Sejak HTML5, sudah ada cara langsung untuk bisa menambahkan video pada halaman web kita; yaitu dengan menggunakan tag <video>. Format video yang didukung antara lain: mp4, ogg, dan webM.

Syntaxnya cukup simpel, seperti di bawah ini:

<video width="320" height="240" src="link-ke-videonya"></video>

atau dengan mengapit element <source> di antara element <video> seperti ini

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

<source src="link-ke-video-nya" type="video/tipe-videonya"/>

</video>

Contoh penggunaannya:

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

<source src="https://go.aws/2W1IK22" />

</video>

Kita akan melihat hasilnya seperti ini

**HTML Video No Controls**

Lho tapi kok videonya tidak jalan? Dan tidak ada tombol play pula.

**Autoplay**

Untuk memulai videonya secara otomatis. Kita bisa tambahkan attribute autoplay di dalam element <video>-nya seperti ini

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

Dengan begitu videonya akan langsung dimulai sesaat setelah halaman web kita ditampilkan di browser.

**Controls**

Bagaimana dengan tombol play atau pause-nya?

Nah kalau kita ingin menambahkan tombol-tombol media tersebut, kita bisa tambahkan atribut controls seperti ini

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

Sekarang ada tombol play, pause, dan fullscreen kan?

HTML Video With Controls

**Poster**

Sebenarnya ada satu lagi attribute yang sering dipakai, yaitu poster.

Misalkan ukuran file video yang kita taruh itu besar dan membutuhkan waktu lama untuk di-load. Daripada hanya menampilkan layar hitam, kita bisa menyuruh element <video> untuk menampilkan sebuah gambar.

Misalnya kita ingin menampilkan gambar seekor kucing selagi video kita di-load, kita bisa melakukannya seperti di bawah ini

<video width="320" height="240" poster="https://skilvul-course-assets.s3-ap-southeast-1.amazonaws.com/lesson/intro-to-html/cat.jpeg" controls>

<source src="https://go.aws/3cKgiaX" />

</video>

Sekarang dari ketika videonya di-load sampai kita memulai, akan ditampilkan gambar seekor kucing. Seperti thumbnail video.

**HTML Video Poster Cat**

**Fallback Video Format**

Oh iya, seperti yang kita bahas di awal, kan ada beberapa format video yang didukung oleh HTML5. Masalahnya tidak semua browser mendukung format video yang sama:

ada yang hanya mendukung mp4 seperti Internet Explorer dan Safari,

ada juga yang mendukung semua format seperti Chrome dan Mozilla.

Bagaimana mengatasi masalah ini?

Kita bisa tambahkan beberapa element <source> di dalam <video> seperti ini

<video width="800" height="600">

<source src="link-ke-videonya" type="video/mp4" />

<source src="link-ke-videonya" type="video/ogg" />

<source src="link-ke-videonya" type="video/webm" />

</video>

Dengan begitu, andaikata browsernya tidak mendukung format mp4, dia akan beralih mencoba yang ada di bawahnya: yaitu ogg dan webm.

Mirip dengan video, HTML5 juga mendukung adanya element yang berisi suara/audio. Untuk menampilkan suara pada halaman sebuah website bisa menggunakan tag <audio>. Format yang didukung pun meliputi mp3, wav, dan ogg.

Syntax penulisannya juga ada dua seperti dengan tag <video>:

<audio controls src="link-ke-file-audio"></audio>

Atau menggunakan tag <source> yang diapit tag <audio>

<audio controls>

<source src="link-ke-file-audio" type="audio/tipe-audionya" />

</audio>

**Fallback Audio Format**

Contoh penggunaannya bisa seperti ini:

<audio controls>

<source src="cat\_meow.ogg" type="audio/ogg" />

<source src="cat\_meow.mp3" type="audio/mp3" />

<source src="cat\_meow.wav" type="audio/wav" />

Your browser does not support the audio element.

</audio>

Lho itu kenapa ada banyak element <source>-nya?

Dengan adanya beberapa <source> dengan type yang berbeda-beda, apabila sebuah browser tidak mendukung file audio yang kita berikan, dia akan secara otomatis membaca file yang berikutnya. Jadi jika kita ingin semua fitur website kita bisa diakses semua orang, sebaiknya kita memberi beberapa jenis file audio; jaga-jaga apabila file audionya tidak didukung.

C**ontrols**

Kalau kalian coba kode yang tadi, kalian bisa lihat ada tombol medianya seperti play, pause, dan timeline audionya. Itu karena kita memberikan attribute controls ke dalam element <audio>-nya.

**Loop**

Jika kita memberikan attribute loop di dalam element <audio> kita, suara yang dimainkan akan terus mengulang tiap kali selesai. Mungkin ini cocok untuk background music yang memang terus dimainkan tanpa henti selama pengguna terus berada di halaman web tersebut.

**Autoplay**

Dengan attribute autoplay, suara akan langsung dimainkan sesaat setelah halaman web selesai di-load. Seperti yang barusan kita bahas, mungkin attribute ini cocok untuk background music pada sebuah website yang tidak perlu dimainkan secara manual oleh pengguna.

Kita juga bisa menampilkan halaman website lainnya di halaman website yang kita buat dengan menggunakan tag <iframe>.

Contoh penggunaannya adalah seperti ini

<iframe></iframe>

**Width dan Height**

Jadi halaman html yang lain itu akan diperlakukan sebagai sebuah element dalam struktur file HTML kita: bisa diatur tinggi-lebarnya juga dengan menambahkan attribute width dan height seperti ini:

<iframe width="lebar iframe" height="tinggi iframe" ></iframe>

**Sumber Konten HTML**

<iframe> itu kan isinya konten HTML, kita bisa menambahkan attribute src untuk memberitahu konten apa yang harus ditampilkan.

Cara menambahkan attribute src ke dalam <iframe> adalah seperti berikut:

<iframe src="link-ke-konten-html"></iframe>

Sekarang halaman web kita akan menampilkan halaman web lain di dalamnya.

HTML Iframe Cat Website

**Scrolling**

Andaikata konten HTML yang kita masukkan ke dalam <iframe> kita itu lebih besar dari ukuran <iframe>-nya sendiri, secara default akan ada scrollbar untuk scroll halamannya.

Nah kita memiliki opsi untuk menghilangkan scrollbar tersebut dengan menambahkan attribute scrolling="no" di dalam <iframe> seperti ini:

<iframe scrolling="no"></iframe>

**Frameborder**

Kalau kalian lihat, di bagian pinggir <iframe> akan ada pinggiran berwarna abu-abu bukan? Kalau kalian merasa itu jelek atau tidak rapi, kalian bisa menambahkan attribute frameborder="0" di dalam

<iframe frameborder="0"></iframe>

**Untuk merangkum cara penggunaan semua attribute di <iframe>, kita bisa melihat contoh berikut:**

<iframe src="https://bit.ly/2JiCStI" width="200" height="300" scrolling="no" frameborder="0"></iframe>

Kode di atas akan menghasilkan <iframe> seperti berikut:

sumber konten yang mengarah ke halaman https://bit.ly/2JiCStI

lebar sebesar 200 dan tinggi sebesar 300

tidak mempunyai scrollbar apabila konten HTML di dalamnya lebih besar dari ukuran <iframe>

tidak mempunyai border yang membungkus element <iframe>

**abel**

**HTML Tag Untuk Tabel**

Sekarang, kita akan belajar cara menampilkan sebuah tabel pada halaman website.

Pada dasarnya, untuk membuat sebuah tabel di HTML cukup membutuhkan tiga tag, yaitu:

1. <table> sebagai element utama.
2. <tr> atau dikenal sebagai **table row** tag, digunakan untuk membuat baris baru di dalam <table>.
3. <td> atau dikenal sebagai **table data** tag, digunakan sebagai *container (wadah)* dari data yang kita mau isi di dalam <tr>.

📝**Catatan**:

Tidak ada pengecualian untuk tipe data yang bisa diisi di dalam tag <td>. Bisa dari tipe paling dasar yaitu sebuah teks, sampai dengan HTML element lainnya seperti <img>, <div>, <p> atau bahkan <table>.

Kita juga bisa menggunakan tag <th> sebagai pengganti <td> untuk membuat header cell (biasanya digunakan untuk menampilkan judul kolom).

Secara standar <th> membuat tulisan di dalamnya menjadi tebal.

Cara penggunaannya seperti ini:

<table>

<tr>

<th>Nama</th>

<th>Nomor Telpon</th>

<th>Negara</th>

</tr>

<tr>

<td>Sarah</td>

<td>0811111111</td>

<td>Indonesia</td>

</tr>

<tr>

<td>Sophia</td>

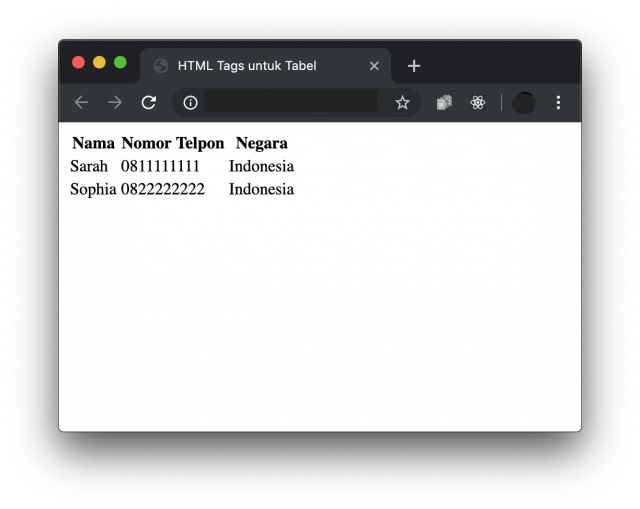
<td>0822222222</td>

<td>Indonesia</td>

</tr>

</table>

Pada browser, kode di atas akan terlihat seperti ini:



Kita juga bisa mengelompokkan <tr> di dalam suatu tabel dengan menggunakan tag table head (<thead>), table body (<tbody>) dan table foot (<tfoot>). Kita bisa gunakan semua atau salah satunya, sesuai kebutuhan.

Contoh penggunaan:

<table>

<thead>

<tr>

<th>Nama</th>

<th>Nomor Telpon</th>

<th>Negara</th>

</tr>

</thead>

<tbody>

<tr>

<td>Sarah</td>

<td>0811111111</td>

<td>Indonesia</td>

</tr>

<tr>

<td>Sophia</td>

<td>0822222222</td>

<td>Indonesia</td>

</tr>

</tbody>

<tfoot>

<tr>

<th>Nama</th>

<th>Nomor Telpon</th>

<th>Negara</th>

</tr>

</tfoot>

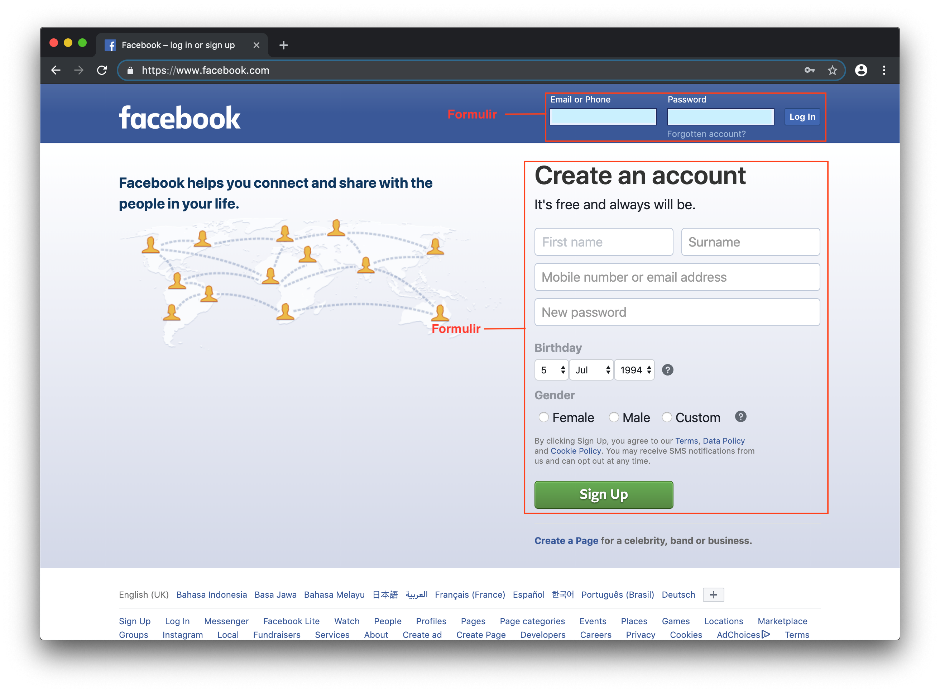
</table>

# Formulir

### HTML Tag Untuk Formulir

Pernah kah kalian membuka website Facebook?

Pertama kali saat kita ingin membuat akun Facebook, pasti kita akan mengisi data diri kita pada halaman formulir, seperti di bawah ini:



Jika kita perhatikan baik-baik, bagian formulir terdiri dari beberapa kotak untuk mengetik nama, nomor handphone, email, dan password. Kemudian juga ada tombol yang berguna untuk mengirim data yang sudah kita ketik di kotak tersebut.

Nah, pada sesi kali ini, kita akan belajar membuat kotak-kotak tersebut beserta tombolnya dan html tag/attribute apa saja yang dibutuhkan untuk membuatnya.

## Formulir

Tag <form> digunakan untuk mengawali pembuatan form.

Contoh penggunaannya:

<form action="/path">

<!-- Elemen pada formulir -->

</form>

## Input

Tag <input> digunakan untuk membuat kotak input dimana user akan mengisi data.

Tag input ada bermacam-macam bentuknya. Ada yang hanya bisa diisi dengan teks, ada yang hanya bisa diisi dengan angka, ada yang berbentuk checkbox, dan lain sebagainya. Untuk menentukan input seperti apa yang ingin kita dapatkan dari user, maka tag <input> harus diberi attribute type.

Di bawah ini adalah contoh-contoh dari tag <input> beserta bermacam-macam attribute type nya:

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="datetime-local">

<input type="email">

<input type="file">

<input type="hidden">

<input type="image">

<input type="month">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="week">

## Select dan Option

Pernahkah kalian melihat pilihan dropdown pada halaman sebuah website? Nah, pilihan tersebut menggunakan tag <select> dan <option> dalam membuatnya.

Contoh penggunaannya:

<select>

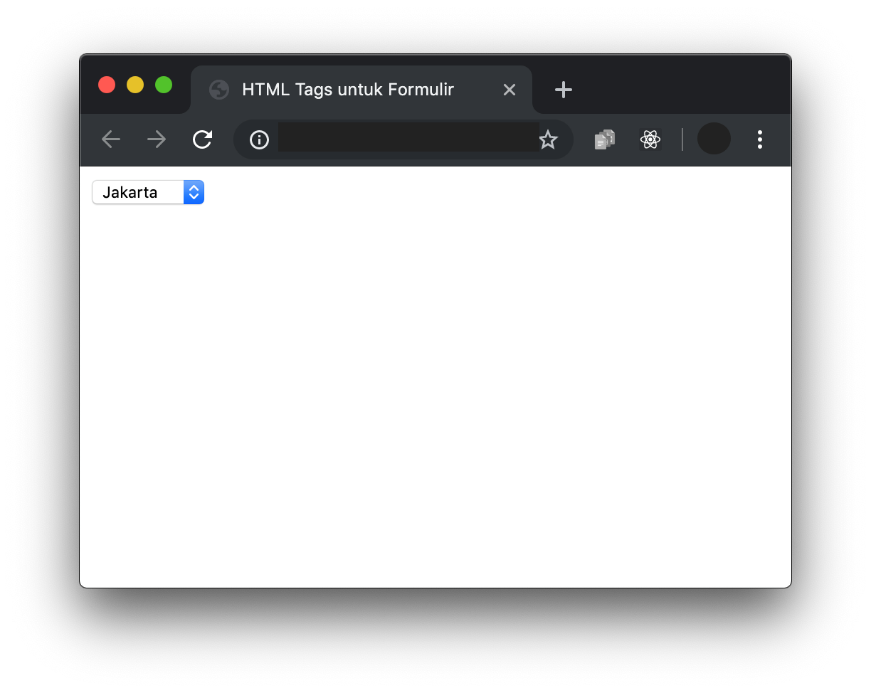
<option value="Jakarta">Jakarta</option>

<option value="Surabaya">Surabaya</option>

<option value="Bandung">Bandung</option>

</select>

Hasil kode di atas pada browser akan muncul seperti ini:



## Fieldset

Tag <fieldset> akan memberikan garis tepi pada element-element HTML yang ingin kita kelompokkan.

Contoh penggunaan:

<fieldset>

<label for="name">Nama: </label>

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

<label for="address">Alamat: </label>

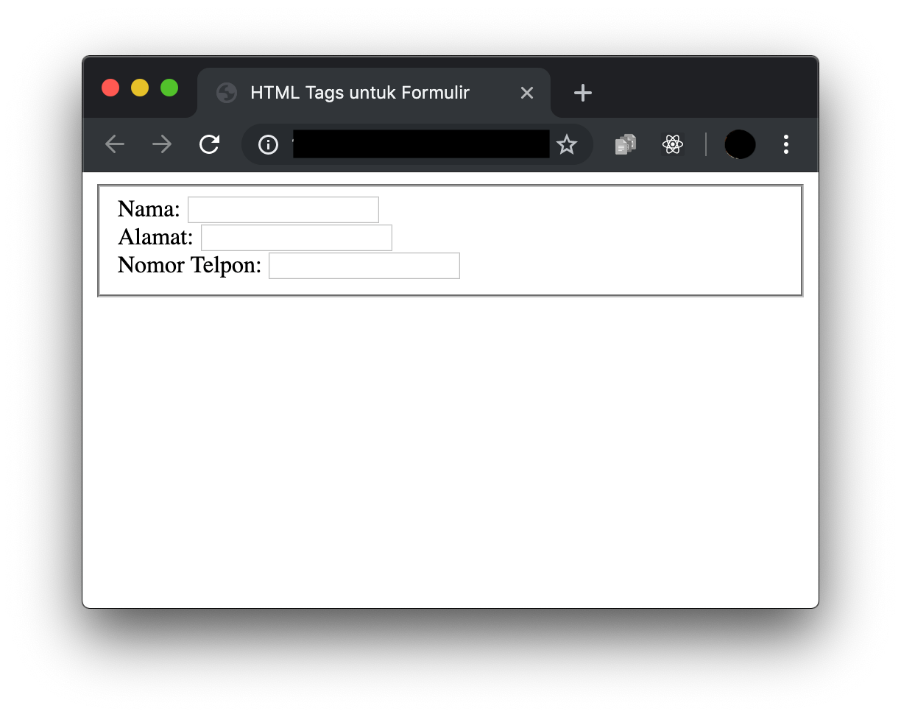
<input type="text" name="address" /><br>

<label for="phoneNumber">Nomor Telpon: </label>

<input type="text" name="phoneNumber" /><br>

</fieldset>

Hasil kode di atas pada browser akan terlihat seperti ini:



## Legend

Tag <legend> digunakan untuk memberikan keterangan pada element <fieldset>.

Contoh penggunaan:

<fieldset>

<legend>Data Pegawai</legend>

<label for="name">Nama: </label>

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

<label for="address">Alamat: </label>

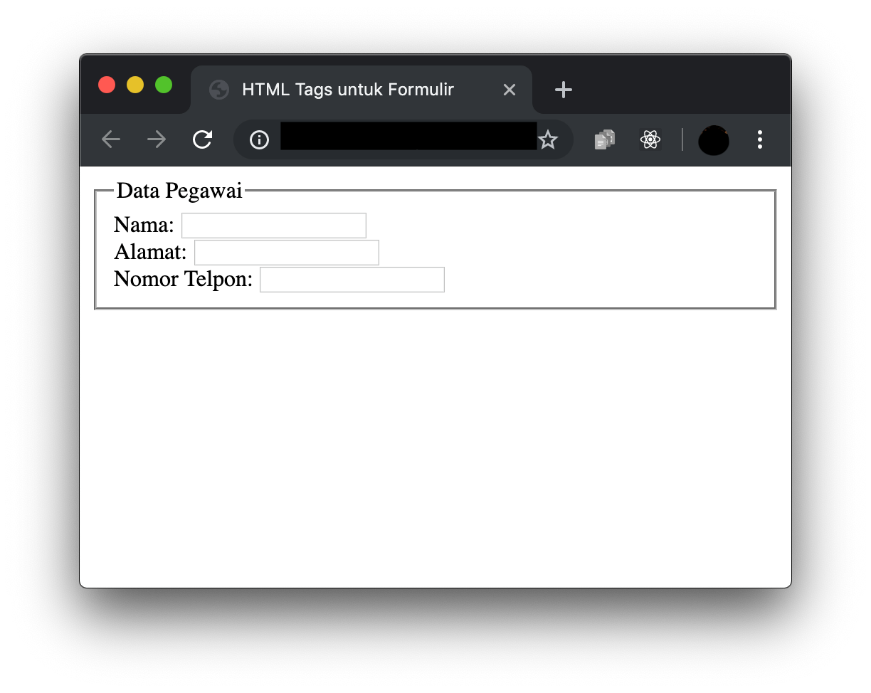
<input type="text" name="address" /><br>

<label for="phoneNumber">Nomor Telpon: </label>

<input type="text" name="phoneNumber" /><br>

</fieldset>

Kode di atas pada browser akan terlihat seperti:



## Text Area

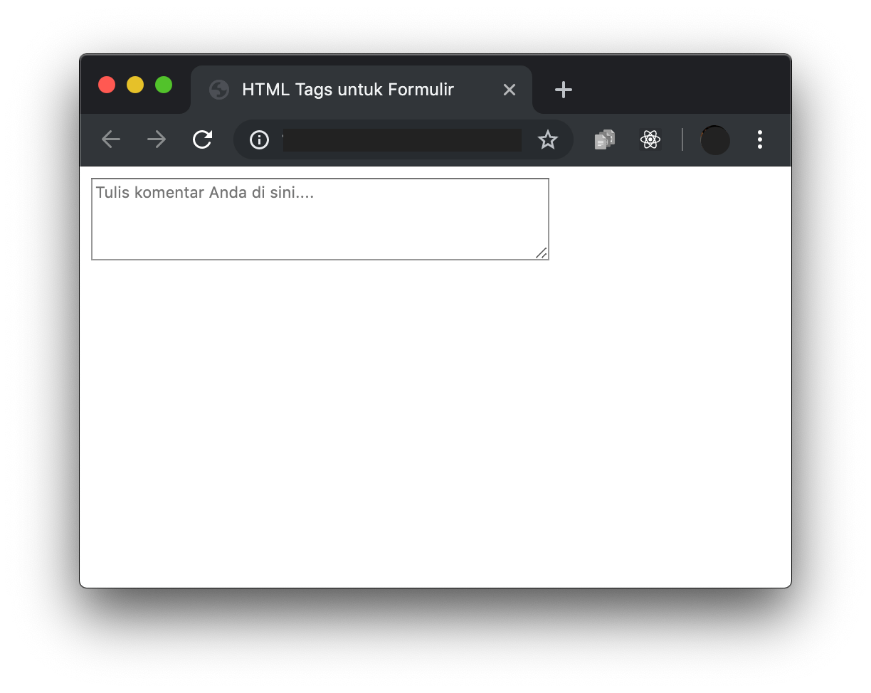
Pernahkah kalian mencoba menulis komentar di facebook maupun youtube? Nah, kalian pasti menulis di kolom komentar yang sudah disediakan. Kita juga bisa membuat kolom komentar tersebut dengan menggunakan tag <textarea>.

Tag <textarea> digunakan untuk menampung input dari user berupa teks yang lebih dari satu baris, dan mempunyai karakter yang sangat banyak (misalnya seperti input berupa paragraf).

Contoh penggunaannya:

<textarea rows="4" cols="50" placeholder="Tulis komentar Anda di sini...."></textarea>

Di browser, kode di atas akan terlihat seperti ini:



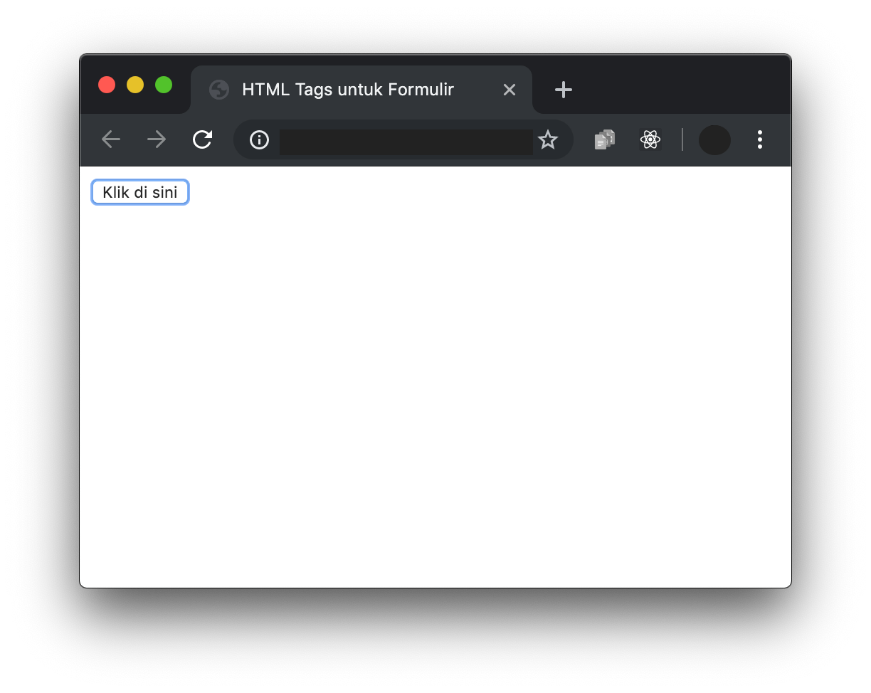
## Tombol

Untuk membuat sebuah tombol yang dapat diklik, maka kita memerlukan tag <button>, dan juga bisa menggunakan tag <input> dengan type button atau submit.

Contoh penggunaannya:

<button>Klik di sini</button>

Maka, kode di atas akan terlihat seperti ini:



## Contoh Formulir Sederhana

<!DOCTYPE html>

<html>

<head>

<title>Belajar Membuat Form pada HTML</title>

</head>

<body>

<form>

<fieldset>

<legend>Formulir Calon Pegawai</legend>

<p>Nama : <input type="text" name="nama"></p>

<p>Alamat : <input type="text" name="alamat"></p>

<p>Password : <input type="password" name="pass"></p>

<p>

Jenis Kelamin :

<input type="radio" name="jk" value="laki-laki" checked />

Laki - Laki

<input type="radio" name="jk" value="perempuan" />

Perempuan

</p>

<p>

Bahasa pemograman yang dikuasai :

<input type="checkbox" name="web"> Web

<input type="checkbox" name="mobile"> MOBILE

<input type="checkbox" name="desk"> DESKTOP

</p>

<p>

Kota asal :

<select>

<option>Semarang</option>

<option>Bandung</option>

</select>

</p>

<p>

Apa yang Membuat Kami harus memilih Anda?:<br>

<textarea rows='10' cols='60' placeholder="Tulis jawaban Anda di sini" ></textarea>

</p>

<p><input type="submit" value="proses"></p>

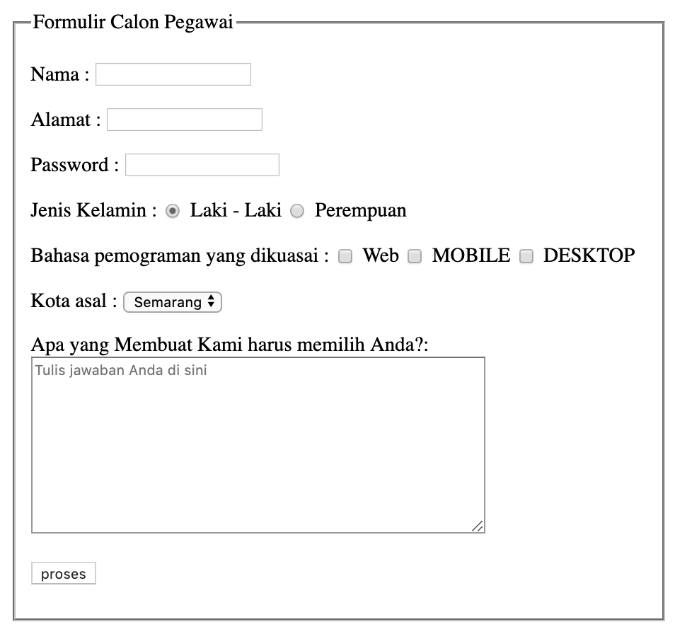
</fieldset>

</form>

</body>

</html>

Maka, kode di atas akan terlihat seperti ini:



# Layout Website

### HTML Tag Untuk Membagi Layout Website

Pada sesi kali ini, kita akan membahas layout dari suatu website.

Layout suatu website biasanya terdiri dari:

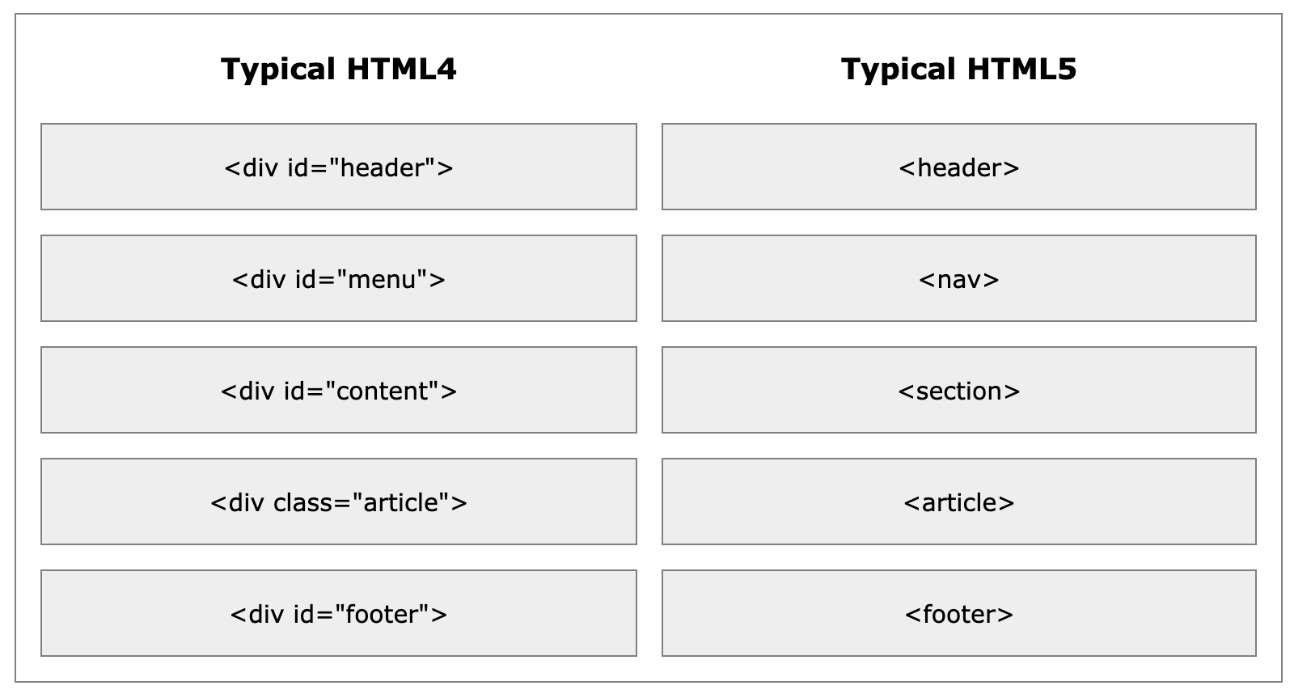
* Header
* Navigation bar
* Main Content
* Footer

Jika diilustrasikan, layout website kurang lebih akan terlihat seperti ini:

Sebelum adanya HTML5, penentuan layout pada halaman website hanya menggunakan tag <div> dan <span> dan dibantu dengan attribute id atau class untuk memperjelas element header, section, main, dan lain-lain. Namun di HTML5, layout pada website lebih diperjelas dengan element tertentu, sehingga memudahkan programmer untuk membaca kode. Penjelasan element ini dinamakan **semantic element**.

Semantic element menjelaskan tujuan dari element tersebut kepada browser dan pemrogram lain yang membaca kodenya. **Semantic element tidak mempengaruhi tampilan dari halaman web**, itu adalah fungsi dari CSS.

Supaya lebih jelas, berikut bandingkan **non-semantic** element (pada HTML4) dengan **semantic** element (pada HTML5):



Berikut adalah beberapa contoh dari semantic element:

* <section> menandakan bagian dalam sebuah halaman web.
* <header> merupakan bagian tajuk dari sebuah halaman web.
* <footer> merupakan bagian halaman web yang terletak di bagian bawah konten utama.
* <article> menandakan sebuah blok teks yang isinya independen terhadap element lain dalam halaman web.
* <nav>adalah bagian yang berisi tautan navigasi utama. Kalian mungkin sering melihat menu navigasi yang berisi tautan ke halaman "Beranda", "Kontak kami", "Galeri", dan lain-lain.
* <aside> adalah bagian di samping konten utama. Kontennya sebaiknya berhubungan dengan element di sebelahnya.

Contoh Penggunaan:

<html>

<head>

<title>Welcome to My Page</title>

</head>

<body>

<!-- nav digunakan untuk tautan navigasi utama -->

<nav>

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

<a href="./signup">Sign Up</a>

<a href="./login">Log In</a>

</nav>

<!-- header adalah bagian tajuk dari sebuah halaman web -->

<header>

<h1>This is my website</h1>

<h3>Please login to continue</h3>

</header>

<!-- aside adalah bagian samping konten utama -->

<aside>

<div>

Hello, anonymous!

</div>

<a href="./terms">Terms & Policy</a>

</aside>

<!-- article menandakan sebuah blok teks yang isinya independen terhadap element lain -->

<article>

<!-- section menandakan bagian dalam sebuah halaman web -->

<section>

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

</section>

<section>

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

</section>

</article>

<!-- footer merupakan bagian halaman web di bawah konten utama -->

<footer>2020 Copyright</footer>

</body>

<html>