juga boleh tidak.

Jadi..

Heading Style adalah style CSS yang diberikan pada heading agar terlihat menarik.

Berikut ini beberapa style yang sering digunakan pada heading:

Heading dengan Garis Bawah

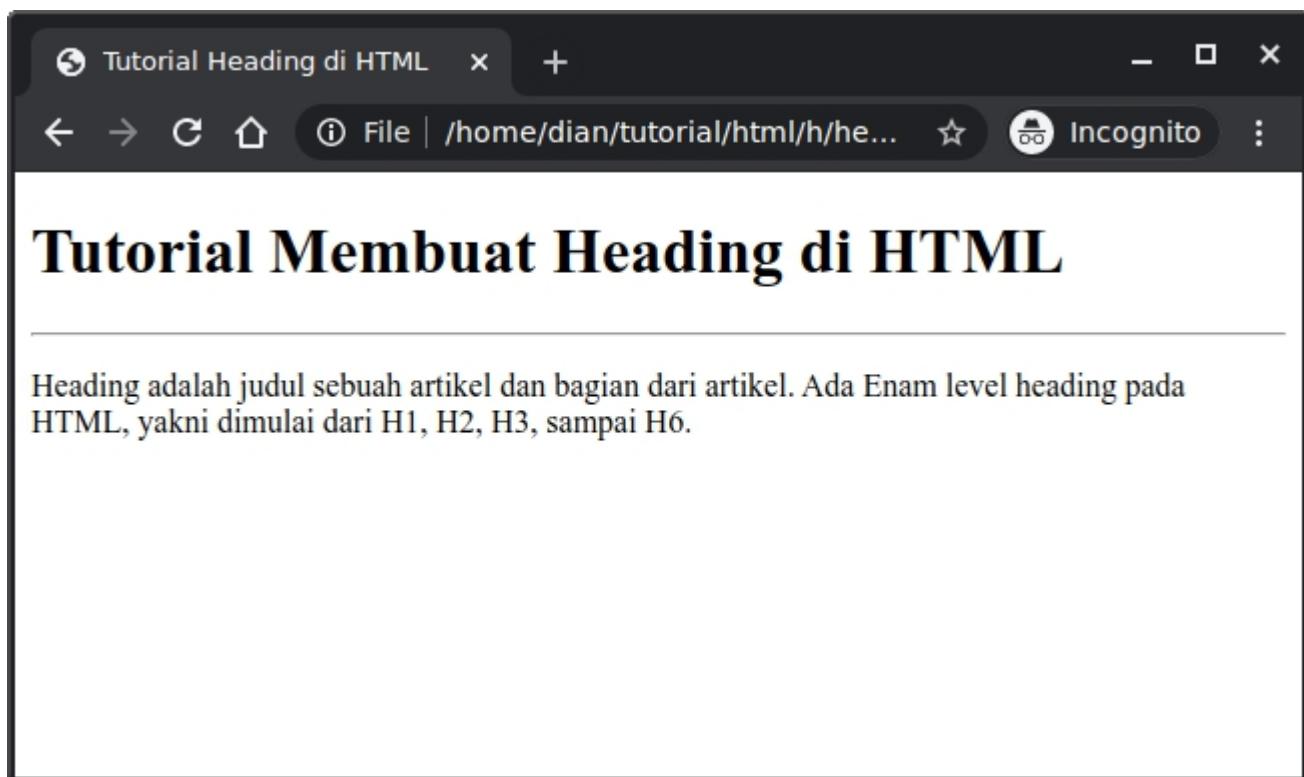
Heading dengan garis bawa bisa kita buat dengan memanfaatkan tag `<hr>` dan juga CSS.

Contoh menggunakan tag `<hr>`:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
```

```
<body>
    <h1>Tutorial Membuat Heading di HTML</h1>
    <hr />
    <p>
        Heading adalah judul sebuah artikel dan bagian dari artikel.
        Ada Enam level heading pada HTML, yakni dimulai dari H1,
        H2, H3, sampai H6.
    </p>
</body>
</html>
```

Hasilnya:

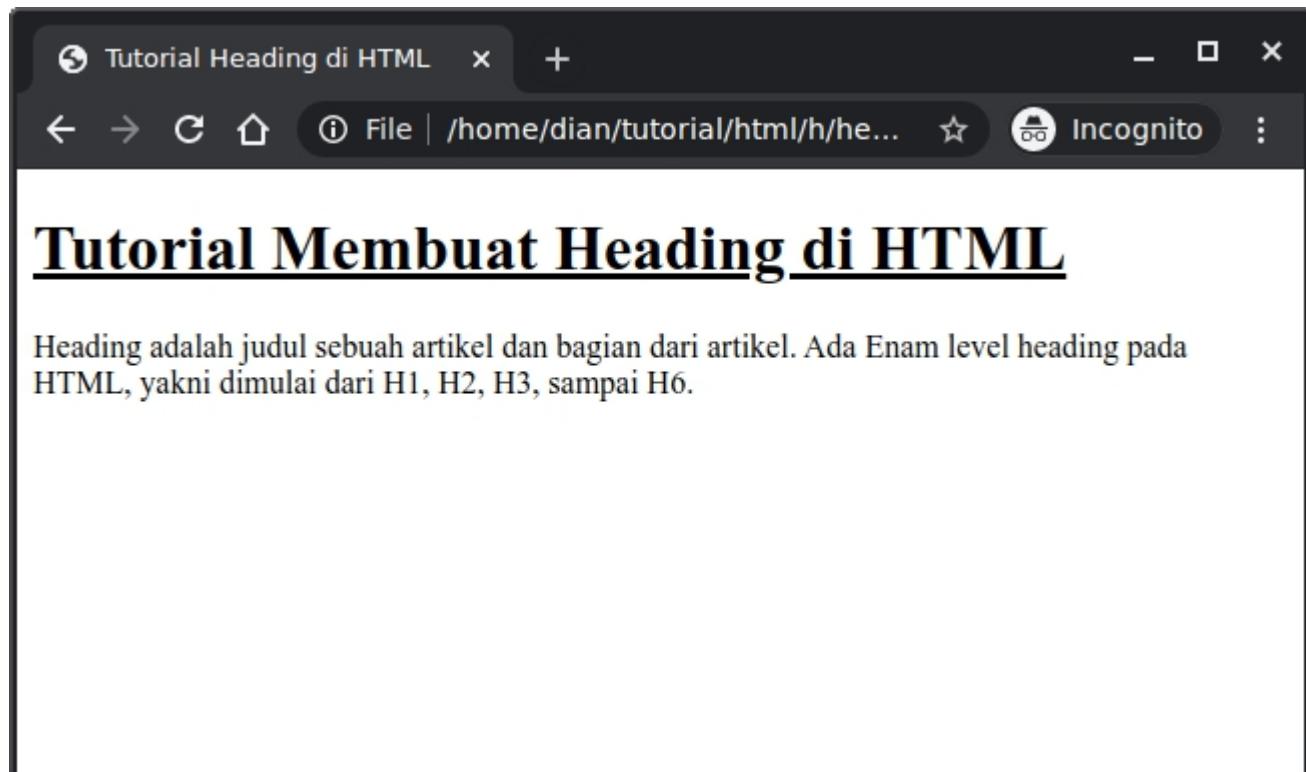


Contoh menggunakan CSS:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Heading di HTML</title>
    </head>
    <body>
        <h1 style="text-decoration: underline;">Tutorial Membuat Heading di HTML</h1>
```

```
<p>  
    Heading adalah judul sebuah artikel dan bagian dari artikel.  
    Ada Enam level heading pada HTML, yakni dimulai dari H1,  
    H2, H3, sampai H6.  
</p>  
</body>  
</html>
```

Hasilnya:



Text Case untuk Heading

Heading kadang ditulis dengan berbagai style case. Ada yang menggunakan huruf besar semua, ada juga yang menggunakan huruf besar di awal kata saja.

Contoh:

```
INI HEADING DENGAN HURUF BESAR SEMUA  
Ini Heading Dengan Huruf Besar Di Depan Saja
```

Nah, untuk membuat style case heading ini, Kita bisa menggunakan CSS dengan atribut **text-transform**.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
  <body>
    <h1 style="text-transform: uppercase;">Heading dengan huruf Besar semua</h1>
    <p>
      Heading adalah judul sebuah artikel dan bagian dari artikel.
      Ada Enam level heading pada HTML, yakni dimulai dari H1, H2, H3, sampai H6.
    </p>
    <h1 style="text-transform: capitalize;">Heading dengan huruf Besar di awal kata</h1>
    <p>
      Properti text-transform berfungsi untuk menentukan style case dari teks. Properti ini bisa diberikan nilai capitalize, uppercase, lowercase, initial, none, inherit.
    </p>
  </body>
</html>
```

Hasilnya:

The screenshot shows a browser window with the title "Tutorial Heading di HTML". The URL in the address bar is "/home/dian/tutorial/html/h/hea...". The main content area displays the text "HEADING DENGAN HURUF BESAR SEMUA" in large, bold, black capital letters. Below this, a smaller paragraph states: "Heading adalah judul sebuah artikel dan bagian dari artikel. Ada Enam level heading pada HTML, yakni dimulai dari H1, H2, H3, sampai H6." A section titled "Heading Dengan Huruf Besar Di Awal Kata" is shown in bold black text.

Warna untuk Heading

Sama seperti elemen yang lainnya, heading juga dapat kita berikan warna dengan bantuan CSS. Warna bisa kita berikan pada teks dan latar belakang atau background.

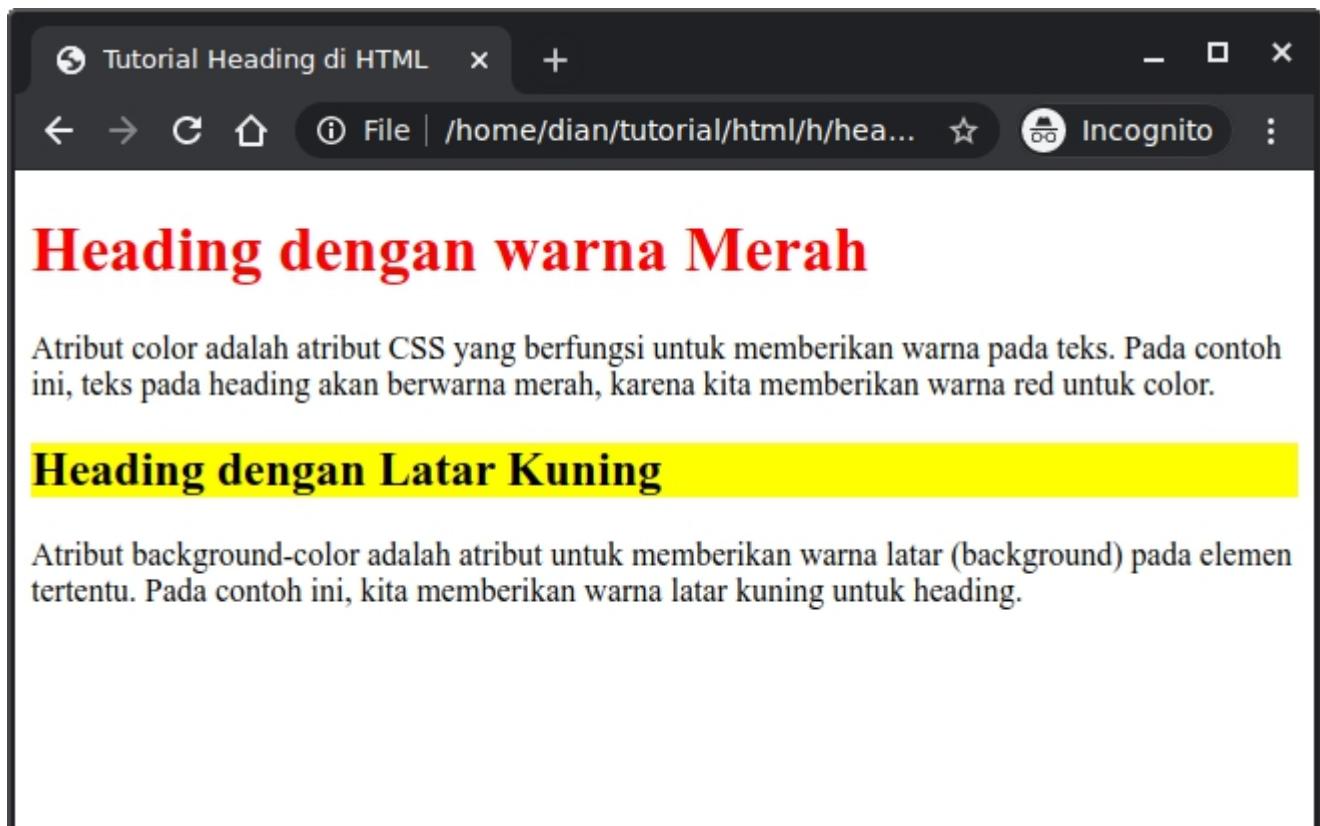
Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
  <body>
    <h1 style="color: red;">Heading dengan warna Merah</h1>
    <p>
      Atribut color adalah atribut CSS yang berfungsi untuk memberikan
      warna pada teks. Pada contoh ini, teks pada heading akan berwarna
      merah, karena kita memberikan warna red untuk color.
    </p>
    <h2 style="background-color: yellow;">Heading dengan Latar Kuning</h2>
    <p>
```

```
Atribut background-color adalah atribut untuk memberikan warna latar (background) pada elemen tertentu. Pada contoh ini, kita memberikan warna latar kuning untuk heading.
```

```
</p>  
</body>  
</html>
```

Hasilnya:



Membuat Komentar di HTML

Apa itu Komentar dalam HTML?

Komentar adalah elemen yang akan diabaikan oleh browser. Ia tidak akan ditampilkan di dalam web.

Komentar biasanya digunakan untuk memberikan informasi tambahan pada kode HTML dan kadang juga digunakan untuk menon-aktifkan beberapa kode HTML.

Cara Membuat Komentar di HTML

Komentar dapat kita buat dengan tag `<!-- -->`.

tag pembuka untuk membuat komentar

<!-- isi komentar -->

Tag penutup

petanikode.com

Contoh:

```
<!DOCTYPE html>

<html lang="en">

  <head>
    <title>Belajar Membuat Komentar di HTML</title>
  </head>

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

Hasilnya:



Komentar tidak akan ditampilkan oleh browser, tapi kita bisa melihatnya dengan cara view source.

Klik kanan pada browser, lalu pilih **view page source**.



Jalan Pintas untuk Membuat Komentar

Jika kamu menggunakan teks editor [Visual Studio Code](#), kamu bisa membuat komentar dengan menekan **Ctrl+%**.

A screenshot of the Visual Studio Code interface, identical to the one above, showing the same HTML code with its comments. The code is as follows:

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          |  <title>Belajar Membuat Komentar di HTML</title>
5      </head>
6      <body>
7          |  <!-- ini adalah komentar -->
8          |  <p>Hello World!</p>
9          |  <!-- ini juga komentar
10         |  dan ditulis dalam dua baris -->
11     </body>
12 </html>
```

The code is color-coded, and the file path "Home > Gian > tutorial > html > Komentar.html" is visible in the top left.

Cara ini lebih cepat dibandingkan dengan harus menulis secara manual tag komentarnya.

Atribut untuk Komentar

Komentar tidak bisa ditambahkan atribut. Jika kamu mencoba menambahkannya, itu akan sia-sia. Soalnya komenter merupakan elemen yang tidak akan diproses oleh browser.

Fungsi Komentar

Komentar memang tidak akan ditampilkan di web, namun bukan berarti ia tidak memiliki fungsi.

Berikut ini beberapa fungsi komenter:

1. Komentar untuk Menjelaskan Arti Tag

Saat belajar HTML, kamu mungkin akan kesulitan mengingat fungsi dan arti dari tag HTML. Karena itu, kamu harus membuat catatan untuk mengingatnya.

Ini bisa kamu lakukan dengan mencatat di buku, maupun media lain. Tapi, alangkah baiknya memanfaatkan komentar untuk mencatat.

Contoh:

```
<!DOCTYPE html> <!-- ini tag untuk menentukan type dokumen -->

<html lang="en">

    <head>

        <title>Belajar Membuat Komentar di HTML</title>

        <!-- tag title berfungsi untuk membuat judul web dan akan ditampilkan
             pada title bar di browser -->

    </head>

    <body>

        <p>Ini tag paragraf yang aktif</p>

        <!--
            tag <p> adalah tag untuk membuat paragraf
            ...
        -->

    </body>

</html>
```

Jika kita lupa, kita bisa membuka kembali dan mempelajari kode tersebut.

Tapi ingat, jangan buat komentar seperti ini pada proyek websitemu. Karena akan sangat mengganggu dan membuat kode sulit terbaca.

Sebaiknya, komentar yang berisi penejelasan semacam ini dibuat pada kode HTML yang kita pakai untuk belajar. Bukan kode HTML untuk web asli.

2. Komentar untuk Menyimpan Todo List

Komentar kadang sering digunakan juga untuk menyimpan todo list, ini akan membantu kita mengingat apa yang harus dikerjakan pada bagian HTML tertentu.

Contoh:

```
<!DOCTYPE html>

<html lang="en">

    <head>

        <title>Belajar Membuat Komentar di HTML</title>
```

```
<!-- TODO: Tambahkan tag meta di sini -->

</head>

<body>

<h1>Komentar di HTML</h1>

<!-- TODO: Buat konten web di ini -->

</body>

</html>
```

Jika kita sudah mengerjakan apa yang dituliskan di todo list tersebut, kita bisa hapus komentarnya.

3. Komentar untuk Menonaktifkan Kode HTML

Komentar kadang seding digunakan untuk menon-aktifkan kode HTML.

Contoh:

```
<!DOCTYPE html>

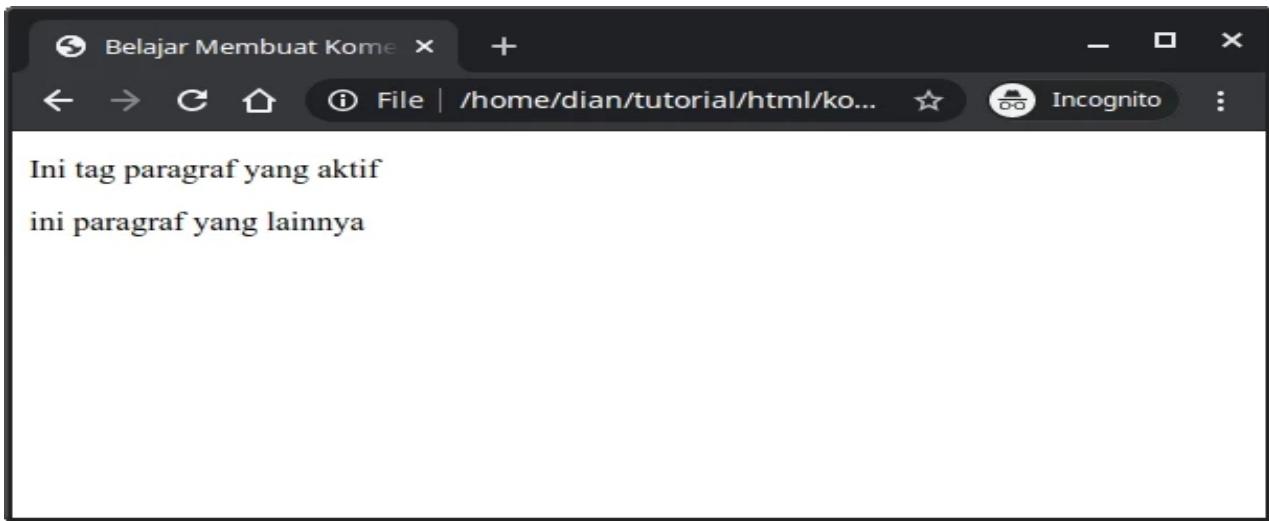
<html lang="en">

  <head>
    <title>Belajar Membuat Komentar di HTML</title>
  </head>

  <body>
    <p>Ini tag paragraf yang aktif</p>
    <!--
      <p>ini tag paragraf yang tidak aktif
      karena berada di dalam komentar
    </p>
    -->
    <p>ini paragraf yang lainnya</p>
  </body>

</html>
```

Hasilnya:



Text Formatting di HTML (Bold, Italic, Underline, dll)

Coba perhatikan kedua paragraf ini:



Kamu lebih tertarik baca yang mana?

..yang terformat, atau yang polos (plain text)?

Teks yang terformat akan lebih mudah dibaca, karena ada penegasan seperti **teks tebal**, miring, dan garis bawah.

Saat kita mencari kata atau kalimat penting.. mata kita akan lebih mudah menemukannya pada teks yang terformat dibandingkan plain text.

Karena itu, dalam membuat web.. ada baiknya menggunakan format teks.

Nah, di HTML terdapat tag-tag yang khusus digunakan untuk text formatting.

Apa saja itu?

Mari kita bahas..

Membuat Teks Tebal

Teks tebal biasanya digunakan untuk memberikan penegasan pada teks tertentu, misalnya seperti judul, subjudul, huruf penting, dll.

Tag yang digunakan untuk membuat teks tebal di HTML adalah tag **** (bold) dan tag ****. Kamu bebas mau pakai yang mana saja, hasilnya akan sama-sama tebal.

Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Tebal di HTML</title>
</head>
<body>
    <h1>Text Tebal di HTML</h1>
    <p>
        <strong>Teks formatting itu penting!</strong> Karena dapat membuat tulisan
        terlihat lebih menarik sehingga akan membuat <b>pengunjung senang</b>
        membacanya.
    </p>
</body>
</html>

```

Hasilnya:



Membuat Teks Miring

Teks miring biasanya digunakan untuk menegaskan sebuah kata atau istilah baru. Teks miring di HTML dapat kita buat dengan tag **<i>** (italic) dan juga tag **** (emphasis).

Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Miring di HTML</title>
</head>
<body>

```

```

<h1>Text Miring di HTML</h1>
<p>
    Gungakan <i>teks miring</i> untuk memberikan penekanan pada teks,
    sehingga akan <em>menarik perhatian</em> pembaca. Biasanya
    digunakan pada <i>istilah asing</i> atau kata serapan dari
    <em>bahasa daerah</em>.
</p>
</body>
</html>

```

Hasilnya:



Membuat Garis Bawah pada Teks

Garis bawah biasanya digunakan untuk menandai teks yang disisipkan atau teks yang memiliki arti penting dibandingkan teks normal lainnya.

Garis bawah di HTML dapat kita buat dengan tag **<u>** (underline) atau juga tag **<ins>** (insert).

Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Garis Bawah di HTML</title>
</head>
<body>
    <h1>Garis Bawah di HTML</h1>
    <p>
        <u>Text formatting itu penting</u>, karena dapat membuat teks terlihat
        lebih menarik dibandingkan <del>text biasa</del> <ins>plain text</ins>.
    </p>
</body>

```

```
</html>
```

Hasilnya:



Pada contoh di atas, kita menggunakan tag **** untuk membuat teks tercoret. Lalu diikuti dengan teks yang ditambahkan (insert).

Membuat Teks Tercoret

Teks tercoret memiliki arti teks yang dihapus. Biasanya untuk memberitahu pembaca bahwa teks tersebut tidak dipakai atau dihapus.

Tag untuk membuat teks tercoret di HTML adalah tag **<s>** (strikethrough) atau bisa juga dengan tag **** (delete).

Contoh:

```
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Tercoret di HTML</title>
</head>

<body>
    <h1>Text Tercoret di HTML</h1>
    <p>
        Coretlah teks yang tidak <s>dibutuhkan</s> terpakai, ini bisa memberitahu
        pembaca tentang perbaikan dari teks tersebut. Kadang juga teks <del>tercoret</del>
        <ins>yang dicoret</ins>, diperbaiki dengan menambahkan teks dengan garis bawah.
    </p>
</body>
</html>
```

Hasilnya:



Membuat Pangkat di HTML

Pangkat biasanya digunakan pada rumus. Ada dua jenis pangkat yang bisa dibuat di HTML, yakni pangkat yang berada di atas (superscript) dan pangkat di bawah (subscript).

Tag untuk membuat pangkat di HTML adalah tag **<sup>** dan **<sub>**.

<sup> untuk membuat pangkat di atas

<sub> untuk membuat pangkat di bawah

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Membuat Pangkat di HTML</title>
</head>
<body>
  <h1>Membuat Pangkat di HTML</h1>
  <p>
    Rumus luas persegi adalah  $S^{sup}2</sup>, dimana <i>S</i> adalah sisi dari
    persegi. Lalu  $O{sub}2</sub> adalah rumus kimia dari oksigen.
  </p>
</body>
</html>$$ 
```

Hasilnya:



Membuat Marker untuk Teks

Marker biasanya digunakan untuk menandai teks yang penting atau kata kunci yang penting. Marker di HTML dapat kita buat dengan tag **<mark>**.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Membuat marker teks di HTML</title>
</head>
<body>
    <h1>Membuat marker teks di HTML</h1>
    <p>
        Marker biasanya digunakan untuk menandai bagian teks yang penting.
        Kalau di dunia nyata, kita <mark>menggunakan stabilo</mark> untuk
        membuat marker.
    </p>
</body>
</html>
```

Hasilnya:



Warna default marker adalah kuning. Warna ini bisa kita ubah dengan style CSS.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Membuat marker teks di HTML</title>
</head>
<body>
    <h1>Membuat marker teks di HTML</h1>
    <p>
```

Marker biasanya digunakan untuk menandai bagian teks yang penting.

Kalau di dunia nyata, kita menggunakan stabilo untuk membuat marker.

Marker biasanya digunakan untuk menandai bagian teks yang penting.

```
</p>
```

```
</body>
```

```
</html>
```

Maka hasilnya, marker akan berwarna pink:



Teks Formatting untuk Komputer

Selain dari teks formatting di atas, ada juga teks formatting yang khusus untuk menandai teks yang berasal dari komputer. Berikut ini beberapa tag yang digunakan untuk memformat teks dari komputer:

<code> untuk menandai bagian dari kode program;

<samp> untuk menandai output dari program komputer;

<kbd> untuk menandai tombol keyboard;

<var> untuk menandai sebuah variabel;

<pre> untuk preformatting teks.

Contoh:

```
<!DOCTYPE html>

<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Text Formatting untuk Teks dari komputer</title>
</head>
<body>
```

<h1>Text Formatting untuk Teks dari komputer</h1>

<p>

Perintah javascript untuk menampilkan teks ke console adalah

<code>console.log()</code>. Kita juga bisa menampilkan isi variabel dengan fungsi ini. Misalkan kita punya variabel <var>name</var>, maka kode programnya bisa ditulis seperti ini:

<pre>

```
var name = "Petani Kode";  
console.log(name);
```

</pre>

Maka hasil outputnya: <samp>Petani Kode</samp>

</p>

<p>

Untuk menjalankan ulang program, lakukan refresh dengan menekan tombol

<kbd>F5</kbd>

</p>

</body>

</html>

Hasilnya:



Menggabungkan Format

Apakah format teks dapat digabungkan?

Misalkan kita ingin membuat teks tebal dan garis bawah, apakah bisa?

Tentu saja bisa.

Caranya:

Ya tinggal dipakai saja tag-tagnya, misal mau menggabungkan bold dengan underline.. maka kita tinggal pakai tag dan <u>.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Penggabungan Format Teks di HTML</title>
</head>
<body>
    <h1>Penggabungan Format Teks di HTML</h1>
    <p>
        Penggabungan format teks bisa dilakukan dengan menuliskan tag-tag yang akan dipakai. Misalkan <b><u>tebal dan garis bawah</u></b>, maka kita tinggal pakai tag <b>b</b> dan tag <b>u</b>.
    </p>
</body>
</html>
```

Hasilnya:



Dalam menggabungkan format, kamu perlu memperhatikan tag mana yang ditulis duluan dan yang terakhir.

Jangan sampai salah menutup..

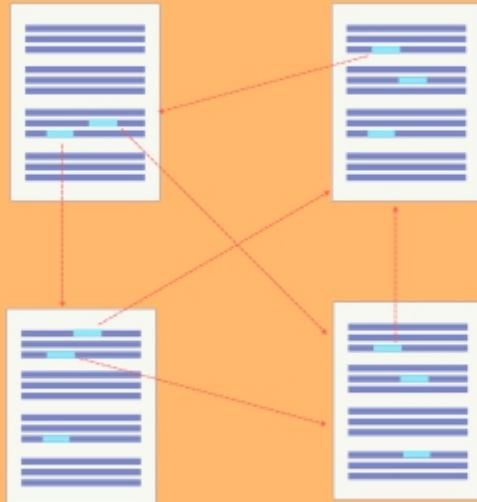
Yang dibuka duluan, harus ditutup terakhir.

Ingatlah konsep “**ibu memasak ubi**”:

Apa itu Link di HTML?

Link atau Hyperlink adalah elemen HTML yang berfungsi menghubungkan suatu halaman web ke halaman web yang lain.

Elemen ini bisa diklik.. dan nanti akan membuka halaman lain sesuai alamat URL yang diberikan.



Link menghubungkan halaman yang satu dengan halaman yang lainnya

petanikode.com

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

Nanti kita akan bahas ini lebih dalam..

Nah! sekarang, bagaimana cara membuat link di HTML?

Silahkan lanjutkan membaca:

Cara Membuat Link di HTML

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

Bentuk sederhananya seperti ini:

The diagram shows the HTML code for a link:

```
<a href="http://petanikode.com/">Petani Kode</a>
```

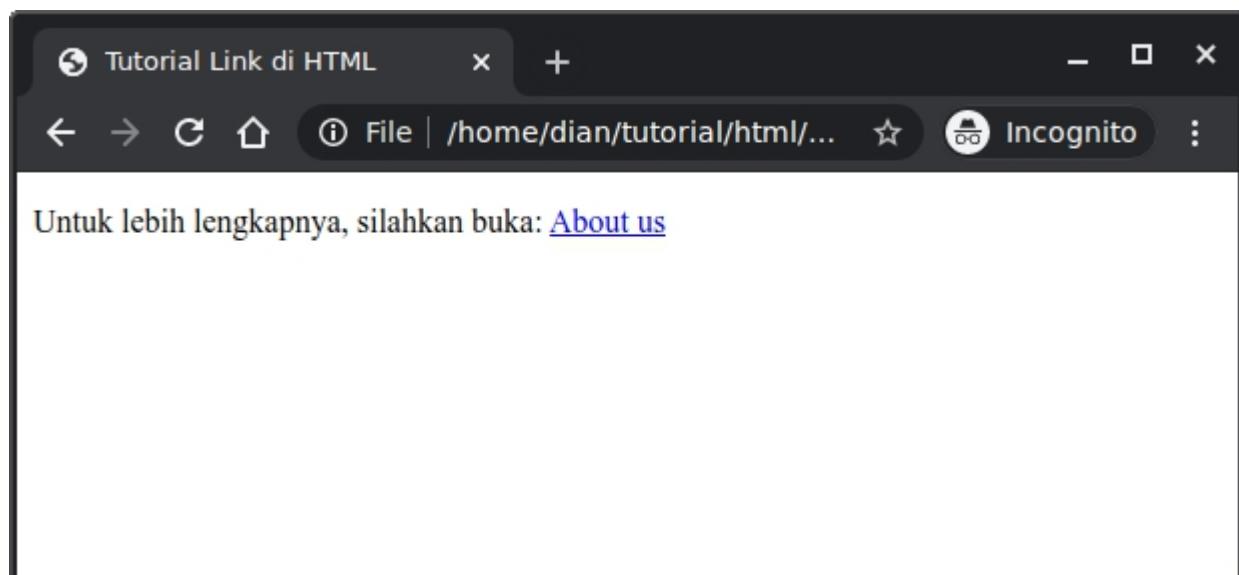
Annotations explain the parts:

- tag untuk membuat link
- Alamat URL tujuan yang akan dibuka saat link diklik
- Tag penutup
- Atribut untuk menentukan alamat URL tujuan dari link
- Teks label untuk link

Contoh: `link.html`

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <p>Untuk lebih lengkapnya, silahkan buka: <a href="about.html">About us</a></p>
    </body>
</html>
```

Hasilnya:

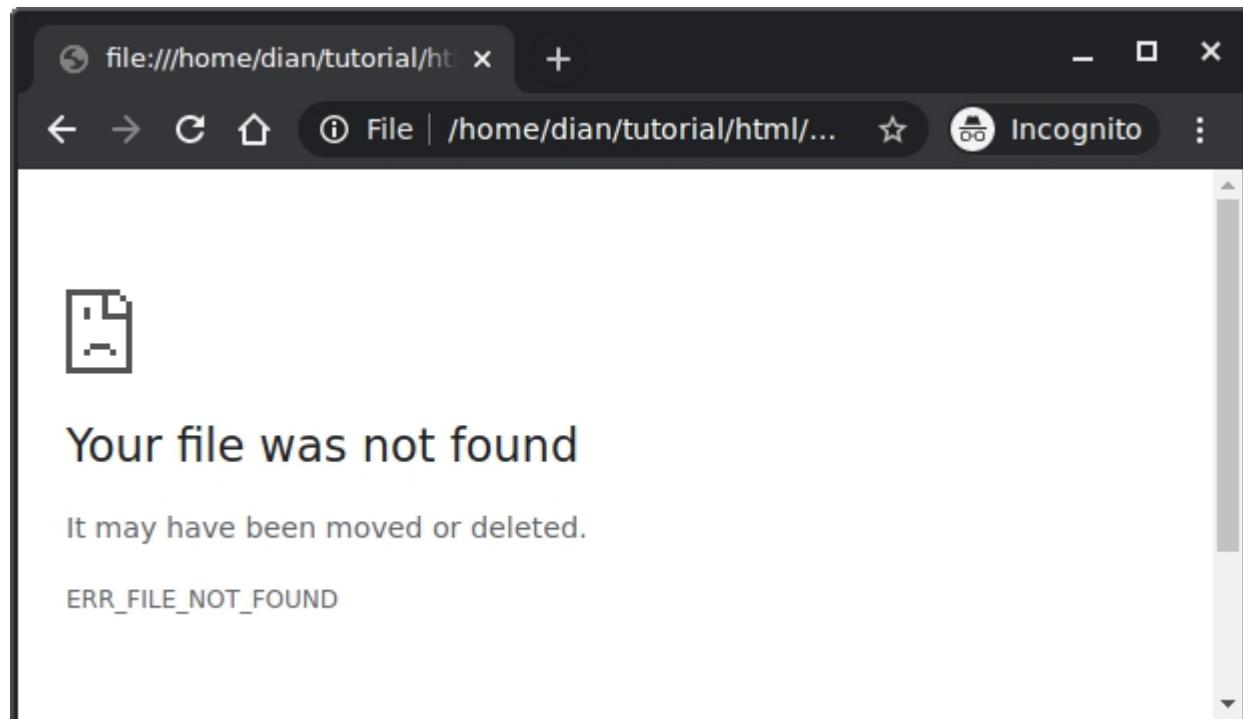


Link akan ditampilkan dengan warna biru dan garis bawah. Ini adalah style standar dari setiap browser.

Tentu saja ini bisa kita ubah-ubah sesuai selera.

Nanti kita akan bahas caranya ya..

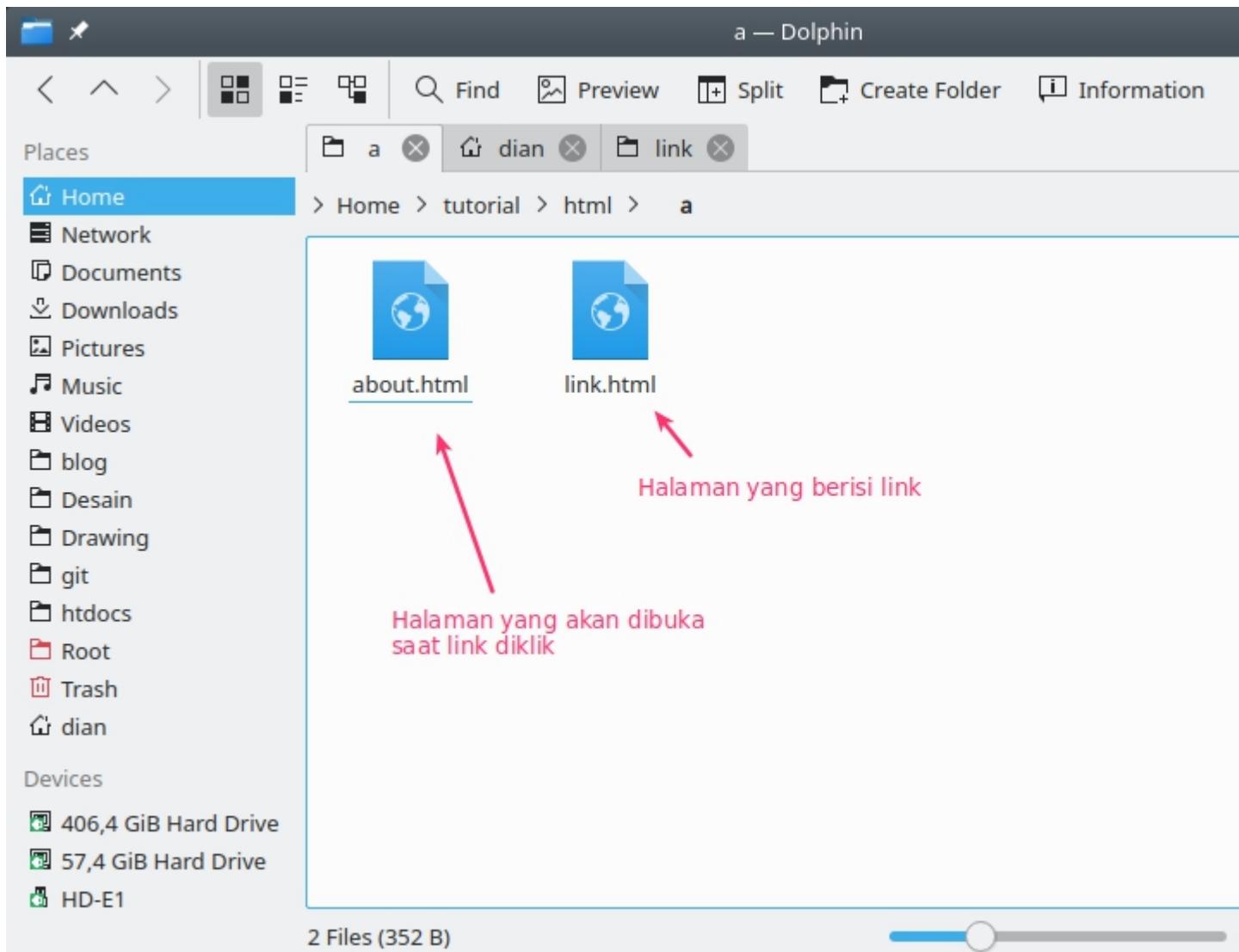
Nah sekarang coba klik link tersebut!



Lha! Kok error?

Link yang kita buat ini akan membuka halaman `about.html`, tapi file ini belum kita buat atau belum ada. Karena itu, pasti error.

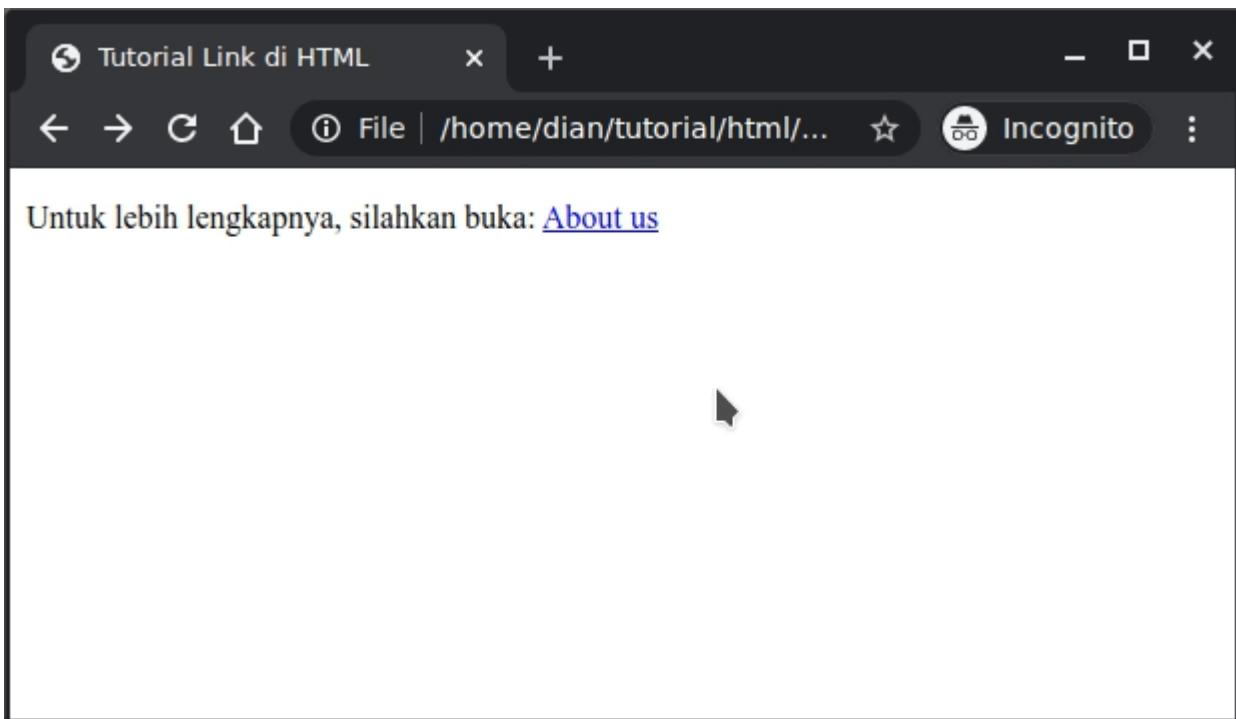
Untuk mengatasinya, kita harus membuat file `about.html` dan disimpan dalam satu folder dengan file `link.html`.



Isi file `about.html` seperti ini:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <h1>About us</h1>
        <p>
            Ini adalah website yang dibuat dengan link.
            Kamu bisa membuka halaman dengan mudah. Tinggal klik-klik saja!
        </p>
        <p>
            <a href="link.html">Kembali ke halaman link</a>
        </p>
    </body>
</html>
```

Nah, sekarang mari kita coba lagi:



Yap!

Halaman `link.html` dan `about.html` berhasil terhubung.

Tapi tunggu dulu..

Mengapa alamat URL pada atribut `href` ditulis dengan nama file?

```
<a href="about.html">About us</a>
```

Bukannya harus alamat URL lengkap yang pakai `http` bla bla?

Ini karena link yang kita buat adalah internal link dan juga file `about.html` masih dalam satu folder dengan file `link.html`. Jadi boleh menuliskan langsung nama filenya, tanpa harus lengkap dengan `http`.

Untuk lebih jelasnya, mari kita pelajari jenis-jenis link dalam HTML.

Jenis-jenis Link pada HTML

Berdasarkan alamat URL yang dituju, link pada HTML dibagi menjadi dua kelompok:

1.

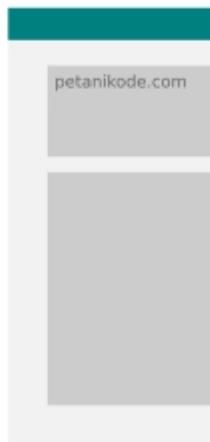
Internal Link: adalah link yang menuju ke domain atau halaman web itu sendiri;

2.

index.html



about.html

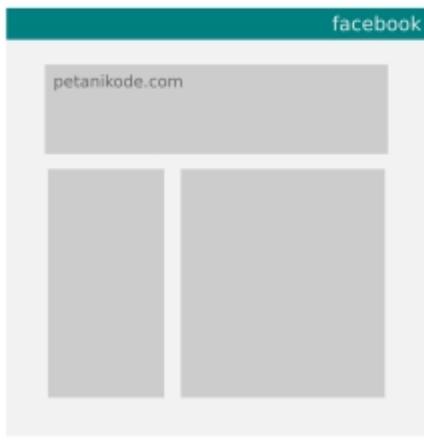


3.
4.
5.

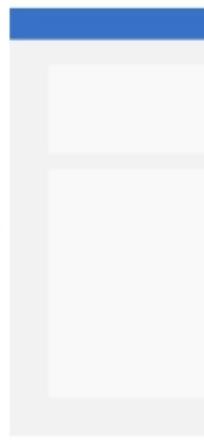
External Link: adalah link yang menuju domain lain.

6.

index.html



facebook.c



7.
8.

Internal link biasanya digunakan untuk menghubungkan halaman yang satu dengan yang lainnya dalam satu website atau domain.

Sementara untuk eksternal link, digunakan untuk membuka web atau domain lain. Misalnya seperti: membuka halaman facebook, membuka chat whatsapp, membuka youtube, dan sebagainya.

Intinya:

Selama link itu membuka web lain, maka ia adalah external. Tapi kalau tetap membuka web itu sendiri maka itu internal.

Biar lebih jelas, mari kita coba membuatnya di HTML:

1. Contoh Internal Link

Buatlah file baru bernama `index.html` kemudian isi dengan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <h1>Selamat Datang</h1>
    <p>
      <a href="index.html" >Home</a> | 
      <a href="contact.html" >Contact</a> | 
      <a href="about.html" >About</a>
    </p>
    <hr>
    <p>
      Selamat datang di websiteku. Coba klik menu di atas,
      maka kamu akan membuka halaman yang berbeda. Semua
      link di atas adalah internal link.
    </p>
    <hr>
    <div>Copyright © 2020 by Petani Kode</div>
  </body>
</html>
```

Kemudian buat lagi file baru bernama `contact.html` dengan isi sebagai berikut:

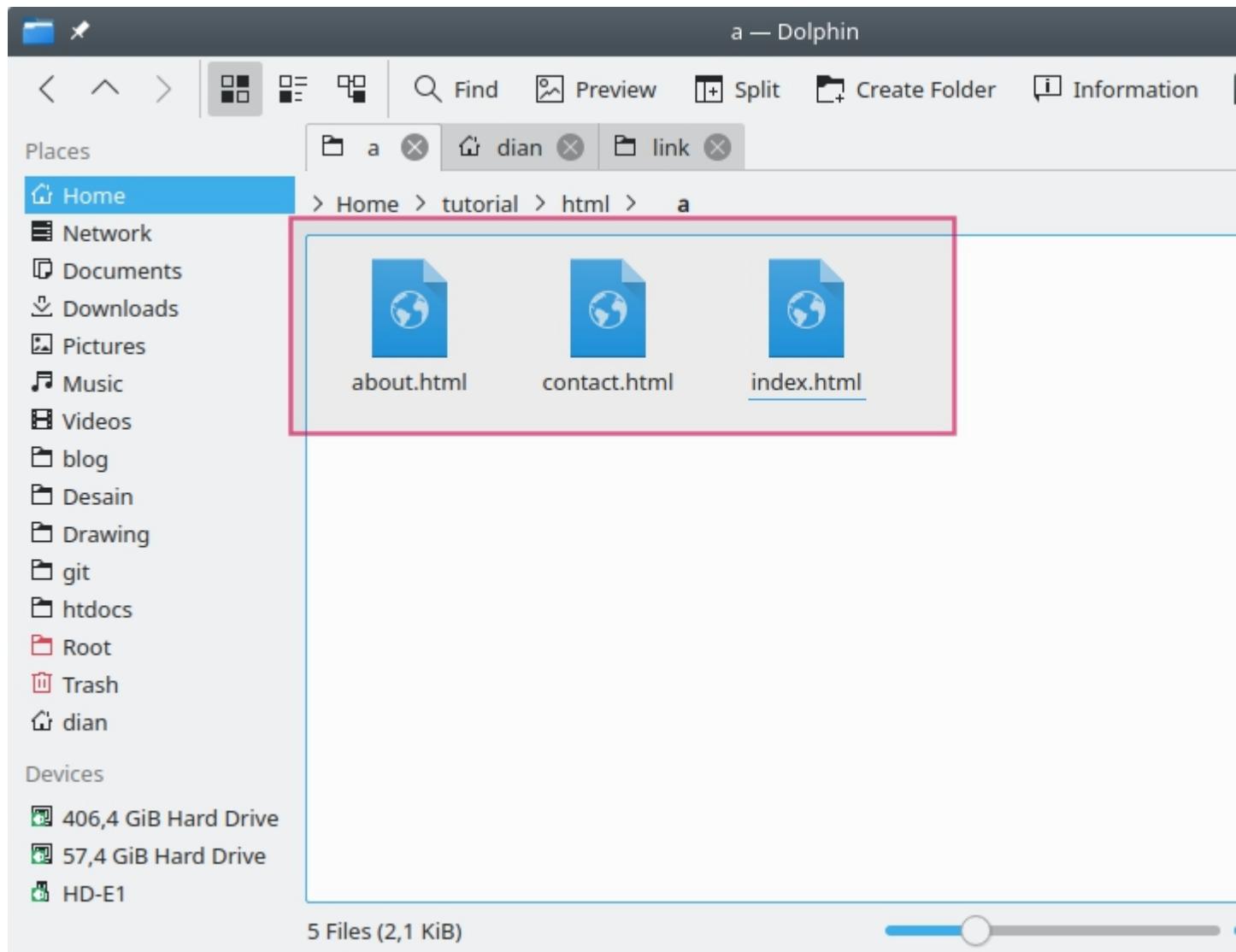
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <h1>Contact</h1>
    <p>
      <a href="index.html" >Home</a> | 
      <a href="contact.html" >Contact</a> | 
      <a href="about.html" >About</a>
    </p>
  </body>
</html>
```

```
</p>
<hr>
<p>
    Kamu bisa menghubungi saya melalui nomer WA: 0871111111
    atau juga alamat email: mail@contoh.com.
</p>
<hr>
<div>Copyright © 2020 by Petani Kode</div>
</body>
</html>
```

Terakhir, buat file `about.html` dengan isi sebagai berikut:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <h1>About</h1>
        <p>
            <a href="index.html" >Home</a> |
            <a href="contact.html" >Contact</a> |
            <a href="about.html" >About</a>
        </p>
        <hr>
        <p>
            Ini adalah halaman about dari website saya.
            Jadi ini adalah contoh cara membuat link internal di HTML.
        </p>
        <hr>
        <div>Copyright © 2020 by Petani Kode</div>
    </body>
</html>
```

Pastikan semua file ini disimpan dalam satu folder.



Setelah itu, coba buka [index.html](#) dengan browser.

Maka hasilnya:

The screenshot shows a web browser window with the title bar "Tutorial Link di HTML". The main content area displays the text "Selamat Datang" in large bold letters. Below it are three blue underlined links: "Home", "Contact", and "About". A horizontal line separates this from a text block that says "Selamat datang di websiteku. Coba klik menu di atas, maka kamu akan membuka halaman yang berbeda. Semua link di atas adalah internal link." Another horizontal line follows, with the text "Copyright © 2020 by Petani Kode".

2. Contoh Eksternal Link

Cara membuat eksternal link sebenarnya sama saja seperti internal link. Perbedaanya terletak pada alamat URL yang diberikan.

Sebagai contoh.. ini adalah eksternal link yang menuju halaman facebook:

```
<a href="https://www.facebook.com/petanikode">Facebook Patani Kode</a>
```

Hasilnya:

[Facebook Patani Kode](https://www.facebook.com/petanikode)

Maka saat link tersebut diklik, kita akan membuka halaman <https://www.facebook.com/petanikode>.

Contoh lain lagi:

Link CTA (call to action) untuk whatsapp.

```
<a href="https://api.whatsapp.com/send?phone=6287111&text=Hi">Chat dengan Saya</a>
```

Hasilnya:

[Chat dengan Saya](https://api.whatsapp.com/send?phone=6287111&text=Hi)

Pada link tersebut, kita memberikan alamat URL whatsapp dan nantinya akan membuka halaman tersebut. Ini juga bisa disebut eksternal link, karena ia akan membuka halaman whatsapp.

Oh iya, di sana kita berikan parameter berupa nomer hp dan juga teks.

Nah, biar lengkap.. kita akan coba tambahkan eksternal link pada file `contact.html`.

Silahkan buka file `contact.html` kemudian ubah kodenya menjadi seperti ini:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <h1>Contact</h1>
        <p>
            <a href="index.html" >Home</a> | 
            <a href="contact.html" >Contact</a> | 
            <a href="about.html" >About</a>
        </p>
        <hr>
        <p>
            Kamu bisa menghubungi saya melalui nomer WA: <a href="https://api.whatsapp.com/send?phone=6287111&text=Hi">+6287111</a> atau juga alamat email: mail@contoh.com.
        </p>
        <p>
            Jangan lupa follow juga fan page saya ya: <a href="https://www.facebook.com/petanikode">Petani Kode</a>
        </p>
        <hr>
        <div>Copyright © 2020 by Petani Kode</div>
    </body>
</html>
```

Hasilnya:

Tes buka link Facebook:

A screenshot of a web browser window titled "Tutorial Link di HTML". The address bar shows the file path: "/home/dian/tutorial/html/...". The main content area has a dark background with white text. It features a large bold heading "Contact". Below it is a horizontal line with a left-pointing arrow icon. Underneath the arrow is the text: "Kamu bisa menghubungi saya melalui nomer WA: [+6287111](#) atau juga alamat email: mail@contoh.com". Another horizontal line follows, with the text: "Jangan lupa follow juga fan page saya ya: [Petani Kode](#)". A third horizontal line at the bottom contains the text: "Copyright © 2020 by Petani Kode".

Tes buka link whatsapp:

A screenshot of a web browser window titled "Tutorial Link di HTML". The address bar shows the file path: "/home/dian/tutorial/html/...". The main content area has a dark background with white text. It features a large bold heading "Contact". Below it is a horizontal line with a left-pointing arrow icon. Underneath the arrow is the text: "Kamu bisa menghubungi saya melalui nomer WA: [+6287111](#) atau juga alamat email: mail@contoh.com". Another horizontal line follows, with the text: "Jangan lupa follow juga fan page saya ya: [Petani Kode](#)". A third horizontal line at the bottom contains the text: "Copyright © 2020 by Petani Kode". A cursor arrow is positioned over the WhatsApp link ([+6287111](#)).

Sudah paham tentang eksternal link?

Oke, sekarang lanjut belajar atribut:

Atribut-atribut untuk Link

Selain atribut `href` terdapat juga beberapa atribut yang sering ditambahkan pada link, seperti: `target`, `title`, `rel`, `style`, dan lain-lain.

Menggunakan Atribut **target**

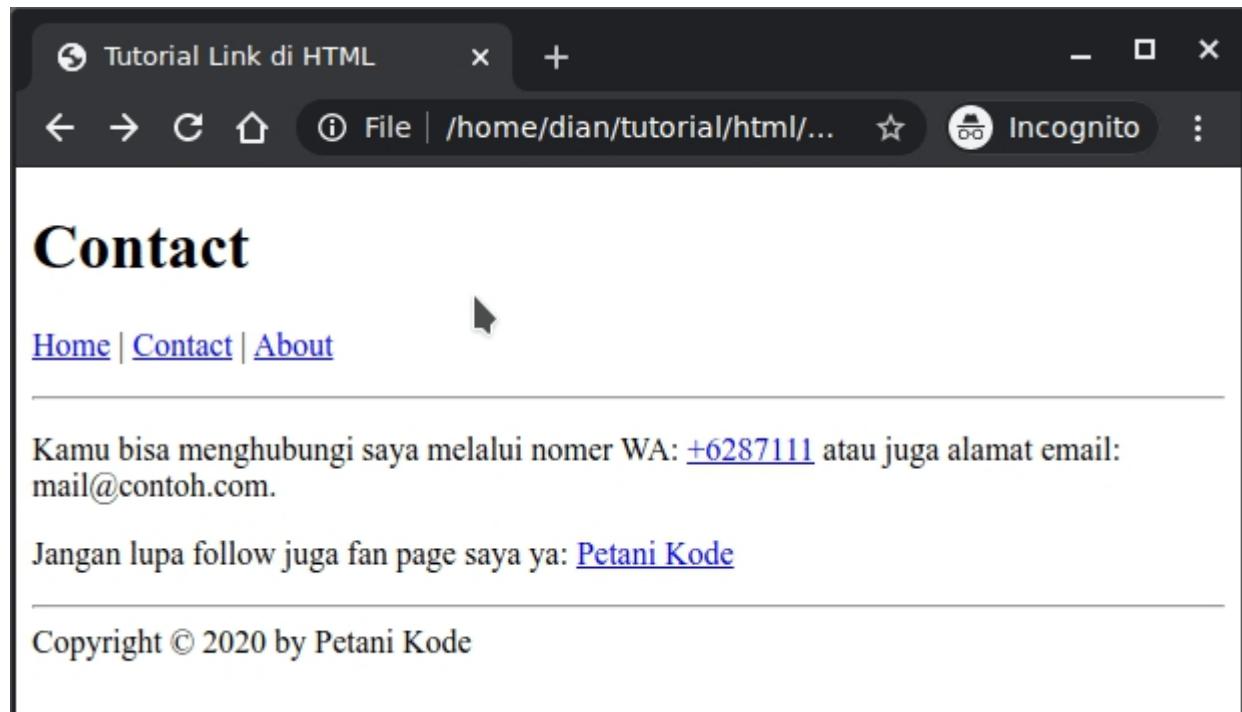
Atribut ini berfungsi untuk menentukan target dari pembukaan link. Ada beberapa target yang biasanya digunakan:

- `_blank` akan membuka link pada jendela atau tab baru;
- `_self` akan membuka link pada halaman itu sendiri (default target);
- `_top` menuju bagian paling atas pada halaman;
- `_parent` membuka link pada frame induk;
- `nama-frame` akan membuka link pada `<iframe>` dengan nama tertentu;

Contoh:

```
<a href='https://www.facebook.com/petanikode'  
target='_blank'>Facebook</a>
```

Hasilnya:



Saat link tersebut diklik, browser akan membuka tab baru. Ini karena kita memberikan atribut target dengan nilai `_blank`.

Berikutnya, kita akan mencoba menampilkan link ke dalam sebuah frame.

Buatlah file baru bernama `link-frame.html`, kemudian isi kodennya seperti ini:

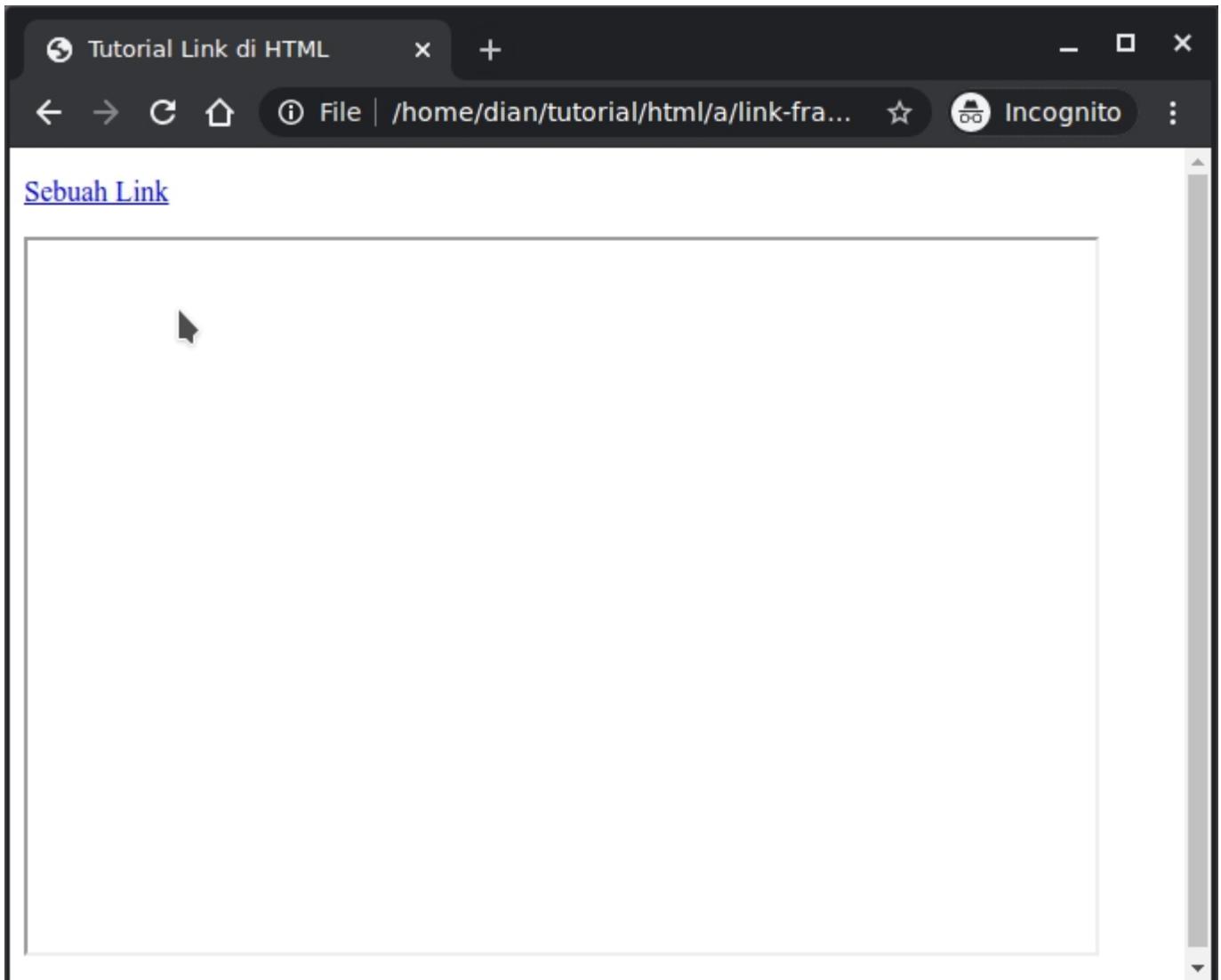
```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <title>Tutorial Link di HTML</title>
```

```
</head>

<body>
  <p>
    <a href="https://www.petanikode.com" target="myframe">Sebuah Link</a>
  </p>
  <p>
    <!-- Frame yang akan menjadi target link -->
    <iframe name="myframe" width="600" height="400"></iframe>
  </p>
</body>

</html>
```

Hasilnya:



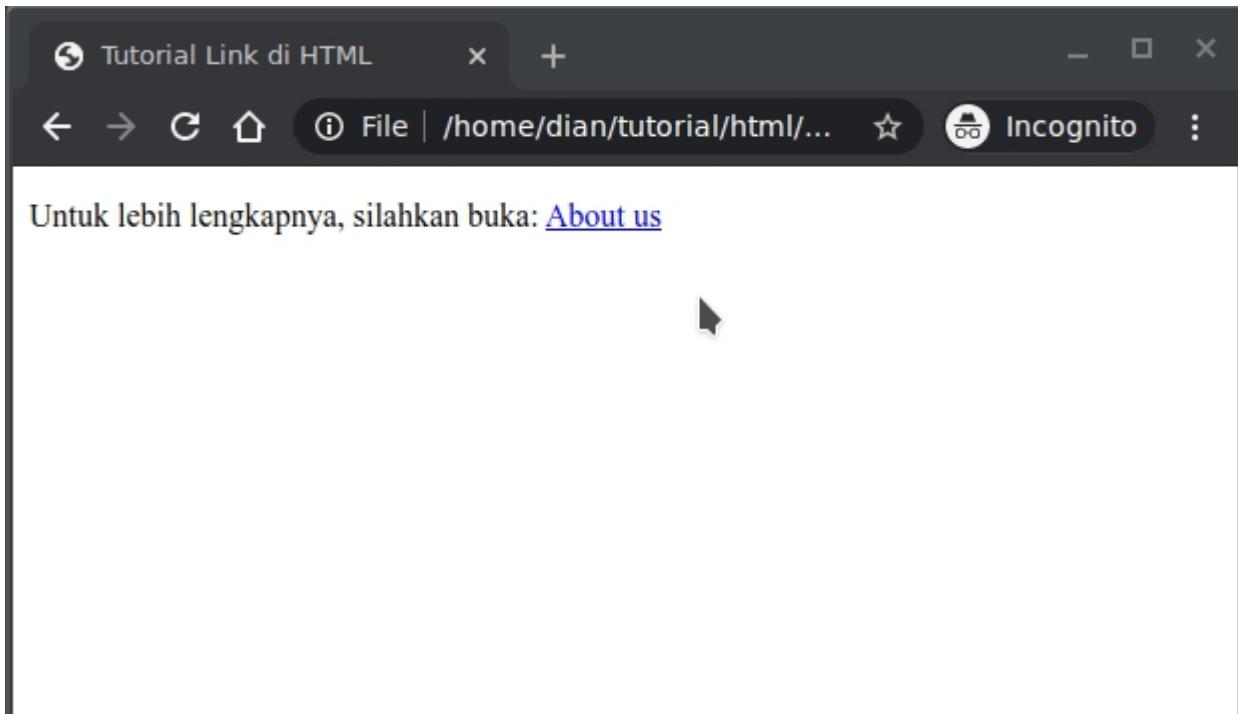
Menggunakan Atribut **title**

Atribut ini berfungsi untuk membuat tooltips. Tooltips adalah informasi tambahan yang akan tampil saat link disentuh pointer atau saat kita menahan tap di hp.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>Untuk lebih lengkapnya, silahkan buka:
      <a href="about.html" title="Menuju ke halaman about">About us</a>
    </p>
  </body>
</html>
```

Hasilnya:



Live Demo:

Untuk lebih lengkapnya, silahkan buka: [About us](#)

Cara Mengubah Warna Link di HTML

Warna default link adalah biru, ini bisa kita ubah dengan style CSS. Warna bisa kita berikan untuk teks dan latar belakang (background).

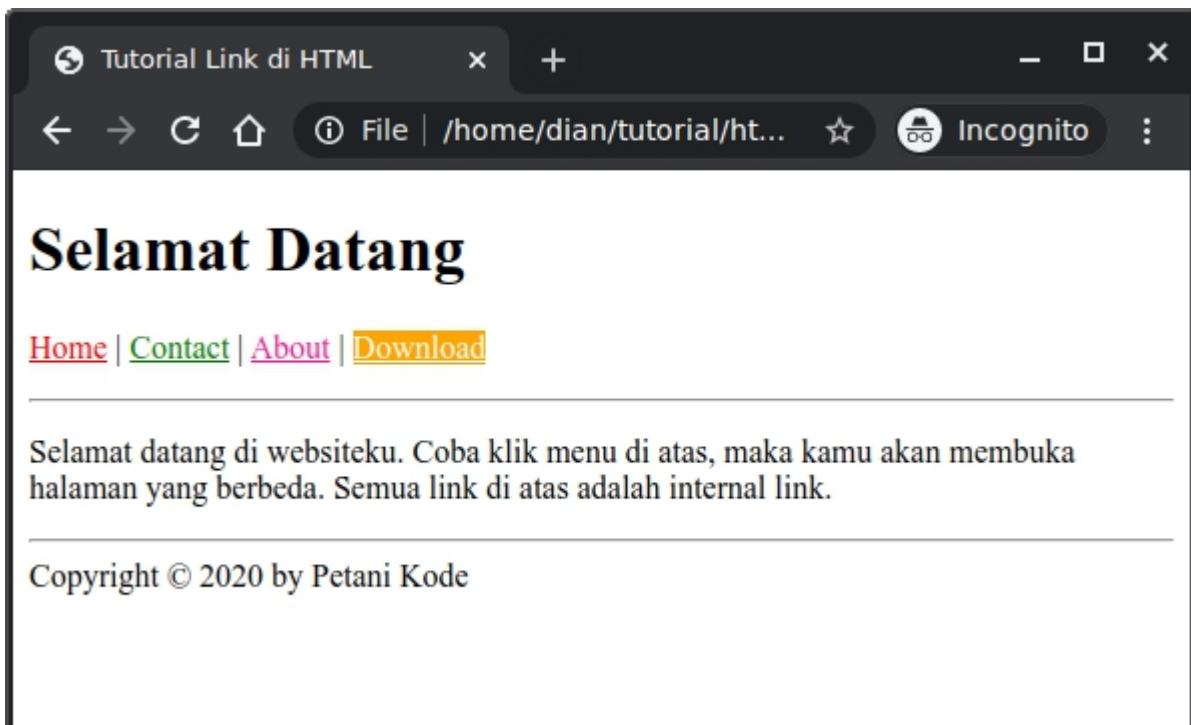
Caranya:

Tambahkan atribut `style` kemudian isi dengan style css untuk mengubah warna, yakni `color` (untuk teks) dan `background-color` (untuk latar).

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <h1>Selamat Datang</h1>
    <p>
      <a href="index.html" style="color:red">Home</a> |
      <a href="contact.html" style="color:green">Contact</a> |
      <a href="about.html" style="color:deeppink;">About</a> |
      <a href="download.html" style="color:white;background-color:
orange;">Download</a>
    </p>
    <hr>
    <p>
      Selamat datang di websiteku. Coba klik menu di atas,
      maka kamu akan membuka halaman yang berbeda. Semua
      link di atas adalah internal link.
    </p>
    <hr>
    <div>Copyright © 2020 by Petani Kode</div>
  </body>
</html>
```

Hasilnya:



Fungsi Menarik Lainnya dari Link

Seperti yang saya bilang sebelumnya.. link tidak hanya digunakan untuk menghubungkan halaman web saja.

Ia juga bisa digunakan untuk beberapa fungsi tertentu seperti:

Membuat Link Buntu

Link buntu adalah link yang tidak akan membuka apa-pun. Ia biasanya digunakan sebagai placeholder atau sampel saja.

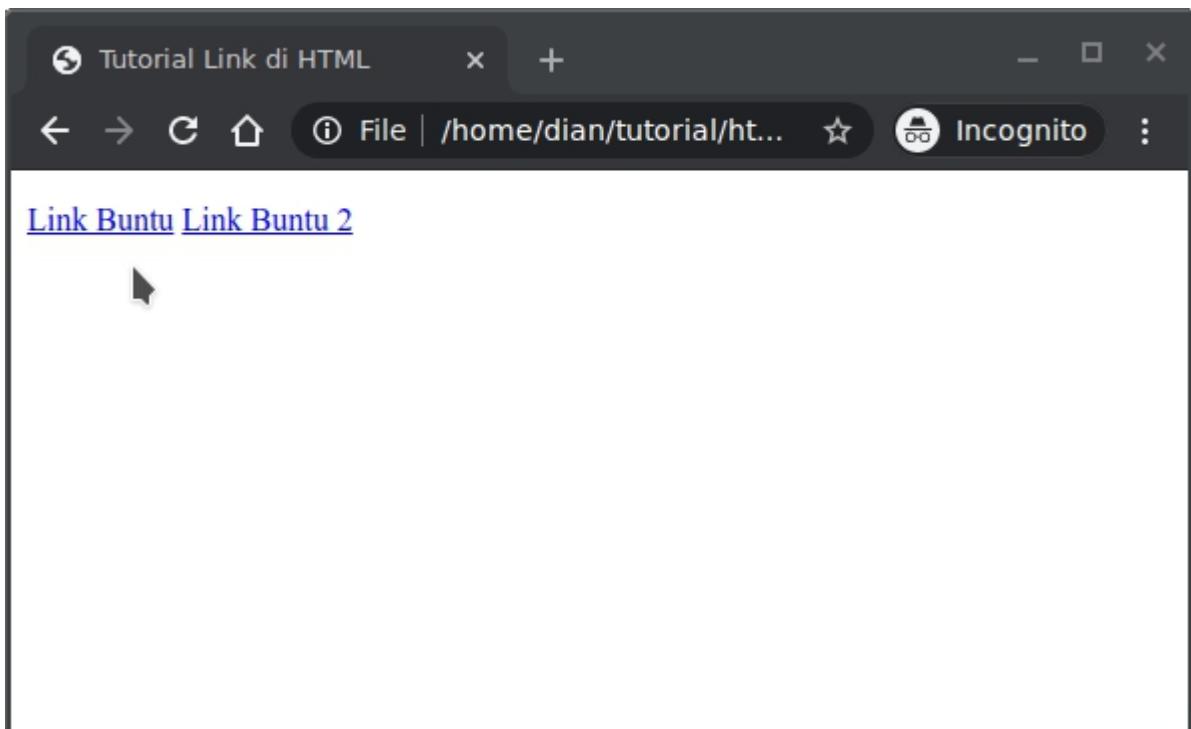
Cara membuatnya:

Tambahkan tanda pagar atau tanda pagar dan tanda seru pada atribut `href`.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <p>          <a href="#">Link Buntu</a>          <a href="#!">Link
Buntu 2</a>      </p>
    </body>
</html>
```

Hasilnya:



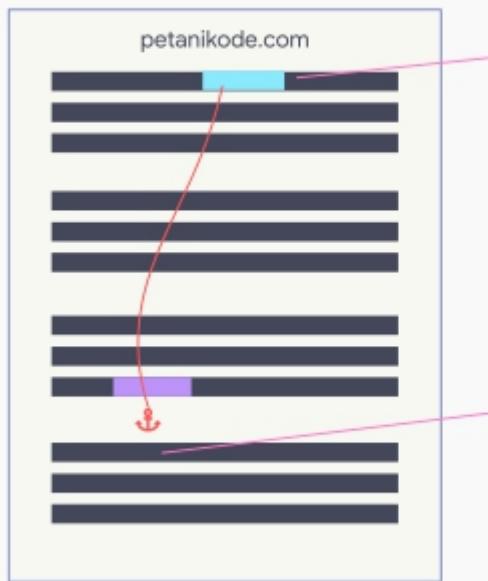
Perhatikan!

Pada contoh tersebut kita mengisi URL-nya dengan tanda pagar. Arti tanda pagar pada URL sebenarnya adalah **anchor** (jangkar).

Jangkar ini nantinya akan membawa kita ke lokasi tertentu di dalam dokumen HTML. Jika hanya diisi pagar saja, maka ia tidak akan kemana-mana.

Membuat Link Anchor

Link anchor adalah link yang menuju ke suatu elemen tertentu. Cara kerjanya seperti saat kita mengaitkan jangkar, lalu ditarik ke sana. Karena itu, dia dinama anchor.



Ini adalah link yang berisi anchor untuk menuju ke elemen dengan id 'penutup'

`Penutup`

Ini adalah elemen dengan id 'penutup', kita nantinya akan dibawa ke sini.

`<h2 id="penutup">Penutup</h2>`

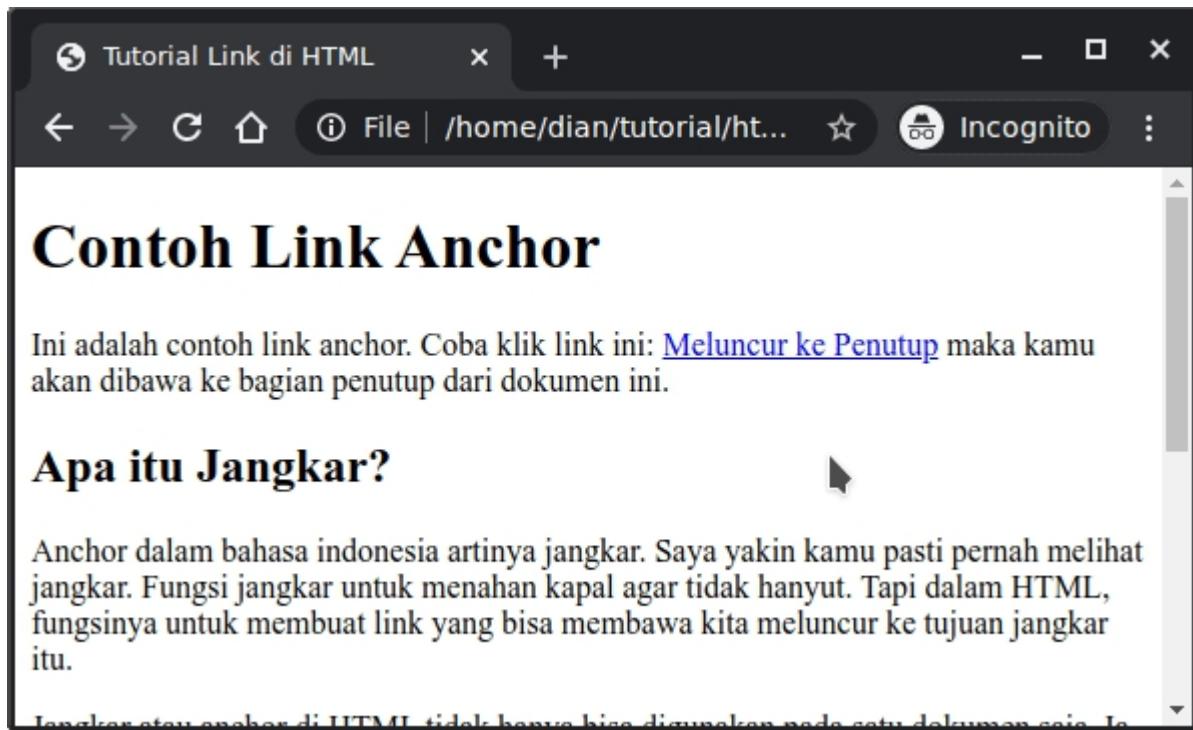
Cara membuat link anchor adalah dengan mengisi alamat URL dengan tanda pagar (#), lalu diisi dengan nama **id** dari elemen yang akan dituju.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Tutorial Link di HTML</title>
    </head>
    <body>
        <h1>Contoh Link Anchor</h1>
        <p>Ini adalah contoh link anchor. Coba klik link ini: <a href="#penutup">Meluncur ke Penutup</a> maka kamu akan dibawa ke bagian penutup dari dokumen ini.</p>
        <h2>Apa itu Jangkar?</h2>
        <p>
            Anchor dalam bahasa indonesia artinya jangkar. Saya yakin kamu pasti pernah melihat jangkar. Fungsi jangkar untuk menahan kapal agar tidak hanyut. Tapi dalam HTML, fungsinya untuk membuat link yang bisa membawa kita meluncur ke tujuan jangkar itu.
        </p>
        <p>
            Jangkar atau anchor di HTML tidak hanya bisa digunakan pada satu dokumen saja. Ia juga bisa digunakan untuk membuka dokumen lain, lalu mengaitkan jangkarnya.
        </p>
    </body>
</html>
```

```
</p>
<p>
    Jangkar biasanya dibuat dengan tanda pagar (#) lalu diikuti
    dengan nama id dari elemen yang ingin dituju. Contohnya #penutup,
    maka link yang menggunakan anchor tersebut akan mencari elemen
    HTML yang memiliki id `penutup` dan membawa kita ke sana.
    Kira-kira begitulah cara kerjanya.
</p>      <h2 id="penutup">Akhir Kata..</h2>      <p>
    Ini adalah penutup dari artikel ini, dan coba perhatikan headingnya
    Di sana kita menggunakan id="penutup". Maka elemen ini akan menjadi
    tujuan dari anchor link.
</p>
<p>          <a href="#top">Kembali ke Atas!</a>           </p>
</body>
</html>
```

Hasilnya:



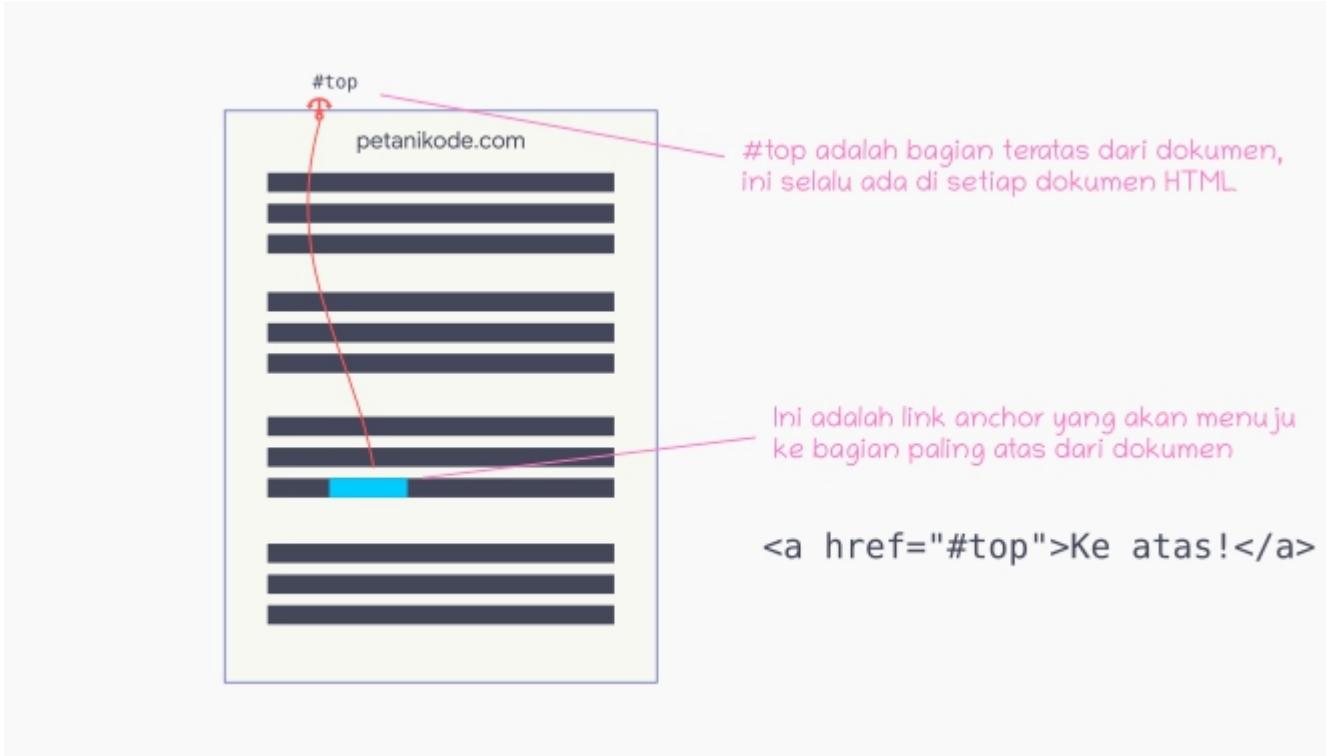
Kalau kita perhatikan, di sana kita menggunakan `#top` sebagai URL.

```
<a href="#top">Kembali ke Atas!</a>
```

Sedangkan elemen yang memiliki `id="top"`, tidak ada di dalam HTML yang kita tulis.

Mengapa link anchor ini bisa menuju ke atas?

Ini karena browser sudah paham, jika ada link anchor yang menuju ke `#top` maka ia akan dibawa ke bagian teratas dari dokumen.



Membuat Link untuk Memanggil Fungsi Javascript

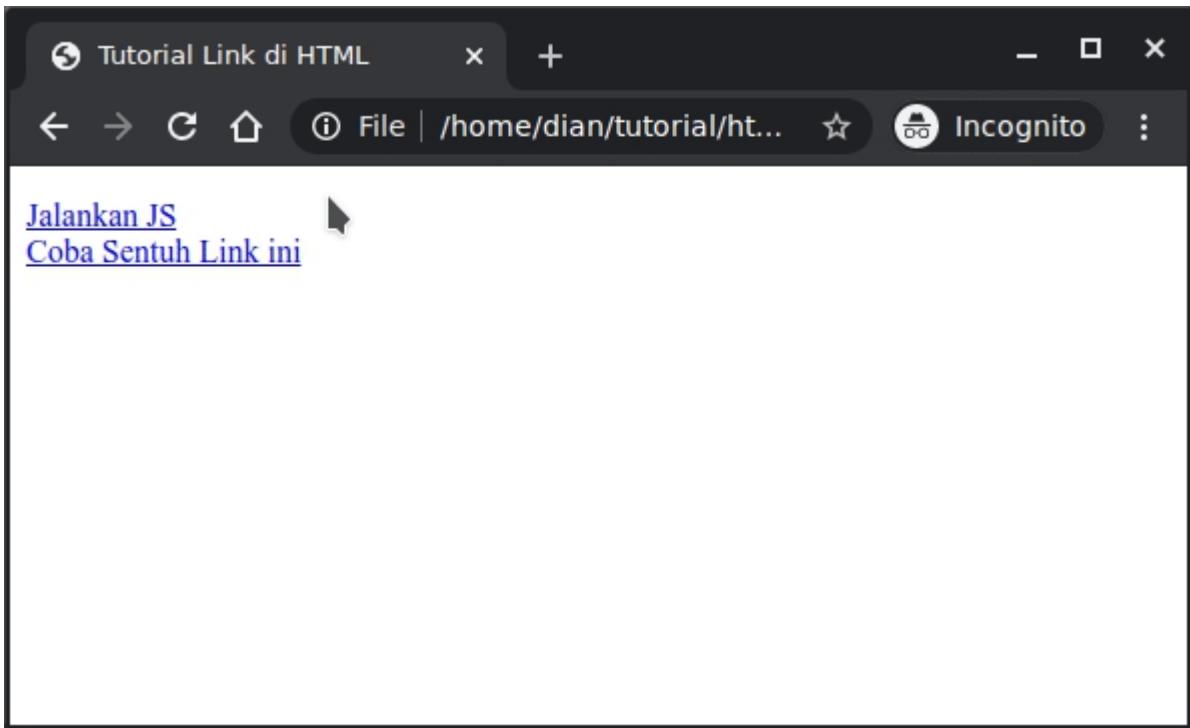
Link dapat juga digunakan untuk memanggil fungsi Javascript.

Pemanggilan fungsi Javascript biasanya dilakukan dengan atribut event seperti `onclick`, `onmouseover`, `onmouseout`, dan sebagainya.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>          <a href="#" onclick="alert('Hello World!')">Jalankan JS</a>      <br>          <a href="#" onmouseover="alert('link sudah kamu sentuh!')">Coba Sentuh Link ini</a>      </p>
  </body>
</html>
```

Hasilnya:



Sebenarnya hampir semua elemen bisa menjalankan fungsi Javascript dengan atribut event.

Nah, khusus untuk link. Dia bisa juga menjalankan fungsi Javascript melalui atribut `href` atau URL.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>          <a href="javascript:alert('Hello World!')">Jalankan JS</a>      </p>
  </body>
</html>
```

Hasilnya akan sama seperti contoh sebelumnya.

Membuat Link dengan Gambar

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

Contoh:

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

Hasilnya:



Semua area gambar akan menjadi link dan jika kita klik, maka akan dibawa ke halaman yang akan dituju oleh link tersebut.

Membuat Link untuk Download File

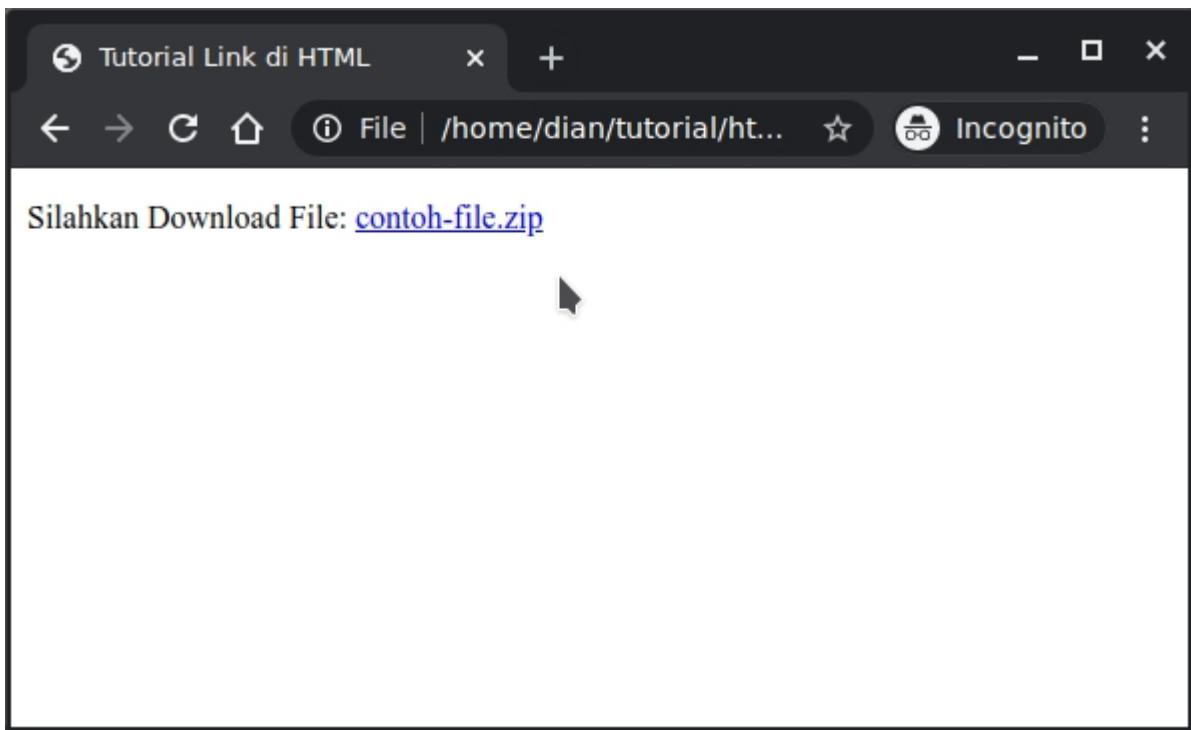
Link kadang juga digunakan sebagai link untuk download file.

Cara membuatnya sangat mudah, kita hanya perlu memasukan alamat URL dari file yang akan didownload.

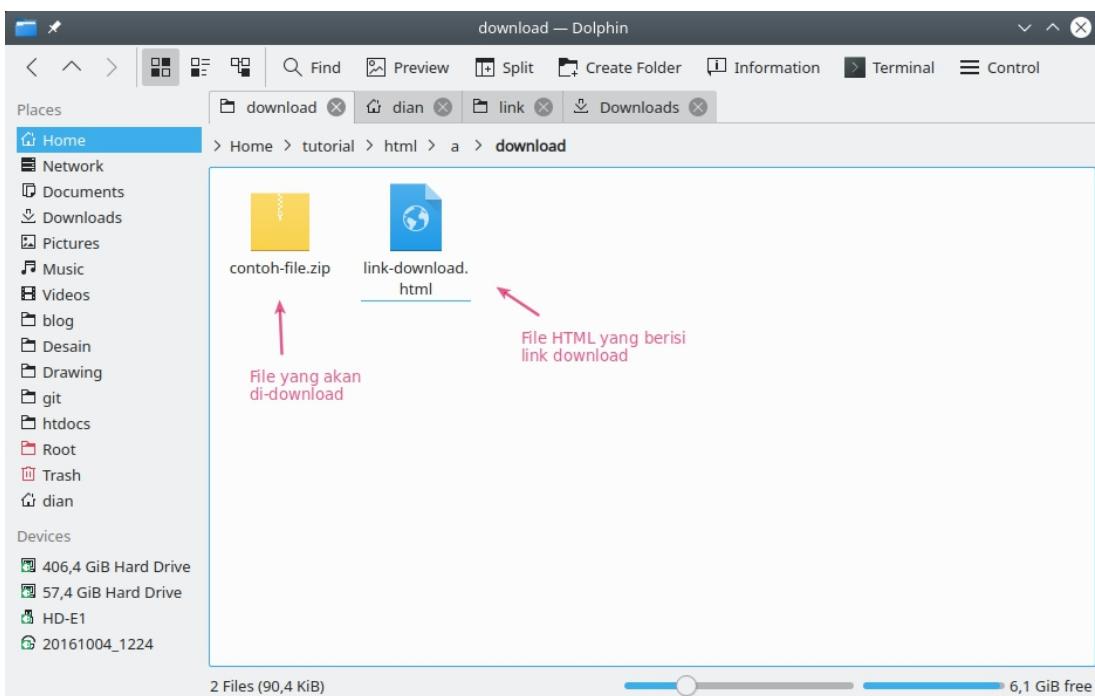
Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>Silahkan Download File: <a href="contoh-file.zip">contoh-
file.zip</a></p>
  </body>
</html>
```

Hasilnya:



Pada contoh di atas, kita menuliskan langsung nama filenya. Ini karena HTML `link-download.html` berada dalam satu folder dengan `contoh-file.zip`.



Lalu, bagaimana kalau filenya berada di tempat lain?

Gampang.. tinggal ganti saja URL-nya.

Contoh:

```
<a href="https://releases.ubuntu.com/20.04/ubuntu-20.04-desktop-amd64.iso">Download Ubuntu</a>
```

Membuat Link untuk Mengirim Email

Link kadang juga digunakan untuk mengirim email.

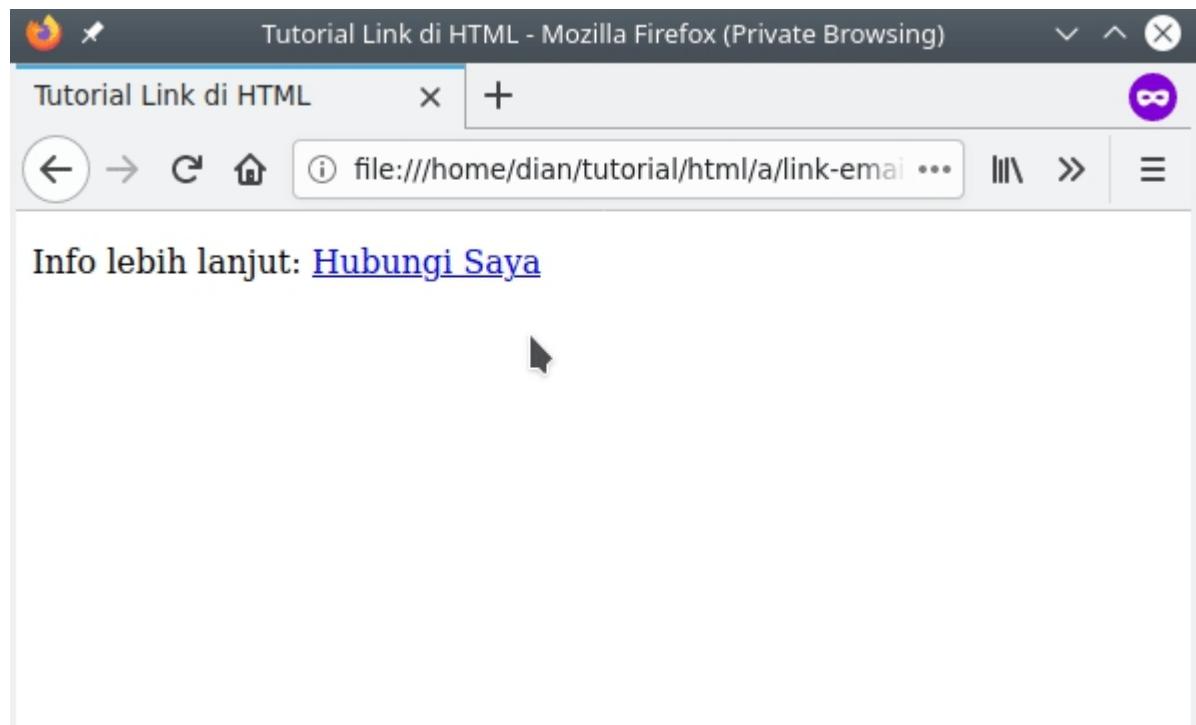
Cara membuatnya:

Ubah URL tujuan dari link dengan `mailto`, lalu diikuti dengan alamat email yang akan menerima email.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>
      Info lebih lanjut:
      <a href="mailto:tes@contoh.com"
          title="kirim email ke Admin">Hubungi Saya</a>
    </p>
  </body>
</html>
```

Hasilnya:



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

Menambahkan Gambar di HTML

Gambar dapat kita tambahkan di HTML dengan menggunakan tag ``. Tag ini memiliki atribut wajib, yakni `src`.

Jika kita tidak mengisi atribut `src`, maka gambar tidak akan ditampilkan.



petanikode.com

Alamat URL gambar pada atribut `src` dapat berupa URL maupun alamat path. Lalu tag `` harus ditutup dengan menambahkan garis miring.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>      </p>
</body>
</html>
```

Hasilnya:

S Contoh Gambar di HTML × +
← → C ⌂ ⌂ File | /home/dian/tutorial/html/img/image.html ☆

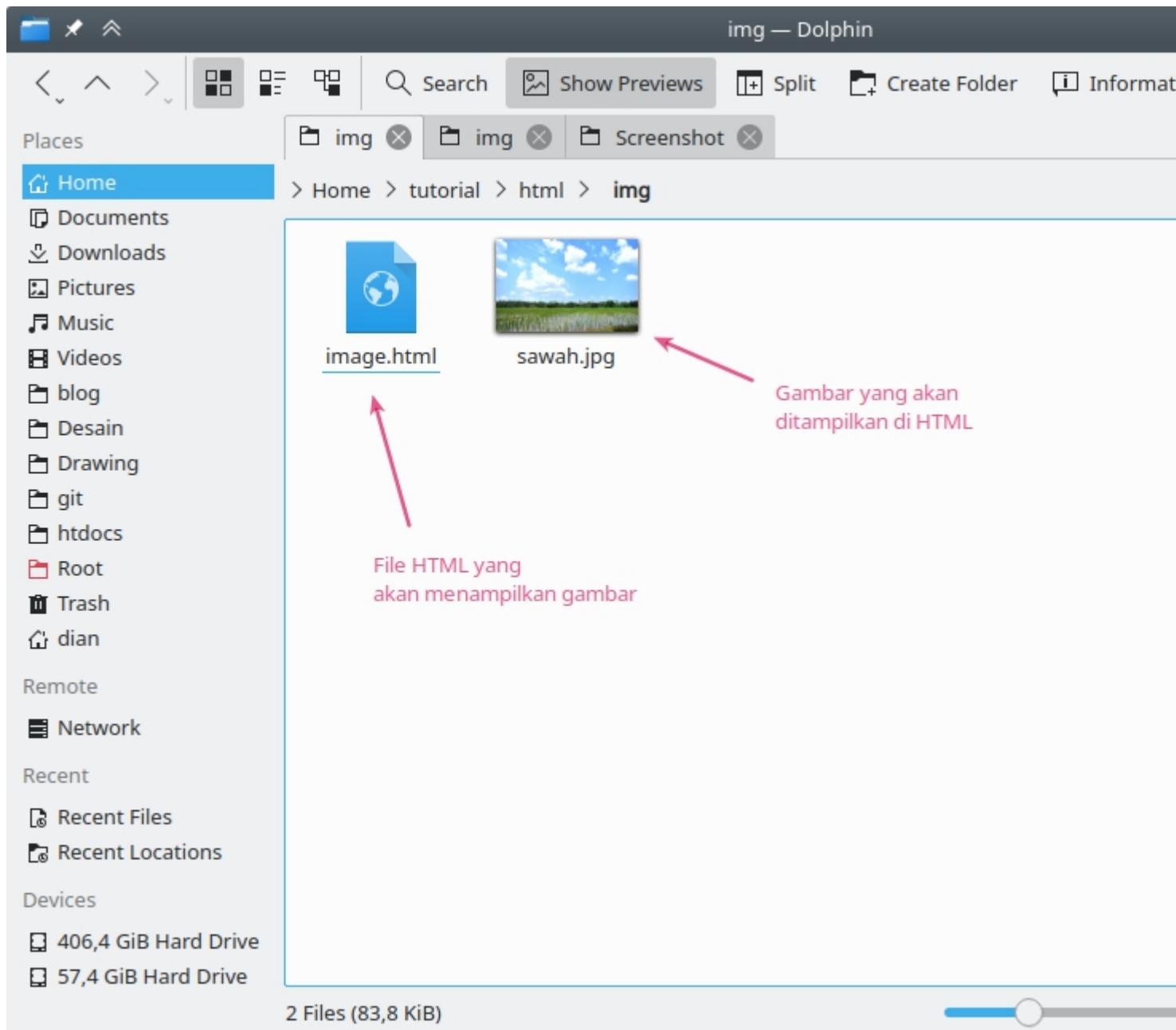
Menampilkan Gambar di HTML

Berikut ini adalah gambar sawah:



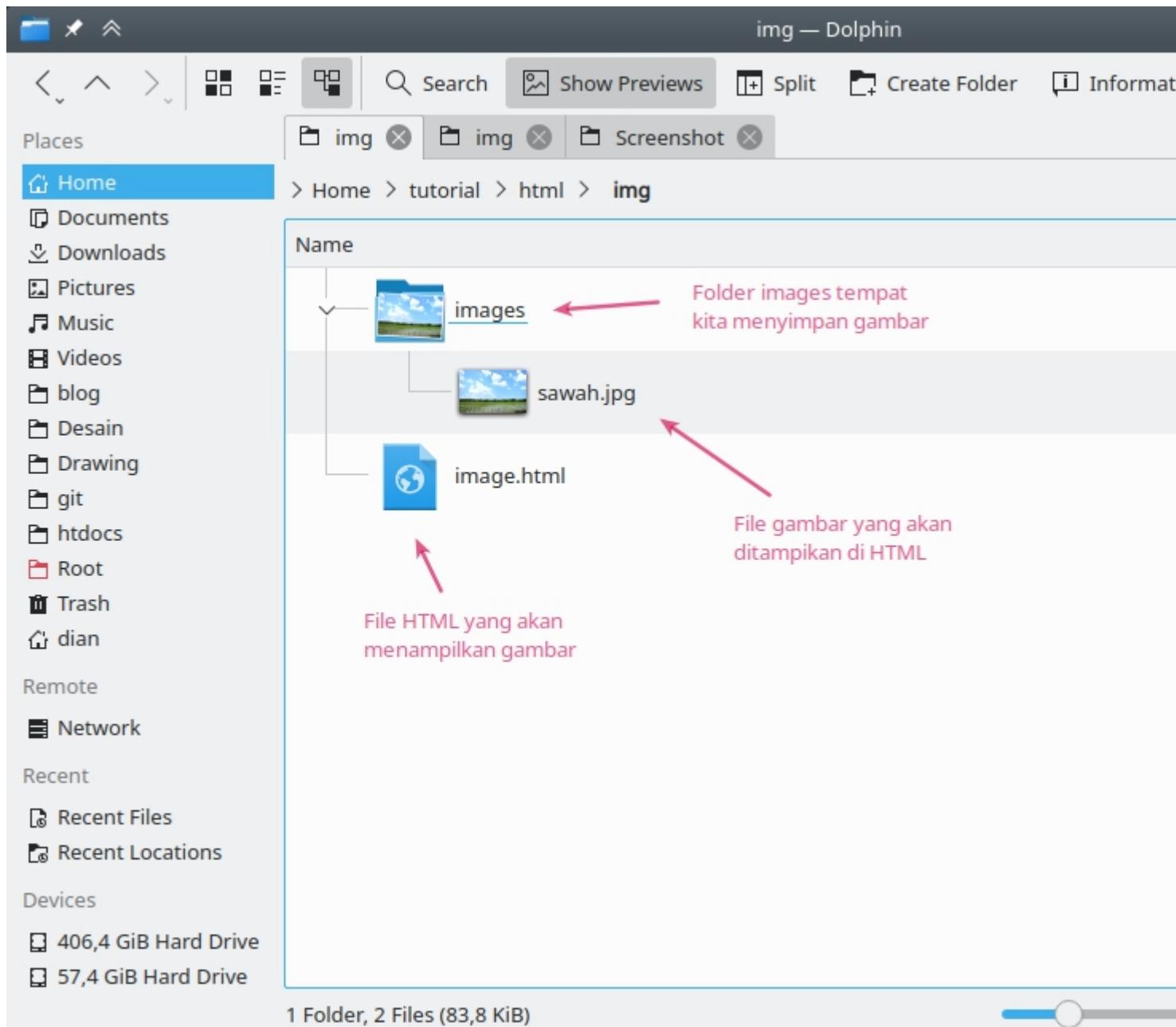
Perhatikan!

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



Apabila file gambarnya tersimpan di folder yang berbeda, maka kita hanya perlu menuliskan alamat path-nya.

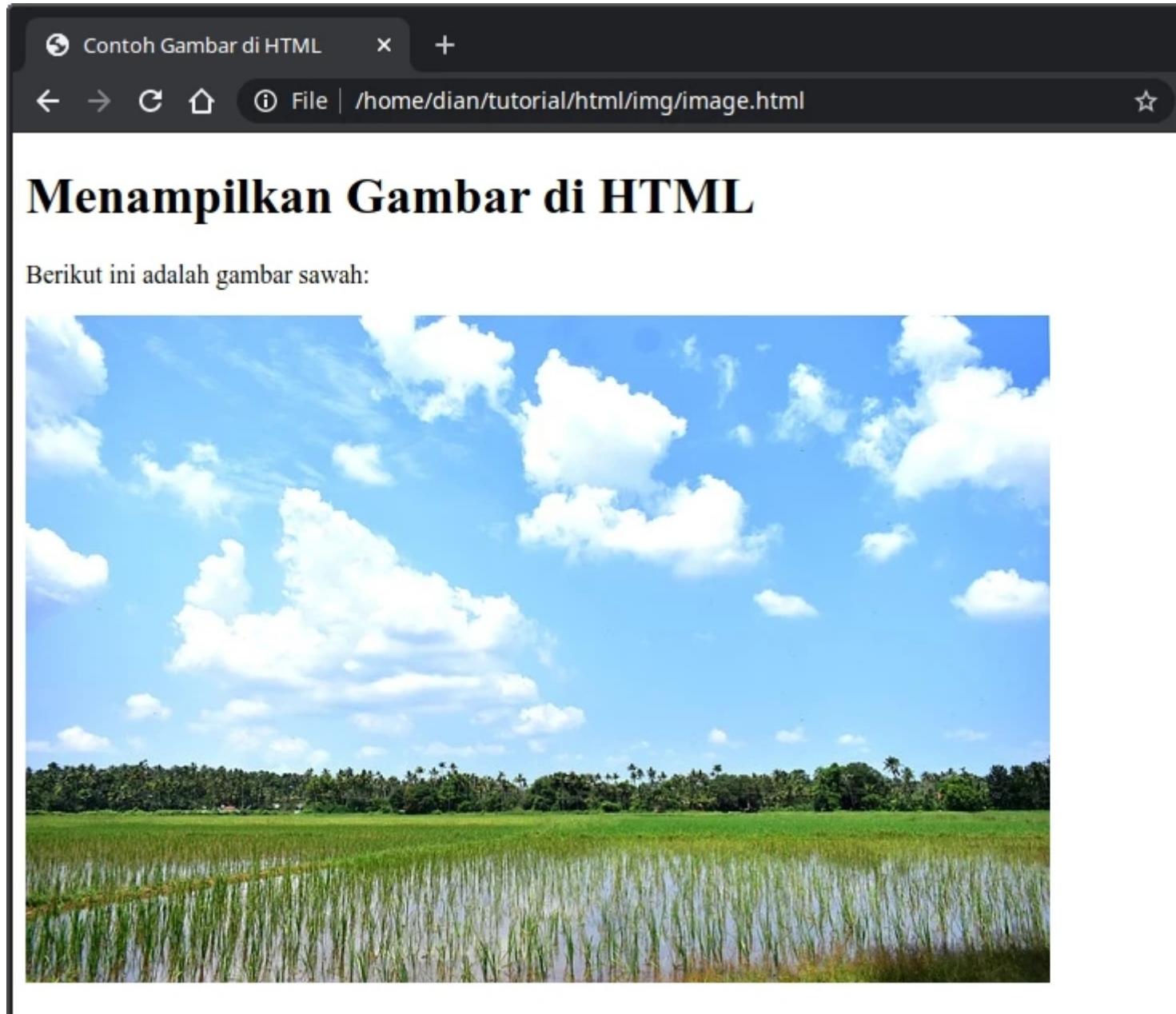
Misalkan, kita akan menyimpan gambar di dalam folder **images**.



Maka di HTML, kita bisa tulis seperti ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>      </p>
</body>
</html>
```

Hasilnya akan sama seperti contoh di atas.



Menampilkan Gambar di HTML

Berikut ini adalah gambar sawah:



Lalu bagaimana kalau gambarnya berada di internet atau website lain?

Nah, jika kita menggunakan gambar dari website lain..

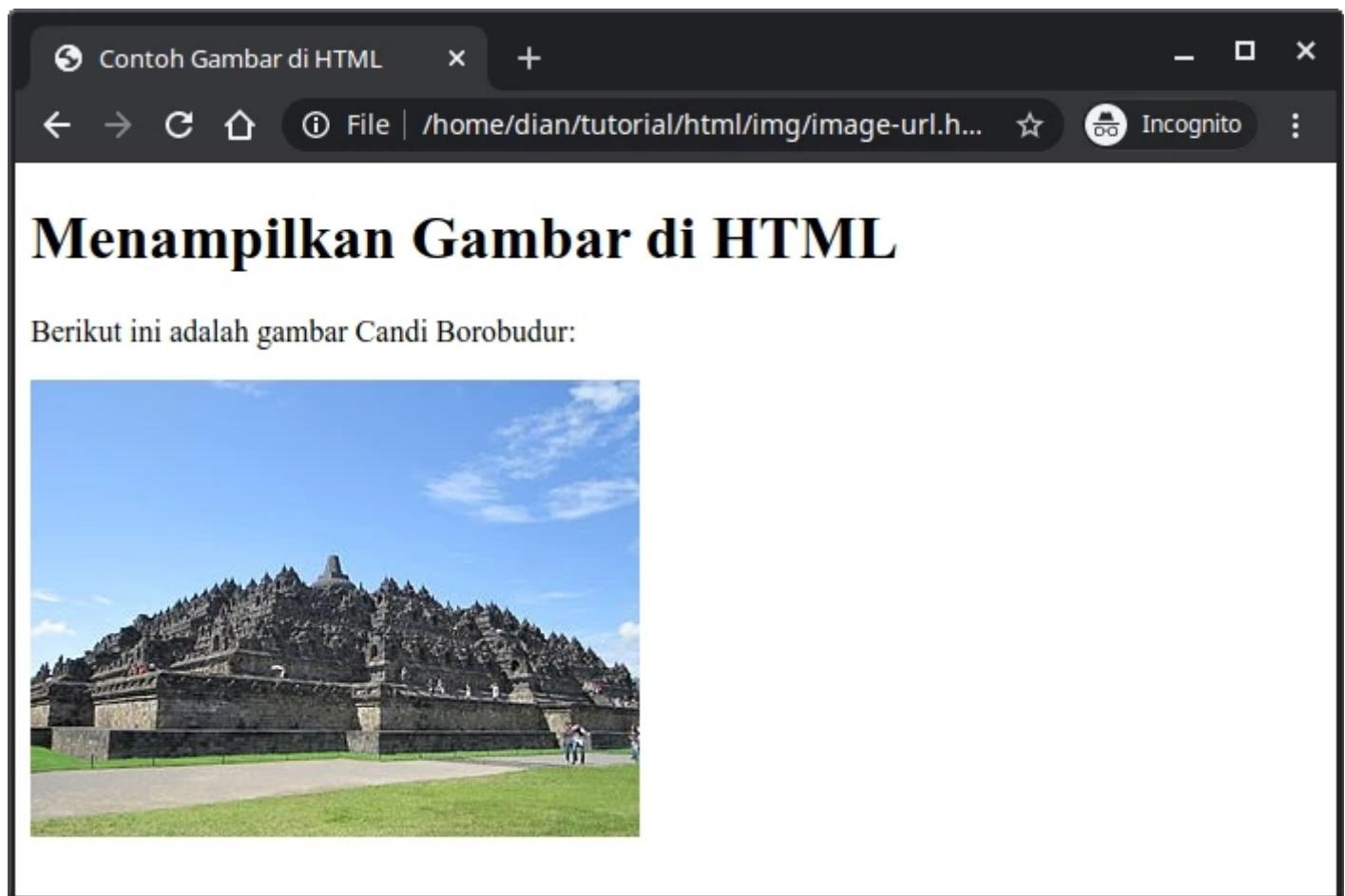
..kita harus menuliskan alamat URL lengkap dari gambar tersebut.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
```

```
</head>
<body>
    <h1>Menampilkan Gambar di HTML</h1>
    <p>Berikut ini adalah gambar Candi Borobudur:</p>
    <p>            </p>
</body>
</html>
```

Hasilnya:



Gampang kan?

Nah, berikutnya kita akan pelajar format gambar apa saja yang bisa digunakan di HTML dan juga atribut lainnya untuk ``.

Silahkan lanjutkan..

Format File Gambar untuk HTML

Tidak semua format file gambar dapat ditampilkan di HTML. Karena tiap file gambar memiliki tujuan masing-masing.

Misalnya file gambar dengan format PSD, ia adalah file gambar untuk Photoshop. Jelas ini tidak akan bisa ditampilkan di HTML.

Lalu, format apa saja yang didukung oleh HTML?

Berikut ini daftar beberapa format gambar yang sering digunakan dalam web:

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

Format file ini juga akan bergantung pada versi browser yang digunakan.

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

Atribut untuk Tag

Ada beberapa atribut yang sering digunakan pada tag :

- `alt` untuk teks alternatif untuk gambar;
- `width` untuk menentukan lebar gambar;
- `height` untuk menentukan tinggi gambar;
- `style` untuk menentukan style CSS untuk gambar.

Mari kita coba satu per satu..

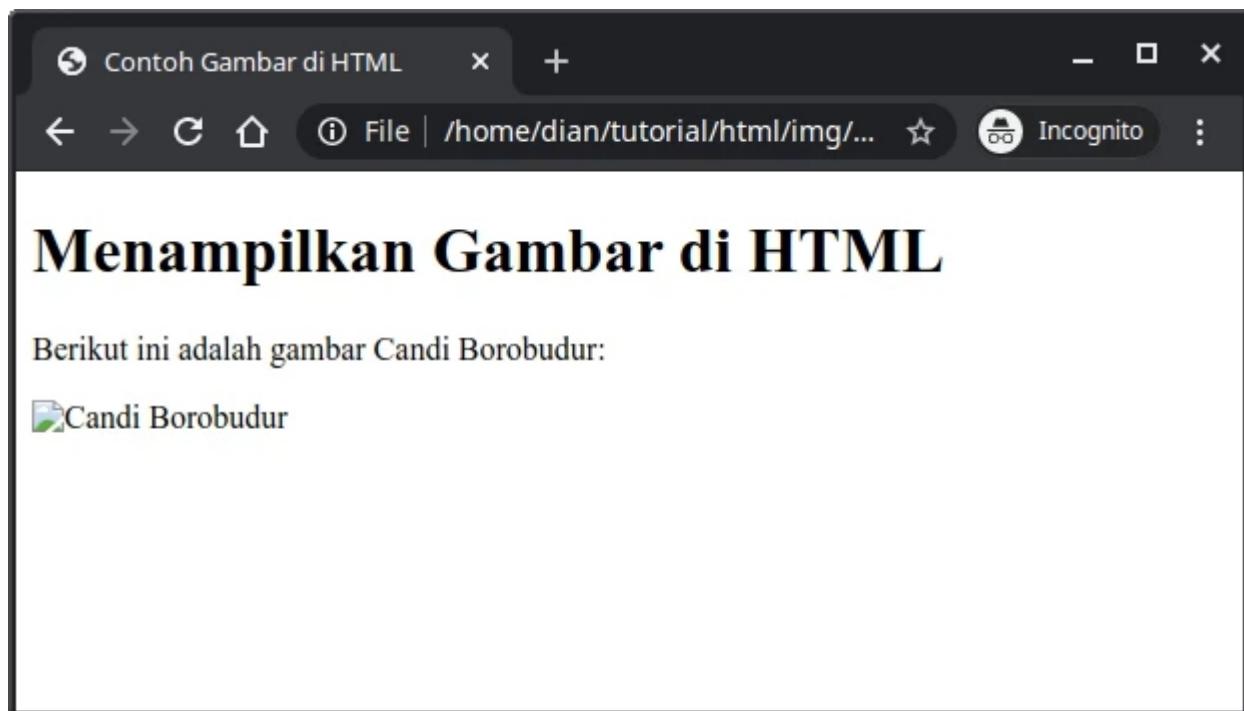
Atribut `alt`

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

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar Candi Borobudur:</p>
  <p>    <img src="" alt="Candi Borobudur" />  </p>
</body>
</html>
```

Hasilnya:



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

Apakah kita wajib menggunakan atribut `alt`?

Sebenarnya tidak wajib, tapi sebaiknya tetap menggunakan atribut ini. Karena, nantinya teks alternatif akan dibaca oleh screen reader.

Screen reader adalah aplikasi pembaca yang digunakan oleh tuna netra agar bisa mengetahui informasi yang ada di komputer.

Selain itu, atribut `alt` juga sering digunakan untuk meletakan kata kunci untuk SEO.

Atribut `width` dan `height`

Atribut `width` dan `height` adalah atribut yang digunakan untuk menentukan lebar dan tinggi dari gambar.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>
<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>              </p>
</body>
</html>
```

Hasilnya:



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

Atribut `width` dan `height` juga tidak wajib. Tapi baiknya ditambahkan agar ukuran gambar konsisten.

Atribut `style`

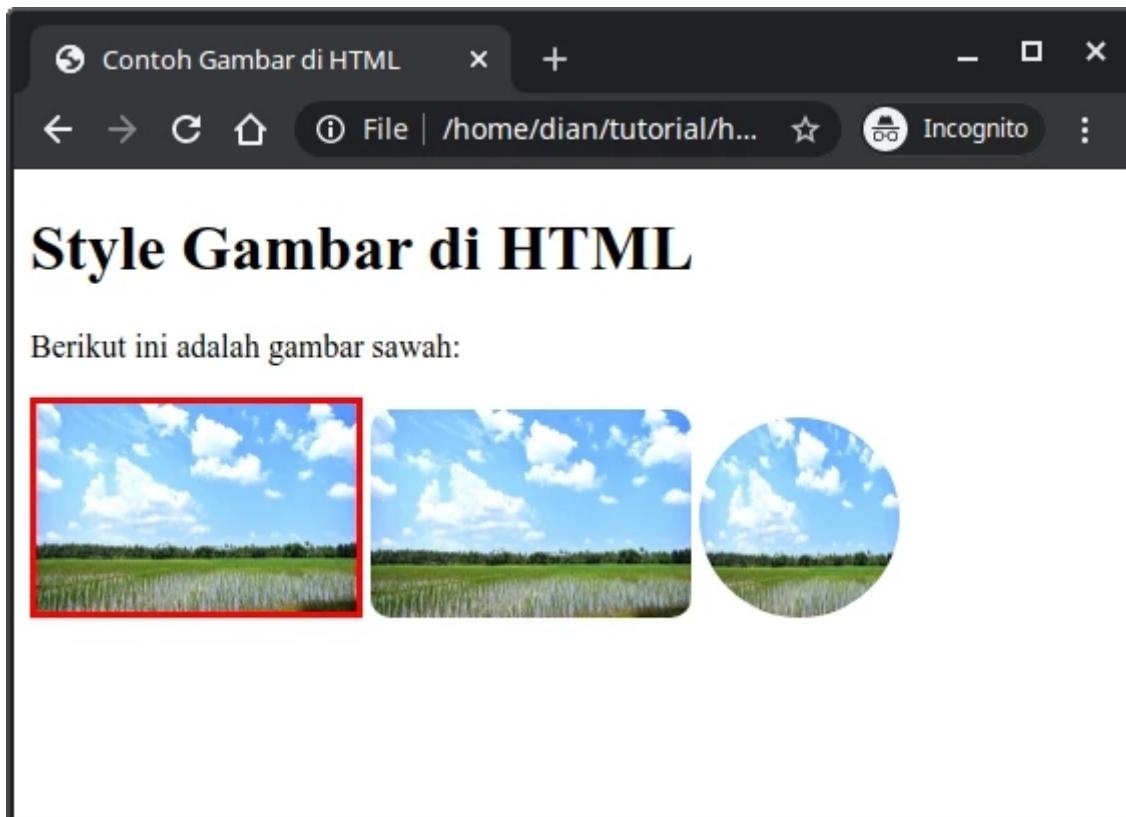
Atribut `style` merupakan atribut untuk menambahkan style CSS pada sebuah elemen. Atribut ini sering digunakan pada gambar untuk memberikan style atau efek tertentu.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>
<body>
  <h1>Style Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>    
        
  </p>
```

```
</body>  
</html>
```

Hasilnya:



Perhatikan gambar di atas ..

Gambar pertama kita berikan style garis (border) dengan ukuran `3px`, jenis solid, dan warnanya adalah merah.

Lalu gambar kedua, kita berikan style `border-radius` untuk menciptkan lengkungan pada pojok gambar.

Pada gambar ketiga, kita berikan nilai `100%` pada `border-radius` yang akan menciptakan lingkaran.

Membuat Gambar Background

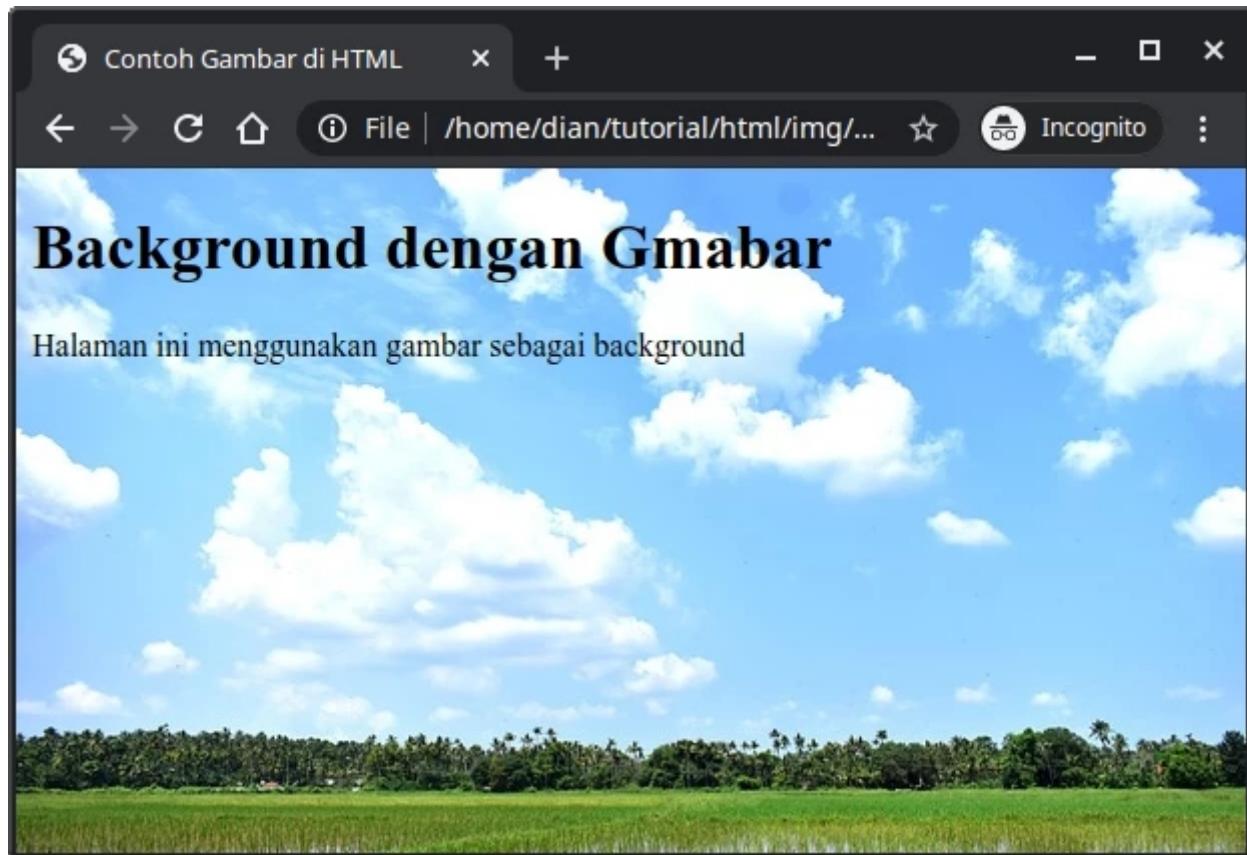
Sebenarnya ini sudah masuk ke dalam pembahasan tentang CSS, tapi tidak apa-apa kita bahas di sini.

Agar gambar bisa ditampilkan sebagai background, maka kita harus menggunakan CSS dengan atribut `background-image` lalu diisi dengan URL dari gambar.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head><body style="background-image: url(images/sawah.jpg);">  <h1>Background
dengan Gmabar</h1>
  <p>Halaman ini menggunakan gambar sebagai background</p>
</body>
</html>
```

Hasilnya:



Membuat Link dengan Gambar

Membuat gambar sebagai link pernah kita bahas di [Tutorial membuat link di HTML](#), tapi tidak apa-apa.. kita akan bahas lagi biar semakin paham.

Baiklah...

Cara membuat link dengan gambar adalah dengan menggabungkan tag [<a>](#) dan tag

Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contoh Gambar di HTML</title>
</head>
<body>
    <h1>Gambar Sebagai Link</h1>
    <p>Coba klik gambar ini:</p>
    <p>
        <a href="https://www.petanikode.com">
            
        </a>
    </p>
</body>
</html>

```

Hasilnya:

Membuat Link pada Bagian Gambar

Link dapat kita buat pada bagian tertentu pada gambar dengan menggunakan tag `<map>` dan `<area>`.

Contoh: [1](#)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contoh Gambar di HTML</title>
</head>
<body>
    <h1>Gambar Sebagai Link</h1>
    <p>Coba sentuh dan klik gambar ini:</p>
    <p>
        
        <map name="workmap">
            <area shape="rect" coords="34,44,270,350" title="Computer" href="#">
            <area shape="rect" coords="290,172,333,250" title="Phone" href="#">
            <area shape="circle" coords="337,300,44" title="Coffee" href="#">
        </map>
    </p>

```

```
</body>  
</html>
```

Hasilnya:

Tag Tambahan untuk Gambar di HTML

Pada HTML versi 5, ada beberapa tag tambahan yang bisa digunakan untuk membantu tag ``.

Diantaranya:

`<figure>` untuk membungkus gambar dan teks caption-nya;

`<picture>` untuk menentukan jenis gambar pada ukuran layar yang berbeda.

Mari kita coba:

Tag `<figure>`

Tag figure berfungsi untuk membungkus tag `` atau gambar dengan teks caption. Teks caption adalah teks yang menjelaskan tentang gambar. Teks caption bisa dibuat dengan tag `<figcaption>`.



petanikode.com

Contoh:

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

```
<meta charset="UTF-8">
<title>Contoh Gambar di HTML</title>
</head>
<body>
    <h1>Menampilkan Gambar di HTML</h1>
    <p>Berikut ini adalah gambar sawah:</p>
    <p>        <figure>        
<figcaption>Landscape sawah dan langit</figcaption>        </figure>    </p>
</body>
</html>
```

Hasilnya:



Tag `<picture>`

Di era mobile seperti saat ini, website tidak hanya dibuka melalui komputer saja, tapi juga smartphone.

Ukuran layar komputer dengan smartphone tentunya berbeda. Karena itu, website harus bersifat responsif agar bisa menyesuaikan diri dengan media yang digunakan.

Nah, tag `<picture>` hadir untuk mengatasi masalah ini. Tag ini berfungsi untuk menentukan gambar mana yang akan ditampilkan pada ukuran layar tertentu.

```

tag pembuka untuk
membuat picture
|
|   ukuran layar media yang
|   akan digunakan
|
<picture>
  <source media="(min-width: 650px)" srcset="sawah.jpg">
  <source media="(min-width: 465px)" srcset="sawah-sm.jpg">
  
</picture>

Jika tag <picture> tidak didukung
maka gambar akan ditampilkan
seperti biasa

```

alamat path atau URL
gambar yang akan ditampilkan

petanikode.com

Mari kita coba contohnya:

```

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

<head>
  <meta charset="UTF-8">
  <title>Contoh Gambar di HTML</title>
</head>

<body>
  <h1>Menampilkan Gambar di HTML</h1>
  <p>Berikut ini adalah gambar sawah:</p>
  <p>
    <picture>      <source media="(min-width: 650px)" srcset="images/sawah.jpg">
    <source media="(min-width: 450px)" srcset="images/sawah-sm.jpg">          </picture>
  </p>

</body>
</html>

```

Elemen List di HTML

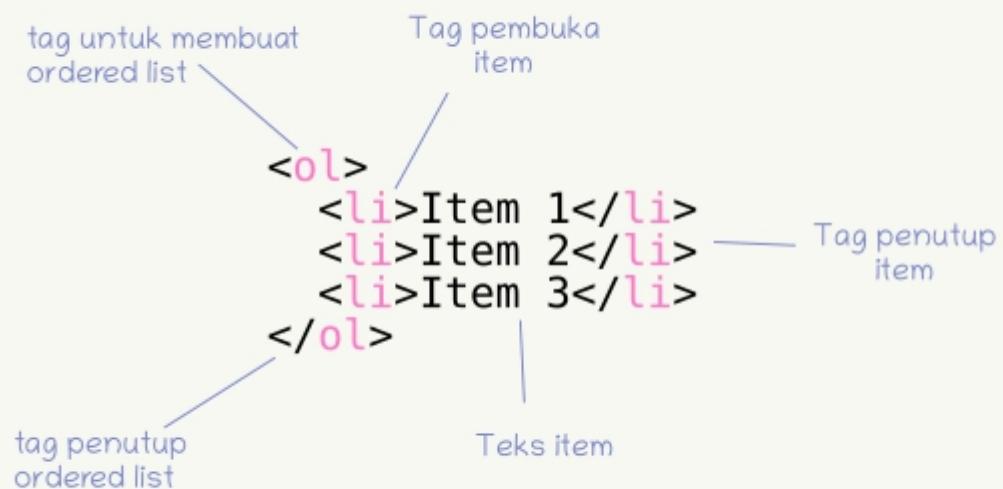
HTML sudah menyediakan elemen untuk membuat list. Ada tiga macam jenis list yang bisa dibuat di HTML:

1. **Ordered List** adalah list yang terurut;
2. **Unordered List** adalah list yang tak terurut;
3. dan **Descriptiona List** adalah list yang berisi definisi.

Mari kita bahas satu-per-satu...

1. Ordered List di HTML

Ordered list dibuat dengan tag ``. Lalu di dalamnya diisi dengan item-item yang akan dimasukkan ke dalam list. Item dibuat dengan tag `` (list item).



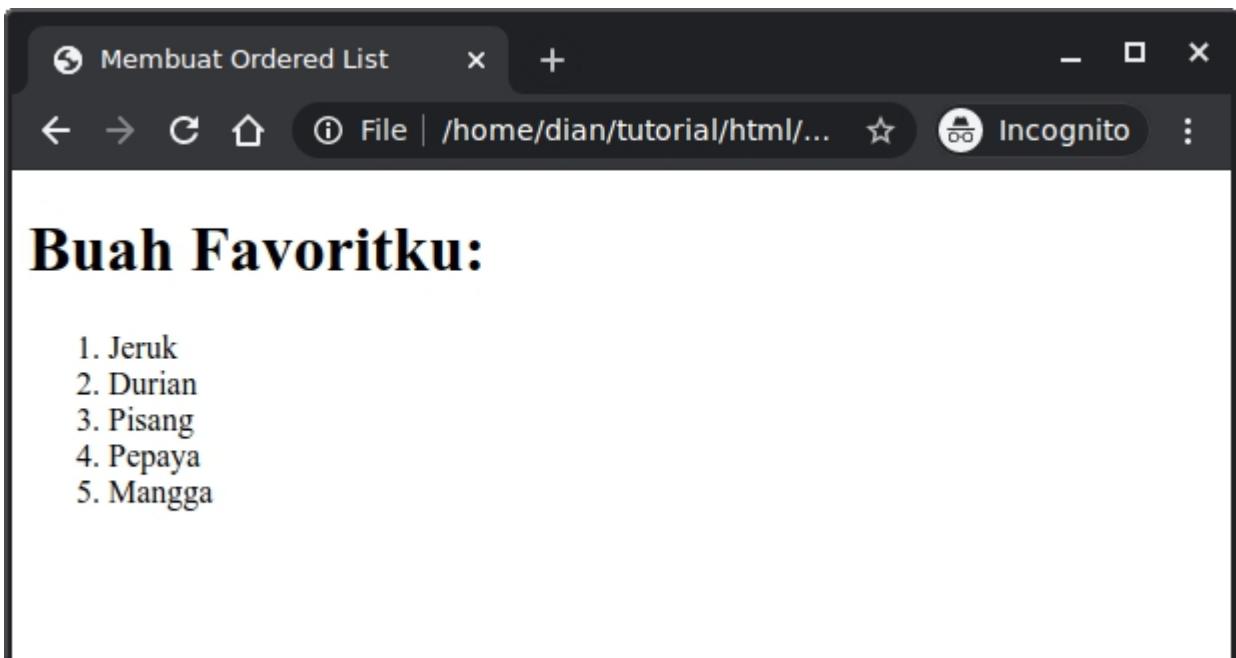
petanikode.com

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Membuat Ordered List</title>
</head>

<body>
    <h1>Buah Favoritku:</h1>      <ol>          <li>Jeruk</li>
    <li>Durian</li>           <li>Pisang</li>        <li>Pepaya</li>
    <li>Mangga</li>       </ol></body>
</html>
```

Hasilnya:



List diatas diurutkan berdasarkan angka dari 1, 2, 3, sampai 5.

Lalu bagaimana kalau kita ingin menggunakan huruf seperti a, b, c atau angka romawi seperti I, II, III?

Gampang..

Untuk membuat yang seperti itu, kita bisa menggunakan atribut `type` pada tag `ol` dan berikut ini nilai yang bisa diberikan:

- a untuk alfabet a, b, c, dan seterusnya;
- A untuk alfabet A, B, C, dan seterusnya;
- i untuk angka romwari i, ii, iii, dan seterusnya;
- I untuk angka romwari I, II, III, dan seterusnya.

Contoh:

```
<!DOCTYPE html>
<html>
<head>
    <title>Ordered List dengan Atribut Type</title>
</head>

<body>
    <h3>List dengan type alfabet</h3> <ol type='a'> <li>Tutorial List
di HTML</li>
        <li>Tutorial Link di HTML</li>
        <li>Tutorial Tabel di HTML</li>
    </ol>
```

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

Hasilnya:

A screenshot of a web browser window titled "Ordered List dengan Atribut". The address bar shows the file path: "/home/dian/tutorial/html/list/list-ol-type.html". The page content displays an ordered list with type "a".

List dengan type alfabet

- a. Tutorial List di HTML
- b. Tutorial Link di HTML
- c. Tutorial Tabel di HTML

List dengan type alfabet kapital (huruf besar)

- A. Tutorial List di HTML
- B. Tutorial Link di HTML
- C. Tutorial Tabel di HTML

List dengan type romawi

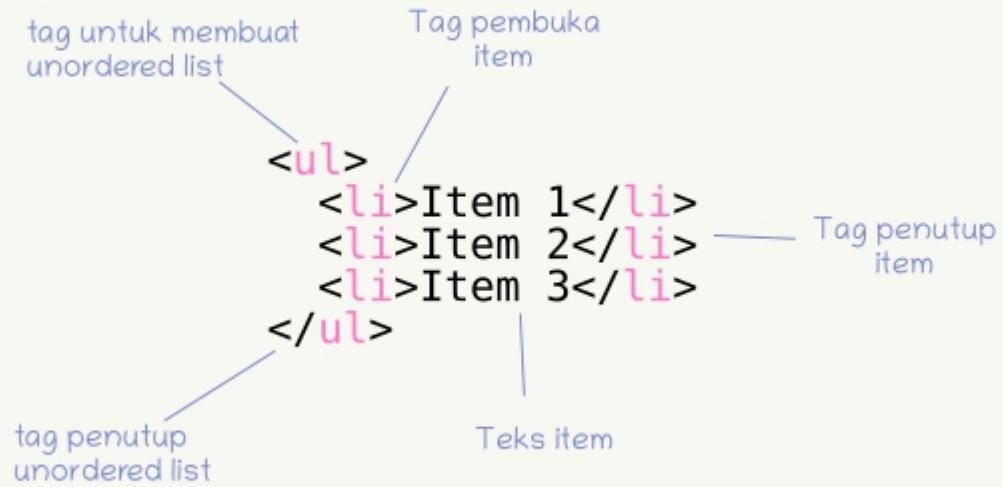
- i. Tutorial List di HTML
- ii. Tutorial Link di HTML
- iii. Tutorial Tabel di HTML

List dengan type romawi kapital

- I. Tutorial List di HTML
- II. Tutorial Link di HTML
- III. Tutorial Tabel di HTML

2.Unordered List di HTML

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



petanikode.com

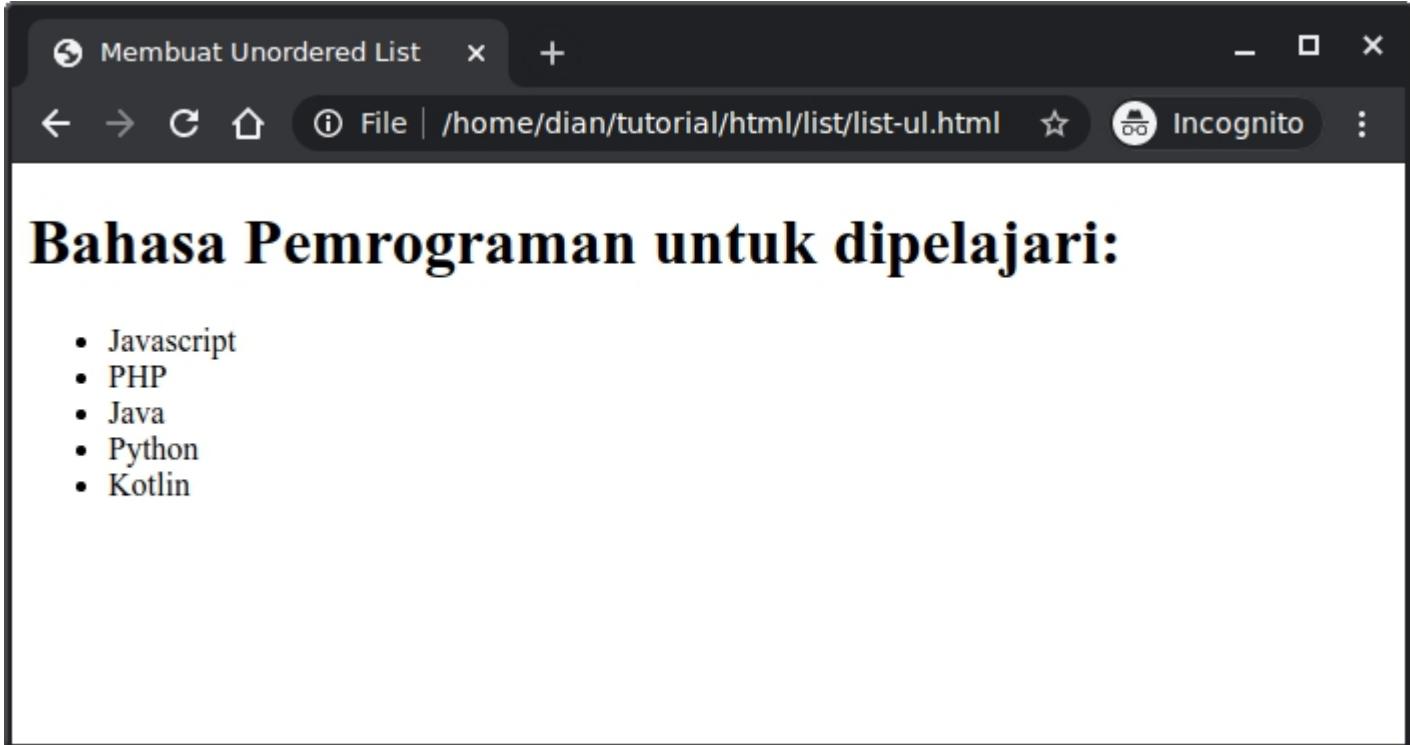
Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Membuat Unordered List</title>
</head>

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

Hasilnya:



Secara default simbol yang digunakan oleh unordered list adalah lingkaran kecil (disc). Kita juga bisa menggantinya dengan atribut **type**.

Berikut ini nilai yang bisa diberikan untuk atribut **type**:

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

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Membuat Unordered List</title>
</head>

<body>
    <h1>Bahasa Pemrograman untuk dipelajari:</h1>      <ul type="square">
<li>Javascript</li>
    <li>PHP</li>
    <li>Java</li>
    <li>Python</li>
    <li>Kotlin</li>
</ul>
    <h1>Framework untuk dipelajari:</h1>      <ul type="circle">
<li>Vuejs</li>
```

```
<li>Svelte</li>
<li>Reactjs</li>
</ul>
<h1>Tools untuk dipelajari:</h1> <ul type="none"> <li>Gulp</li>
    <li>NPM</li>
    <li>Js Lint</li>
</ul>
<h1>Pelajari juga:</h1> <ul type="disc"> <li>JSON</li>
    <li>XML</li>
    <li>Markdown</li>
</ul>
</body>
</html>
```

Hasilnya:

Bahasa Pemrograman untuk dipelajari:

- Javascript
- PHP
- Java
- Python
- Kotlin

Framework untuk dipelajari:

- Vuejs
- Svelte
- Reactjs

Tools untuk dipelajari:

Gulp
NPM
Js Lint

Pelajari juga:

- JSON
- XML
- Markdown

Selain menggunakan type, kita juga bisa menggunakan gambar.

Ini dilakukan dengan style CSS.

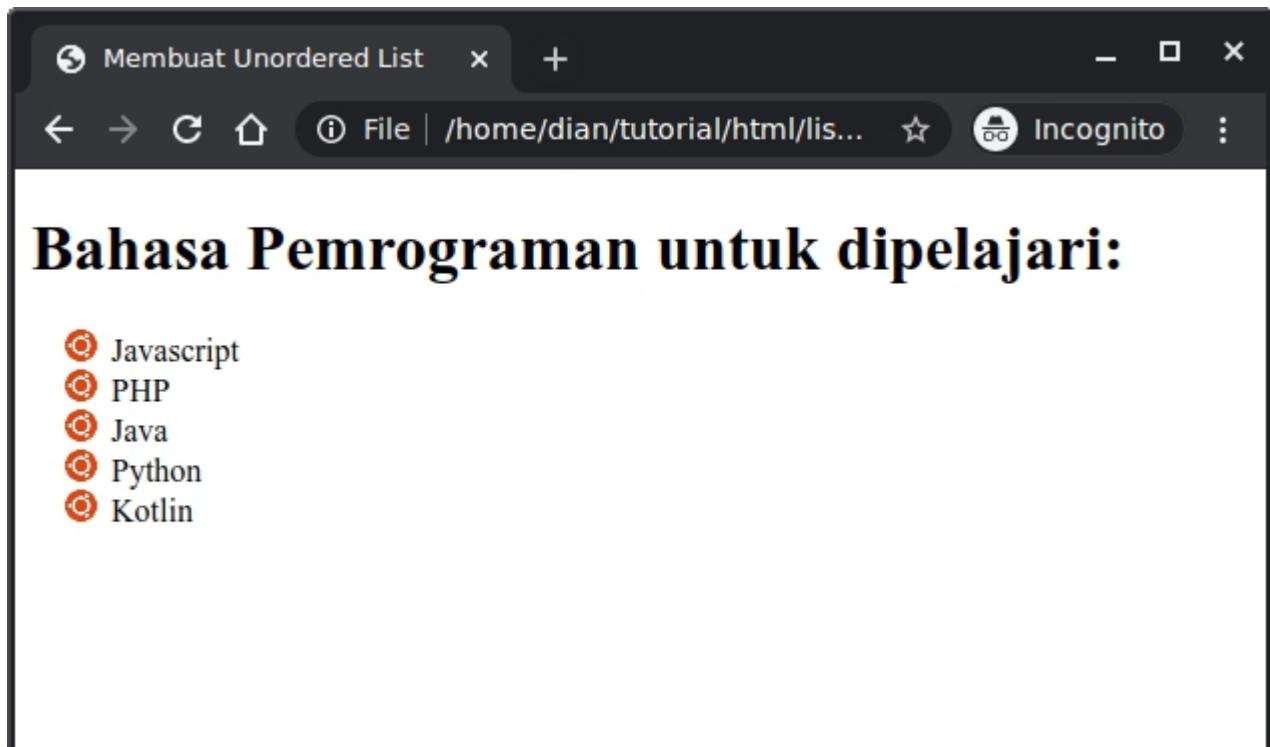
Contohnya:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Membuat Unordered List</title>
</head>

<body>
```

```
<h1>Bahasa Pemrograman untuk dipelajari:</h1>    <ul style="list-style-image: url("https://assets.ubuntu.com/sites/ubuntu/latest/u/img/favicon.ico")">
    <li>Javascript</li>
    <li>PHP</li>
    <li>Java</li>
    <li>Python</li>
    <li>Kotlin</li>
</ul>
</body>
</html>
```

Hasilnya:



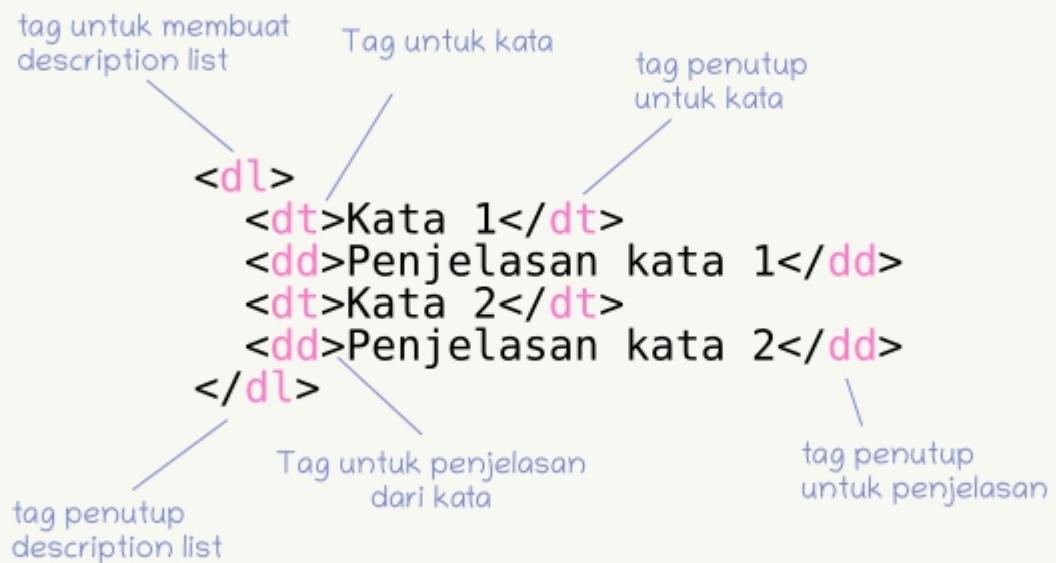
3. Description List di HTML

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

Ada tiga tag yang digunakan untuk membuat description list:

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

Format penulisannya seperti ini:



petanikode.com

Contoh:

```

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

<head>
    <title>Membuat Description List</title>
</head>

<body>
    <h1>Daftar istilah:</h1>
    <dl>
        <dt>Kopi</dt>
        <dd>Sebuah minuman berwarna hitam. Menurut pendapat lain kopi adalah air yang dimasak sampai gosong.</dd>
        <dt>Kopi Black Magic</dt>
        <dd>Kopi hitam atau kopi tradisional yang dibuat dengan mantra-mantra.</dd>
        <dt>White Coffee</dt>
        <dd>Kopi berwarna putih, kandungan kafeinnya sedikit.</dd>
        <dt>Kopi++</dt>
        <dd>Kopi ini mampu meningkatkan imajinasi 99 kali lipat.</dd>
    </dl>
</body>

</html>

```

Hasilnya:

The screenshot shows a web browser window with the title "Membuat Description List". The address bar indicates the file is located at "/home/dian/tutorial/html/lis...". The page content is a list of coffee types with their descriptions:

- Kopi**
Sebuah minuman berwarna hitam. Menurut pendapat lain kopi adalah air yang dimasak sampai gosong.
- Kopi Black Magic**
Kopi hitam atau kopi tradisional yang dibuat dengan mantra-mantra.
- White Coffee**
Kopi berwarna putih, kandungan kafeinnya sedikit.
- Kopi++**
Kopi ini mampu meningkatkan imajinasi 99 kali lipat.

List di dalam List (Nested List)

List juga dapat dibuat di dalam list, misalkan kita ingin menggabungkan ordered list dengan unordered list.

Caranya, list yang di dalam ditulis di dalam tag .

Contoh:

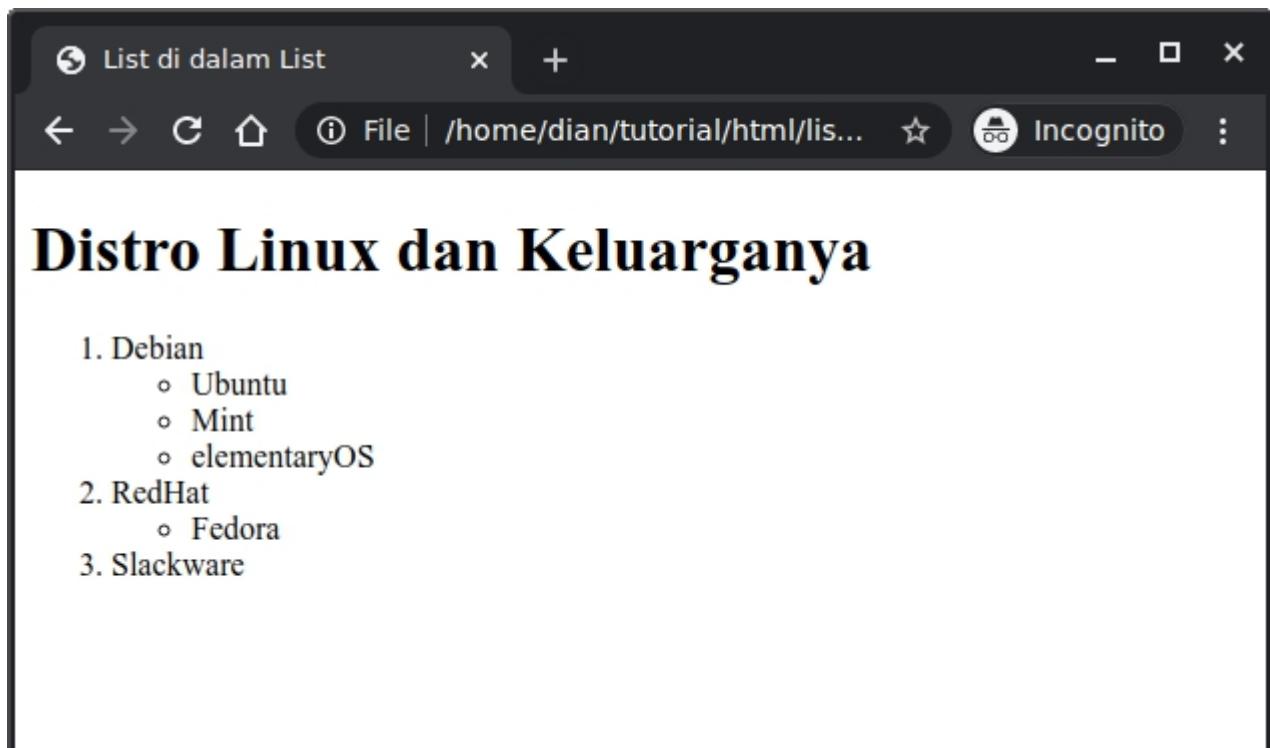
```
<!DOCTYPE html>
<html lang='en'>

<head>
    <title>List di dalam List</title>
</head>

<body>
    <h1>Distro Linux dan Keluarganya</h1>
    <ol>
        <li>Debian<ul><li>Ubuntu</li><li>Mint</li><li>elementaryOS</li></ul></li>
        <li>RedHat<ul><li>Fedora</li></ul></li>
        <li>Slackware</li>
    </ol>
</body>
```

```
</html>
```

Hasilnya:



Tag untuk Membuat Tabel di HTML

Ada beberapa tag yang harus diingat untuk membuat tabel di HTML:

1. Tag `<table>` untuk membungkus tabelnya
2. Tag `<thead>` untuk membungkus bagian kepala tabel
3. Tag `<tbody>` untuk membungkus bagian body dari tabel
4. Tag `<tr>` (tabel row) untuk membuat baris
5. Tag `<td>` (table data) untuk membuat sel
6. Tag `<th>` (table head) untuk membuat judul pada header

Tag yang paling penting untuk diingat adalah tag `<table>`, `<tr>`, dan `<td>`.

Sementara tag yang lain adalah tambahan (opsional), boleh digunakan boleh tidak.

Mari kita lihat contohnya:

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

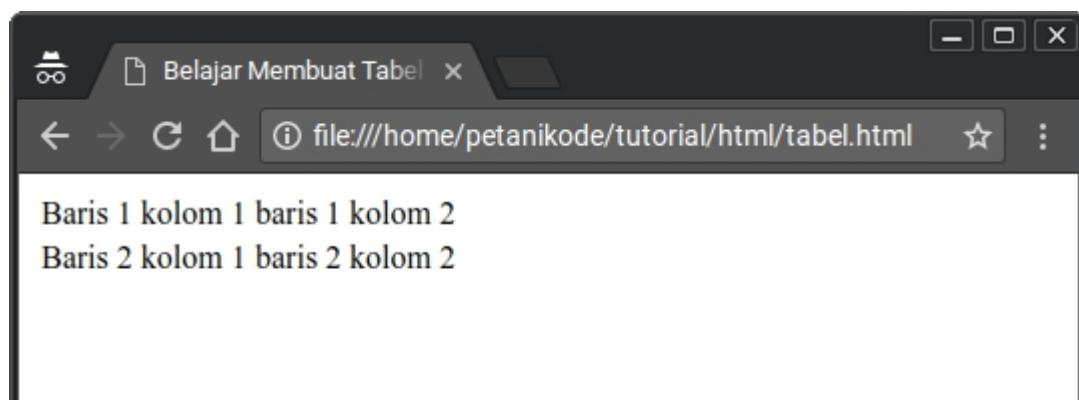
```

<table>
  <tr>
    <td>Baris 1 kolom 1</td>
    <td>baris 1 kolom 2</td>
  </tr>
  <tr>
    <td>Baris 2 kolom 1</td>
    <td>baris 2 kolom 2</td>
  </tr>
</table>

</body>
</html>

```

Hasilnya:



Kok tidak ada garisnya?

Iya, karena kita tidak menambahkan atribut `border` pada tabelnya.

Agar Tabelnya memiliki garis, silahkan tambahkan atribut `border="1"` di dalam tag `<table>`.

Sehingga akan menjadi seperti ini:

```

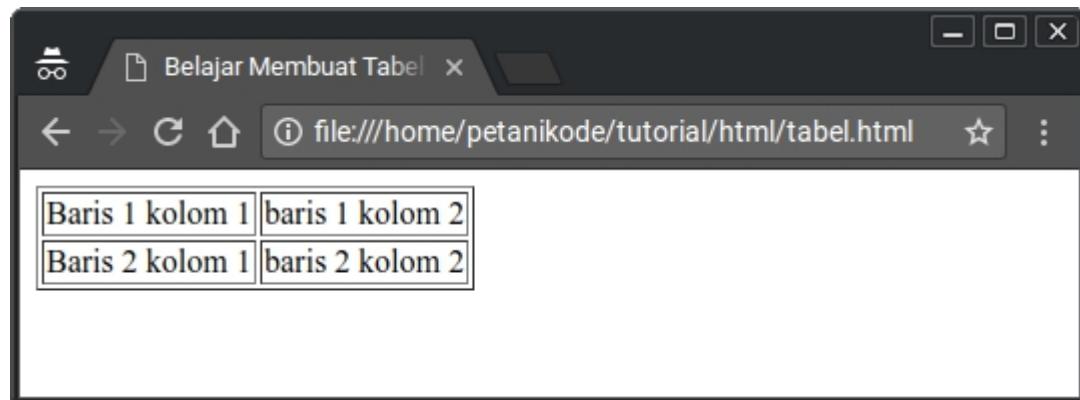
<table border="1">
  <tr>
    <td>Baris 1 kolom 1</td>
    <td>baris 1 kolom 2</td>
  </tr>
  <tr>
    <td>Baris 2 kolom 1</td>
    <td>baris 2 kolom 2</td>
  </tr>
</table>

```

Nilai "1" pada atribut **border** adalah ukuran garisnya. Semakin besar ukurannya, maka akan semakin besar pula ukuran garisnya.

Nilai "1" adalah ukuran garis yang paling kecil.

Hasilnya akan seperti ini:



Mengatur Jarak Sel dengan Cellpadding

Atribut **cellpadding** adalah atribut untuk mengatur jarak teks dengan garis di dalam sel.

Atribut ini dapat kita berikan kepada tag **<table>**.

Contoh:

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

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

Maka hasilnya:

Baris 1 kolom 1	baris 1 kolom 2
Baris 2 kolom 1	baris 2 kolom 2

Menambahkan Warna pada Sel dan Baris

Untuk menambahkan warna pada sel dan baris, kita bisa menambahkan atribut **bgcolor** di dalam tag **<td>** (untuk sel) atau **<tr>** (untuk baris).

Contoh:

```
<table border="1" cellpadding="10">
  <tr>
    <td bgcolor="yellow">Baris 1 kolom 1</td>
    <td>baris 1 kolom 2</td>
  </tr>
  <tr bgcolor="#00ff80">
    <td>Baris 2 kolom 1</td>
    <td>baris 2 kolom 2</td>
  </tr>
</table>
```

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

Maka hasilnya akan seperti ini:

Baris 1 kolom 1	baris 1 kolom 2
Baris 2 kolom 1	baris 2 kolom 2

Menggabungkan Sel Tabel

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

rowspan untuk menggabungkan baris;
colspan untuk menggabungkan kolom.

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

The diagram shows a 4x3 grid of cells. The first column has two light gray cells. The second column has three cells: a dark purple one at the top labeled "rowspan = \"2\"", a white one below it, and another white one at the bottom. The third column has two white cells. Handwritten annotations explain: "sel ini telah bergabung secara colspan" points to the second column's first cell; "sel ini telah bergabung secara rowspan" points to the second column's first cell; and "colspan = \"2\"" is written above the second column's first cell.

Mari kita lihat contohnya:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Belajar Membuat Tabel HTML</title>
    </head>
    <body>
        <table border="1">
            <tr>
                <th rowspan="2" bgcolor="yellow">Bulan</th>
                <th colspan="2" bgcolor="#00ff80">Hasil Panen</th>
            </tr>
            <tr>
                <th>Padi</th>
                <th>Kacang</th>
            </tr>
            <tr>
                <td>Januari</td>
                <td>500 Kg</td>
                <td>231 Kg</td>
            </tr>
            <tr>
```

```

<td>Februari</td>
<td>342 Kg</td>
<td>423 Kg</td>
</tr>
<tr>
    <td>Maret</td>
    <td>432 Kg</td>
    <td>124 Kg</td>
</tr>
<tr>
    <td>April</td>
    <td>453 Kg</td>
    <td>523 Kg</td>
</tr>
</table>

</body>
</html>

```

Hasilnya:

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

Menyisipkan Elemen yang Lain ke dalam Sel

Di salam sel `<td>` dan `<th>`, kita bisa menyisipkan elemen HTML yang lain, seperti gambar, link, video, `list`, dsb.

Contoh:

```

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

```

```
<body>
  <table border="1">
    <tr>
      <th colspan="3" bgcolor="yellow">Produk Unggulan</th>
    </tr>
    <tr>
      <td rowspan="4" width="200" />
      
    </tr>
    <tr>
      <td>Nama</td>
      <td>Benih Kode</td>
    </tr>
    <tr>
      <td>Harga</td>
      <td>Rp 192.000</td>
    </tr>
    <tr>
      <td>Fitur</td>
      <td>
        <ul>
          <li>Dilengkapi Dokumentasi</li>
          <li>Ukuran: 31MB</li>
          <li>Masa Tanam: 6 Bulan</li>
          <li>Lisensi: MIT</li>
        </ul>
      </td>
    </tr>
  </table>

</body>
</html>
```

Hasilnya:

Produk Unggulan	
	Nama Benih Kode
Harga Rp 192.000	
Fitur	<ul style="list-style-type: none"> • Dilengkapi Dokumentasi • Ukuran: 31MB • Masa Tanam: 6 Bulan • Lisensi: MIT

Cara Membuat Form di HTML

Form di HTML dapat kita buat dengan tag `<form>`.

Tag ini memiliki beberapa atribut yang harus diberikan, seperti:

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

Contoh:

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

Untuk atribut `action`, kita dapat mengisinya dengan alaman URL dari endpoint yang akan memproses form.

Secara sederhana,—pada contoh di atas— kita akan menyuruh file `prosess.php` untuk memproses data form.

Ini nanti akan kita pelajari pada PHP.

Kode HTML di atas, tidak akan menghasilkan apa-apa.

Karena kita belum membuat field-nya.

Apa itu Field?

Field adalah ruas yang dapat diisi dengan data.

Contoh field:

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

Field memiliki beberapa atribut yang harus diberikan:

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

Latihan: Membuat Form Login

Sebagai latihan, mari kita buat form login.

Pada form login, terdapat beberapa field dan elemen:

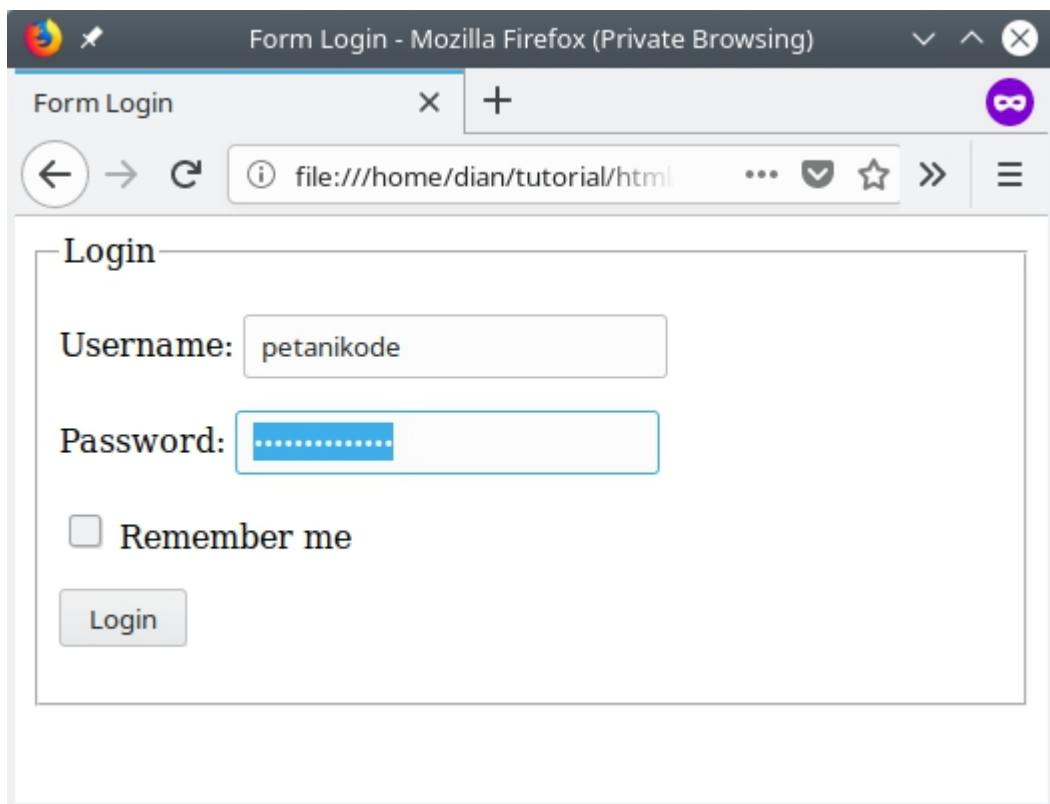
1. Field untuk input username atau email;
2. Field untuk input password;
3. Checkbok untuk remember me;
4. Tombol untuk login.

Berikut ini kodennya:

```
<!DOCTYPE html>
<html>
<head>
    <title>Form Login</title>
</head>
<body>
    <form action="login.php" method="POST">
        <fieldset>
            <legend>Login</legend>
            <p>
                <label>Username:</label>
                <input type="text" name="username" placeholder="username..." />
            </p>
            <p>
                <label>Password:</label>
                <input type="password" name="password" placeholder="password..." />
            </p>
            <p>
                <label><input type="checkbox" name="remember" value="remember" />
Remember me</label>
            </p>
    </form>
</body>
</html>
```

```
<p>
    <input type="submit" name="submit" value="Login" />
</p>
</fieldset>
</form>
</body>
</html>
```

Hasilnya:



Sekarang perhatikan!

Pada kode di atas, kita membuat empat buah filed:

1. input `username` dengan tipe `text`;
2. input `password` dengan tipe `password`;
3. input `remember` dengan tipe `checkbox`;
4. input `submit` dengan tipe `submit`;

Lalu ketiga filed ini dibungkus ke dalam tag `<fieldset>`.

Nanti tag `<fieldset>` ini akan membuat sebuah garis.

Di dalam tag `<fieldset>`, kita membuat tag `<legend>` untuk memberikan teks pada fieldset.

Lalu, perhatikan juga atirbut yang digunakan pada setiap field.

Atribut `placeholder` untuk menampilkan teks sementara (placeholder);
Atribut `value` untuk memberikan nilai default pada field.

Setiap field kita bungkus dalam tag `<p>` agar terlihat rapi dan juga kita berikan sebuah label dengan tag `<label>`.

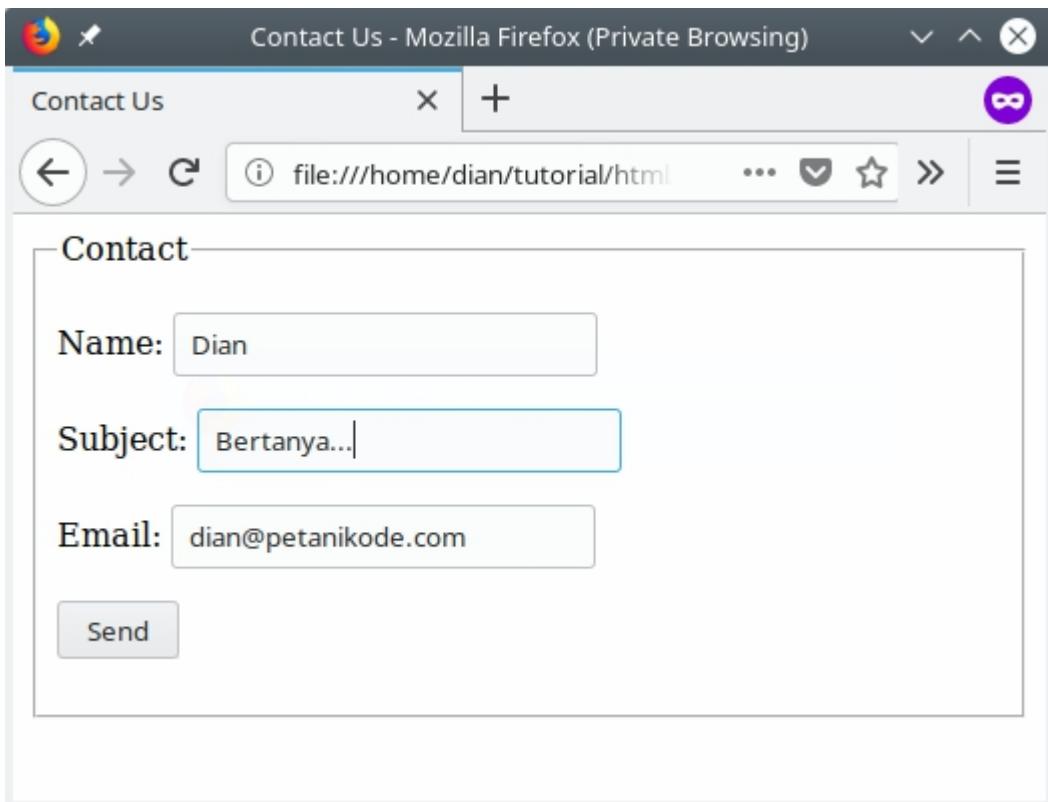
Latihan: Membuat Form Contact

Latihan Selanjutnya, kita akan membuat form contact. Form ini berfungsi untuk menghubungi atau kontak admin.

Silahkan ikuti kode berikut:

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Us</title>
</head>
<body>
    <form action="contact.php" method="POST">
        <fieldset>
            <legend>Contact</legend>
            <p>
                <label>Name:</label>
                <input type="text" name="name" placeholder="your name..." />
            </p>
            <p>
                <label>Subject:</label>
                <input type="text" name="subject" placeholder="subject..." />
            </p>
            <p>
                <label>Email:</label>
                <input type="email" name="email" placeholder="your email..." />
            </p>
            <p>
                <input type="submit" name="submit" value="Send" />
            </p>
        </fieldset>
    </form>
</body>
</html>
```

Hasilnya:



Contact Us - Mozilla Firefox (Private Browsing)

Contact Us

Name: Dian

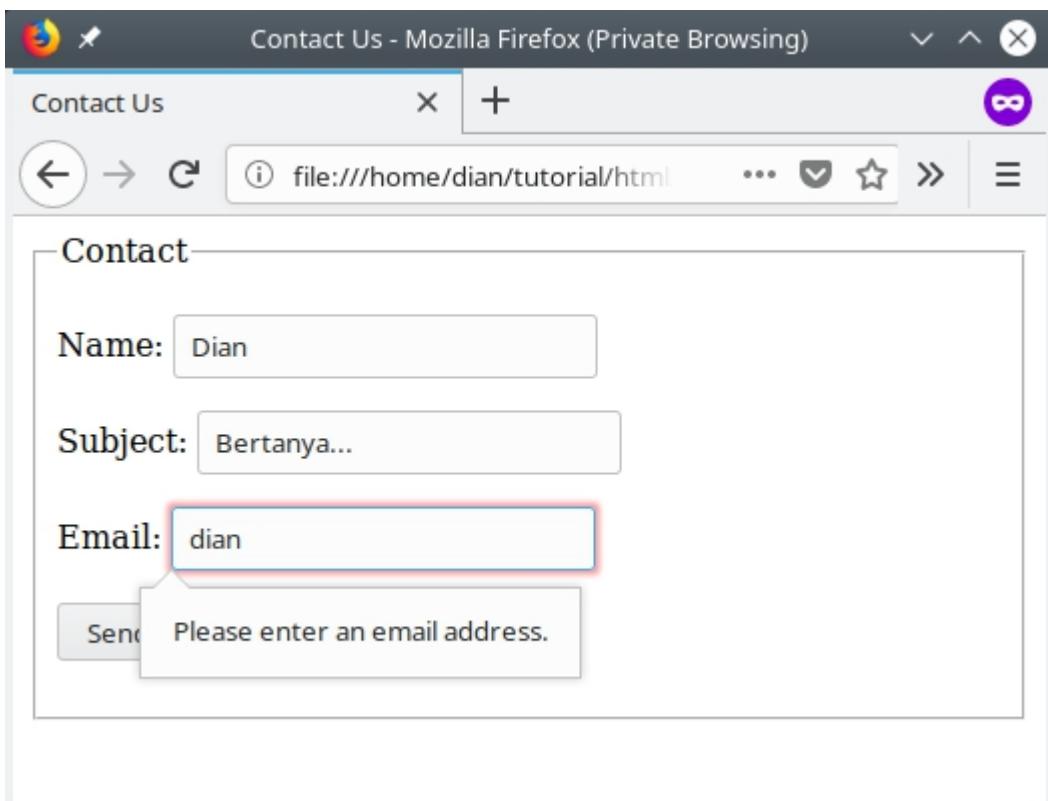
Subject: Bertanya...

Email: dian@petanikode.com

Send

Pada contoh di atas, kita memberikan `type="email"` untuk field `email`, agar field ini harus diisi dengan email saja.

Coba saja isi dengan yang lain, lalu klik **Send**...maka akan muncul pesan peringatan.



Contact Us - Mozilla Firefox (Private Browsing)

Contact Us

Name: Dian

Subject: Bertanya...

Email: dian

Send Please enter an email address.

Latihan: Membuat Form Register

Semakin banyak latihan, semakin bagus.

Berikutnya kita akan coba membuat form registrasi.

Form ini berisi field untuk:

Input nama lengkap;
Input username;
Input email;
Input password;
Input jenis kelamin;
Input Agama;
Input Biografi.
dsb.

Mari kita buat...

```
<!DOCTYPE html>
<html>
<head>
    <title>Registrasi</title>
</head>
<body>
    <form action="contact.php" method="POST">
        <fieldset>
            <legend>Registrasi</legend>
            <p>
                <label>Nama:</label>
                <input type="text" name="nama" placeholder="Nama lengkap..." />
            </p>
            <p>
                <label>Username:</label>
                <input type="text" name="username" placeholder="Username..." />
            </p>
            <p>
                <label>Email:</label>
                <input type="email" name="email" placeholder="Your email..." />
            </p>
            <p>
                <label>Password:</label>
                <input type="password" name="password" placeholder="Passowrd..." />
            </p>
            <p>
```

```
<label>Jenis kelamin:</label>
    <label><input type="radio" name="jenis_kelamin" value="laki-laki">
/> Laki-laki</label>
    <label><input type="radio" name="jenis_kelamin" value="perempuan">
/> Perempuan</label>
</p>
<p>
    <label>Agama:</label>
    <select name="agama">
        <option value="islam">Islam</option>
        <option value="kristen">Kristen</option>
        <option value="hindu">Hindu</option>
        <option value="budha">Budha</option>
    </select>
</p>
<p>
    <label>Biografi:</label>
    <textarea name="biografi"></textarea>
</p>
<p>
    <input type="submit" name="submit" value="Daftar" />
</p>
</fieldset>
</form>
</body>
</html>
```

Hasilnya:

The screenshot shows a Mozilla Firefox browser window with a private browsing session. The title bar reads "Registrasi - Mozilla Firefox (Private Browsing)". The address bar shows the URL "file:///home/dian/tutorial/html/form/". The main content area displays a registration form with the following fields:

- Nama:** Petani Kode
- Username:** petanikode
- Email:** info@petanikode.com
- Password:** (redacted)
- Jenis kelamin:** Laki-laki Perempuan
- Agama:** Islam
- Biografi:** Petani kode adalah sebuah blog pemrograman yang menggunakan linux

A "Daftar" button is located at the bottom left of the form.

Apa saja field baru yang ada di form tersebut?

1. Field **radio**;
2. Field **<select><option>**;
3. Field **<textarea>**.

Apa bedanya **radio** dengan **checkbox**?

Jika kita ingin agar pengunjung memilih salah satu, maka kita gunakan **radio**.

Tapi kalau kita ingin pengunjung memilih lebih dari satu, maka kita gunakan **checkbox**.

Lalu untuk **<select></option>**, sifatnya sama seperti **radio**. Cuma dia bentuknya berbeda.

Lalu untuk menginputkan teks yang panjang, kita gunakan tag **<textarea>**.

Latihan: Membuat Form Tingkat Lanjut

Field-field di atas merupakan jenis field yang sering digunakan dalam pembuatan form.

Masih ada jenis field lagi yang belum kita coba, seperti `meter`, `color`, `url`, `number`, `date`, `datetime`, dsb.

Mari kita coba beberapa:

```
<!DOCTYPE html>
<html>
<head>
    <title>Form HTML</title>
</head>
<body>
    <form action="contact.php" method="POST">
        <fieldset>
            <legend>Form</legend>
            <p>
                <label>Alamat Web:</label>
                <input type="url" name="name" placeholder="Masukan URL Web..." />
            </p>
            <p>
                <label>Tanggal Lahir:</label>
                <input type="date" name="tanggal" />
            </p>
            <p>
                <label>Umur:</label>
                <input type="number" min="10" max="90" name="umur" />
            </p>
            <p>
                <input type="submit" name="submit" value="Send" />
            </p>
        </fieldset>
    </form>
</body>
</html>
```

Hasilnya:

Form HTML - Mozilla Firefox (Private Browsing)

Form HTML

Alamat Web: <https://www.petanikode.com>

Tanggal Lahir:

Umur:

Send

A calendar modal is displayed over the form fields. The calendar shows the month of September 2018. The date **17** is highlighted with a gray background, indicating it is the selected date.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Apabila di browser anda tidak tampil kalender seperti di atas, coba gunakan browser versi terbaru.

Apa itu Elemen Semantik?

Jadi gini..

Di awal-awal hadirnya HTML dulu, elemen semantik belum ada.

Orang-orang membuat layout dengan menggunakan **tag yang salah**.

Ada yang membuatnya dengan tag **<table>** dan ada juga dengan tag **<div>**.

Ini sebenarnya tidak sepenuhnya salah, karena membuat layout dengan kedua tag itu benar-benar bisa.

Tapi..

Ini bukanlah cara yang baik dan akan membuat kode HTML kita sulit terbaca.

Karena itu.. hadirlah elemen semantik sebagai solusi.

Elemen semantik mulai ditambahkan pada HTML 5.

Elemen semantik adalah elemen-elemen yang **menyatakan makna** atau tujuan dari elemen itu sendiri. [1](#)

Misalnya tag `<footer>`, tag ini digunakan untuk membuat elemen footer atau bagian kaki dari web.

Jangan gunakan tag ini di bagian paling atas, karena maknanya sudah jelas untuk footer.

Jadi tidak akan ada lagi yang namanya penyalahgunaan tag. Karena setiap tag sudah punya tujuan masing-masing.

Berikut ini daftar elemen-elemen semantik:

- `<article>` — untuk membuat elemen artikel;
- `<aside>` — untuk membuat elemen bagian samping;
- `<details>` — untuk membuat elemen datail atau spoiler;
- `<figcaption>` — untuk membuat teks caption pada figure;
- `<figure>` — untuk membuat figur atau gambar pada artikel;
- `<footer>` — untuk membuat elemen bagian kaki dari web;
- `<header>` — untuk mebuat kepala kop dari web;
- `<main>` — untuk membuat elemen utama;
- `<mark>` — untuk menandai teks;
- `<nav>` — untuk membuat navigasi;
- `<section>` — untuk membuat bagian artikel;
- `<summary>` — untuk membuat ringkasan artikel atau isi spoiler;
- `<time>` — untuk membuat elemen yang menyatakan waktu;

dan masih banyak lagi.

Mengapa Harus Pakai Elemen Semantik?

Salah satu keuntungan menggunakan elemen semantik adalah dokumen HTML kita akan **mudah dibaca**, baik itu oleh manusia maupun mesin.

Coba perhatikan kode berikut:

```
<div id="header"></div>
<div class="section">
    <div class="article">
        <div class="figure">
            <img>
            <div class="figcaption"></div>
        </div>
    </div>
```

```
</div>
<div id="footer"></div>
```

Ini adalah contoh layout yang dibuat dengan tag `<div>`. Tag ini memang bagus untuk membungkus elemen HTML.

Kamu mungkin akan mudah paham dengan membaca nama-nama class yang diberikan pada elemen `<div>`.

Ada `<div>` yang bertugas untuk membuat elemen header, article, footer, dan sebagainya.

Sekilas, tidak ada masalah dengan ini..

Tapi nanti kalau elemen `<div>` nya sudah banyak, kita akan kesulitan membacanya.

Apa apaan ini??



```
<div>
  <div>
    <div>
      <div>
        <div>
          <div>
            <div>
              <div>
                <div>
                  <div>
                    <div>
                      <div>
                        Petanikode
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Sekarang coba bandingkan dengan kode ini:

```
<header></header>
<section>
  <article>
    <figure>
      <img>
      <figcaption></figcaption>
    </figure>
  </article>
</section>
<footer></footer>
```

Ini tentunya akan lebih mudah dibaca.

Oh iya, elemen semantik juga **bagus untuk SEO**. Jadi kalau mau websitemu disukai mesin pencari, sebaiknya gunakanlah elemen ini.

Membuat Layout dengan Elemen Semantik

Nah, sekarang mari kita belajar membuat layout dengan elemen semantik.

Buatlah file baru dengan nama `layout.html`, kemudian isi dengan kode berikut:

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

<head>
  <meta charset="UTF-8">
  <title>Contoh Layout dengan Elemen Semantik</title>
</head>

<body>

  <header>
    <h1>Belajar Elemen Semantik di HTML</h1>
  </header>

  <nav>
    <a href="#">Home</a> |
    <a href="#">About</a> |
    <a href="#">Contact</a>
  </nav>

  <article>
    <h1>Tutorial Semantik Elemen untuk Pemula</h1>
    <p>Semantik elemen adalah elemen yang memiliki makna dan tujuan.  

      Tujuannya agar kode HTML mudah dibaca dan tidak ada penyalahgunaan tag.  

      Elemen semantik bagus untuk SEO dan juga dapat meningkatkan accessibility</p>
  </article>

  <footer>
    Copyright © 2020 by Petani Kode
  </footer>

</body>

</html>
```

Setelah itu, buka dengan web browser.

Maka hasilnya:



The screenshot shows a web browser window titled "Contoh Layout dengan Elemen Semantik". The address bar indicates the file is located at "/home/dian/tutorial/html/semantik/layout.html". The page content is as follows:

Belajar Elemen Semantik di HTML

[Home](#) | [About](#) | [Contact](#)

Tutorial Semantik Elemen untuk Pemula

Semantik elemen adalah elemen yang memiliki makna dan tujuan. Tujuannya agar kode HTML mudah dibaca dan tidak ada penyalahgunaan tag. Elemen semantik bagus untuk SEO dan juga dapat meningkatkan accessibility.

Copyright © 2020 by Petani Kode

Hasilnya akan terlihat biasa saja dan sama seperti contoh-contoh sebelumnya. Ini karena kita belum memberikan style CSS.

Karena itu, mari kita coba memberikan style CSS untuk elemen semantik.

Style untuk Elemen Semantik

Cara memberikan style untuk elemen semantik sama saja seperti memberikan style pada elemen lainnya. Tinggal membuat atribut `style`, lalu mengisinya dengan kode style CSS.

Oke, sekarang coba ubah contoh yang tadi menjadi seperti ini:

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

<head>
  <meta charset="UTF-8">
  <title>Contoh Layout dengan Elemen Semantik</title>
</head>
<body style="background-color: silver;">  <header style="background-color: white;">    <h1>Belajar Elemen Semantik di HTML</h1>
  </header>

  <nav>
    <a href="#">Home</a> |
    <a href="#">About</a> |
```

```

<a href="#">Contact</a>
</nav>
<article style="background-color: white;">      <h1>Tutorial Semantik Elemen
untuk Pemula</h1>
    <p>Semantik elemen adalah elemen yang memiliki makna dan tujuan.
        Tujuannya agar kode HTML mudah dibaca dan tidak ada penyalahgunaan tag.
        Elemen semantik bagus untuk SEO dan juga dapat meningkatkan accessibility
    </p>
</article>
<footer style="background-color: white;">      Copyright © 2020 by Petani
Kode
</footer>

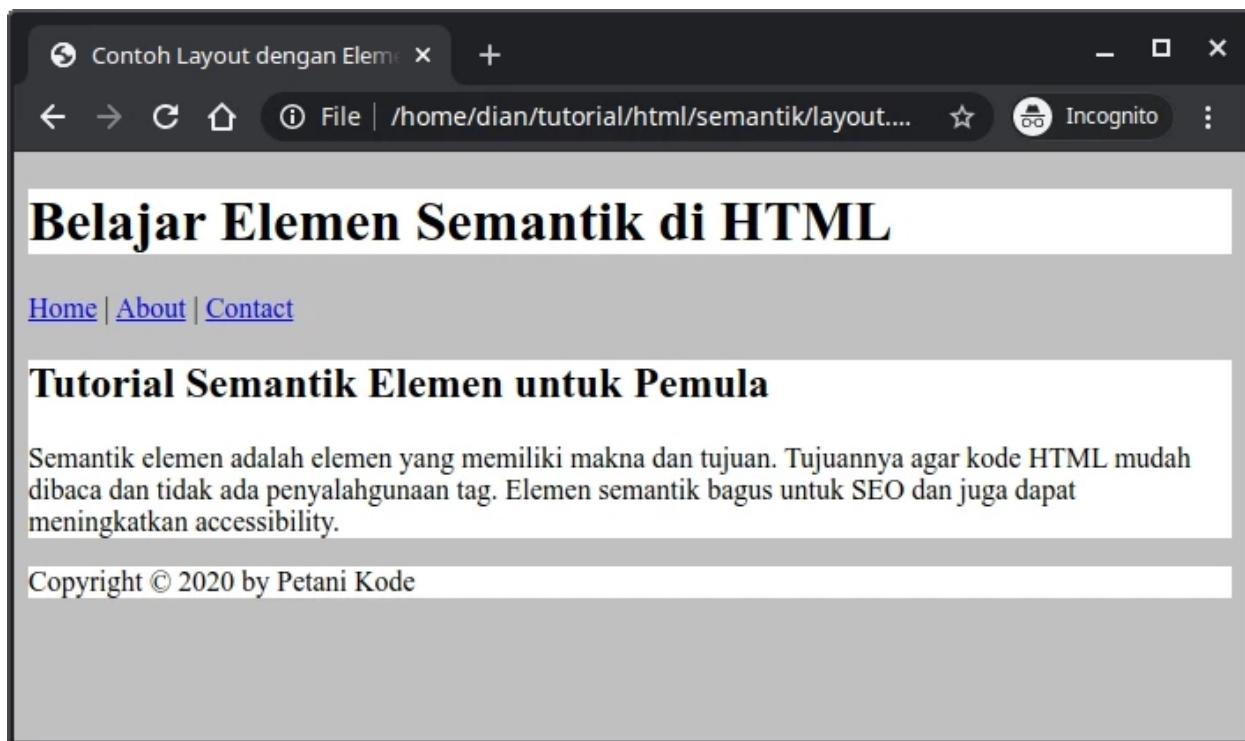
</body>

</html>

```

Kita memberikan warna latar untuk elemen body dan juga beberapa elemen semantik.

Maka hasilnya:



Mencoba Elemen Semantik Lainnya

Tidak semua elemen semantik bertujuan untuk membuat layout, ada juga beberapa elemen semantik lainnya yang bertujuan untuk membuat elemen tertentu.

Mari kita pelajari elemen semantik lainnya..

Elemen <detail> dan <summary>

Tag <detail> dan <summary> merupakan tag untuk membuat elemen spoiler.

Tag <detail> akan berisi semua detail konten, lalu tag <summary> akan menjadi tombol yang bisa diklik untuk menampilkan detail isinya.



petanikode.com

Contoh:

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

<head>
  <meta charset="UTF-8">
  <title>Contoh Elemen Semantik</title>
</head>

<body>

  <article>
    <h1>Laptop Terbaik untuk Programmer</h1>
    <details>      <summary>Lihat Spesifikasi</summary>      Prosesor: Intel
Core i9, RAM 32GB, SSD 1TB, HDD 4TB      </details>  </article>

</body>

</html>
```

Hasilnya:

Elemen <time>

Tag <time> merupakan tag untuk membuat elemen waktu. Elemen waktu biasanya dibutuhkan untuk menampilkan waktu.

Misalnya:

Waktu saat sebuah postingan dibut, waktu saat sebuah pesan dibaca, waktu keberangkatan, dan lain sebagainya.

Contoh:

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

<head>
  <meta charset="UTF-8">
  <title>Contoh Elemen Semantik</title>
</head>

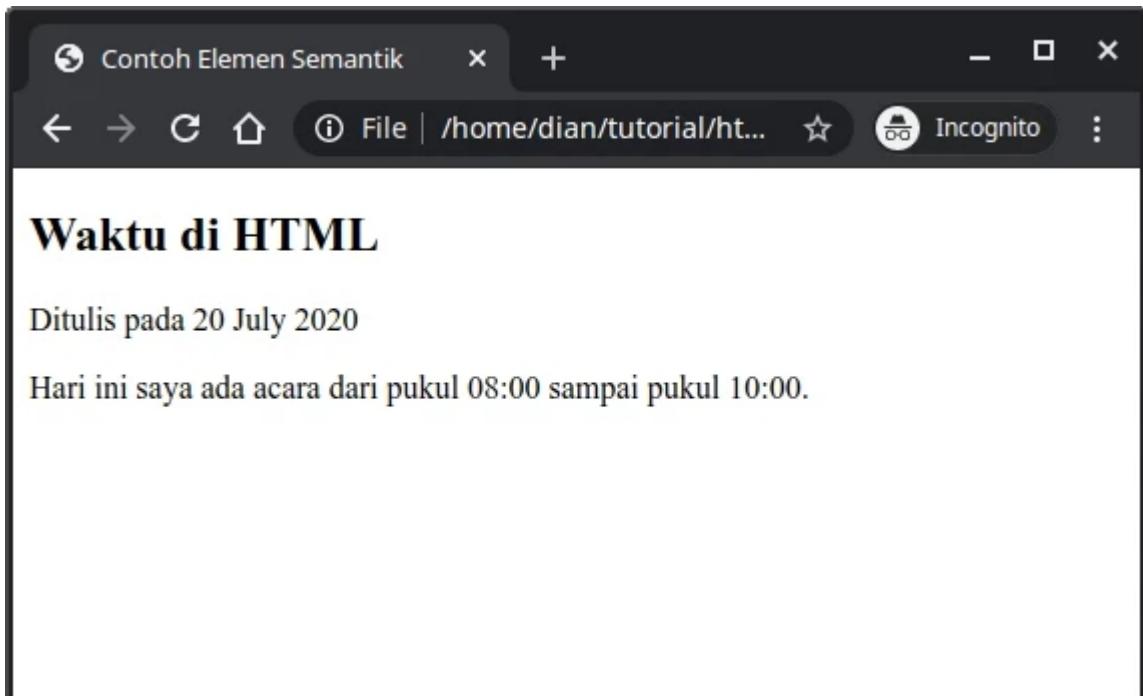
<body>

  <article>
    <h1>Waktu di HTML</h1>

    <p> Ditulis pada <time datetime="2020-20-07">20 July 2020</time>
  </p>
    <p> Hari ini saya ada acara dari pukul <time>08:00</time> sampai pukul
<time>10:00</time>.</p>
  </article>

</body>
</html>
```

Hasilnya:



Elemen `<time>` akan ditampilkan apa adanya. Atribut `datetime` berfungsi untuk memberikan nilai tanggal dan waktu yang nantinya akan dibaca oleh program.

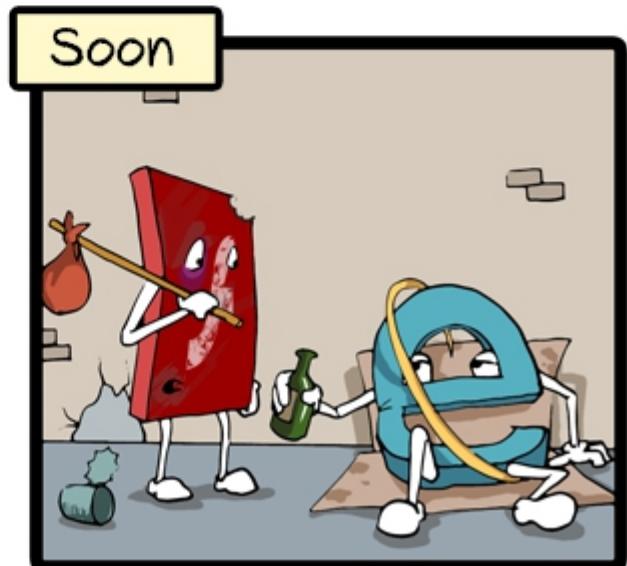
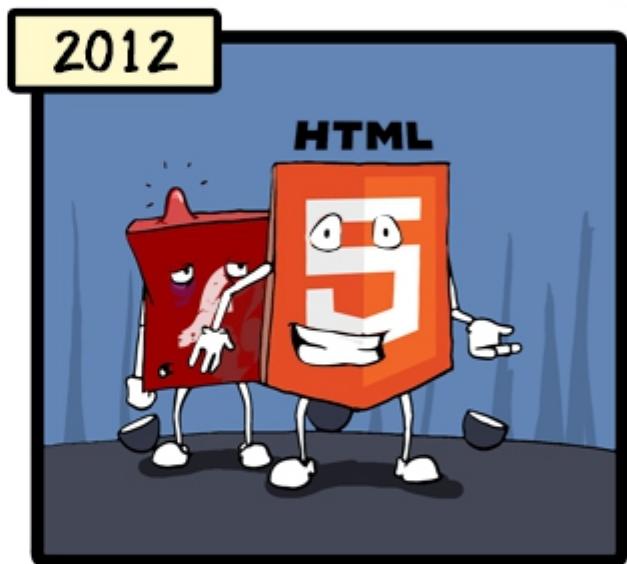
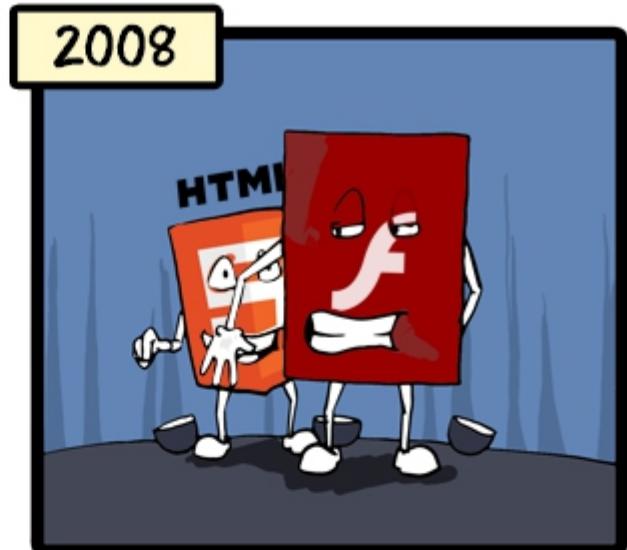
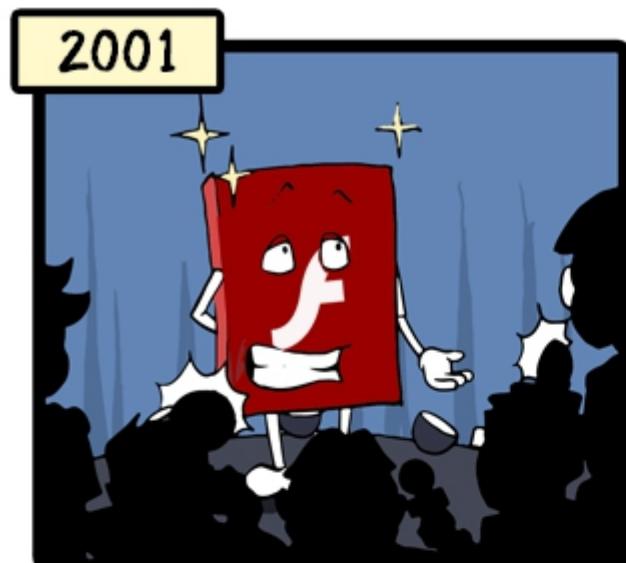
Cara Menambahkan Video di HTML

Kita membutuhkan sebuah media player untuk menampilkan video di HTML.

Dulu..

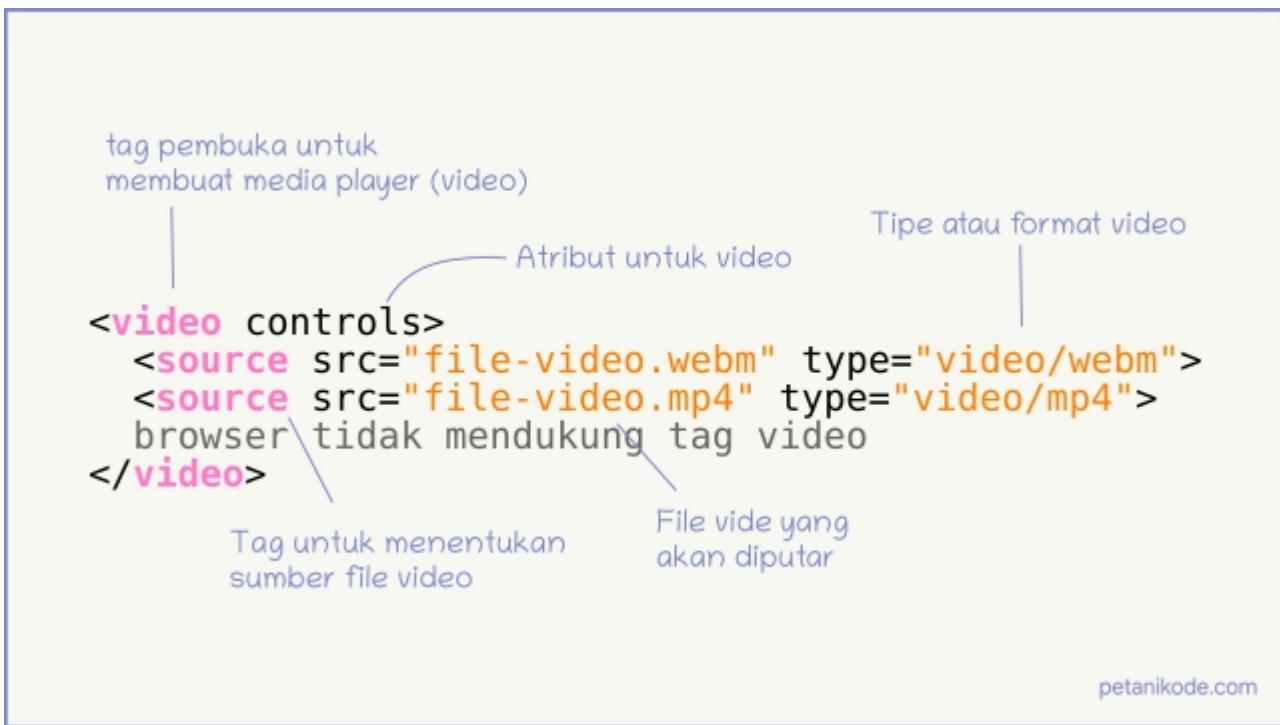
Sebelum ada HTML 5, media player di HTML dibuat dengan program eksternal seperti adobe flash.

Namun, kini sudah tidak dipakai lagi.



CommitStrip.com

HTML sekarang punya tag baru untuk membuat media player, yakni tag <video>.



Jika tag video di buka pada browser yang tidak mendukungnya, maka teks **browser tidak mendukung tag video** akan ditampilkan.

Tapi, kalau mendukung.. videonya yang akan ditampilkan.

Contoh:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Video di HTML</title>
</head>
<body>
  <h1>Contoh Video di HTML</h1> <video controls> <source src="cat-herd.webm" type="video/webm" /> Browsermu tidak mendukung tag ini, upgrade donk! </video></body>
</html>

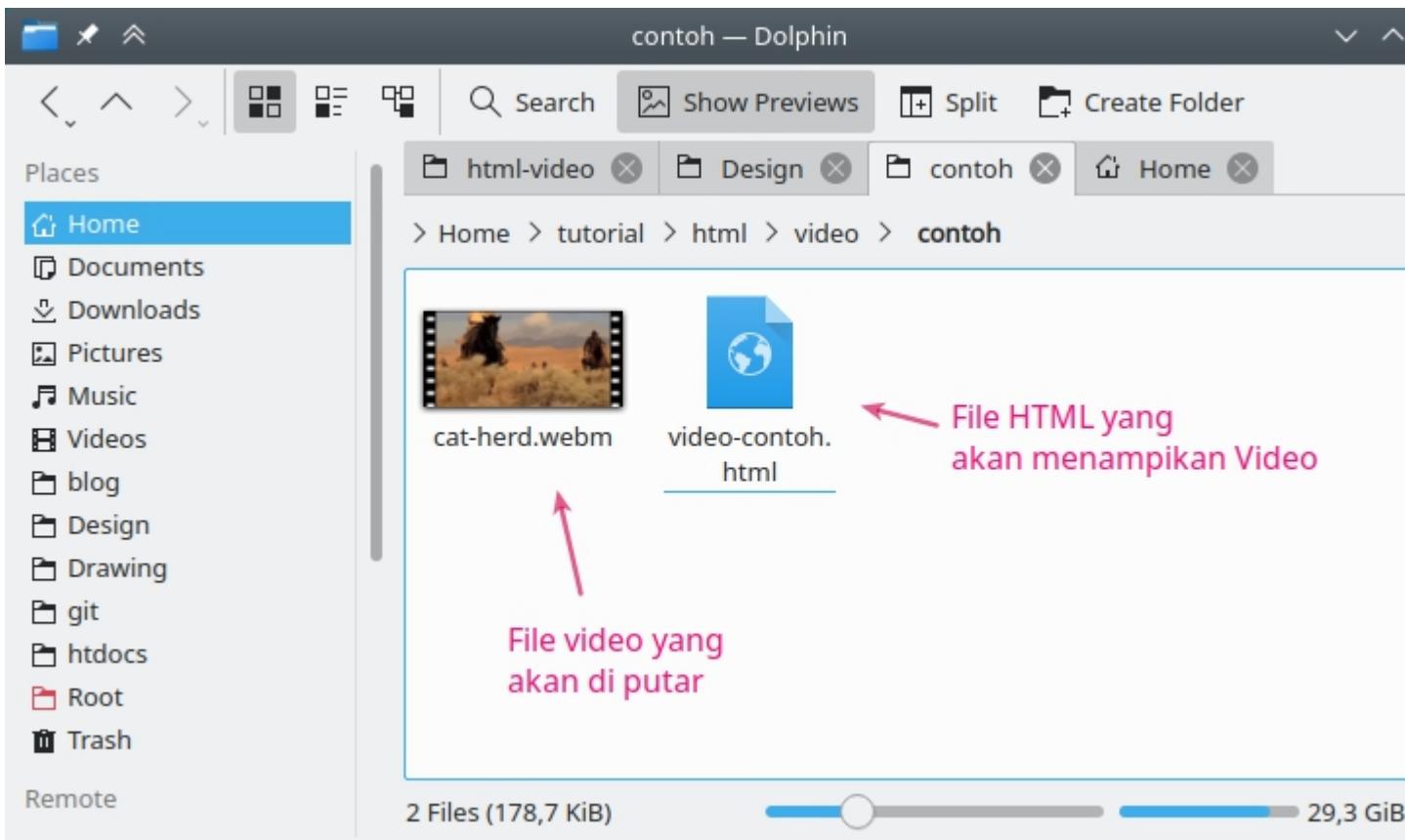
```

File videonya kamu bisa download di: [cat-herd.webm](#)

Hasilnya:

Live Demo

Pada contoh di atas, kita menuliskan secara langsung nama video yang akan ditampilkan. Ini karena videonya berada dalam satu folder dengan file HTML.



Jika video tersebut tersimpan di folder yang berbeda, maka perlu ditulis alamat folder atau path-nya.

Misalkan, saya meneruh videonya di dalam folder **video**. Maka, alamat path yang digunakan adalah:

```
<video><source src="video/nama-video.mp4"></video>
```

..dan jika videonya berada di website yang berbeda, maka kita harus mengisi atribut **src** dengan URL lengkap dari video.

```
<video><source src="https://www.petanikode.com/img/html-video/cat-herd.webm"></video>
```

Format Video yang Didukung

Tidak semua format video dapat ditampilkan di HTML. Berikut ini beberapa format video yang didukung: [1](#)

Format FILE

MP4

WebM

Ogg

Media Type

video/mp4

video/webm

video/ogg

Jika kamu punya video dengan format yang berbeda dari ketiga format tersebut, maka kamu harus mengubahnya agar bisa ditambahkan ke HTML.

Atribut untuk Video

Tag `<video>` punya beberapa atribut yang bisa diberikan:

Nama Atribut	Nilai	Fungsi
<code>autoplay</code>	<code>true/false</code>	Agar video diputar otomatis
<code>controls</code>	<code>true/false</code>	Untuk mengaktifkan control video player
<code>loop</code>	<code>true/false</code>	Untuk memutar video terus menerus
<code>muted</code>	<code>true/false</code>	Untuk menonaktifkan audio
<code>poster</code>	Image Path	Untuk menentukan gambar cover dari video
<code>width & height</code>	angka	Untuk menentukan tinggi dan lebar video
<code>playsinline</code>	<code>true/false</code>	Untuk memutar video secara ‘inline’

Jika atribut bernilai `true`, maka ia boleh ditulis namanya saja.

Contoh:

```
<video loop="true"><source src="video.mp4" />
</video>
```

Bisa disingkat menjadi:

```
<video loop><source src="video.mp4" />
</video>
```

Jika nilai atribut bernilai `false`, maka atribut tersebut boleh tidak ditulis atau juga boleh ditulis.

Contoh:

```
<video loop="false"><source src="video.mp4" />
</video>
<!-- boleh juga seperti ini: --&gt;&lt;video&gt;&lt;source src="video.mp4" /&gt;
&lt;/video&gt;</pre>
```

Nah, untuk atribut lainnya, kamu bisa cek di [MDN: The Video Embed element](#).

Video Sebagai Gambar Animasi Gif

Banyak website modern saat ini menggunakan video sebagai ganti dari animasi gif. Bahkan juga Google menyarankan untuk menggunakan video daripada gif.[2](#)

Mengapa?

Karena ukuran file dari video jauh lebih kecil dibandingkan dengan gif dan juga tentunya akan mempengaruhi kecepatan download.

Nggak percaya?

Mari kita bandingkan:

Name	Size	Modified	
cat-herd.gif	3,6 MiB	26/05/18 08.02	
cat-herd.mp4	390,7 KiB	22/07/20 17.30	
cat-herd.webm	178,4 KiB	22/07/20 17.30	

File **cat-herd.gif** punya ukuran **3,6 MB** setelah saya convert formatnya menjadi MP4 dan Webm, ukurannya menjadi sangat kecil.

Terbukti kan, file video lebih kecil daripada gif.

Lalu, bagaimana cara membuat video menjadi gambar gif di HTML.

Caranya sama seperti menambahkan video biasa, tapi kita harus mengaktifkan beberapa atribut seperti **autoplay**, **muted**, **playsinline** dan **loop**.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Video di HTML</title>
</head>
<body>
```

```
<h1>Contoh Video sebagai gif</h1> <video autoplay loop muted playsinline>
<source src="cat-herd.webm" type="video/webm" /> <source src="cat-herd.mp4"
type="video/mp4" /> </video> </body>
</html>
```

Hasilnya:

Live Demo

Subtitle untuk Video

Subtitle adalah teks yang akan ditampilkan dalam video. Biasanya digunakan untuk terjemahan atau alih bahasa dari video dan juga untuk membantu tuna rungu (orang tuli) untuk menyerap informasi pada video.

Subtitle pada HTML dapat kita tambahkan dengan tag `<track>`. Tag ini memiliki atribut `src` yang akan digunakan untuk menambahkan file subtitle.

```
<track src="sub-id.vtt" kind="subtitles" srclang="id" label="Indonesia"/>
```

The diagram illustrates the structure of the `<track>` tag with various annotations:

- `<track`: tag pembuka untuk menambahkan subtitle
- `src="sub-id.vtt"`: File Subtitle
- `kind="subtitles"`: Tipe subtitle
- `srclang="id"`: kode bahasa
- `label="Indonesia"`: Label yang akan ditampilkan pada Media Player
- `Atribut yang sering digunakan`: A bracket groups `src`, `kind`, `srclang`, and `label`.

petanikode.com

Format file subtitle untuk video di dalam HTML adalah WebVTT (`.vtt`) atau Web Video Text Track. File `.vtt` ini bisa dibuat dengan teks editor.

Jika kamu punya subtitle dengan format SubRip Text (`.srt`), kamu bisa mengubahnya menjadi `.vtt` di [srt2vtt](#).

Sekarang mari kita coba contohnya: [video-subtitle.html](#)

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

```
<meta charset="UTF-8">
<title>Contoh Video di HTML</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
</head>
<body>
  <h1>Video Subtitle</h1>
  <video controls>
    <source src="cat-herd.webm" type="video/webm"/>
    <source src="cat-herd.mp4" type="video/mp4"/>
    <track src="cat-herd-id.vtt" kind="subtitles" srclang="id"
label="Indonesia"/>
  </video>
</body>
</html>
```

Dan berikut ini isi file: `cat-herd-id.vtt`

```
WEBVTT

0
00:00:00.000 --> 00:00:03.000
Para penunggang kuda.

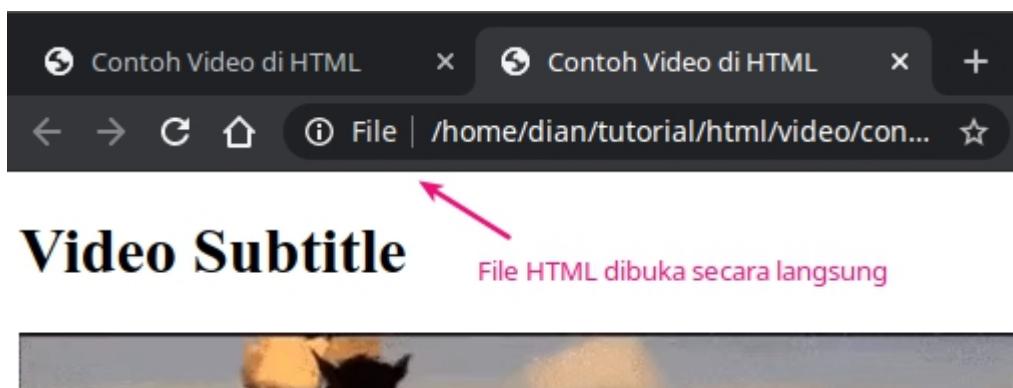
1
00:00:04.000 --> 00:00:08.000
Kucing berlari.
```

Hasilnya:

Catatan penting:

Subtitle tidak akan ditampilkan jika kita membuka file HTML secara langsung dari browser.

Coba perhatikan di bagian address bar, jika di sana ada tulisan **File..** berarti kita membuka file HTML secara langsung.



Namun, jika di address bar ada HTTP atau HTTPS.. itu artinya kita membuka file HTML melalui web server.

..dan juga jika format file **.vtt** tidak benar, subtitle tidak akan ditampilkan.

Pastikan formatnya valid, silahkan gunakan [Live WebVTT Validator](#) untuk pengecekan.

Menambahkan Video dari Youtube

Saat tidak ingin repot-repot mengubah format video, kita bisa manfaatkan Youtube.

Tinggal upload saja videonya ke Youtube. Nanti kita akan dapat id unik dari video. Id unik ini bisa kita dapatkan dari URL video.

Contoh: **N7iY-jNWeFc**

Youtube sendiri sudah punya media player, jadi kita tidak perlu membuatnya dengan tag **<video>**.

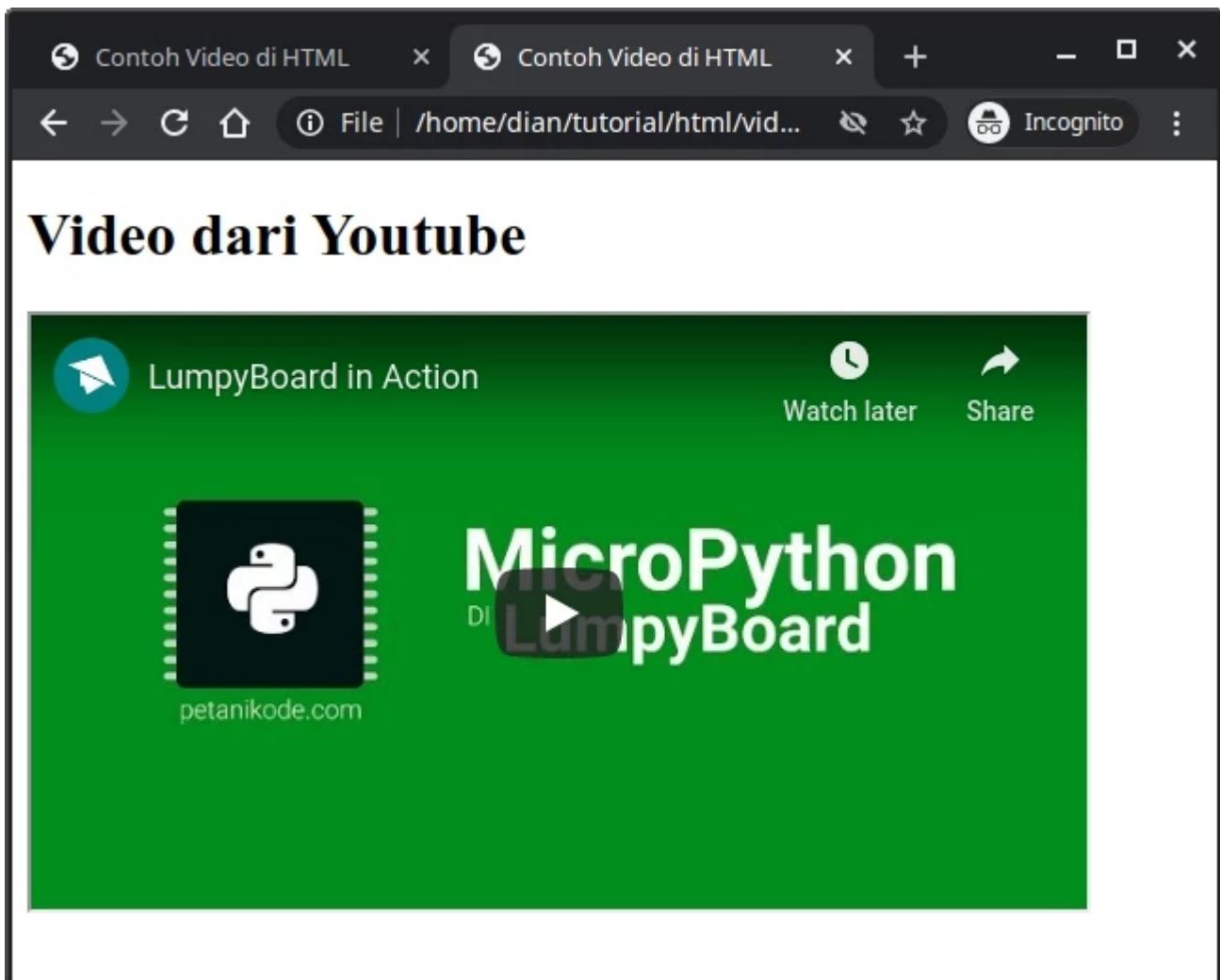
Tag yang kita butuhkan untuk menambahkan video dari Youtube adalah **<iframe>**. Tag ini sebenarnya berfungsi untuk menambahkan halaman lain dalam sebuah frame.

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Video di HTML</title>
</head>
<body>
  <h1>Video dari Youtube</h1>  <iframe width="560" height="315"
src="https://www.youtube.com/embed/N7iY-jNWeFc"></iframe></body>
</html>
```

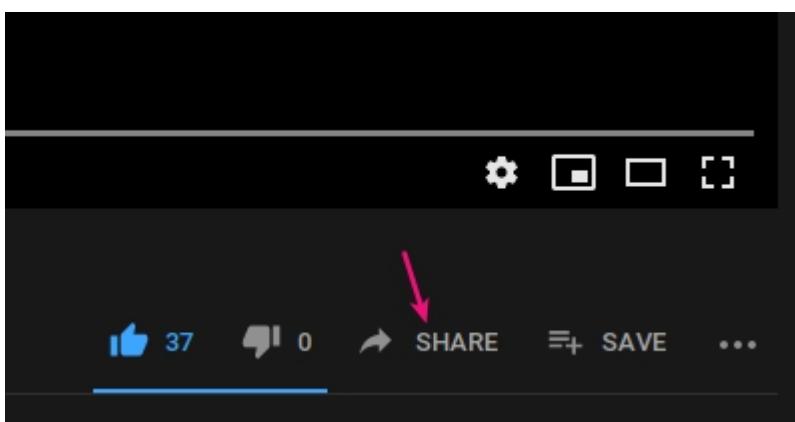
Perhatikan URL yang digunakan pada atribut **src**, di sana kita menggunakan **/embed/**. Ini adalah halaman yang digunakan khusus untuk embed video dari Youtube.

Hasilnya:

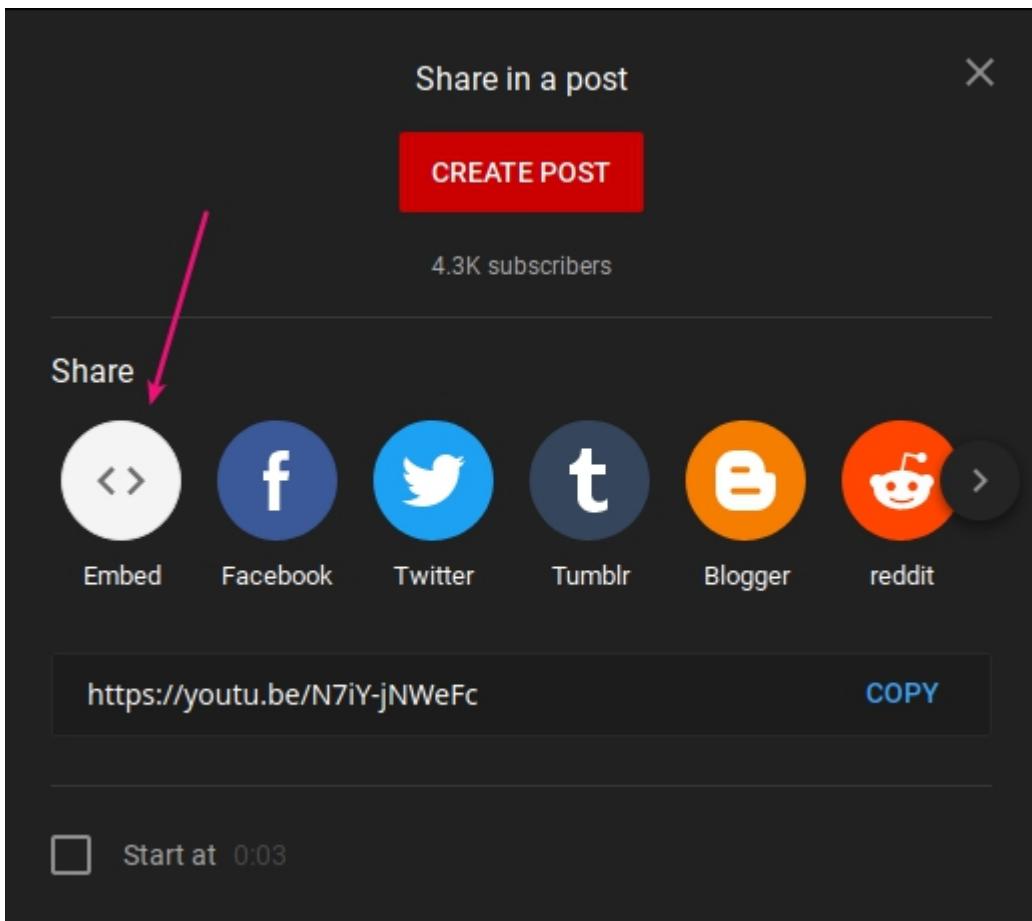


Sebenarnya ada cara gampangnya..

Pada video Youtube yang ingin kita embed, klik saja tombol share.



Maka akan muncul opsi untuk share video, pilih saja embed..



..dan kita akan mendapatkan kode HTML untuk embed videonya.

Kode ini bisa kita copy ke file HTML.

Cara Menambahkan Audio di HTML

Sebelum adanya HTML 5, audio di HTML ditambahkan dengan program eksternal seperti flash player.

Namun, kini HTML sudah punya tag sendiri yakni `<audio>`.

Tag `<audio>` adalah tag untuk membuat audio player. Lalu kita bisa memberikan file audio yang akan diputar dengan tag `<source>`.

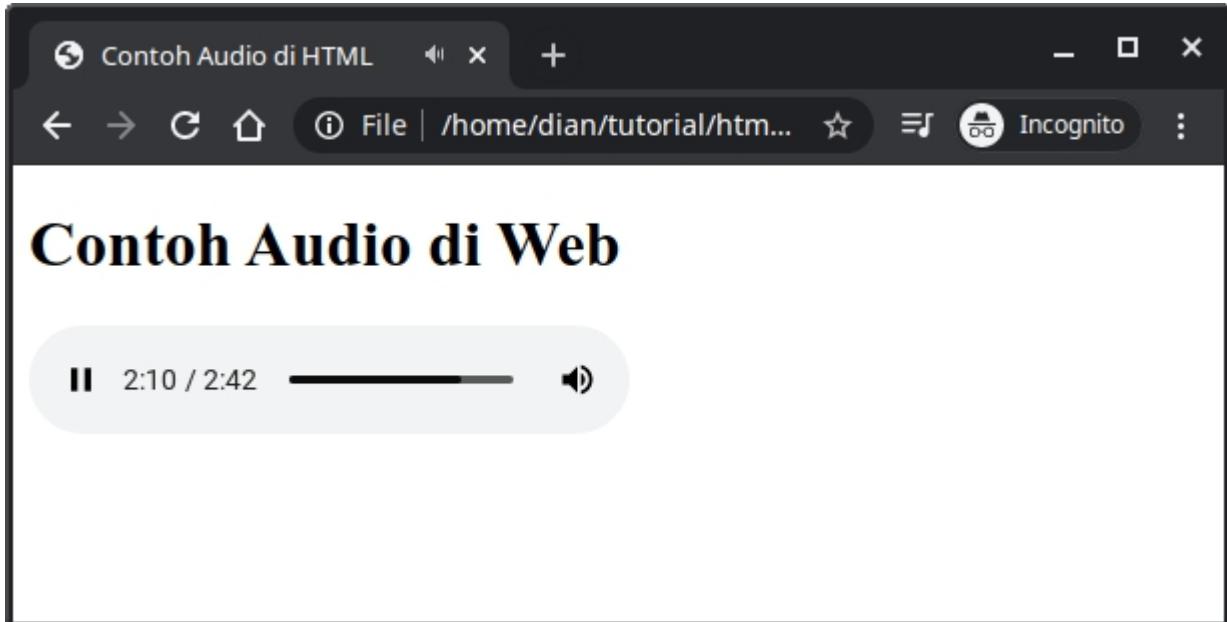
Contoh: `contoh-audio.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Contoh Audio di HTML</title>
</head>
```

```
<body>
  <h1>Contoh Audio di Web</h1> <audio controls> <source src="Ngoni.mp3"
type="audio/mpeg"> Browsermu tidak mendukung tag audio, upgrade donk!
</audio></body>
</html>
```

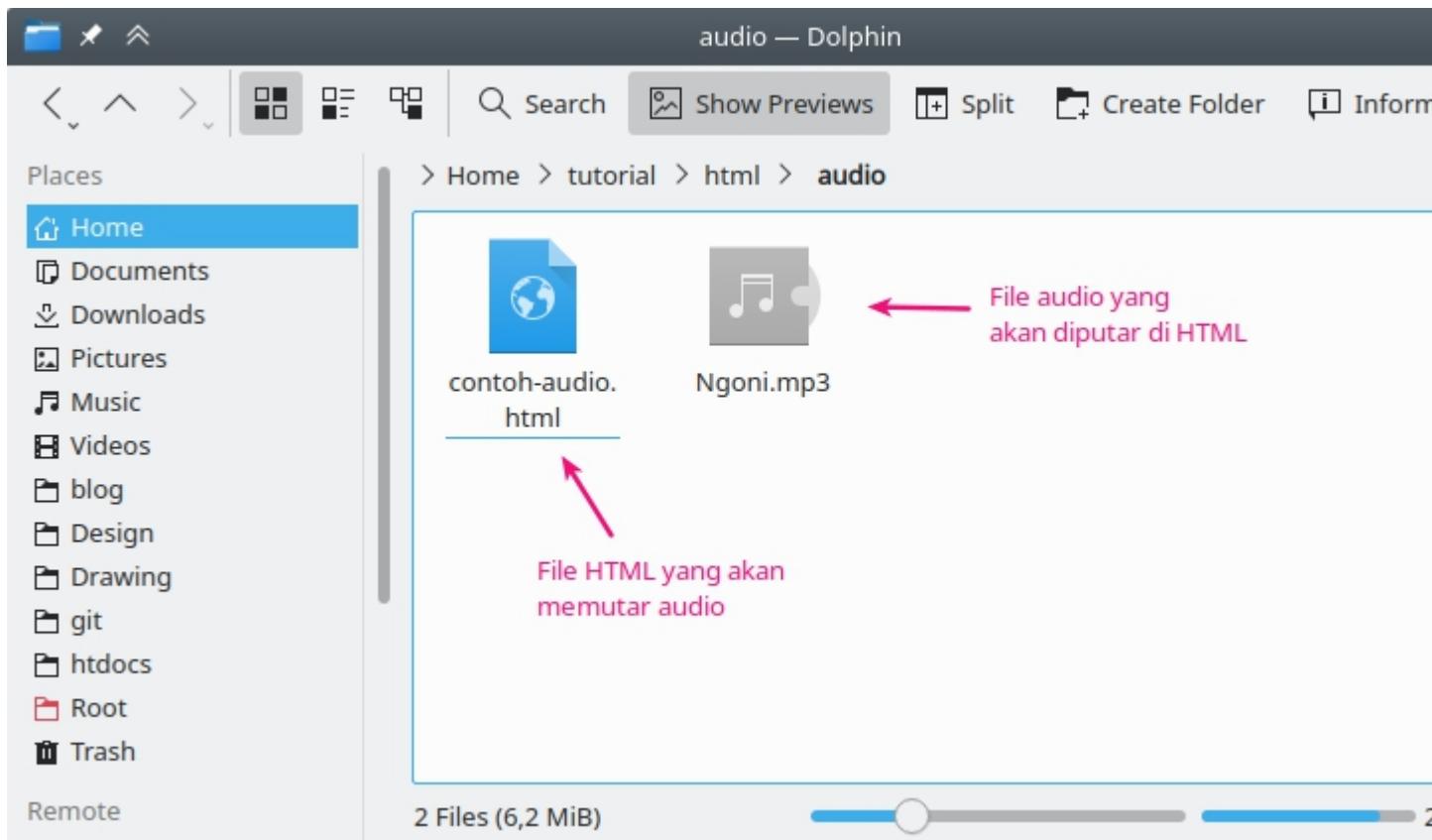
File audionya bisa kamu download di: [Ngoni.mp3](#) (6.2 MB)

Hasilnya:



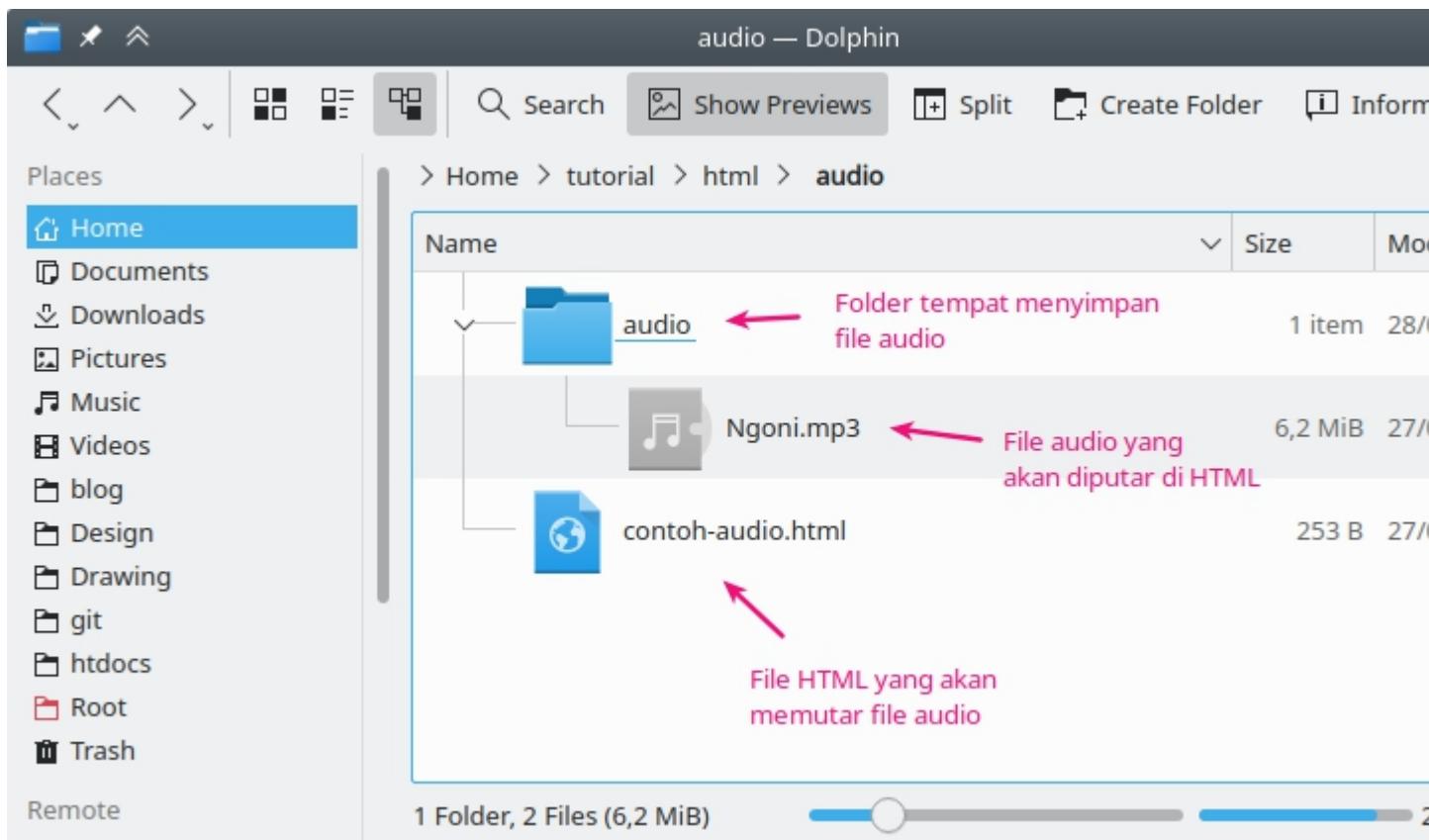
Perhatikan!

Pada atribut `src`, kita menulis langsung nama file audio yang akan diputar. Ini karena kita menaruh file tersebut dalam satu folder yang sama dengan file HTML.



Jika file audio tersimpan di folder yang berbeda, maka harus ditulis alamat path menuju folder tersebut.

Misalkan, saya menaruhnya di dalam folder **audio**,



maka atribut `src` bisa diisi seperti ini:

```
<audio controls>    <source src="audio/Ngoni.mp3" type="audio/mpeg">  
Browsermu tidak mendukung tag audio, upgrade donk!  
</audio>
```

...dan jika file audionya tersimpan di website yang berbeda, maka kita harus mengisinya dengan alamat URL.

```
<audio controls>    <source src="https://github.com/petanikode/belajar-  
html/raw/master/audio/Ngoni.mp3" type="audio/mpeg">    Browsermu tidak  
mendukung tag audio, upgrade donk!  
</audio>
```

Format File Audio yang didukung

Audio player di HTML tidak mendukung semua jenis format file audio. Berikut ini daftar format file audio yang bisa diputar di HTML. [1](#)

Format	Container	MIME type
PCM	WAV	audio/wav
MP3	MP3	audio/mpeg
AAC	MP4	audio/mp4
AAC	AAC	audio/aac
AAC	AAC	audio/aacp
Vorbis	Ogg	audio/ogg
Vorbis	WebM	audio/webm
Opus	Ogg	audio/ogg
Opus	WebM	audio/webm
FLAC	FLAC	audio/flac
FLAC	Ogg	audio/ogg

Format file yang biasanya digunakan adalah MP3 dan MP4 (M4A), karena ukuran filenya relatif kecil. Sementara format FLAC adalah format file audio dengan kualitas tinggi dan ukuran filenya relatif lebih besar.

Atribut untuk Audio

Tag `<audio>` punya beberapa atribut yang sering digunakan:

1. controls

Atribut ini berfungsi untuk mengaktifkan tombol kontrol seperti tombol play, pause, stop, scroll, dan volume).

Contoh penggunaan:

```
<audio controls="true">      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
<!-- atau -->
<audio controls>      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
```

Jika bernilai `true`, maka nilainya boleh tidak diisi. Nilai `true` artinya, kita akan mengaktifkan tombol kontrol dan jika nilainya `false` maka artinya tombol kontrol tidak diaktifkan.

2. autoplay

Atribut ini berfungsi untuk memutar audio secara otomatis. Nilai yang bisa diberikan pada atribut ini adalah `true` dan `false`.

Nilai `true` artinya kita akan memutar audio secara otomatis, dan `false` artinya audio tidak akan diputar secara otomatis.

Contoh:

```
<audio autoplay="true">      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
<!-- atau -->
<audio autoplay>      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
```

Catatan: Atribut ini mungkin saja tidak akan bekerja pada Google Chrome, karena ada [perubahan policy](#) (kebijakan) dalam memutar audio secara otomatis.

3. loop

Atribut `loop` berfungsi untuk mengulang-ulang pemutaran audio. Ini seperti repeat one. Nilai yang bisa diberikan adalah `true` dan `false`.

Contoh:

```
<audio loop="true">      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
<!-- atau -->
<audio loop>      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
```

4. muted

Atribut ini berfungsi untuk mensenyapkan audio. Nilai yang bisa diberikan adalah **true** dan **false**.

Contoh:

```
<audio muted="true">      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
<!-- atau -->
<audio muted>      <source src="audio/Ngoni.mp3" type="audio/mpeg">
    Browsermu tidak mendukung tag audio, upgrade donk!
</audio>
```

Audio Sebagai Background

Audio kadang sering digunakan sebagai background. Biasanya menggunakan musik.

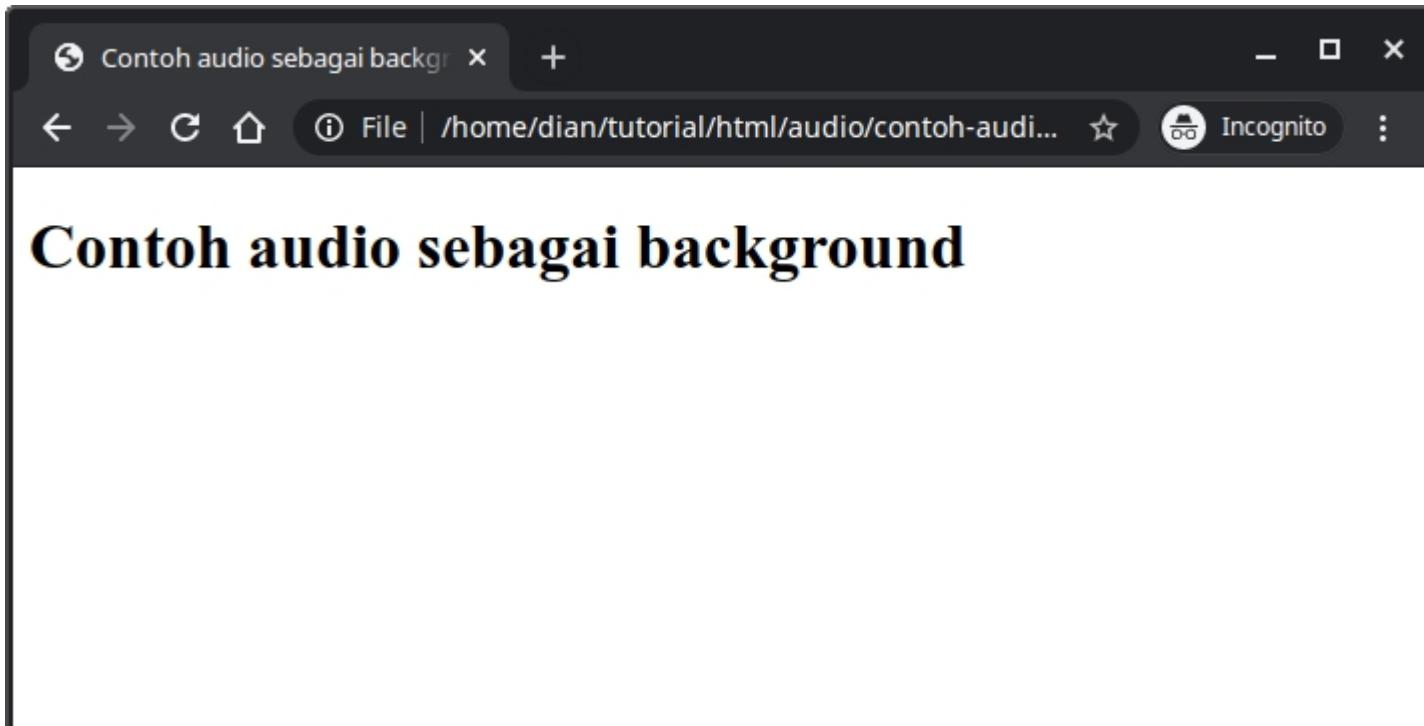
Tujuan menambahkan background agar menambah kesan tertentu pada web. Misalkan, ingin membuat pengunjung merasa santai.. maka kita bisa memutar audio musik yang santai.

Cara membuat musik sebagai background adalah dengan menambahkan atribut **autoplay**, **loop**, dan menghilangkan kontrol (**hidden**).

Contoh:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Contoh audio sebagai background</title>
</head>
<body>
    <h1>Contoh audio sebagai background</h1> <audio hidden autoplay loop>
<source src="Ngoni.mp3" type="audio/mpeg"> Browsermu tidak mendukung tag
audio </audio></body>
</html>
```

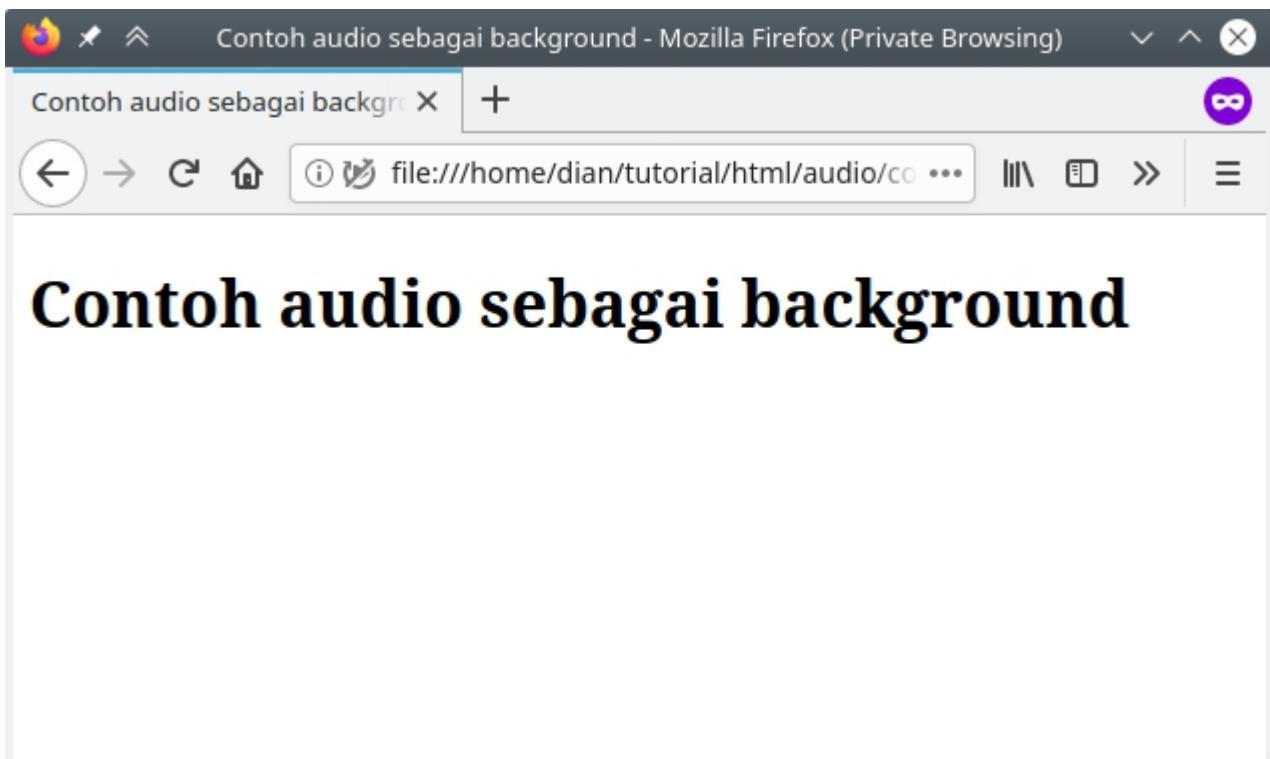
Hasilnya:



Nggak ada suaranya?

Sekali lagi, atribut `autoplay` mungkin tidak bekerja di Google Chrome karena ada pembaharuan kebijakan dalam memutar audio.

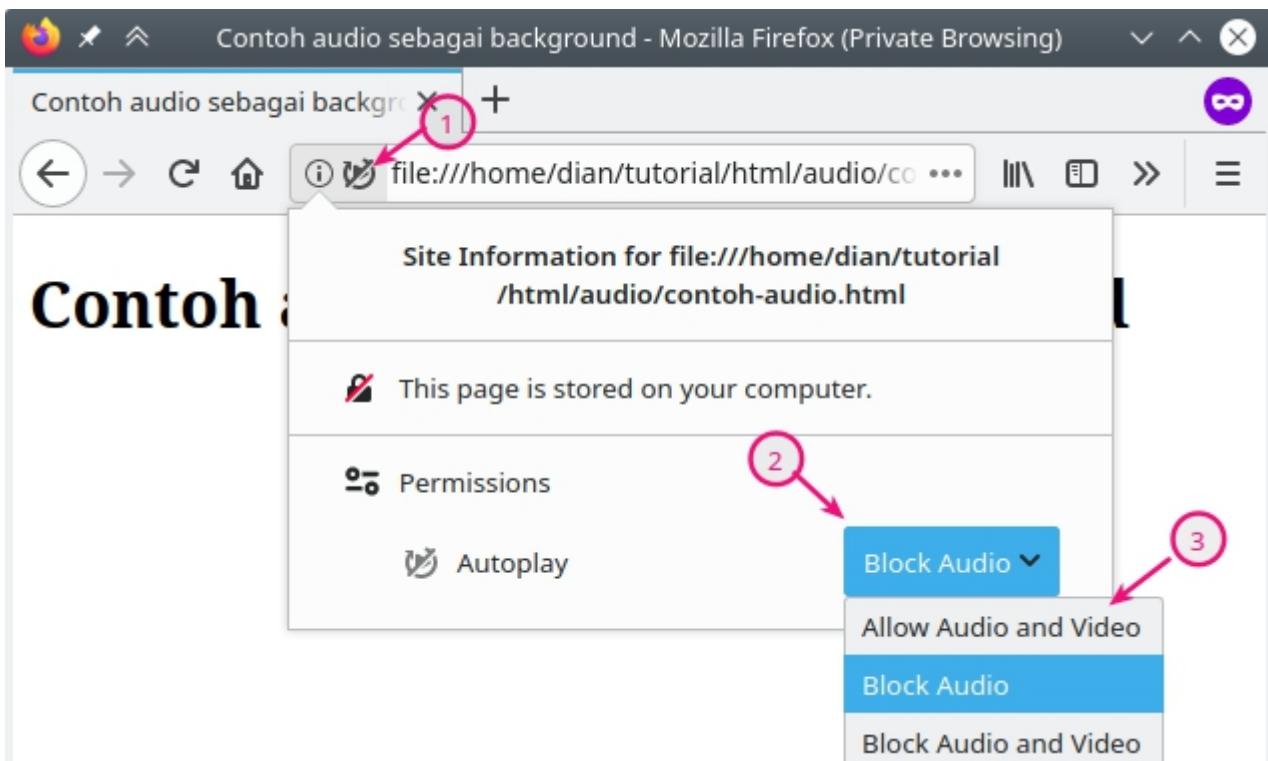
Sekarang kita coba buka dengan Mozilla Firefox.



Hasilnya sama saja, tidak ada suaranya.

Ini karena **autoplay** diblokir otomatis.

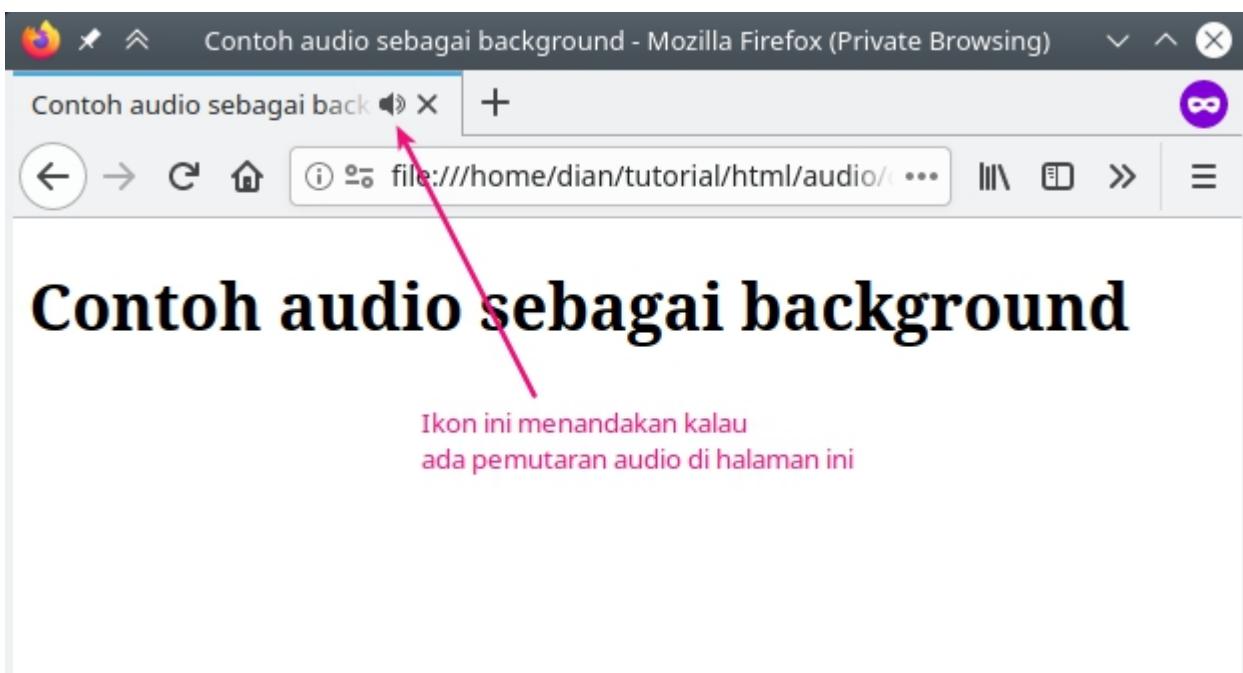
Sekarang kita coba ubah izin untuk memutar audio secara otomatis. Klik ikon **autoplay** pada address bar di dekat ikon **(i)**, sehingga muncul seperti ini:



Pada bagian Autoplay, ubah Block Audio menjadi Allow Audio and Video.

Setelah itu, coba refresh atau reload.

Maka suaranya akan terdengar.



Step 1 – Membuat Desain Web

Pembuatan web dimulai dengan desain. Kalau tidak ada desain, nanti kita akan kesulitan dan tidak akan tahu mau buat apa.

Biasanya desain web dikerjakan oleh desainer, setelah itu diserahkan ke programmer untuk diubah menjadi HTML.

Pada project ini, kita akan membuat tiga halaman web, yakni home, contact, dan about. Desain yang digunakan adalah desain dalam bentuk wireframe atau sketsa.

Berikut ini desain untuk tiap-tiap halaman:

1. Desain Halaman Home



Nama Lengkap
(Jabatan atau title)

Overview

Hi, saya adalah web developer. Saat ini sedang belajar HTML di Petani Kode.

Skill	Pengalaman
<ul style="list-style-type: none">- HTML (Expert)- CSS (Beginner)- Javascript (Beginner)	<ul style="list-style-type: none">- Pernah bekerja di _____- Menjadi leader di komunitas- Kontributor open source

Copyright 2020 @ Nama pemilik.

Halaman home adalah halaman utama yang akan dibuka pertama kali saat pengunjung membuka website. Halaman ini berisi menu, foto, teks, dan tabel.

2. Desain Halaman Contact

Contact Me

Email

alamat email

Pesan

Tulis pesan anda...

Kirim

Copyright 2020 @ Nama pemilik.

Halaman contact adalah halaman yang berisi form untuk menghubungi pemilik website.

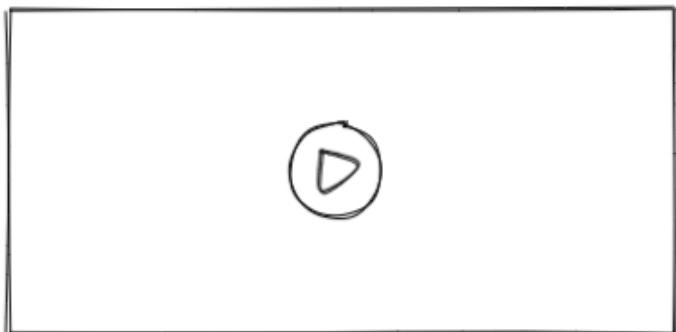
3. Desain Halaman About

About Me

Hi, saya adalah web developer. Saat ini sedang belajar HTML di Petani Kode.

Saya memang masih baru dalam web development. Karena itu, saya tidak akan pernah berhenti belajar.

Saya ingin menguasai bahasa HTML, CSS, dan Javascript. Simak video lengkap tentang saya:



Copyright 2020 @ Nama pemilik.

Halaman about adalah halaman yang berisi informasi lengkap tentang website.

Step 2 – Memulai Project Web

Silahkan buat folder baru dengan nama **websiteku**.

Setelah itu buat folder **image** dan **video** di dalam folder **websiteku**.

Folder **image** akan kita gunakan untuk menyimpan gambar dan **video** untuk menyimpan video.

Sehingga nanti struktur folder kita akan menjadi seperti ini:

```
image
  o  foto-profile.jpg
video
```

- o `video-about.webm`

- `cv-name.pdf`
- `index.html`
- `contact.html`
- `about.html`
- `favicon.ico`

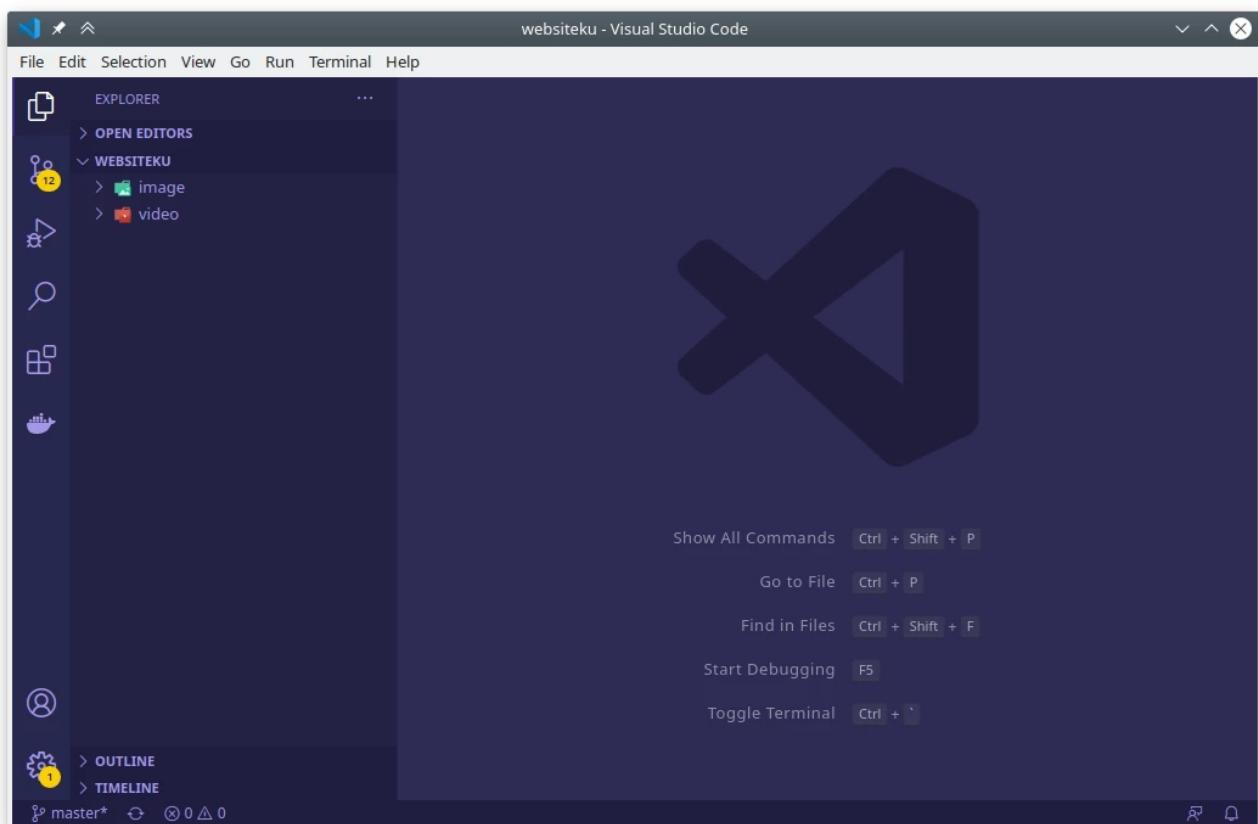
File yang perlu kamu persiapkan di tahapan ini adalah `foto-profile.jpg` dan `video-about.webm`.

Berikutnya, kita akan mulai menulis kode. Silahkan buka folder `websiteku` dengan Visual Studio Code.

Caranya:

Klik menu **File**, lalu pilih **Open Folder** dan carilah folder `websiteku`.

Dengan begini, kita sudah siap untuk menulis kode.



Step 3 – Membuat Halaman Home

Silahkan buat file baru bernama `index.html` di dalam folder `websiteku`.

Setelah itu, isi dengan kode berikut:

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ahmad Muhardian Personal Website</title>
</head>

<body>
    <nav>
        <a href="index.html">Home</a> |
        <a href="cv-dian.pdf">Download CV</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About me</a>
    </nav>

    <hr />

    <header style="text-align: center;">
        
        <h1>Ahmad Muhardian</h1>
        <p>(Web Developer)</p>
    </header>

    <hr />

    <article style="text-align: center;">
        <h2>Overview</h2>
        <p>
            Hi, saya adalah web developer yang berdomisili di Jakarta.
            Saat ini sedang belajar HTML di Petnai Kode
        </p>
    </article>

    <div style="max-width: 600px; margin: 3em auto">
        <table border="1" width="100%">
            <thead>
                <tr>
                    <th>Skill</th>
                    <th>Pengalaman</th>
                </tr>
            </thead>
```

```
<tbody>
  <tr>
    <td>
      <ul>
        <li>HTML (Expert)</li>
        <li>CSS (Beginner)</li>
        <li>Javascript (Beginner)</li>
      </ul>
    </td>
    <td>
      <ul>
        <li>Freelancer di Internet</li>
        <li>Leader Local Linux Community</li>
        <li>Leader Local Linux Community</li>
      </ul>
    </td>
  </tr>
</tbody>
</table>
</div>

<hr>
<footer style="text-align: center;">
  <p>Copyright © 2020 Ahmad Muhardian.</p>
</footer>
</body>
</html>
```

Jangan lupa untuk mengubah nama **Ahmad Muardian** dengan nama kamu.

Jika kita coba membuka file **index.html**, maka hasilnya akan seperti ini:

A screenshot of a web browser window displaying a personal website. The address bar shows the URL as `/home/dian/tutorial/html/websiteku/index.html`. The page content includes:

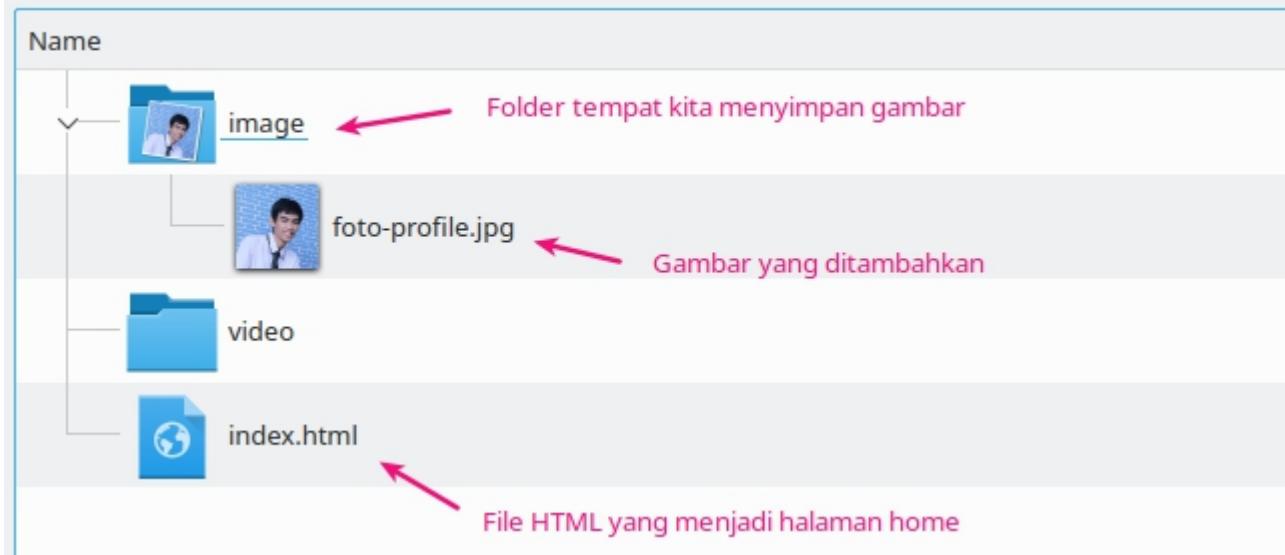
- A large empty rectangular box where a profile picture would normally be displayed.
- The name **Ahmad Muhardian** in bold black font.
- The title **(Web Developer)** in smaller black font.
- A section titled **Overview** containing the text: "Hi, saya adalah web developer yang berdomisili di Jakarta. Saat ini sedang belajar HTML di Petnai Kode".
- A table with two columns:

Skill	Pengalaman
<ul style="list-style-type: none">HTML (Expert)CSS (Beginner)Javascript (Beginner)	<ul style="list-style-type: none">Freelancer di InternetLeader Local Linux CommunityLeader Local Linux Community
- At the bottom, a copyright notice: "Copyright © 2020 Ahmad Muhardian."

Gambarnya tidak bisa tampil karena kita belum menambahkan file gambar di dalam folder [image](#).

Silahkan tambahkan file gambar dengan nama [foto-profile.jpg](#). Pastikan gambar yang ditambahkan memiliki ukuran persegi atau rasio 1:1. Pada proyek ini, saya menggunakan gambar dengan ukuran 200x200 piksel.

Download file gambar: [foto-profile.jpg](#)



Setelah itu, coba refresh halaman home atau `index.html`.

Maka hasilnya:

Ahmad Muhardian Personal

Home | Download CV | Contact | About me



Ahmad Muhardian

(Web Developer)

Overview

Hi, saya adalah web developer yang berdomisili di Jakarta. Saat ini sedang belajar HTML di Petnai Kode

Skill	Pengalaman
<ul style="list-style-type: none">HTML (Expert)CSS (Beginner)Javascript (Beginner)	<ul style="list-style-type: none">Freelancer di InternetLeader Local Linux CommunityLeader Local Linux Community

Copyright © 2020 Ahmad Muhardian.

Step 4 – Membuat Halaman About Me

Buatlah file HTML baru dengan nama `about.html`.

Kemudian isi dengan kode berikut:

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ahmad Muhardian Personal Website</title>
</head>

<body>
    <nav>
        <a href="index.html">Home</a> |
        <a href="cv-dian.pdf">Download CV</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About me</a>
    </nav>

    <hr />

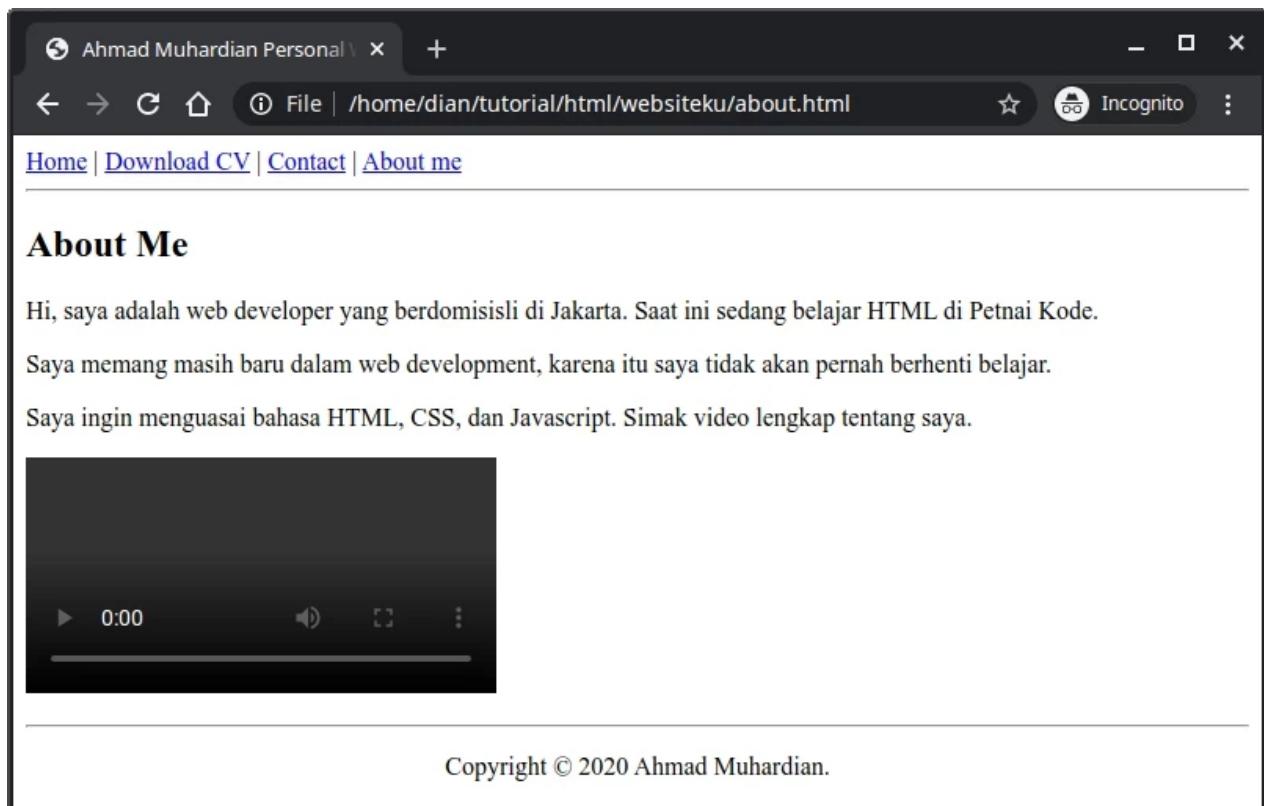
    <article>
        <h1>About Me</h1>
        <p>
            Hi, saya adalah web developer yang berdomisili di Jakarta.
            Saat ini sedang belajar HTML di Petnai Kode.
        </p>
        <p>
            Saya memang masih baru dalam web development, karena itu
            saya tidak akan pernah berhenti belajar.
        </p>
        <p>
            Saya ingin menguasai bahasa HTML, CSS, dan Javascript.
            Simak video lengkap tentang saya.
        </p>
        <p>
            <video controls>
                <source src="video/video-about.webm" type="video/webm"/>
            </video>
        </p>
    </article>

    <hr>
    <footer style="text-align: center;">
        <p>Copyright © 2020 Ahmad Muhardian.</p>
    </footer>
```

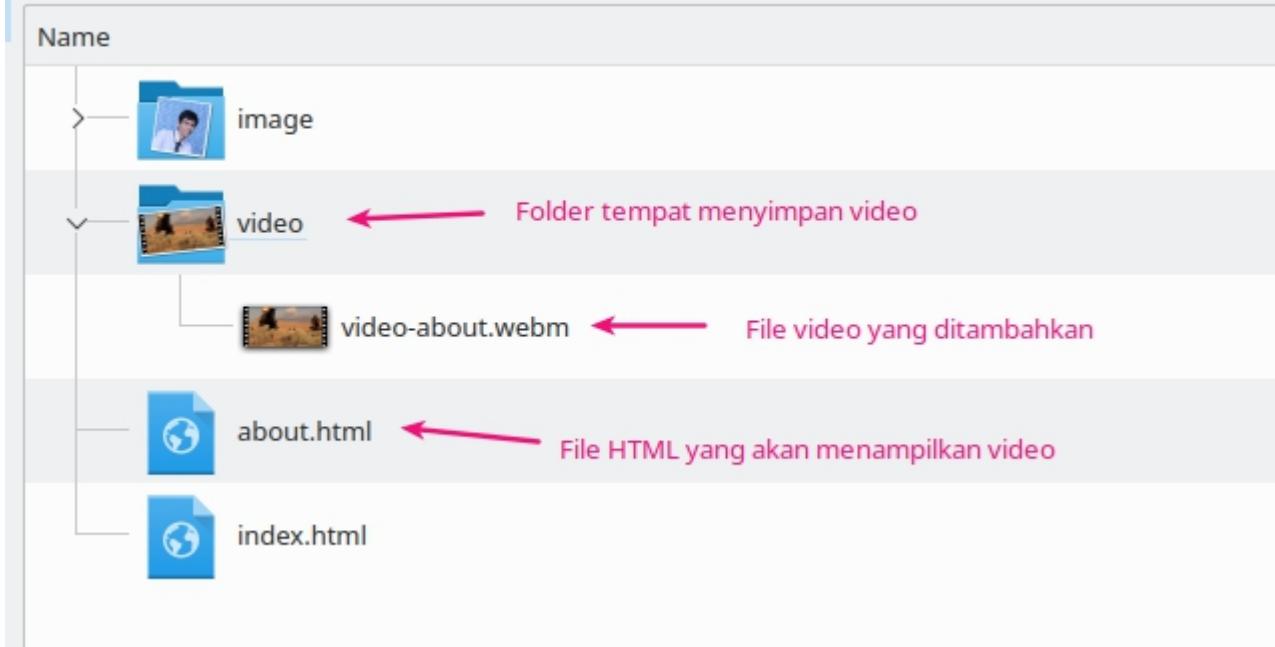
```
</body>  
</html>
```

Sama seperti halaman home, halaman ini juga memiliki konten berupa video untuk ditampilkan. Tapi file videonya belum ada.

Sudah pasti videonya tidak akan bisa ditampilkan:



Karena itu, silahkan tambahkan file video-nya ke dalam folder **video** dengan nama **video-about.webm**.



Jika kamu belum punya filenya, silahkan download di link ini:

[video-about.webm](#)

Setelah itu, coba buka dan refresh halaman about.

Maka hasilnya:

The screenshot shows a web browser window with the following details:

- Title Bar:** Ahmad Muhardian Personal
- Address Bar:** File | /home/dian/tutorial/html/websiteku/about.html
- Content Area:** Displays the 'about.html' page with the following text:

[Home](#) | [Download CV](#) | [Contact](#) | [About me](#)

About Me

Hi, saya adalah web developer yang berdomisili di Jakarta. Saat ini sedang belajar HTML di Petnai Kode. Saya memang masih baru dalam web development, karena itu saya tidak akan pernah berhenti belajar. Saya ingin menguasai bahasa HTML, CSS, dan Javascript. Simak video lengkap tentang saya.

Below the text is a video player showing a scene from a movie with two cowboys on horseback in a field. The video player interface includes a play button, a progress bar showing 0:00 / 0:08, and other control icons.

Step 5 – Membuat Halaman Contact

Buatlah file baru dengan nama `contact.html`.

Kemudian isi dengan kode berikut:

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Ahmad Muhardian Personal Website</title>
</head>

<body>
    <nav>
        <a href="index.html">Home</a> |
        <a href="cv-dian.pdf">Download CV</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About me</a>
    </nav>

    <hr />

    <div>
        <h1>Contact Me</h1>
        <form>
            <label for="email">Email</label><br />
            <input type="email" name="email" placeholder="alamat email"/>
            <br />
            <label for="message">Pesan</label><br />
            <textarea name="message" placeholder="Tulis pesan anda..." rows="4" cols="80"></textarea>
            <br />
            <br />
            <input type="submit" value="Kirim" />
        </form>
    </div>

    <hr>
    <footer style="text-align: center;">
        <p>Copyright © 2020 Ahmad Muhardian.</p>
    </footer>
</body>
</html>
```

Hasilnya:

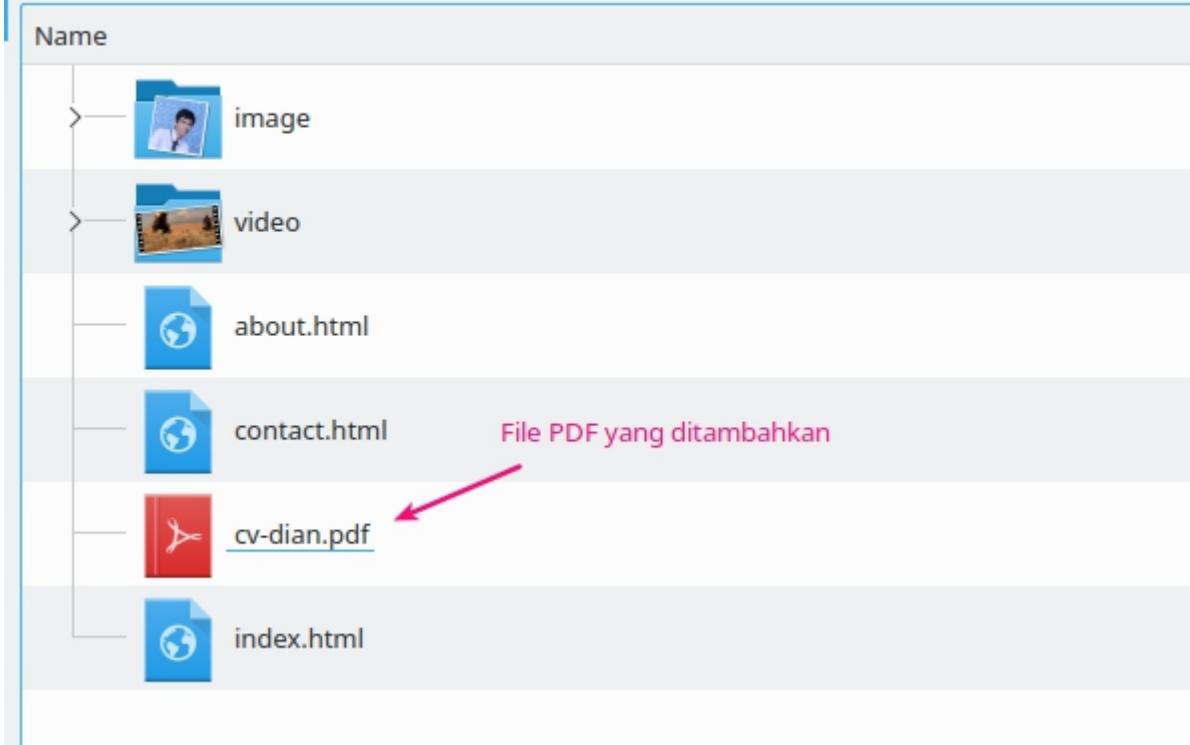
A screenshot of a web browser window titled "Ahmad Muhardian Personal". The address bar shows the URL "/home/dian/tutorial/html/websiteku/contact.html". The page content includes a header with links to "Home", "Download CV", "Contact", and "About me". Below this is a section titled "Contact Me". It contains fields for "Email" (with placeholder "alamat email") and "Pesan" (with placeholder "Tulis pesan anda..."). A "Kirim" button is at the bottom left. At the bottom right, it says "Copyright © 2020 Ahmad Muhardian."

Form contact ini belum bisa berfungsi, karena kita belum membuat kode untuk mengirim data.

Step 6 – Membuat Fitur Download CV

Fitur ini sebenarnya paling gampang dibuat. Kita hanya perlu menambahkan file **cv-dian.pdf** ke dalam folder websiteku.

Download file: [cv-dian.pdf](#)



Setelah itu, coba klik menu **Download CV**. Jika PDF-nya terbuka, maka link ini sudah benar.

Step 7 – Membuat Ikon untuk Web

Agar websitenya terlihat menarik, kita akan membuat ikon atau favicon. Silahkan buka favicon-generator.org kemudian pilih gambar yang akan dijadikan ikon.

The screenshot shows the 'Generate your favicon.ico' page of the favicon-generator.org website. The interface includes:

1. An 'Image:' input field containing 'dian-foto-profile.jpg' with a 'Browse...' button.
2. Two radio button options: 'Generate Icons for Web, Android, Microsoft, and iOS (iPhone and iPad) Apps' (unchecked) and 'Generate only 16x16 favicon.ico' (checked).
3. A 'Create Favicon' button.

Below these options are two checkboxes:

- Maintain Image Dimensions (don't resize to be square)
- Include your favicon.ico in the public gallery.

Setelah itu, kita akan mendapatkan link download dan juga kode HTML yang harus ditambahkan ke dalam tag `<head>` agar ikon bisa ditampilkan.

 Download

Klik untuk download

 Download the generated favicon and save to the root directory of your site. Then include the following code in the head of your HTM

What's next?

Copy dan paste kode ini ke dalam tag <head>

```
<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
<link rel="icon" href="/favicon.ico" type="image/x-icon">
```

Not happy with the result? Use our icon editor to tweak your favicon or bookmark this link and return later to edit your favicon.

Generate another favicon

Setelah kita mendapatkan ikon, simpan ikonnya ke dalam folder **websiteku**.

> Home > tutorial > html > **websiteku**

Name
 image
 video
 about.html
 contact.html
 cv-dian.pdf
 favicon.ico
 index.html

Terakhir, tugas kita tinggal membuat kode HTML untuk menampilkan ikon.
Silahkan copy kode berikut:

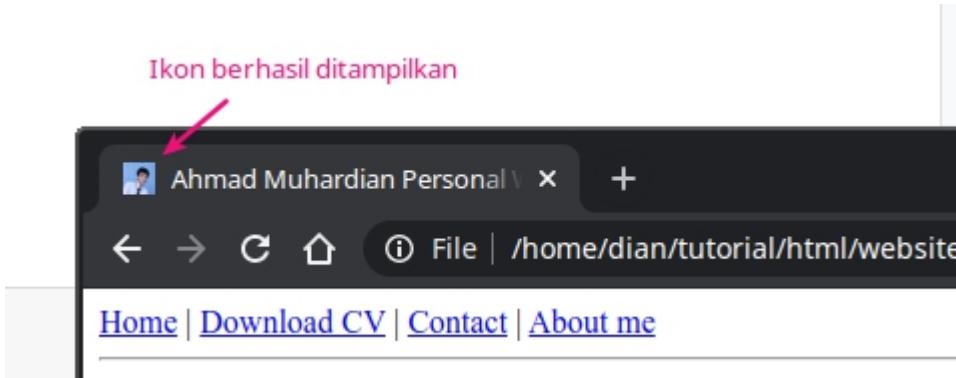
```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Kemudian paste di dalam tag **<head>** pada setiap halaman.

The screenshot shows the Visual Studio Code interface with the file 'index.html' open. The code includes meta tags for charset and viewport, a title, and a navigation bar with links to Home, Download CV, Contact, and About me. A specific line of code for a favicon is highlighted.

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0"
<title>Ahmad Muhardian Personal Website</title>
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

Maka hasilnya:



Apa itu Netlify?

Netlify adalah penyedia layanan hosting untuk static site dengan berbagai fitur menarik.

Netlify menyediakan semua kebutuhan untuk workflow web development zaman now.

Mengapa saya bilang zaman now?

Karena di Netlify tidak menggunakan FTP seperti yang biasa kita temukan pada shared hosting.

Netlify menggunakan [Git](#) dan CI/CD untuk melakukan deployment. Kedua alat ini sangat membantu dalam melakukan deployment.

Selain itu, Netlify juga menyediakan fitur untuk mengelola DNS, SSL atau HTTPS diberikan gratis, form, function, dan lain-lain.

Soal Harga bagaimana?

Tenang saja, kabar baiknya Netlify sangat bersahabat dengan kantong mahasiswa. Cek saja di [halaman pricing](#) untuk melihat paket hosting yang tersedia.

STARTER	PRO	BUSINESS	ENTERPRISE
<p>The basics for personal projects, hobby sites, or experiments.</p> <p>Free to get started</p> <ul style="list-style-type: none">✓ Automated builds from Git✓ Deploy to global Edge network✓ Site previews for every push✓ Instant rollbacks to any version✓ Deploy static assets & dynamic serverless functions <p>Start for free</p>	<p>Advanced features and support for building pilot projects or small sites.</p> <p>\$ 19 per member /month</p> <ul style="list-style-type: none">✓ Starter features, plus:✓ Password-protected sites✓ Slack & email notifications✓ Audit logs with 7-day history✓ 4x bandwidth & 3x build minutes✓ Shared environment variables✓ Multiple account owners✓ Email support <p>Buy Pro</p>	<p>Collaboration, security, & compliance for scaling production sites & web apps.</p> <p>\$ 99 per member /month</p> <ul style="list-style-type: none">✓ Pro features, plus:✓ SAML single sign-on✓ Role-based access control✓ Audit logs with full history✓ Unlimited Forms, Identity, & Functions✓ Unlimited Analytics for all sites✓ Prioritize time-sensitive builds✓ Billing administrator role✓ Self-hosted GitHub/GitLab <p>Buy Business</p>	<p>A custom plan and services for mission-critical web properties</p> <p>Contact us</p> <p>Starting at \$1,500/month. Tailored to team and performance requirements</p> <ul style="list-style-type: none">✓ 99.99% uptime SLA✓ 24x7x365 premium support✓ Enterprise-grade global edge network✓ High-performance builds with SLAs✓ Custom contracts & invoicing✓ Security & compliance review✓ Pentesting and load testing <p>Contact us</p>

Ada paket Starter, yang ini **gratis**. Tapi jika ingin berlangganan yang Pro juga bisa.

Saran saya, mulai dulu pakai paket Starter. Kalau ingin fitur yang lebih, baru upgrade ke Pro.

Oh iya, web petanikode juga pernah pakai yang starter dan sudah berjalan hampir selama 2 tahun.

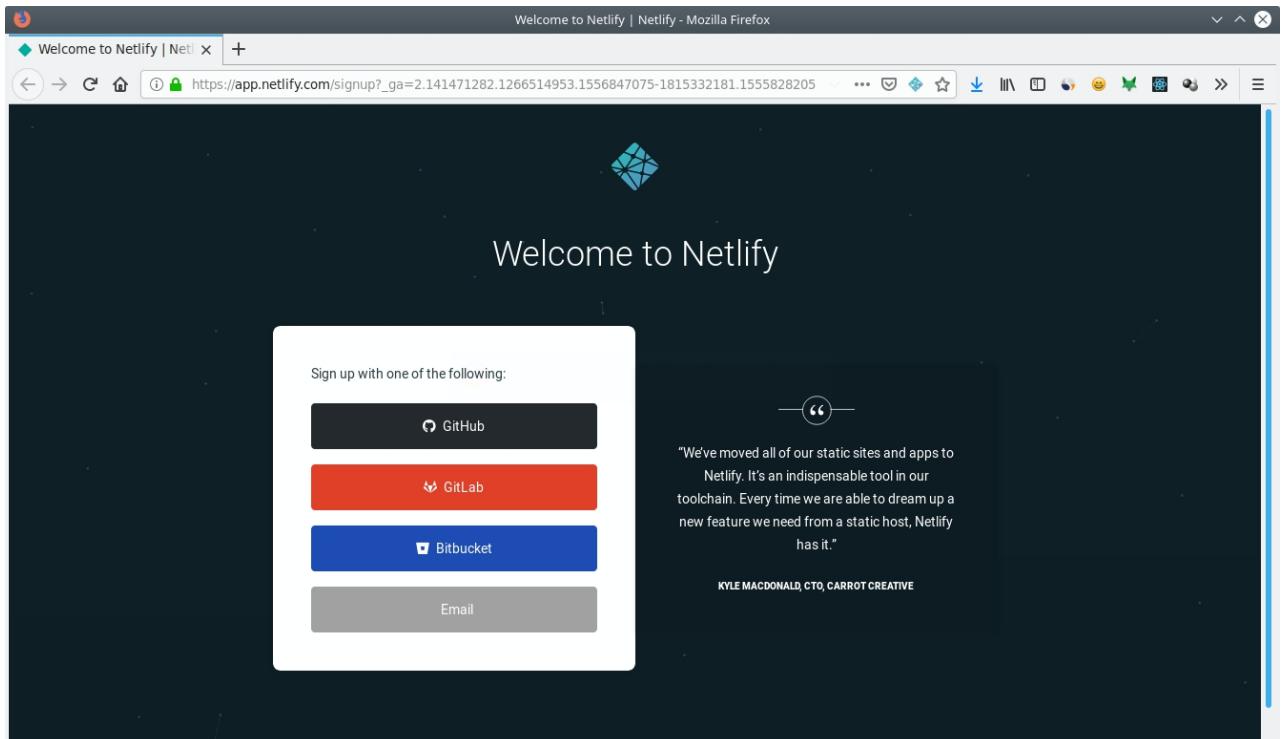
Tapi karena ramainya pengunjung membuat bandwidth-nya cepat habis. Netlify memberikan jatah 100GB bandwidth per bulan untuk paket starter.

Jika melewati batasan bandwidth tersebut, websitemu tidak akan bisa dibuka.

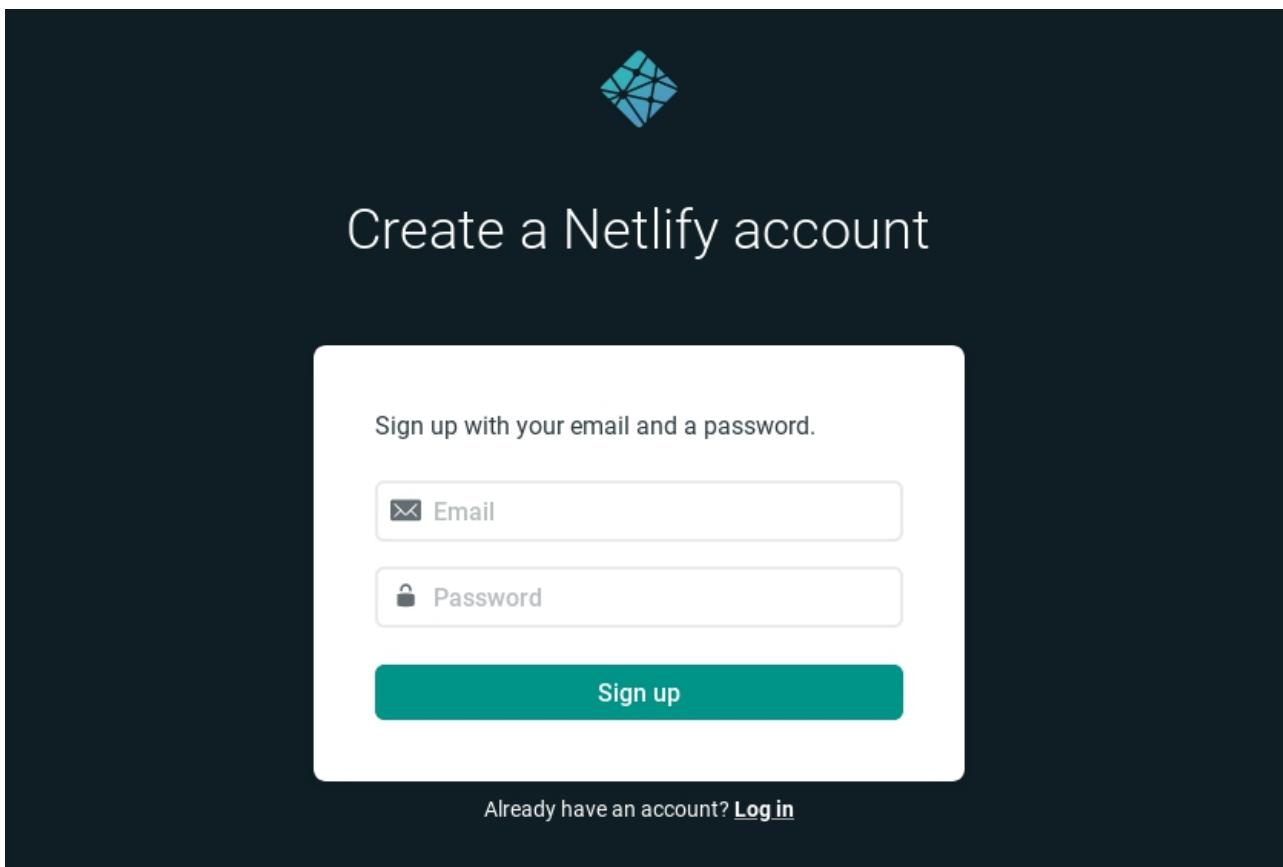
Cara Mendaftar Akun Netlify

Silahkan buka halaman [pendaftaran akun Netlify](#).

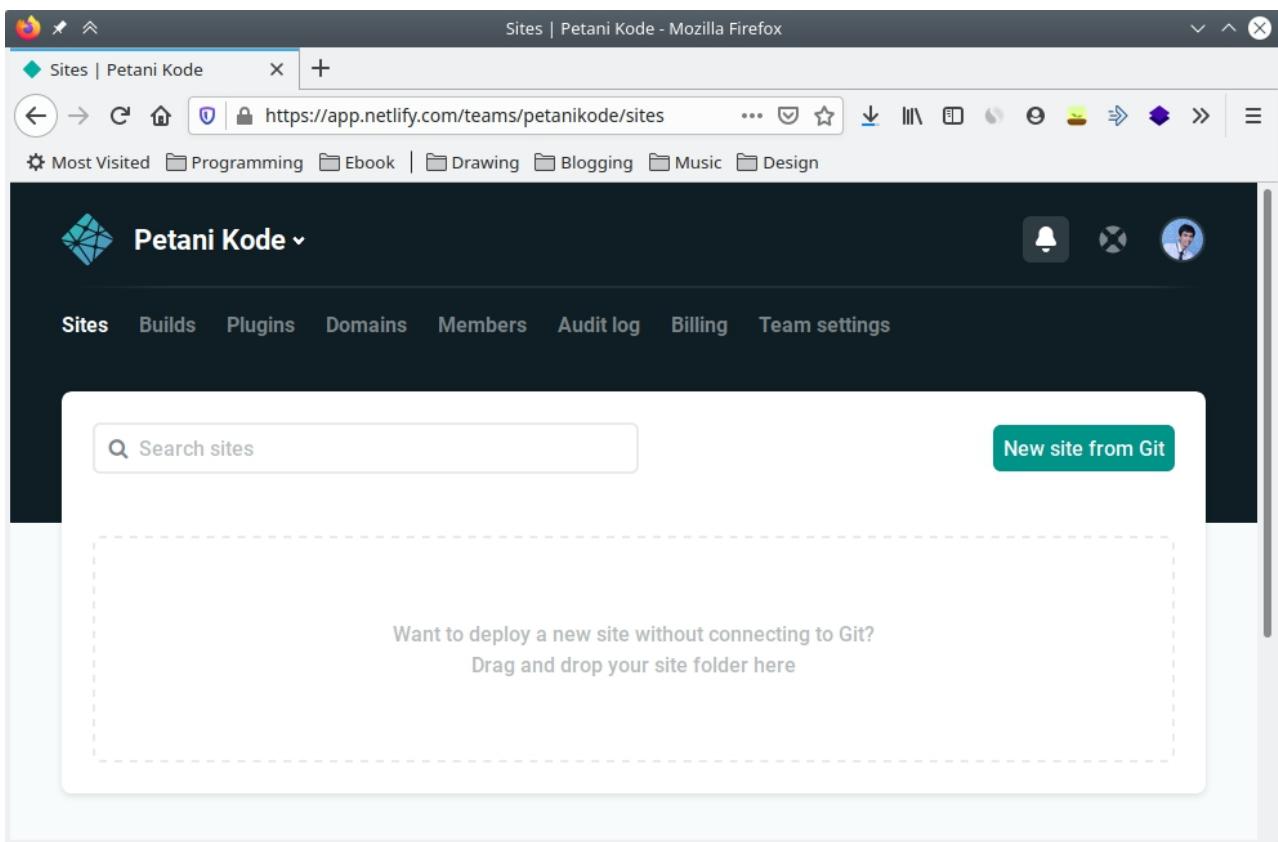
Kemudian, kita bisa daftar dengan akun Github, Gitlab, Bitbucket, dan Email. Silahkan pilih salah satu metode pendaftaran yang akan kamu gunakan.



Jika kita menggunakan email, maka kita akan diminta alamat email dan password.



Setelah sukses mendaftar, kita akan memiliki halaman dashboard seperti berikut.



Upload Website ke Netlify

Netlify tidak seperti shared hosting yang harus pakai FTP untuk upload website.

Di sini kita bisa gunakan dua cara:

1. Menggunakan Git
2. Drag/Drop

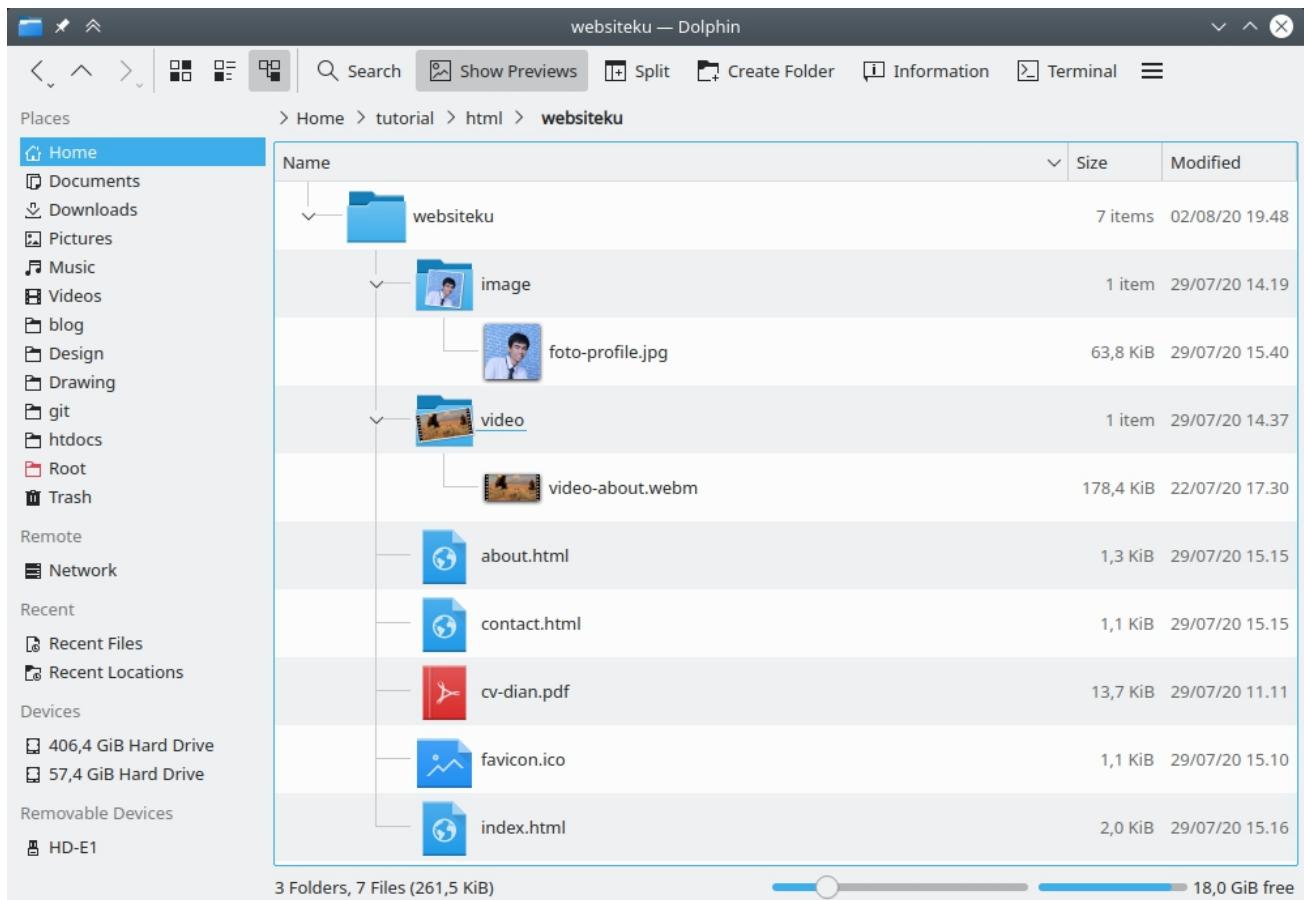
Yang paling gampang adalah **drag/drop**, jika kamu ingin menggunakan Git.. pastikan sudah paham dulu tentang git.

O ya, kamu bisa belajar Git di:

[Tutorial Git untuk Pemula](#)

Tapi, karena kondisi kita di sini belum tau Git.. maka kita akan gunakan teknik drag/drop.

Baiklah, siapkan terlebih dahulu website yang akan di-upload. Kita akan menggunakan **websiteku** yang sudah buat pada [project web di tutorial sebelumnya](#).



Silahkan drag/drop folder **websiteku** ke dashboard Netlify dan tunggu sampai proses upload selesai.

Tunggu sampai status production-nya menjadi **published**.

The screenshot shows the "Production deploys" section of the Netlify dashboard. It displays the following information:

- Production Published**
- 7:53 PM: No deploy message

Ini artinya, website kita sudah berhasil di-upload dan dipublikasi di Netlify.

Wow, benar kan..

Tidak sampai 30 detik, website kita sudah go online

Keren ya.

Jika kita perhatikan, website yang diupload akan mendapatkan alamat domain acak seperti silly-blackwell-3e7430.netlify.app.

Domain ini gratis diberikan oleh Netlify..

The screenshot shows the Netlify dashboard for a project named 'silly-blackwell-3e7430'. At the top, there's a navigation bar with tabs: Overview, Deployments, Plugins, Functions, Identity, Forms, Large Media, and Split Testing. Below the navigation, the project name 'silly-blackwell-3e7430' is displayed, followed by a URL link: <https://silly-blackwell-3e7430.netlify.app>. A red arrow points from the text '..dan kita juga bisa mengubahnya sesuai selera.' to this URL link. Below the URL, it says 'Manual deploys. Last published at 7:53 PM.' Underneath the URL, there are two buttons: 'Site settings' and 'Domain settings'. To the right of the main content area, there's a smaller preview window showing a user profile for 'Ahmad Muhardian'.

..dan kita juga bisa mengubahnya sesuai selera.

Sekarang coba klik alamat domain tersebut, maka kita akan bisa melihat hasil website yang baru saja kita upload.

Ahmad Muhardian Personal Website - Mozilla Firefox

Overview | silly-blackwell- X Ahmad Muhardian Person +

https://silly-blackwell-3e7430.netlify.app

Most Visited Programming Ebook | Drawing Blogging Music Design

[Home](#) | [Download CV](#) | [Contact](#) | [About me](#)



Ahmad Muhardian

(Web Developer)

Overview

Hi, saya adalah web developer yang berdomisili di Jakarta. Saat ini sedang belajar HTML di Petnai Kode

Skill	Pengalaman
<ul style="list-style-type: none">• HTML (Expert)• CSS (Beginner)• Javascript (Beginner)	<ul style="list-style-type: none">• Freelancer di Internet• Leader Local Linux Community• Leader Local Linux Community

Copyright © 2020 Ahmad Muhardian.

Gampang banget kaan...

Nah, sekarang kita akan coba mengubah nama domainnya agar mudah diingat.

Mengubah Nama Domain

Silahkan masuk ke **Domain Settings**.

silly-blackwell-3e7430

- <https://silly-blackwell-3e7430.netlify.app>

Manual deploys. Last published at 7:53 PM.

Site settings **Domain settings**

Getting started

Pada bagian sebelah kanan, klik **option** kemudian pilih **Edit site name**.

Domains

Use your own domain for your Netlify site for free

Custom domains

By default, your site is always accessible via a Netlify subdomain based on the site name. Custom domains allow you to access your site via one or more non-Netlify domain names.

- **silly-blackwell-3e7430.netlify.app**
Default subdomain

Options ▾

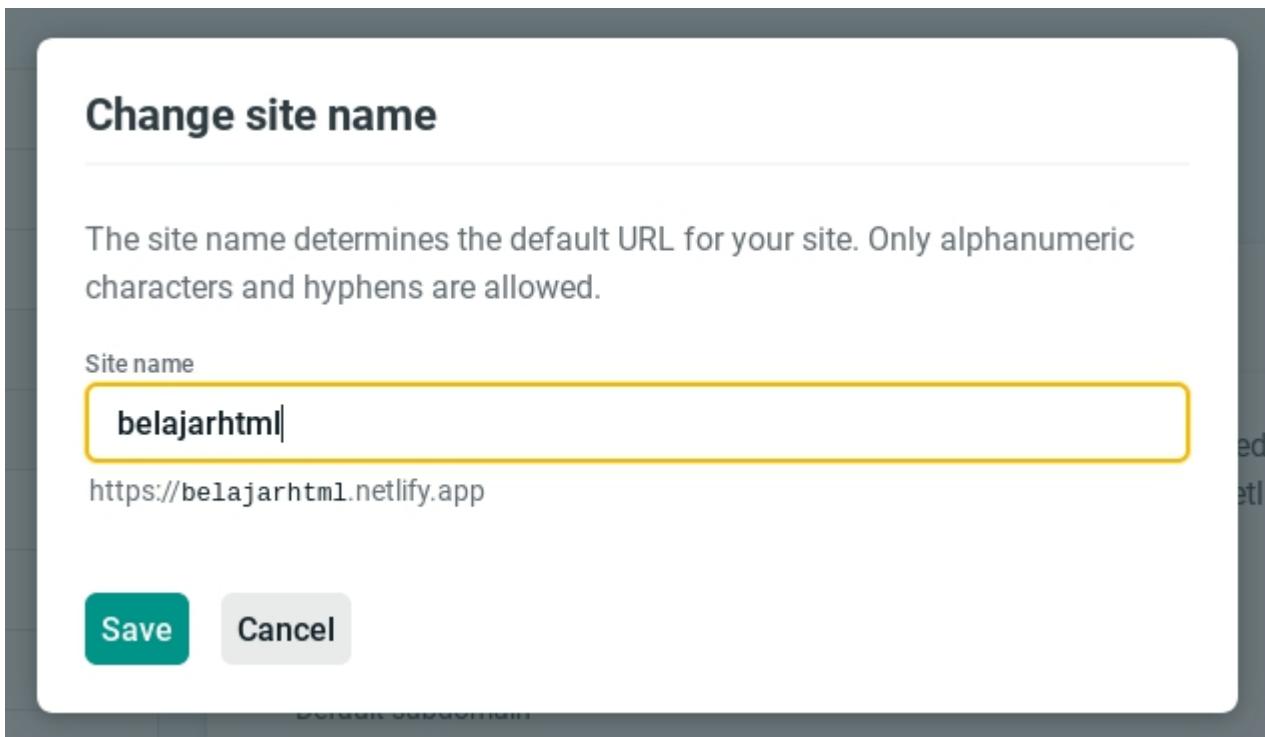
Edit site name

[Learn more about custom domains in the docs ↗](#)

Add custom domain

Pada jendela pop-up yang muncul, ubahlah nama domainnya dengan nama yang kamu inginkan.

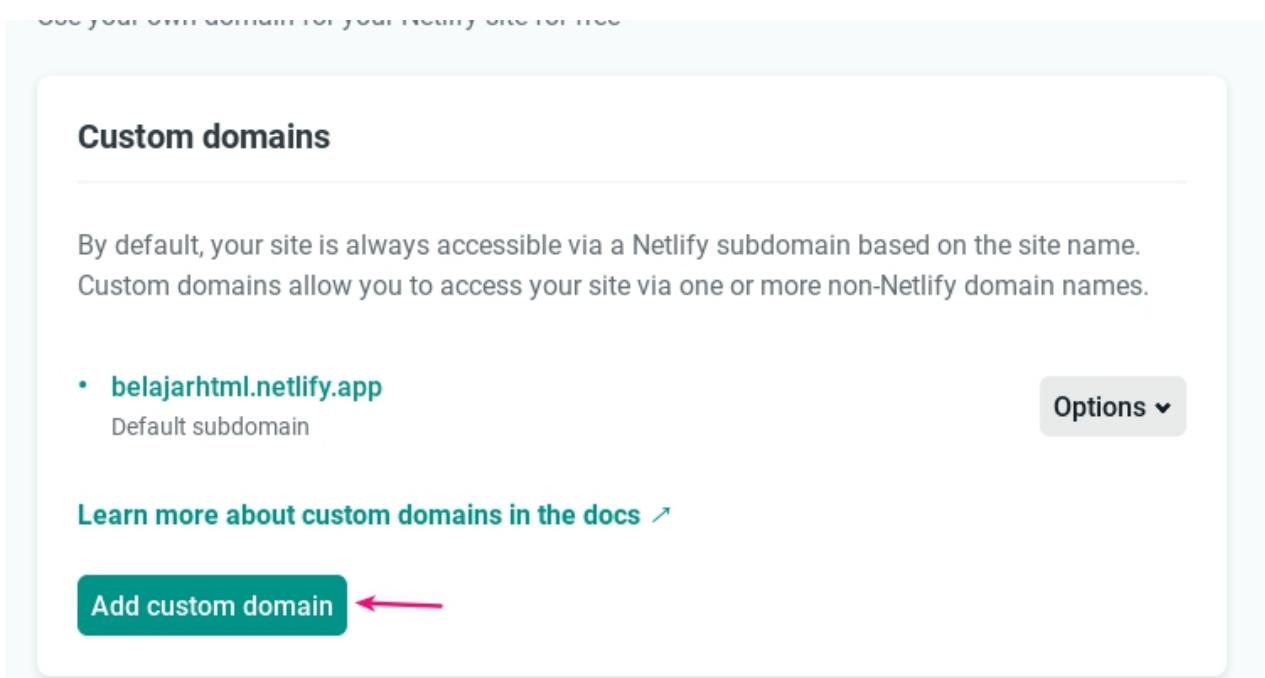
Sebagai contoh, saya akan mengubah namanya dengan **belajarhtml**.



Jika nama domain tersebut, belum diambil orang atau belum ada yang pakai.. maka saya akan bisa menggunakannya.

Jadi nanti websitenya akan bisa dibuka melalui <https://belajarhtml.netlify.app>.

Oh iya, kita juga bisa menambahkan custom domain dengan mengklik **add custom domain**.



Jika kamu sudah punya alamat domain, kamu bisa pointing alamat Name Server-nya ke server Netlify.

Cara Update Website di Netlify

Website sudah berhasil kita upload.

Tapi, bagaimana cara update jika ada perubahan?

Caranya sangat gampang..

Kita hanya perlu mengupload ulang folder websitenya.

Silahkan masuk ke menu **Deploys**.

The screenshot shows the Netlify dashboard for the 'Petani Kode > belajarhtml' site. The 'Deploy' tab is highlighted with a pink arrow pointing to it from the text 'Klik ini'. The page displays the 'Deploys for belajarhtml' section, which includes a link to the deployed site (<https://belajarhtml.netlify.app>). It also shows that 'Auto publishing is on' and 'Deploys are published automatically'. At the bottom, there are three buttons: 'Deploy settings', 'Notifications', and 'Stop auto publishing'.

Setelah halaman deploys terbuka, di sana kita bisa upload ulang websitenya.

Deploys for belajarhtml

- https://belajarhtml.netlify.app

Manual deploys.

Auto publishing is on. Deploys are published automatically.

Deploy settings Notifications Stop auto publishing

Deploys

Production Published Yesterday at 7:53 PM >

No deploy message

Drop Folder Website ke sini

Need to update your site?
Drag and drop your site folder here

High-Performance Edge

Website akan di-deploy ulang dengan yang baru.

Konfigurasi Netlify Form

Netlify form adalah fitur yang diberikan netlify untuk mengelola form. Fitur ini memungkinkan pengunjung web untuk mengisi form dan mengirimnya ke pemilik website.

Cara mengaktifkan netlify form sangat gampang, kita hanya perlu menambahkan atribut `netlify` pada tag `<form>`.

```
<form method="POST" netlify>
</form>
<!-- atau bisa juga --&gt;

&lt;form method="POST" netlify="true"&gt;
&lt;/form&gt;</pre>
```

Oke sekarang mari kita ubah lagi website yang sudah kita buat. Buka file `contact.html`, kemudian ubah kode di bagian tag `<form>` menjadi seperti ini:

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

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ahmad Muhardian Personal Website</title>

<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
<link rel="icon" href="favicon.ico" type="image/x-icon">
</head>

<body>
    <nav>
        <a href="index.html">Home</a> |
        <a href="cv-dian.pdf">Download CV</a> |
        <a href="contact.html">Contact</a> |
        <a href="about.html">About me</a>
    </nav>

    <hr />

    <div>
        <h1>Contact Me</h1> <form method="POST" netlify>
        <label for="email">Email</label><br />
            <input type="email" name="email" placeholder="alamat email" />
            <br />
            <label for="message">Pesan</label><br />
            <textarea name="message" placeholder="Tulis pesan anda..." rows="4" cols="80"></textarea>
            <br />
            <br />
            <input type="submit" value="Kirim" />
        </form>
    </div>

    <hr>
    <footer style="text-align: center;">
        <p>Copyright © 2020 Ahmad Muhardian.</p>
    </footer>
</body>

</html>

```

Setelah itu, upload ulang websitenya di halaman deploys. Tunggulah sampai status deploy-nya menjadi **published**.

The screenshot shows the 'Deploys' section of the Netlify dashboard. It lists two production deployments:

- Production Published**: Today at 12:19 PM, status: **Published**, message: No deploy message.
- Production**: Yesterday at 7:53 PM, status: **Pending**, message: No deploy message.

Nah, sekarang coba [buka websitenya](#) dan isilah form kontak.

Jika berhasil tampil halaman **Thank you!** berarti data yang kita inputkan berhasil terkirim ke pemilik website.

Untuk memastikan sudah terkirim atau tidak, kita bisa cek di menu form atau inbox dari email yang digunakan untuk mendaftar netlify.

The screenshot shows the 'Forms' section of the Netlify dashboard. It displays the following information:

- Forms**: Shows 1 form collecting data.
- Forms Level 0**: A green button.
- Active forms**: Shows 1 active form submission named "form 1".
- Last submission at 12:31 PM (a minute ago)**: The details of the most recent submission.

Two pink arrows point to specific elements: arrow 1 points to the 'Forms Level 0' button, and arrow 2 points to the 'form 1' entry in the 'Active forms' list.

Berikut ini contoh isi pesan yang kita terima dari form.

Verified submissions ▾[Delete](#)[Mark as spam](#)[Expand all](#) ini adalah percobaan form hehe

12:31 PM ⚡

Email dian@petanikode.com

Message ini adalah percobaan form hehe

Received today at 12:31 PM from 114.124.149.47

Sejauh ini website kita sudah berjalan dengan baik