

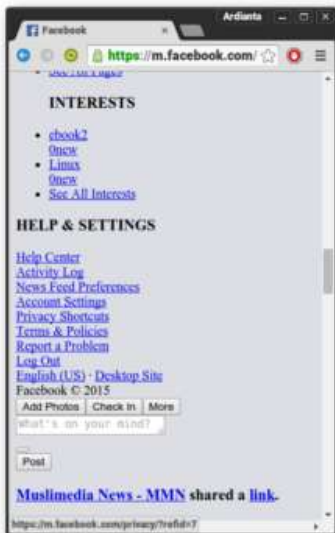
PENGENALAN CSS

CSS adalah bahasa kedua setelah **HTML** yang harus dipelajari seorang *web developer*.

Pengetahuan tentang CSS sangatlah penting bagi seorang web developer, karena dengan CSS kita bisa:

- Halaman *landing page* yang menarik;
 - Template atau tema blog;
 - Mengubah PSD (Desain web) menjadi HTML;
 - dan sebagainya.
-
- ZAContohnya, bagaimana tampilan facebook tanpa CSS?
 - Mungkin bisa seperti ini:

Tanpa CSS



Dengan CSS



Apa Itu CSS? Pengertian dan Cara Kerjanya

CSS adalah bahasa Cascading Style Sheet dan biasanya digunakan untuk mengatur tampilan elemen yang tertulis dalam bahasa markup, seperti [HTML](#). CSS berfungsi untuk memisahkan konten dari tampilan visualnya di situs.

CSS dibuat dan dikembangkan oleh **W3C** ([World Wide Web Consortium](#)) pada tahun 1996 untuk alasan yang sederhana. Dulu HTML tidak dilengkapi dengan tags yang berfungsi untuk memformat halaman. Anda hanya perlu menulis markup untuk situs.

Tags, seperti ****, diperkenalkan di HTML versi 3.2, dan ketika itu menyebabkan banyak masalah bagi developer. Karena website memiliki berbagai font, warna background, dan style, maka untuk menulis kembali (*rewrite*) kode memerlukan proses yang sangat panjang dan sulit. Oleh sebab itu, W3C membuat CSS untuk menyelesaikan masalah ini.

HTML dan CSS memiliki keterikatan yang erat. Karena HTML adalah bahasa markup (fondasi situs) dan CSS memperbaiki style (untuk semua aspek yang terkait dengan tampilan website), maka kedua bahasa pemrograman ini harus berjalan beriringan.

Tak hanya penting di sisi teknis, CSS juga berpengaruh pada tampilan sebuah website. Tentunya Anda tidak ingin website terlihat seperti situs yang ‘telanjang’ tanpa modifikasi apa pun, bukan?

Kelebihan CSS

Mudah bagi Anda untuk mengetahui mana website yang menggunakan CSS dan mana yang tidak.

Anda pasti pernah membuka website, tapi sesaat kemudian gagal loading dan malah menampilkan background putih yang didominasi teks hitam dan biru. Situasi ini menggambarkan bahwa CSS pada situs tersebut tidak dapat di-load dengan benar atau situs sama sekali tidak memiliki CSS. Umumnya, situs seperti ini hanya menerapkan HTML.

Sebelum menggunakan CSS, semua *styling* harus disertakan ke dalam markup HTML. Itu berarti Anda harus mendeskripsikan semua background, warna font, *alignment*, dan lain-lain secara terpisah.

Dengan CSS, Anda dapat mengatur tampilan semua aspek pada file yang berbeda, lalu menentukan *style*, kemudian mengintegrasikan file CSS di atas markup HTML. Alhasil, markup HTML bisa lebih mudah di-*maintain*.

Singkatnya, dengan CSS, Anda tidak perlu mendeskripsikan tampilan dari masing-masing elemen secara berulang-ulang. Anda tidak membuang-buang waktu, kode yang digunakan pun lebih singkat, dan error dapat diminimalisir.

Karena opsi kustomisasi yang ada hampir tak terbatas, CSS memungkinkan Anda untuk menerapkan berbagai macam *style* pada satu halaman HTML.

Cara Kerja CSS

CSS menggunakan bahasa Inggris sederhana berbasis syntax yang dilengkapi dengan sekumpulan *rule* yang mengaturnya. Seperti yang telah kami sebutkan sebelumnya, HTML tidak dibuat untuk menerapkan elemen *style*, hanya markup halaman saja. HTML dirancang semata-mata untuk mendeskripsikan konten. Sebagai contoh: **<p>This is a paragraph.</p>**.

Satu yang menjadi pertanyaannya sekarang, bagaimana caranya *style* paragraf? Struktur syntax CSS cukup sederhana. Struktur ini memiliki *selector* dan *declaration block*. Pilih elemen yang diinginkan, kemudian deklarasi (*declare*) yang harus Anda lakukan terhadap elemen tersebut. Sangat mudah, kan?

Ada berbagai *rule* yang harus diingat. Meskipun demikian, *rule* struktur cukup simpel dan sederhana.

Selector mengarah ke elemen HTML yang ingin Anda ubah tampilannya. *Declaration block* memuat satu atau lebih banyak deklarasi (*declaration*) yang dipisahkan dengan tanda titik koma.

Setiap deklarasi menyertakan nama dan value dari properti CSS, yang dipisahkan dengan tanda koma. Umumnya deklarasi CSS diakhiri dengan tanda titik koma, sedangkan declaration block dikelilingi oleh tanda kurung kurawal.

Mari kita lihat contoh di bawah ini:

Semua elemen **<p>** ditandai dengan warna biru dan hurufnya ditebalkan (*bold*).

```
<style>
  p {
    color: blue;
    text-weight: bold;
  }
</style>
```

Contoh lain, semua elemen **<p>** diposisikan di tengah (center-aligned), dengan lebar 16x, dan berwarna pink.

```
<style>
  p {
    text-align: center;
    font-size: 16px;
    color: pink;
  }
</style>
```

CSS Style Inline, Internal, dan External

Penulisan kode CSS dalam HTML dibagi menjadi tiga cara, internal, *inline*, dan eksternal. Pembagian ini berdasarkan letak kode CSS tersebut ditulis.

1. Internal CSS

Internal CSS adalah kode CSS yang ditulis di dalam tag `<style>`. Internal CSS juga dikenal dengan sebutan *Embedded CSS*.

Tag `<style>` biasanya ditulis di dalam tag `<head>`. Bisa juga ditulis di dalam `<body>`, namun lebih banyak ditulis di dalam `<head>`.

Contoh:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Contoh Internal CSS</title>
5   <!-- penulisan internal css dalam tag head -->
6   <style type="text/css">
7     {
8       font-family: serif;
9       line-height: 1.75em;
10      font-size: 18px;
11    }
12    {
13      font-family: sans;
14      color: orange;
15    }
16  </style>
17 </head>
18
19 <body>
20   <!-- penulisan internal css dalam tag body -->
21   <style type="text/css">
22     {
23       font-family: sans;
24       color: #333;
25     }
26   </style>
27   <h2>Ini judul artikel</h2>
28   <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonkan <code><!--internal css--></code>
29     Seperti namanya, <code><!--inline css--></code> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
30 </body>
31 </html>
32
```

Hasilnya:

Text justifikasi serifik

Text adalah perangkat yang membuat isi artikel. Perangkat ini berguna untuk menghasilkan output. Perangkat ini memiliki kemampuan untuk menghasilkan output yang berbeda-beda. Perangkat ini juga memiliki kemampuan untuk menghasilkan output yang berbeda-beda. Perangkat ini juga memiliki kemampuan untuk menghasilkan output yang berbeda-beda.

2. Eksternal CSS

Eksternal CSS adalah kode CSS yang ditulis terpisah dengan kode HTML. Eksternal CSS ditulis di sebuah file khusus yang berekstensi `.css`.

Sebagai contoh, saya akan membuat sebuah file bernama `style-ku.css`. Berikut ini cuplikan isi file `style-ku.css`

```
p {
  font-family: serif;
  line-height: 1.75em;
}

i {
  font-family: sans;
  color: orange;
}

h2 {
  font-family: sans;
  color: #333;
}
```

Untuk menggunakan CSS tersebut dalam HTML, kita perlu mengimpornya. Ada beberapa cara memasukkan kode CSS dari berkas eksternal:

Pertama menggunakan tag `<link>`

```
<link rel="stylesheet" type="text/css" href="style-ku.css">
```

Atau bisa juga menggunakan `@import`

```
<style type="text/css">
@import "style-ku.css";
</style>
```

Penulisan pada HTML versi lengkapnya seperti ini:

```

<!DOCTYPE html>
<html>
<head>
  <title>Contoh Eksternal CSS</title>
  <link rel="stylesheet" type="text/css" href="style-ku.css">
</head>
<body>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonkan <i>internal css</i>. Seperti namanya, <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
</body>
</html>

```

Hasilnya pun akan sama seperti contoh internal CSS, karena kode CSS-nya sama. Hanya saja berbeda tempat penulisannya.

3. Inline CSS

Inline CSS adalah kode CSS yang ditulis langsung pada atribut elemen HTML. Setiap elemen HTML memiliki atribut **style**, di sana lah inline CSS ditulis. Contohnya seperti ini:

```

<h2 style="color:red; font-family:sans;">Ini judul artikel</h2>

```

Contoh lengkap:

```

<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inline CSS</title>
</head>
<body>
  <h2 style="color:red;font-family:sans;">Ini judul artikel</h2>
  <p style="color:red;font-family:sans;">Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonkan <i>internal css</i>. Seperti namanya, <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file HTML.</p>
</body>
</html>

```

Hasilnya:

Ini judul artikel

Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemonkan *internal css*. Seperti namanya, *inline CSS* adalah kode CSS yang ditulis langsung dalam file HTML.

Pengertian Selector CSS

Selector adalah sebuah tag html yang digunakan pada css agar element tersebut dapat dimanipulasi atau ditambahkan gaya pada css, selector dapat berupa tag html itu sendiri, atau bisa juga berupa nilai dari atribut class atau id pada tag html.

Didalam selector terdapat dua bagian penting yang perlu dipahami, bagian tersebut adalah property dan value, antara kedua bagian tersebut adalah inti dari css, bagaimana sebuah element html memiliki gaya desain yang menarik, untuk dapat melakukan hal tersebut diatur oleh *property* dan *value*.

Apa itu Property?

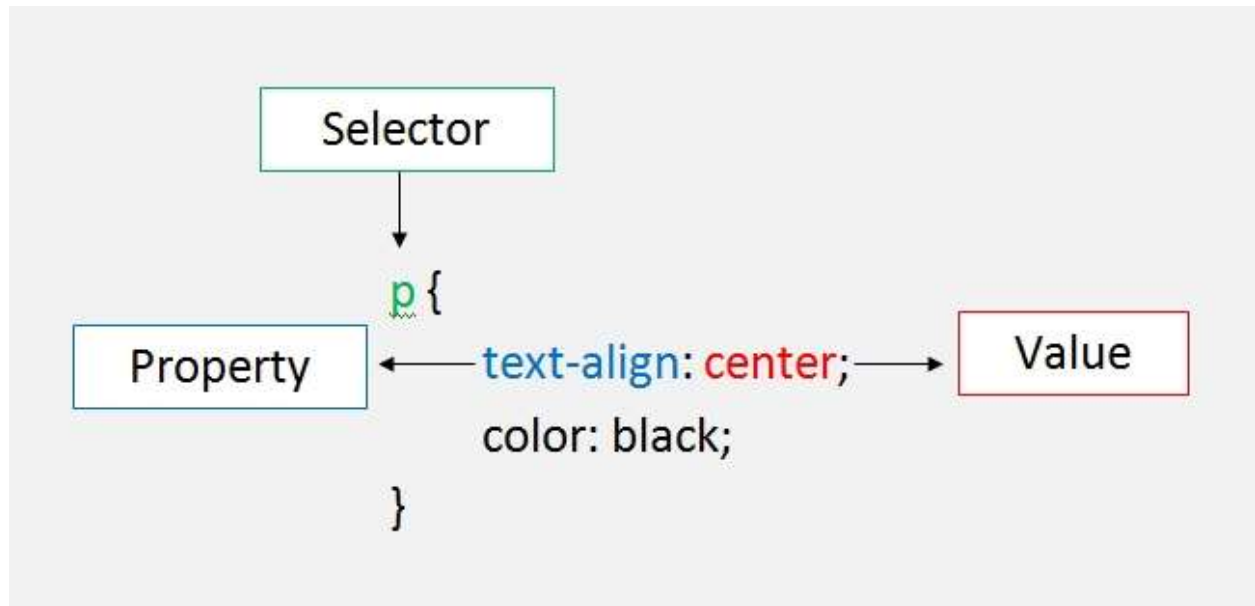
Property adalah jenis gaya, atau nama gaya untuk menentukan style pada tag html, dengan property lah sebuah tag html dapat diubah menjadi sifat css, misalnya seperti menambahkan color, background, text-align dan lain sebagainya. Banyak sekali property yang bisa digunakan untuk memanipulasi element html agar desain web lebih menarik.

Lalu, pengertian Value?

Value jika di bahasa indonesiakan artinya adalah nilai, ya.. sama saja seperti di css, value berfungsi untuk menentukan nilai dari property pada saat mendekrasi css, misalkan property color, maka value yang harus diberikan adalah nama warna, dapat rupa nama warnanya langsung atau juga berupa kode hexa. Contohnya sebagai berikut:


```
selector { property: value; }
```

Dari keterangan diatas seharusnya dapat mudah dipahami, namun penulis sudah menyiapkan gambar untuk memperjelas maksud dari keterangan tersebut, yaitu tentang pengertian selector, property dan value. Berikut ini gambarnya..



Gambar diatas seharusnya sudah lebih memperjelas keterangan sebelumnya, jadi bisa dipastikan belajar css seharusnya sangat mudah.

Memahami 5 Macam Selektor pada CSS

Hal dasar yang harus kita ketahui setelah memahami [cara menulis HTML di CSS](#) adalah selektor.

Selektor adalah katakunci dan simbol yang digunakan pada CSS untuk menyeleksi atau memilih elemen HTML.

Ada 5 macam selektor di CSS:

1. Selektor Tag
2. Selektor Class
3. Selektor ID
4. Selektor Atribut
5. Selektor Universal

1. Selektor Tag

Selektor Tag disebut juga *Type Selector*. Selektor ini akan memilih elemen berdasarkan nama tag.

Contoh:

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

Artinya: Pilih semua elemen `<p>` lalu atur warna teksnya menjadi biru.

Kalau misalnya kita punya struktur HTML seperti ini:

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

Maka akan menghasilkan output seperti ini:

Helo apa kabar, saya sedang belajar CSS

Materi yang sedang kita bahas adalah Selektor

Saat ini sedang dibahas tentang selektor tag

Selektor tag adalah selektor untuk memilih elemen HTML berdasarkan nama tagnya

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

2. Selektor Class

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

Contoh:

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

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

Contoh:

```
<p>Saya sedang belajar <b class="pink">CSS</b> di petanikode.com.</p>
```

Hasilnya:

Saya sedang belajar **CSS** di petanikode.com.

Selektor class dapat kita gunakan pada elemen yang kita inginkan.

Sebuah elemen HTML dapat menggunakan satu atau lebih class.

Contoh:

```
.text-white {  
  color: white;  
}  
  
.bg-teal {  
  background: teal;  
}
```

Lalu di HTML kita gunakan kedua class tersebut pada satu elemen:

```
<h2 class="text-white bg-teal">Tutorial CSS untuk Pemula</h2>
```

Hasilnya:

Tutorial CSS untuk Pemula

3. Selektor ID

Selektor ID hampir sama dengan class. Bedanya, ID bersifat unik. Hanya boleh digunakan oleh satu elemen saja.

Selektor ID ditandai dengan tanda pagar (#) di depannya.

Contoh:

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

Lalu pada kode HTML:

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

Hasilnya:

A screenshot of a web browser window. The browser's address bar is empty. The main content area shows a teal-colored header bar. Inside this bar, the text "Selamat Datang di Website Saya" is displayed in a white, serif font, centered horizontally. The text is bold. To the right of the teal bar, there is a vertical scrollbar with a grey track and a white slider.

4. Selektor Atribut

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

Contoh selektor Atribut:

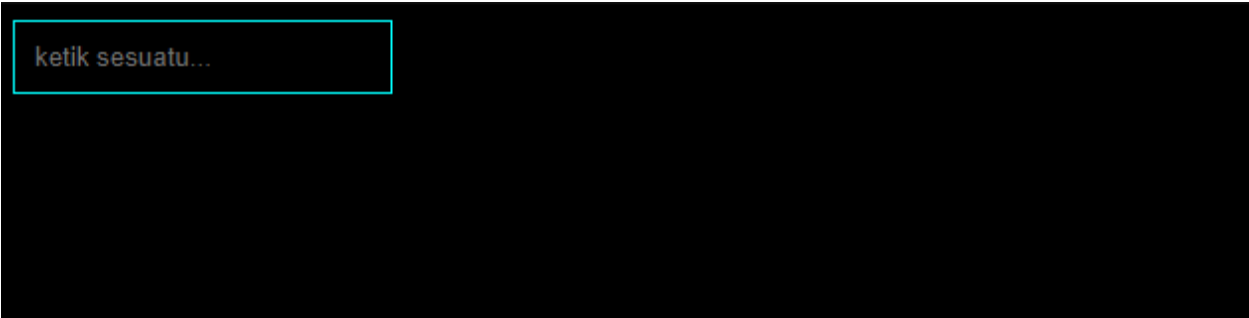
```
input[type=text] {  
  background: none;  
  color: cyan;  
  padding: 10px;  
  border: 1px solid cyan;  
}
```

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

Contoh kode HTML:

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

Hasilnya:



5. Selektor Universal

Selektor universal adalah selektor yang digunakan untuk menyeleksi semua elemen pada jangkauan (*scope*) tertentu.

Contoh:

```
* {  
  border: 1px solid grey;  
}
```

Artinya semua elemen akan memiliki garis **solid** dengan ukuran **1px** dan berwarna **grey**.

Hasilnya:

Helo apa kabar, saya sedang belajar CSS
Materi yang sedang kita bahas adalah Selektor
Saat ini sedang dibahas tentang selektor tag
Selektor tag adalah selektor untuk memilih elemen HTML berdasarkan nama tagnya

Selektor universal biasanya digunakan untuk me-reset CSS.

Kenapa harus di-reset?

Pada halaman HTML, ada beberapa CSS bawaan browser seperti **padding** dan **margin** pada elemen tertentu.

Reset bertujuan untuk menghilangkan **padding** dan **margin** tersebut.

Contoh CSS reset:

```
* {  
  padding: 0;  
  margin: 0;  
}
```

Maka semua elemen tidak akan memiliki *padding* dan *margin*.

Atribut pada CSS beserta Fungsinya

Dalam css banyak sekali atribut-atribut yang terdapat maka sebab itu anda harus tau apa-apa saja atribut di dalamnya dan juga pengrtian dan fungsinya supaya lebih mudah dalam belajar CSS. Dan berikut di bawah ini atribut beserta fungsinya:

- * animation : untuk menambahkan efek animasi pada website.
- * azimuth : digunakan untuk memberikan kesan bahwa suara tersebut berasal dari arah horisontal tertentu.
- * font-size : untuk menentukan ukuran font.
- * font-height : digunakan untuk ketebalan huruf.
- * font-family : digunakan untuk memilih jenis font.
- * font-style : digunakan untuk membuat font memiliki tampilan teks
- * text-decoration : digunakan untuk mendekorasi text
- * direction : digunakan untuk direksi teks
- * text-shadow : digunakan untuk menambah bayangan pada teks
- * text-transform : digunakan untuk menentukan besar kecil nya huruf.
- * letter-spacing : digunakan untuk memberika tambahan space antara kata.
- * word-spacing : digunakan untuk memberikan tambahan space antara kata.
- * line-height : digunakan untuk menentukan tinggi baris diantara text.
- * text-align : digunakan untuk mengatur posisi text pada element secara horizontal.

- * vertical-align : untuk mengatur warna pada teks yang dibuat
- * text-indent : digunakan untuk mengatur indent baris pertama
- * color : digunakan untuk mengatur warna pada teks yang dibuat.
- * background-color : digunakan untuk menentukan warna latar belakang element html.
- * background-image : digunakan untuk menentukan lokasi file yang akan dijadikan background image.
- * background-position : digunakan untuk menentukan posisi dari background.
- * background-repeat : digunakan untuk perulangan penampilan image.
- * background-attachment : digunakan untuk menentukan apakah background dapat di scroll bersama dokumen atau tidak.
- * border : property ini digunakan untuk membuat bingkai disekitar element.
- * border-width : digunakan untuk menentukan ketebalan bingkai.
- * border-style : digunakan untuk menentukan border bingkai.
- * border-color : digunakan untuk mengatur warna pada bingkai.
- * border-top : digunakan untuk mengatur jenis style border pada bagian bawah selector.
- * border-bottom : digunakan untuk mengatur jenis style border pada bagian atas selector.
- * border-left : digunakan untuk mengatur jenis style border pada bagian kiri selector.
- * border-right : digunakan untuk mengatur jenis style border pada bagian kanan selector.

* border-collapse : menetapkan apakah sel tabel harus memiliki perbatasan mereka sendiri atau berbagi perbatasan.

* list-style : digunakan untuk memilih bentuk list yang akan digunakan pada point-point teks.

* list-style-image : digunakan untuk mengganti bentuk list standar dengan image atau ikon sesuai keinginan kita.

* list-style-position : digunakan untuk mengatur posisi list apakah secara inside. maka list akan ditampilkan kedalam. sedangkan outside. list akan ditampilkan ke luar.

*.clear : Digunakan untuk menghentikan elemen dari membungkus seluruh elemen mengambang.

*.clip : Menentukan area yang akan diberikan suatu efek.

*.content : Digunakan untuk memasukkan isi sebelum ataupun sesudah elemen.

*.cursor : Digunakan untuk menentukan jenis kursor yang akan digunakan.

*.orphans : Menentukan jumlah minimum baris paragraf yang harus ditinggalkan di bagian bawah halman ketika halaman web dicetak.

*.overflow : Menentukan apakah isi dari elemen block level dipotong ketika lebih besar dari elemen induk.

*.pitch : Menentukan rata-rata pitch (frekuensi) dari suara yang berbicara.

*.stress : Mengontrol jumlah infleksi yang dihasilkan dari penanda stres.

*.table-layout : Menentukan lebar dari table yang akan dihitung.

*.visibility : Menentukan apakah element tersebut terlihat.

*.volume : Digunakan untuk menentukan rata-rata dari suara volume.

*.height : Menentukan tinggi suatu elemen.

*.width : Menentukan lebar dari suatu elemen.

*.margin : Digunakan untuk memberikan spasi disisi luar element.

*.padding : Digunakan untuk memberikan spasi pada sisi dalam sebuah element.

* z-index : untuk mengatur posisi depan belakang suatu element atau untuk mengatur susunan layer pada browser.

Kurang Lebih hanya segitu kode yang dapat saya berikan kepada sobat sekalian. Silahkan sobat visit w3school jika ingin belajar css lebih dalam, selain css di w3school juga terdapat beberapa bahasa pemrograman selain css ada html, javascript, php, sql dll. Cukup sampai disini Tips yang dapat saya bagikan tentang Daftar Atribut Dalam CSS Beserta Fungsi Nya Semoga yang telah saya sampaikan di atas bisa bermanfaat untuk sobat sekalian.

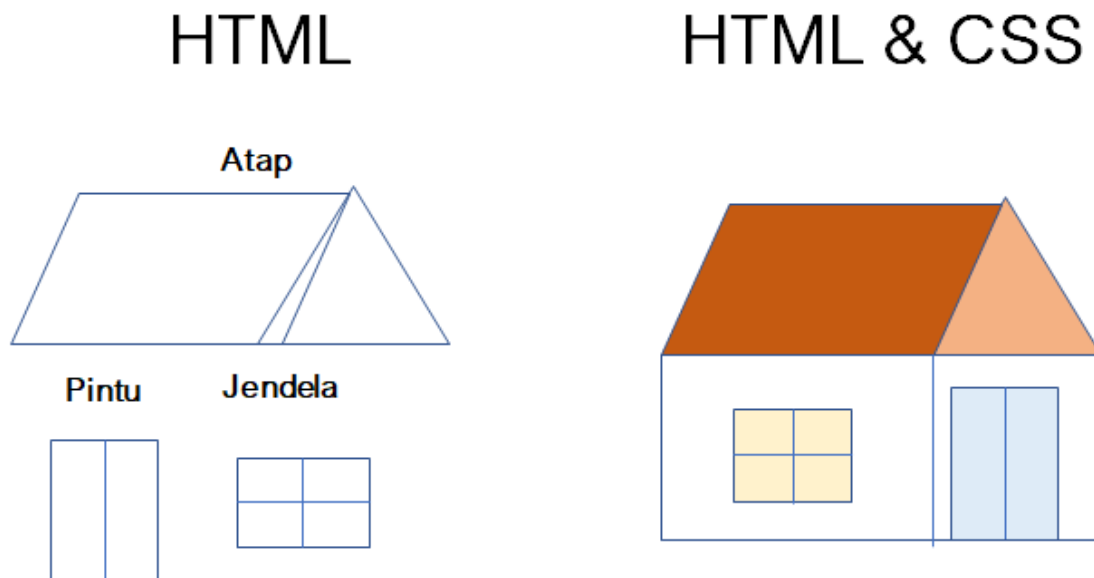
Cara Membuat Website dengan HTML dan CSS

Sebetulnya saat ini untuk membuat website itu sangat mudah, bahkan ada beberapa CMS yang telah menyediakan fitur website builder sehingga kita bahkan tidak perlu sedikitpun untuk ngoding. Namun sangat baik jika kalian memahami cara membuat website dari nol tanpa bantuan tools atau CMS yang ada saat ini.

Secara umum website yang sering kita jumpai di internet menggunakan bahasa markup HTML dan CSS (*Cascading Style Sheet*) serta bahasa pemrograman javascript dan PHP. Namun untuk membuat website sederhana bersifat statis cukup dengan HTML dan CSS saja.

Kenapa Harus Belajar CSS Juga?

Untuk menjawab pertanyaan ini perhatikan ilustrasi gambar di bawah ini:



Saat membangun sebuah website kita ilustrasikan seperti membuat kerangka bangunan rumah, html ibaratnya komponen-komponen untuk membangun sebuah

rumah sedangkan CSS akan membuat komponen-komponen tersebut menjadi satu kesatuan yang utuh sehingga rumah tersebut bisa dibangun dengan pas dan menarik.

CSS kita gunakan untuk mengatur jarak antar elemen, ukuran (tinggi, lebar), ukuran dan jenis font, warna, efek hover dan masih banyak lagi.

Jadi kita tidak bisa hanya memakai html saja untuk membuat website, kita membutuhkan css juga.

Oke, Saya harus mulai dari mana ?

Untuk mulai langkah awal kita butuh *tools* untuk coding dan menjalankan hasil codingan. Secara sederhana yang kita butuhkan hanya teks editor dan browser.

1. Teks Editor



Notepad



Notepad++



Sublime Text

Teks editor ini akan menjadi *tools* untuk kalian membuat kode-kode HTML dan CSS yang di butuhkan untuk membangun sebuah halaman website.

Notepad menjadi aplikasi yang paling standar dan sudah tersedia di platform windows. Jika kalian ingin yang profesional sebaiknya kalian pakai notepad++ atau [sublime text](#)

3. Dari ketiganya yang paling saya rekomendasikan adalah sublime text 3.

Semua aplikasi di atas bisa kalian download secara gratis di internet.

2. Browser

Website html yang kita buat melalui teks editor selanjutnya akan kita jalankan di aplikasi browser.



Google Chrome



Mozilla Firefox

Browser yang saya rekomendasikan adalah **chrome** dan **firefox** kalian bisa memilih salah satunya.

Mengenal Struktur Dasar Dokumen HTML

Secara umum struktur dasar dari suatu dokumen HTML adalah seperti berikut:

```
<!DOCTYPE html>

<html>
<head>
    <title>Membuat Web dengan HTML dan CSS - Kelasprogrammer.com</title>
</head>
<body>
    <!--Elemen yang ditampilkan di halaman browser -->
```

```
</body>
```

```
</html>
```

Di dalam HTML versi ke-5 setiap membuat dokumen html harus diawali dengan sintak **<!DOCTYPE html>** sintak ini berfungsi untuk memberitahu kepada browser bahwa dokumen html yang dibuat menggunakan versi ke-5, fungsi lain sekaligus membantu browser untuk menampilkan setiap elemen html dengan baik (karena menggunakan versi terbaru).

Selanjutnya dokumen HTML selalu di mulai dengan tag pembuka **<html>** dan kemudian di akhiri dengan penutup **</html>**

Setiap tag di html di desain memiliki pasangan, ada tag pembuka dan tag penutup, walaupun ada juga beberapa tag yang single (berdiri sendiri) seperti **
** untuk membuat baris baru **<hr>** untuk membuat garis horizontal dan lainnya.

Tag selanjutnya adalah tag **<head>** yang berfungsi sebagai heading dari dokumen html yang kita buat. Pada bagian **<head>** bisa kita buat judul dokumen, inisialisasi css, deskripsi dan meta tag lainnya.

Di dalam tag **<head>** terdapat tag **<title>** untuk membuat judul yang nantinya akan di tampilkan pada menu tab di browser.

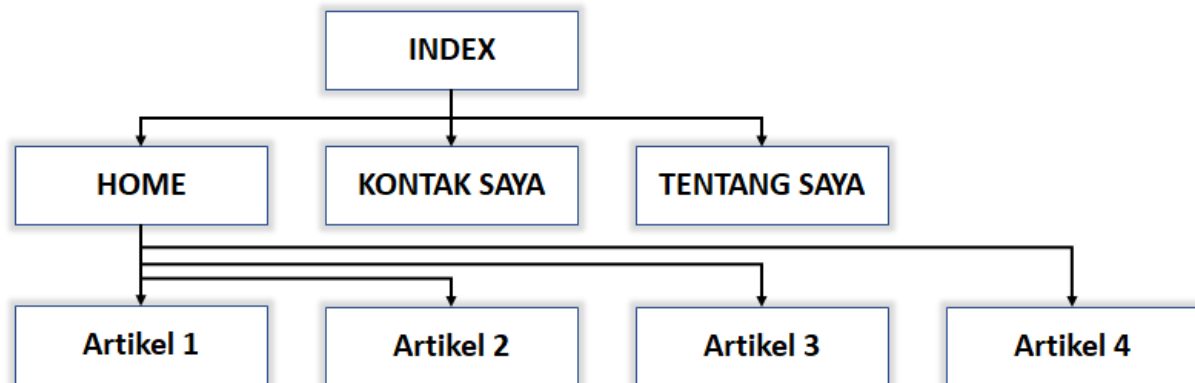
Bagian utama dalam struktur dokumen html adalah bagian **<body>** karena di bagian ini berkaitan dengan elemen yang tampil dan di lihat oleh pengguna, kita akan banyak membuat kode-kode html nya.

Menentukan Struktur navigasi

Struktur navigasi web adalah adalah suatu struktur untuk menggambarkan bagaimana suatu halaman di dalam website yang saling berhubungan antara satu dengan lainnya.

Penting untuk membuat struktur navigasi sebelum kita membuat website. Berikut ini contoh navigasi halaman website yang akan kita buat:

STRUKTUR NAVIGASI HALAMAN WEBSITE



Saat pertama kali pengguna mengakses website, halaman index akan menjadi halaman utama yang terdiri dari menu home, kontak saya dan tentang saya, halaman home bisa kita gunakan sebagai halaman index yang berisi konten artikel-artikel pada website kita.

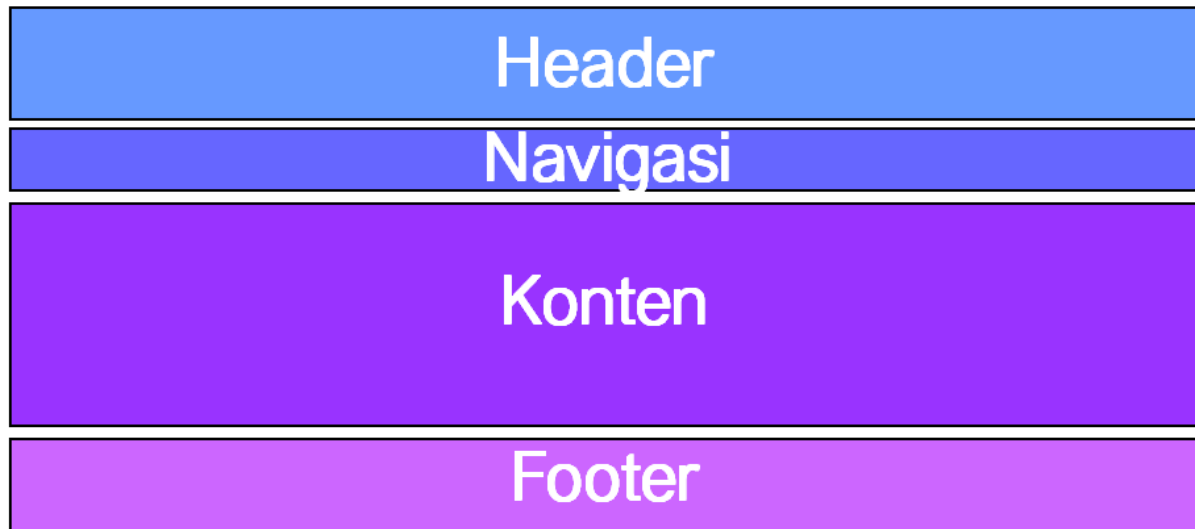
Membuat Layout Website

Layout adalah tata letak dari setiap komponen elemen di dalam website. Layout berfungsi untuk memetakan bagian-bagian dalam halaman website. Dengan desain layout yang baik akan memberikan kenyamanan kepada pengunjung.

Secara umum website memiliki 4 bagian layout yaitu

- Header – Berisi judul website
- Navigasi – Untuk membuat menu navigasi
- Konten – Bagian utama yang menjadi sasaran informasi yang ingin di sampaikan ke pengunjung

- Footer – Bagian paling bawah biasa berisi keterangan tambahan seperti informasi kepemilikan atau copyright.



Untuk membuat layout di HTML 5 kita bisa menggunakan tag-tag berikut:

- Untuk membuat header menggunakan tag **<header>**
- Membuat navigasi menggunakan tag **<nav>**
- Membuat konten/artikel menggunakan tag **<article>**
- Membuat footer menggunakan tag **<footer>**

```
<!DOCTYPE html>

<html>

<head>

    <title>Contoh Membuat Layout - www.kelasprogrammer.com</title>

</head>

<body>

    <header>

        <p>Header</p>
```

```
</header>

<nav>

    <p>Navbar</p>

</nav>

<article>

    <p>Konten</p>

</article>


<footer>

    <p>Footer</p>

</footer>

</body>

</html>
```

Agar bisa terlihat lebih jelas tambahkan kode CSS di dalam tag **<head>** seperti berikut:

```
<style>

    body {

        width: 65%;

        height: auto;

        background: white;

        margin:auto;

        margin-top: 5px;

        padding-bottom: 5px;

    }


    p {

        text-align: center;

        font-size: x-large;

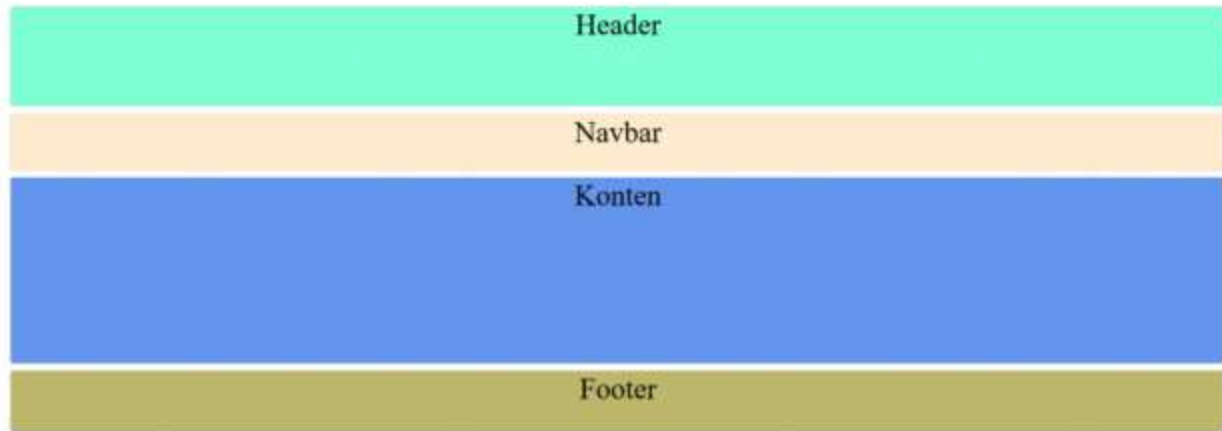
        margin:auto;

    }


```

```
header,nav,article,footer {  
    border-radius: 3px;  
    margin: 5px 5px 5px 5px;  
    border: 1px solid gainsboro;  
  
}  
  
header {  
    height: 80px;  
    background-color: aquamarine;  
}  
  
nav {  
    height: 45px;  
    background-color: blanchedalmond;  
}  
  
article {  
    height: 150px;  
    background-color: cornflowerblue;  
}  
  
footer {  
    height: 50px;  
    background-color: darkkhaki;  
}  
  
</style>
```

Hasil:



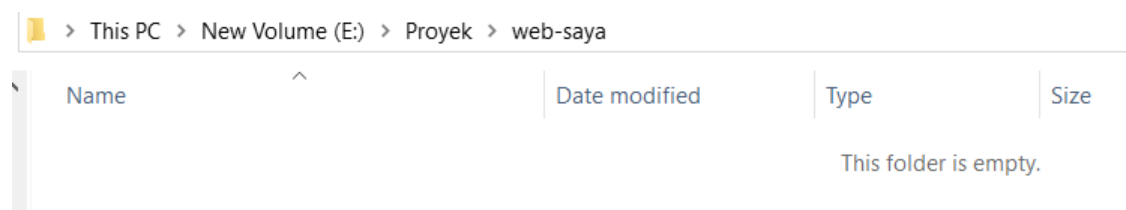
Studi Kasus : Membuat Website Sederhana Dengan HTML dan CSS

Setelah kalian memahami pembahasan diatas , selanjutnya kita akan belajar membuat website sederhana.

Website yang kita buat bersifat statis karena hanya menggunakan HTML dan CSS, kedepannya kalian bisa kembangkan dengan [belajar javascript](#), [PHP](#), atau [python](#) untuk membuat website dinamis.

Langkah 1 : Buatlah folder project web

Buatlah sebuah folder baru dalam di rektori kalian, contoh dibawah ini saya membuat folder web-saya



Selanjutnya buatlah folder gambar yang berisi gambar untuk kebutuhan konten website



Perhatikan untuk nama gambar pastikan tidak ada spasi jika nama lebih dari satu kata pakailah tanda penghubung – (*strip*) atau _ (*underscore*).

Kenali setiap ekstensi gambar apakah menggunakan format .png atau jpeg karena nantinya kita akan sertakan gambar-gambar ini dalam sintak html menggunakan tag ****.

Langkah 2 : Buatlah file index melalui teks editor

Buatlah sebuah file dengan nama index.html melalui teks editor yang kalian miliki, saran saya kalian bisa menggunakan **sublime text** atau teks editor lainnya yang kalian miliki dan sukai.

```
<!DOCTYPE html>

<html>

<head>

    <title>Beranda</title>

    <link rel="stylesheet" type="text/css" href="style.css">

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

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

</head>

<body>

    <div class="container">

        <header>

            <div class="logo">
```

Web Saya

</div>

</header>

<nav>

Beranda

Kontak Saya

Tentang Saya

</nav>

<article>

<div class="konten">

<div class="judul">

Penyu

</div>

<p>Penyu adalah hewan yang banyak di temukan bertelur di beberapa pesisir pantai di Indonesia sejak dahulu.</p>

</div>

<div class="konten">

<div class="judul">

Ikan Badut

</div>

<p>Ikan badut ini dapat dikenali dengan warna jingganya ikan ini tumbuh mencapai 8 cm serta termasuk dalam ikan terpopuler didunia. </p>

</div>

<div class="konten">

<div class="judul">

Lion Fish

</div>

```
<p>Lionfish atau volitans adalah ikan berbisa milik keluarga yang
Scorpaenidae terjemahan harfiah berarti ikan kalajengking.</p>

</div>

<div class="konten">

  <a href="moorish-idol.html"></a>

  <div class="judul">

    <a href="moorish-idol.html">Moorish Idol</a>

  </div>

  <p>Ikan Moorish Idol adalah salah satu ikan yang cukup populer di kalangan
  ara hobiis ikan hias karena bentuknya yang unik dan pola warnanya yang menarik.</p>

</div>

</article>


<footer>

  Copyright 2020 web-saya.com

</footer>

</div>

</body>

</html>
```

Jika kita menjalankan file **index.html** di browser maka hasilnya akan seperti pada gambar berikut:

Web Saya

- [Beranda](#)
- [Kontak Saya](#)
- [Tentang Saya](#)



[Penyu](#)

Penyu adalah hewan yang banyak di temukan bertelur di beberapa pesisir pantai di Indonesia sejak dahulu.



[Ikan Badut](#)

Ikan badut ini dapat dikenali dengan warna jinganya ikan ini tumbuh mencapai 8 cm serta termasuk dalam ikan terpopuler didunia.



Untuk mengatur tampilan halaman website kita perlu menambahkan style CSS. Coba perhatikan di dalam tag **<head>** saya menyertakan sintak ini:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Sintak ini merupakan cara menyisipkan style css secara eksternal (terpisah dari file html) sehingga kita perlu membuat file baru bernama **style.css**

Langkah 3 : Membuat Style CSS

Buat file dengan nama **style.css** dengan isi kode seperti berikut:

```
body{  
    font-family: Arial, Helvetica, sans-serif;  
    background: #e6e6e6;  
    padding: 0;  
    margin:0;  
  
}
```

```
.container{  
    width: 65%;  
    height: auto;  
    background: white;  
    margin:auto;  
    margin-top: 5px;  
    padding-bottom: 5px;  
  
}
```

```
.logo {  
    font-size: xx-large;  
    padding: 30px 0 30px 10px;  
  
}
```

```
nav ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #1a75ff;  
    border: 1px solid gainsboro;  
  
}
```

```
nav li {  
    float: left;
```

```
}
```

```
nav li a{  
    display: inline-block;  
    color: #ffffff;  
    text-align: center;  
    padding: 14px 15px;  
    text-decoration: none;  
}
```

```
nav li a:hover {  
    opacity: 0.9;  
    background-color: #3385ff;  
}
```

```
article {  
    width: 98.8%;  
    height: auto;  
    border-radius: 3px;  
    margin: 5px 5px 5px 5px;  
    border: 1px solid gainsboro;  
    clear: both;  
}
```

```
article p {  
    padding: 5px 6px 5px 5px;  
    line-height: 1.5;  
}
```

```
article .konten {  
    width: 23.5%;  
    height: auto;  
    margin: 5px 5px 5px 5px;
```

```
border-radius: 3px;

border: 1px solid gainsboro;

float: left;

}
```

```
article .isi {

width: 98.5%;

height: auto;

margin: 5px 5px 5px 5px;

border-radius: 3px;

border: 1px solid gainsboro;

float: left;

}
```

```
article .isi .judul {

font-size: xx-large;

margin-bottom: 10px;

margin-left: 10px;

margin-top: 10px;

}
```

```
article .isi p {

margin-top: 0;

}
```

```
article .isi img {

float: left;

margin: 5px 5px 5px 5px;

}
```

```
article .konten img {

width: 95.8%;

height: 150px;

margin: 5px 5px 5px 5px;
```

```
}

article .konten:hover {

    opacity:0.9;

}

article .konten .judul a {

    margin-left: 5px;

    text-decoration: none;

    font-size:x-large;

    color:#020101;

}

article .konten a:hover {

    opacity:0.9;

}

footer {

    clear: both;

    width: 98.8%;

    height: auto;

    text-align: center;

    margin: 5px 5px 5px 5px;

    border-radius: 3px;

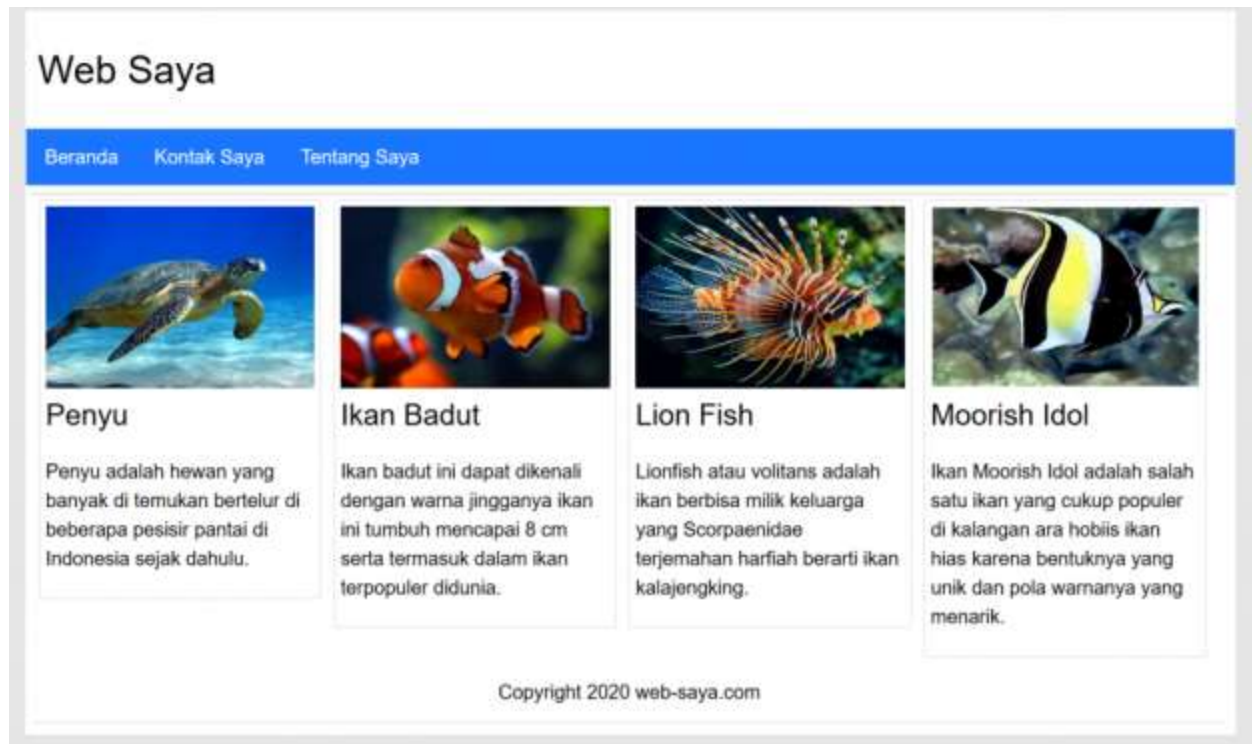
    border-bottom: 1px solid gainsboro;

    padding-top: 15px;

    padding-bottom: 15px;

}
```

Hasil



Dapat di lihat setelah kita menambahkan style CSS tampilan website menjadi lebih menarik.

Pada tahap ini kita baru membuat satu halaman yaitu halaman index selanjutnya masih ada halaman lainnya yang perlu dibuat.

Langkah 4 : Membuat Halaman Artikel

Agar ketika pengguna mengklik salah satu konten artikel untuk dapat di baca secara keseluruhan isi konten, maka kita perlu membuat halaman untuk masing-masing konten tersebut.

Halaman yang harus kita buat adalah

- penyu.html

- ikan-badut.html
- lion-fish.html
- moorish-idol.html

Contoh : Membuat Halaman ikan-badut.html

```
<!DOCTYPE html>

<html>

<head>

    <title>Ikan Badut</title>

    <link rel="stylesheet" type="text/css" href="style.css">

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

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

</head>

<body>

    <div class="container">

        <header>

            <div class="logo">

                Web Saya

            </div>

        </header>

        <nav>

            <ul>

                <li><a href="index.html">Beranda</a></li>

                <li><a href="kontak-saya.html">Kontak Saya</a></li>

                <li><a href="tentang-saya.html">Tentang Saya</a></li>

            </ul>

        </nav>

        <article>

            <div class="isi">

                <div class="judul">
```

Ikan Badut

</div>

<p>Ikan badut ini dapat dikenali dengan warna jingganya ikan ini tumbuh mencapai 8 cm serta termasuk dalam ikan terpopuler didunia. Hewan ini masuk kedalam Genus Amphiprion. Ikan giru atau lebih dikenal dengan sebutan ikan badut adalah ikan dari anak suku Amphiprioninae dalam suku Pomacentridae. Ada 28 spesies yang biasa dikenali, salah satunya adalah genus Premnas, sementara sisanya termasuk dalam genus Amphiprion. </p>

<p>Mereka tersebar di lautan Pasifik, Laut Merah, lautan India, dan karang besar Australia. Di alam bebas mereka bersimbiosis dengan anemon laut. Anemon akan melindungi Ikan badut dari pemangsa dan Ikan badut akan membersihkan Anemon dengan memakan sisa - sisa makanan Anemon. Ikan badut berwarna kuning, jingga, kemerahan atau kehitaman. </p>

<p>Spesies terbesar mencapai panjang 18 cm, sementara yang terkecil hanya 6 cm. Mereka, pada umumnya, dijumpai pada laguna-laguna berbatu di seputar terumbu karang, atau pada daerah koastal dengan kedalaman kurang dari 50 meter dan berair jernih. Di perairan Papua New Guinea, bisa ditemukan ikan badut tidak kurang dari 8 spesies. Di alam, ikan badut mengkonsumsi zooplankton, udang-udangan dan algae yang dijumpai di habitat mereka.</p>

</div>

</article>

<footer>

Copyright 2020 web-saya.com

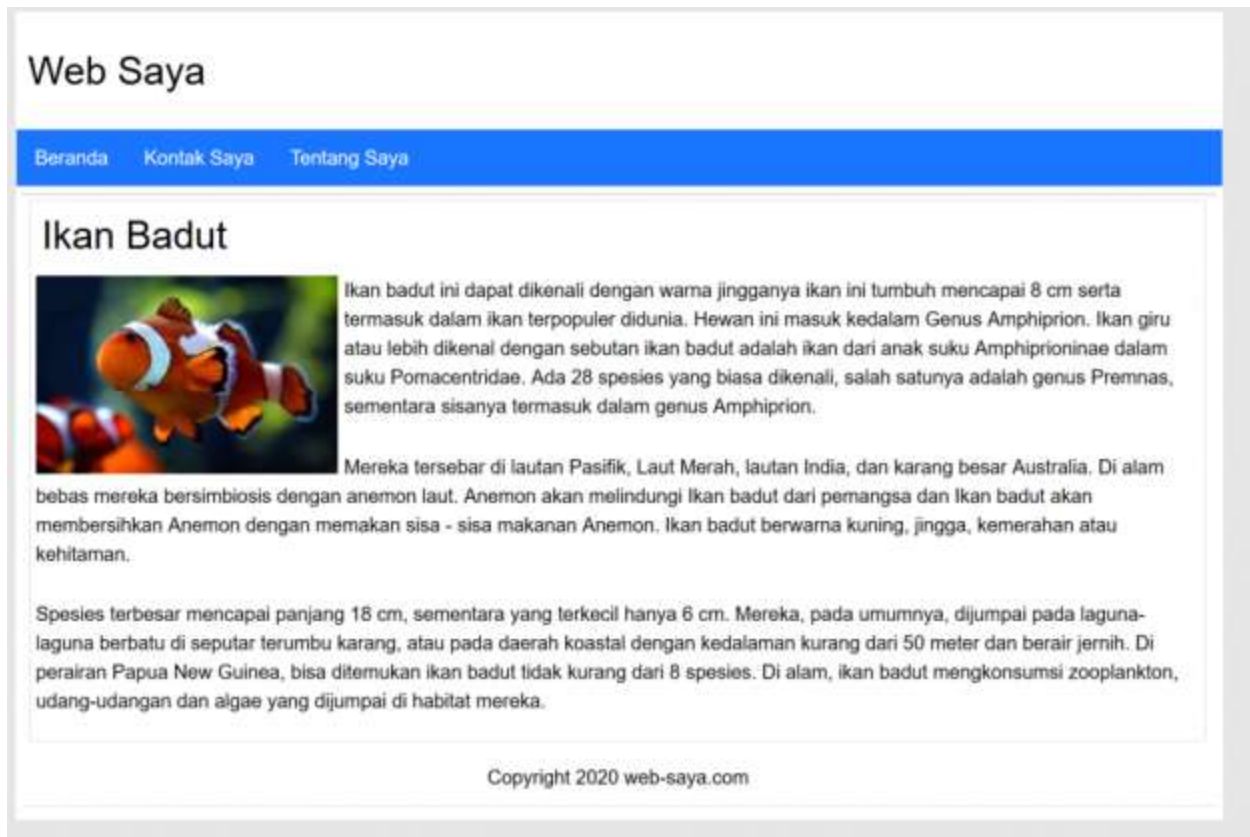
</footer>

</div>

</body>

</html>

Hasil



Kalian bisa lakukan cara yang sama dengan cara meng-copy dari file ikan-badut.html lalu mengganti dengan file lainnya.

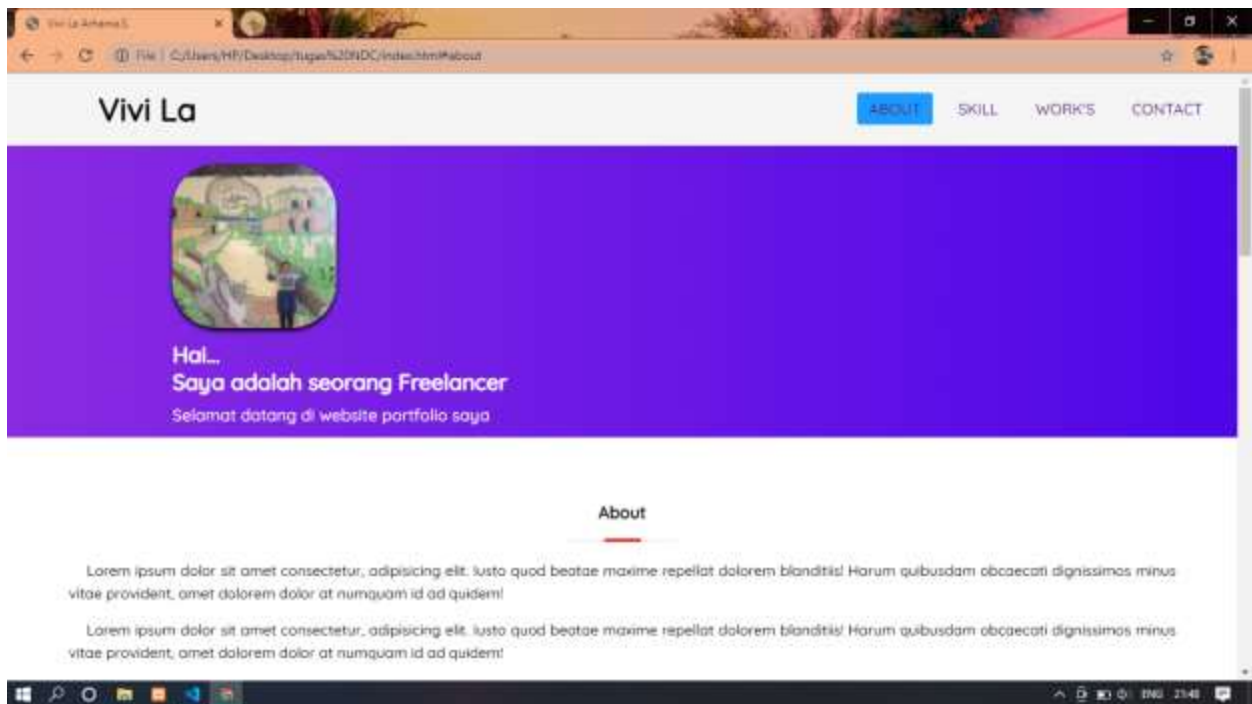
Langkah 5 : Membuat Halaman Kontak & Tentang saya

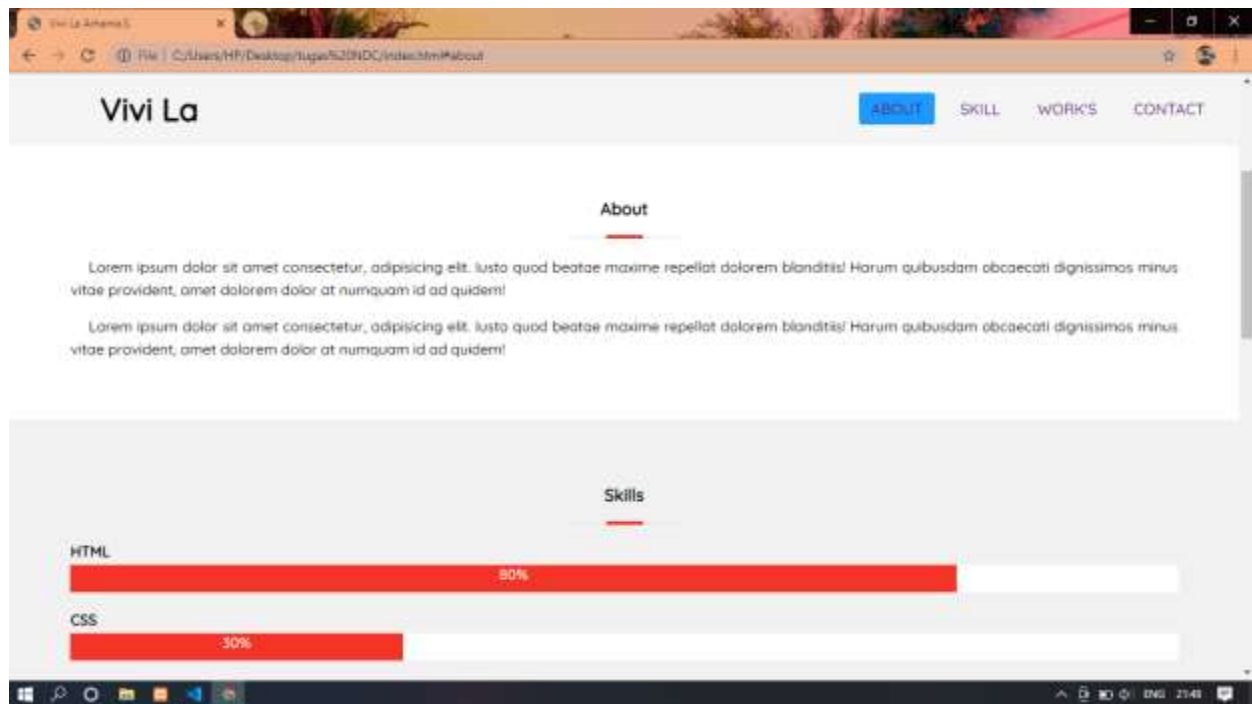
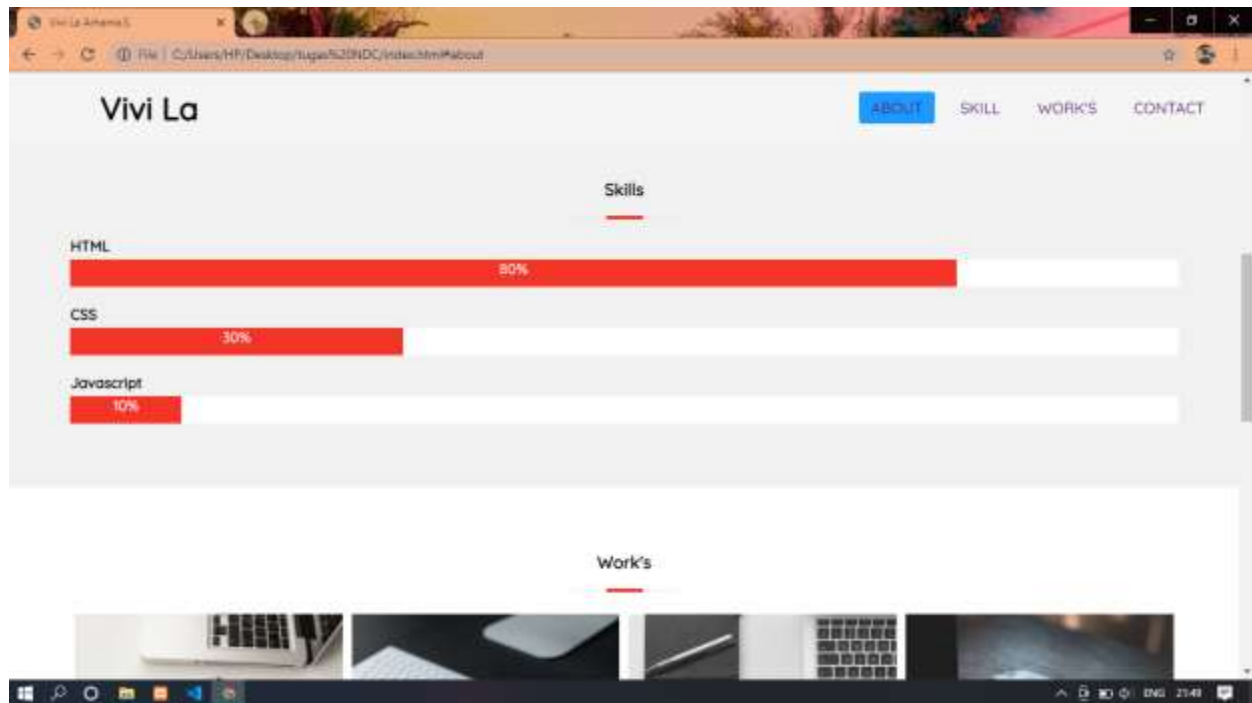
Selain membuat halaman diatas, kita juga perlu membuat dua halaman lainnya yaitu:

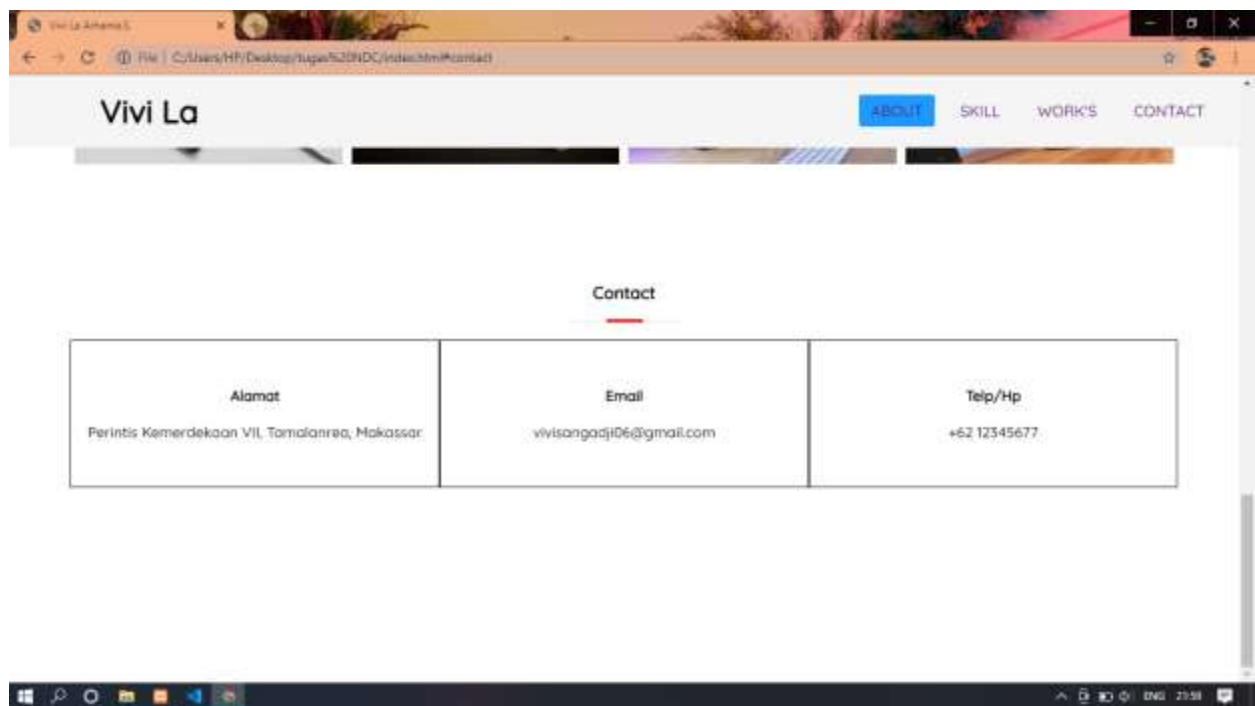
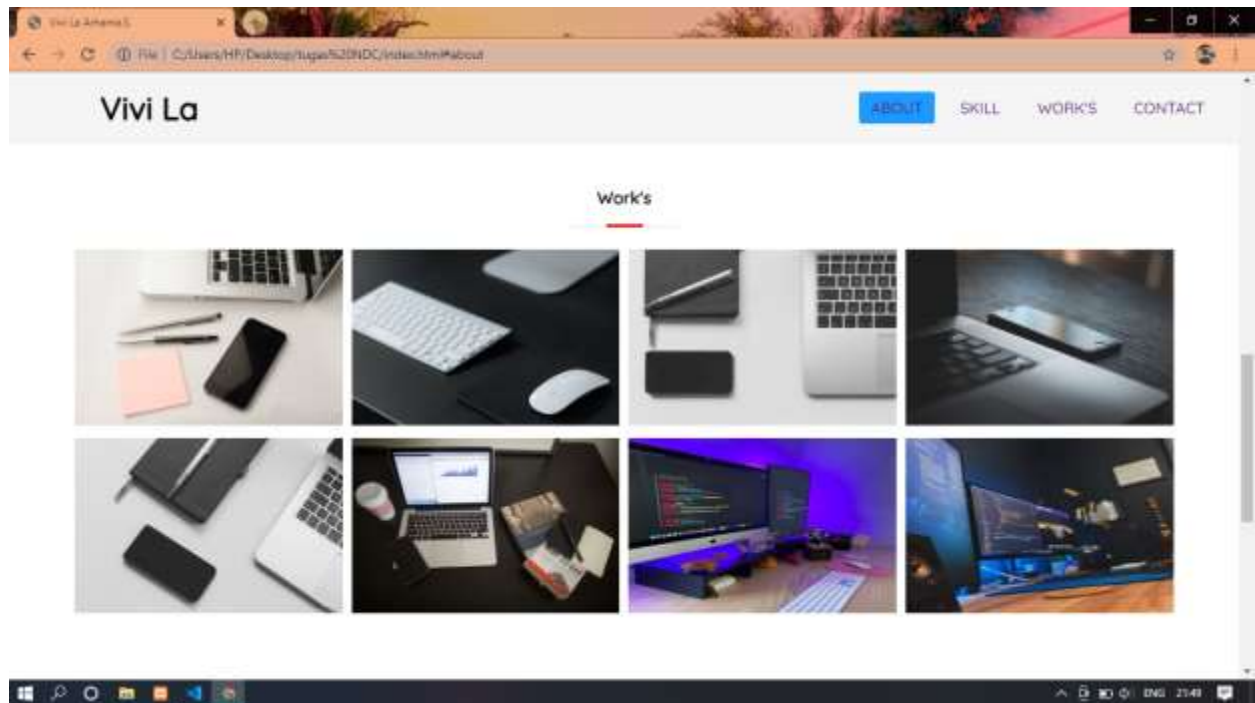
- kontak-saya.html
- tentang-saya.html

Ketika penjung mengklik menu kontak saya pengunjung tersebut akan di arahkan ke halaman **kontak-saya.html** begitu juga dengan halaman tentang saya. Kalian bisa memuat profil kontak kalian pada kedua halaman ini.

TAMPILAN WEBSITE PORTFOLIO







SCRIPT INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Vivi La Arhama.S.</title>
  <link rel="stylesheet" href="css/style.css">
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@500&displ
ay=swap" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.1/css/all.min.css">
</head>
<body>
  <!-- bagian navbar -->
  <nav>
    <input type="checkbox" id="check">
    <label for="check" class="checkbtn"><i class="fas fa-bars"></i></label>
    <label class="logo" for="">Vivi La</label>
    <ul>
      <li><a class="active" href="#about">About</a></li>
      <li><a href="#skill">SKill</a></li>
      <li><a href="#work">Work's</a></li>
      <li><a href="#contact">Contact</a></li>
    </ul>
  </nav>

  <!-- bagian banner -->
  <section class="banner">
    <div class="container">
      <div class="banner-left">
        
        <h2>Hai... <br> Saya adalah seorang <span class="efek-
mengetik"></span></h2>
        <p>Selamat datang di website portfolio saya</p>
      </div>
    </div>
  </section>

  <!-- bagian about -->
  <section id="about">
```

```
    <div class="container">
      <h3>About</h3>
      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Iusto quod beatae maxime repellat dolorem blanditiis! Harum quibusdam obcaecati dignissimos minus vitae provident, amet dolorem dolor at numquam id ad quidem!</p>
      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Iusto quod beatae maxime repellat dolorem blanditiis! Harum quibusdam obcaecati dignissimos minus vitae provident, amet dolorem dolor at numquam id ad quidem!</p>
    </div>
  </section>
```

```
<!-- bagian skills -->
<section id="skill">
  <div class="container">
    <h3>Skills</h3>
    <h4>HTML</h4>
    <div class="bar">
      <span class="nilai html">80%</span>
    </div>

    <h4>CSS</h4>
    <div class="bar">
      <span class="nilai css">30%</span>
    </div>

    <h4>Javascript</h4>
    <div class="bar">
      <span class="nilai js">10%</span>
    </div>
  </div>
</section>
```

```
<!-- bagian work's -->
<section id="work">
  <div class="container">
    <h3>Work's</h3>
    <div class="col-4">
      <a href="#">
        
      </a>
    </div>

    <div class="col-4">
```

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

    <div class="col-4">
        <a href="#">
            
        </a>
    </div>

    <div class="col-4">
        <a href="#">
            
        </a>
    </div>
    <div class="col-4">
        <a href="#">
            
        </a>
    </div>

    <div class="col-4">
        <a href="#">
            
        </a>
    </div>

    <div class="col-4">
        <a href="#">
            
        </a>
    </div>

    <div class="col-4">
        <a href="#">
            
        </a>
    </div>
</div>
</section>

<!-- bagian contact -->
<section id="contact">
```

```

        <div class="container">
            <h3>Contact</h3>
            <div class="col-3">
                <h4>Alamat</h4>
                <p>Perintis Kemerdekaan VII, Tamalanrea, Makassar</p>
            </div>

            <div class="col-3">
                <h4>Email</h4>
                <p>vivisangadji06@gmail.com</p>
            </div>

            <div class="col-3">
                <h4>Telp/Hp</h4>
                <p>+62 12345677</p>
            </div>
        </div>
    </section>

<script src="js/javascript.js"></script>
</body>
</html>

```

SCRIPT STYLE.CSS

```

* {
    padding: 0;
    margin: 0;
    text-decoration: none;
    box-sizing: border-box;
}
html {
    scroll-behavior: smooth;
}
body {
    font-family: 'Quicksand', sans-serif;
}

/* Navbar */
nav {
    background: whitesmoke;

```



```
    height: 80px;
    width: 100%;
    position: fixed;
    margin-bottom: 1px;
}

label.logo {
    font-size: 35px;
    line-height: 80px;
    padding: 0 100px;
    font-weight: bold;
}

nav ul {
    float: right;
    margin-right: 20px;
}

nav ul li {
    display: inline-block;
    line-height: 80px;
    margin: 0 5px;
}

nav ul li a {
    font-size: 17px;
    padding: 7px 13px;
    border-radius: 3px;
    text-transform: uppercase;
}

a.active, a:hover {
    background: #1b9bff;
    transition: .5s;
}

.checkbtn {
    font-size: 30px;
    float: right;
    line-height: 80px;
    margin-right: 40px;
    cursor: pointer;
    display: none;
}

#check {
    display: none;
}
```

```
/* Untuk container */
.container {
  width: 90%;
  margin: 0 auto;
}

.container::after {
  content: '';
  display: block;
  clear: both;
}

/* Bagian Banner */
.banner {
  background-image: linear-
gradient(to right,rgba(138,43,226,1),rgba(77, 5, 232, 1));
  padding: 100px;
  height: 400px;
}

.banner img {
  width: 180px;
  border-radius: 50px;
  box-shadow: 0 3px 5px #000;
  margin-bottom: 10px;
}

.banner h2,
.banner p, .banner span {
  color: #ffff;
}

.banner h2 {
  margin-bottom: 10px;
}

.banner p {
  font-size: 18px;
}

.banner-left{
  padding-left: 24px;
}
```

```
/* untuk Tag Section */
section {
    padding: 50px 0;
}

section h3 {
    padding-bottom: 20px;
    text-align: center;
    margin: 20px;
    position: relative;
}

section h3::before {
    content: '';
    display: block;
    position: absolute;
    bottom: 0;
    width: 120px;
    height: 1px;
    background-color: #dddd;
    left: calc(50% - 60px);
}

section h3::after {
    content: '';
    display: block;
    position: absolute;
    bottom: -1px;
    width: 40px;
    height: 4px;
    background-color: #F53428;
    left: calc(50% - 20px);
}

#about p {
    text-align: justify;
    text-indent: 20px;
    line-height: 25px;
    margin-bottom: 15px;
}

#skill,
.contact {
```

```
        background-color: #f0f1f0;
    }

#skill .bar {
    height: 30px;
    background-color: #ffff;
    margin: 5px 0 20px 0;
}

.bar .nilai {
    height: 30px;
    background-color: #F53428;
    display: inline-block;
    text-align: center;
    color: #fff;
}

.html {
    width: 80%;
}

.css {
    width: 30%;
}

.js {
    width: 10%
}

.col-4 {
    width: 25%;
    box-sizing: border-box;
    padding: 5px;
    float: left;
    text-align: center;
}

.col-4 img {
    width: 100%;
}

.col-4 a:hover img {
    transform: scale(1.2);
}
```

```
.col-3 {
    width: 33.33%;
    border: 1px solid;
    box-sizing: border-box;
    float: left;
    text-align: center;
    padding: 50px 10px;
}

.col-3 h4 {
    margin-bottom: 20px;
}

#contact {
    height: 510px;
}

/* Responsive Breakpoint */

@media (max-width: 952px) {
    label.logo{
        font-size: 30px;
        padding-left: 50px;
    }
    nav ul li a {
        font-size: 16px;
    }
}

/* Ukuran Tablet */
@media screen and (max-width: 858px) {
    .checkbtn {
        display: block;
    }
    ul {
        position: fixed;
        width: 80%;
        height: 385px;
        background-image: linear-
gradient(to right,rgba(138,43,226,1),rgba(77, 5, 232, 1));
        top: 80px;
        left: -100%;
    }
}
```

```
        text-align: center;
        transition: all .5s;
    }
    nav ul li {
        display: block;
        margin: 50px 0;
        line-height: 30px;
    }
    nav ul li a {
        font-size: 20px;
        color: white;
    }
    a:hover,a.active{
        background: none;
        color: #0082e6;
    }
    #check:checked ~ ul{
        left: 0;
    }
}
```

SCRIPT JAVASCRIPT.JS

```
const txtElement = ['Mahasiswi', 'Freelancer', 'Web Developer'];
let count = 0;
let txtIndex = 0;
let currentTxt = '';
let words = '';

(function mengetik(){

    if(count == txtElement.length){
        count = 0;
    }

    currentTxt = txtElement[count];

    words = currentTxt.slice(0, ++txtIndex);
    document.querySelector('.efek-mengetik').textContent = words;

    if(words.length == currentTxt.length){
        count++;
        txtIndex = 0;
    }

    setTimeout(mengetik, 500);

})();
```