

2.1.3 HTML Layout

Layout pada HTML adalah tata letak atau susunan kerangka sebuah halaman *web*. Ini adalah bagian penting dalam desain *web* yang mengatur cara elemen-elemen konten seperti teks, gambar, dan elemen lainnya ditempatkan dan ditampilkan di halaman *web*. *Layout* bertujuan untuk menciptakan tampilan yang terstruktur, baik, dan responsif agar pengguna dapat dengan mudah berinteraksi dengan konten.

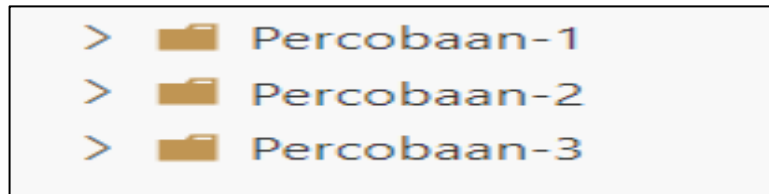
Dalam HTML, *layout* dapat diatur menggunakan berbagai elemen dan teknik. Beberapa elemen yang umum digunakan dalam pembuatan *layout* HTML termasuk tag `<div>` (*divisi*), tag `<table>` (*tabel*), tag `<header>` (*bagian atas*), tag `<footer>` (*bagian bawah*), dan lainnya. Selain itu, CSS (*Cascading Style Sheets*) juga digunakan untuk mengontrol tampilan dan posisi elemen-elemen tersebut.

Mengenal teknik dan elemen-elemen tata letak HTML sangat penting dalam mengembangkan halaman *web* yang menarik dan berfungsi dengan baik. Dengan memahami prinsip-prinsip dasar *layout* HTML, Anda dapat merancang tampilan *web* yang sesuai dengan kebutuhan dan preferensi desain Anda serta memberikan pengalaman yang baik kepada pengguna. Pemahaman ini juga memungkinkan Anda untuk memanfaatkan elemen HTML seperti `<div>`, `<header>`, dan `<footer>` secara optimal, sehingga setiap bagian dari halaman *web* memiliki fungsi dan tampilan yang jelas. Selain itu, dengan menguasai CSS, Anda bisa mengontrol elemen-elemen tersebut agar tampil secara konsisten dan responsif di berbagai perangkat. Teknik seperti penggunaan *grid layout*, *flexbox*, dan *media queries* dalam CSS membantu membuat *layout* yang fleksibel dan dinamis. Dengan kombinasi HTML dan CSS yang tepat, Anda dapat membangun halaman *web* yang tidak hanya enak dipandang tetapi juga mudah dinavigasi, yang pada akhirnya meningkatkan kepuasan pengguna secara keseluruhan. Pemahaman yang baik tentang *layout* HTML memungkinkan desainer *web* untuk menciptakan tampilan yang fungsional sekaligus estetik, sesuai dengan kebutuhan dan tren modern.

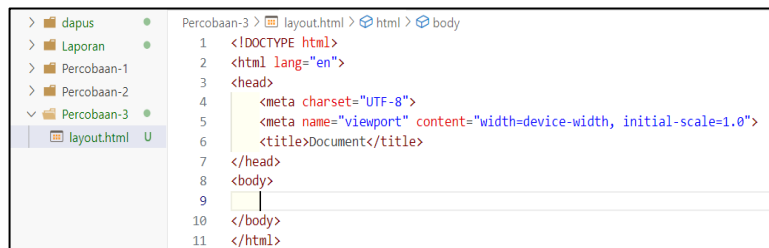
3.1.3 HTML Layout

1. Project

- a. Pada folder utama, buat folder baru dengan nama “Percobaan-3”

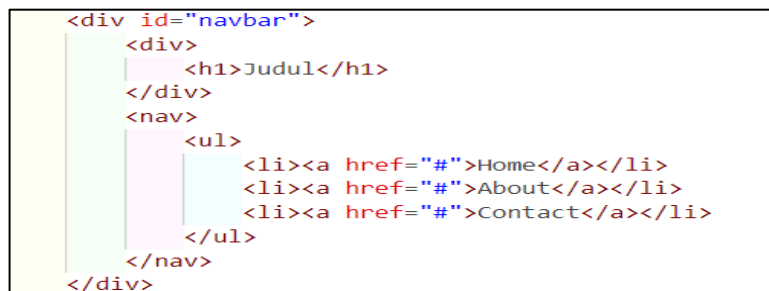


- b. Buka folder tersebut pada *code editor* VS Code, buat folder “Percobaan-3” kemudian buat *file* “layout.html” dan tambahkan struktur dasar HTML 5 kedalamnya.

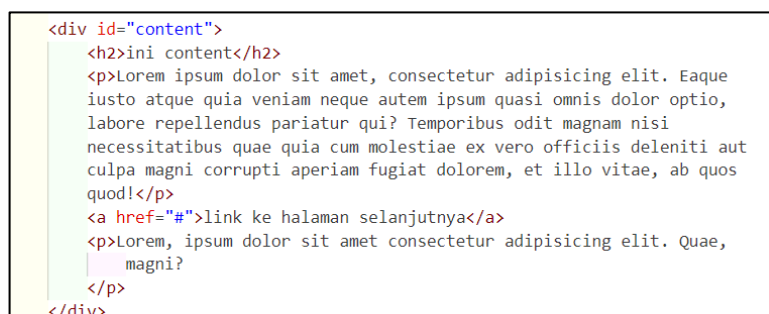


2. Layout

- a. Dalam tag `<body>`, tambahkan kode berikut untuk membuat *layout navigasi*



- b. Tambahkan kode berikut untuk membuat *layout content*.



- c. Tambahkan kode berikut untuk membuat *layout footer*.

```
<div id="footer">  
  <strong>copyright praktikum web 1</strong>  
</div>
```

3. Semantic layout

- a. Dalam *tag body*, tambahkan kode berikut untuk membuat *layout navigasi*.

```
<nav>  
  <section>  
    <h1>judul semantic tag</h1>  
  </section>  
  
  <section>  
    <ul>  
      <li><a href="#">Home</a></li>  
      <li><a href="#">About</a></li>  
      <li><a href="#">Contact</a></li>  
    </ul>  
  </section>  
</nav>
```

- b. Tambahkan kode berikut untuk membuat *layout content*.

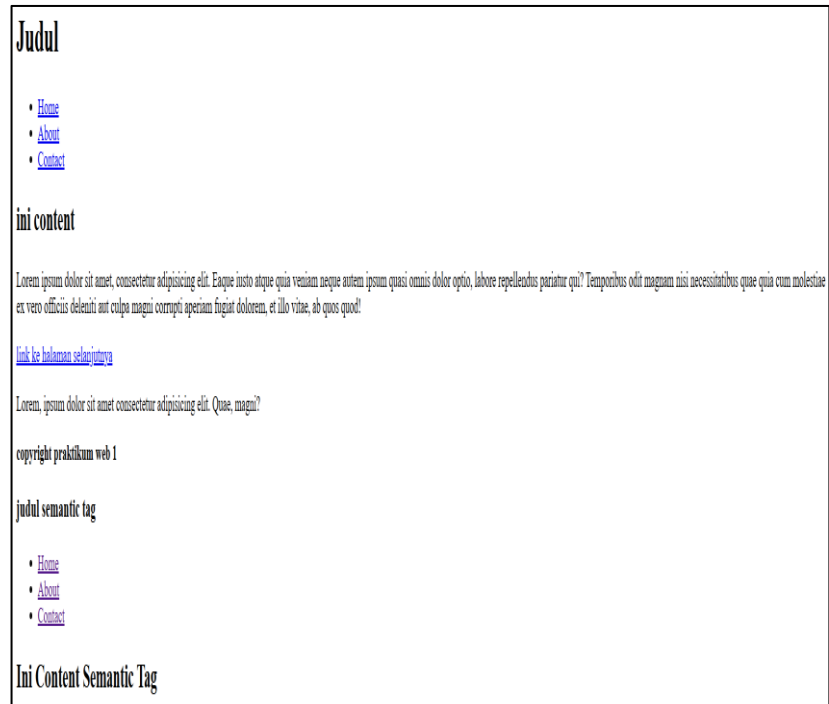
```
<main>  
  <article>  
    <h2>Ini Content Semantic Tag</h2>  
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Eaque  
iusto atque quia veniam neque autem ipsum quasi omnis dolor optio,  
labore repellendus pariatur qui? Temporibus odit magnam nisi  
necessitatibus quae quia cum molestiae ex vero officiis deleniti aut  
culpa magni corrupti aperiam fugiat dolorem, et illo vitae, ab quos  
quod!</p>  
    <a href="#">link ke halaman selanjutnya</a>  
    <p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quae,  
magni?</p>  
  </article>  
</main>
```

- c. Tambahkan kode berikut untuk membuat *layout footer*.

```
<footer>  
  <strong>Copyright Praktikum Web 1</strong>  
</footer>
```

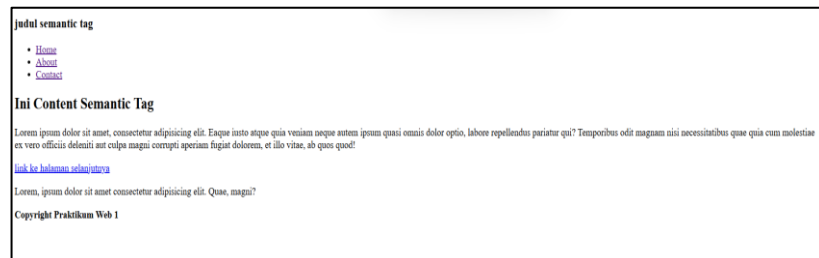
4.1.2 Basic Tag HTML Part 2

1. Layout tag



Berdasarkan kode diatas, setiap elemen dibungkus dalam *tag div* yang diberi atribut *id*, seperti "*id='navbar'*", "*id='content'*", dan "*id='footer'*". Atribut *id* berfungsi sebagai identitas unik bagi setiap elemen, sehingga dapat diidentifikasi secara spesifik dalam CSS atau JavaScript untuk menambahkan gaya atau interaksi. Bagian `<div id="navbar">` berfungsi sebagai navigasi utama dan berisi `<h1>` untuk judul, serta `<nav>` dengan daftar navigasi `` yang berisi tautan seperti "*Home*", "*About*", dan "*Contact*". Bagian `<div id="content">` menampilkan konten utama, dengan subjudul `<h2>` dan beberapa paragraf `<p>` berisi teks contoh "*lorem ipsum*". Ada juga tautan untuk "link ke halaman selanjutnya". Terakhir, `<div id="footer">` berisi informasi hak cipta yang ditandai dengan ``, membuat teks "*copyright praktikum web 1*" menjadi tebal. Struktur ini memisahkan navigasi, konten, dan informasi tambahan di bagian bawah halaman.

2. Semantic tag



Berdasarkan hasil kode diatas, kode HTML tersebut Menggunakan *semantic tag* yang memberikan arti spesifik pada setiap bagian halaman. Bagian *tag* `<nav>` digunakan untuk menampung navigasi dan memiliki dua *section* di dalamnya. *Section* pertama berisi *tag* `<h1>` yang menampilkan judul "judul *semantic tag*". *Tag* `<h1>` ini berfungsi untuk menampilkan judul utama, biasanya dengan ukuran font yang lebih besar. Di dalam *section* kedua, terdapat sebuah daftar tautan menggunakan *tag* `` yang mengelompokkan item-item navigasi dalam bentuk *tag* ``. Masing-masing item daftar ini mengandung tautan ``, yang berfungsi untuk mengarahkan ke halaman lain, meskipun dalam contoh ini menggunakan tautan kosong. Setelahnya, ada elemen *tag* `<main>` yang menandakan bagian utama halaman. Di dalamnya, terdapat *tag* `<article>`, yang menunjukkan sebuah artikel atau konten yang berdiri sendiri. Artikel ini berisi subjudul yang ditandai dengan *tag* `<h2>` dan beberapa paragraf menggunakan *tag* `<p>`. Tautan `` digunakan untuk memberikan navigasi menuju halaman selanjutnya, meskipun *link* tersebut masih kosong. Pada bagian bawah halaman, terdapat elemen *tag* `<footer>`, yang berfungsi untuk menampung informasi hak cipta atau catatan lainnya. Di dalamnya, *tag* `` digunakan untuk menebalkan teks, menandakan teks yang lebih penting. Struktur ini menggunakan *semantic tag* untuk membuat halaman lebih terstruktur dan mudah dipahami.

BAB V

PENUTUP

5.1 Kesimpulan

5.1.1 *Basic Tag HTML Part 1*

Dari hasil percobaan sebelumnya, menunjukkan bahwa struktur dasar HTML terdiri dari berbagai elemen seperti *tag* HTML, *head*, dan *body* yang menentukan tampilan halaman *web*. Penggunaan *tag heading*, paragraf, dan *formatting* memberikan variasi dan penekanan pada teks, sedangkan *tag* komentar membantu pengembang untuk mencatat informasi penting dalam kode tanpa memengaruhi tampilan. *Hyperlink* memungkinkan navigasi antar halaman dan ke URL *eksternal*, sedangkan *tag* gambar menyediakan cara untuk menampilkan media visual dengan atribut yang mendukung. Semua elemen ini berkontribusi pada pembentukan halaman *web* yang interaktif dan informatif.

5.1.2 *Basic Tag HTML Part 2*

Dari percobaan sebelumnya, dapat disimpulkan bahwa HTML memiliki elemen-elemen dasar seperti *list*, tabel, dan *form input*. *List* terbagi menjadi *unordered list* dan *ordered list*, yang masing-masing menggunakan *tag* `` dan `` serta diisi dengan elemen ``. Tabel dibangun dengan baris dan kolom menggunakan *tag* `<tr>`, `<td>`, dan `<th>`, yang dapat diatur ukurannya. *Form input* menyediakan berbagai tipe *input* untuk data pengguna, seperti *text*, *email*, *number* dan lainnya, serta atribut *id* dan *name* untuk identifikasi dan pengumpulan data. Dengan elemen-elemen ini, dapat memungkinkan pembuatan struktur halaman yang interaktif dan fungsional.

5.1.3 *HTML Layout*

Dari percobaan ini, dapat disimpulkan bahwa materi HTML Layout ini adalah penggunaan *semantic tag* seperti `<nav>`, `<section>`, `<article>`, dan `<footer>` membantu memberi makna yang jelas pada struktur halaman. Setiap elemen dibungkus dalam *div* dengan atribut *id* untuk identifikasi yang spesifik, yang memudahkan penerapan gaya atau interaksi dengan CSS atau JavaScript. Elemen-elemen seperti `<h1>`, `<h2>`, ``, dan `<p>`

membantu membentuk konten dan navigasi dengan lebih terstruktur. Struktur ini membuat halaman lebih rapi, mudah dipahami, dan meningkatkan aksesibilitas bagi pengguna maupun mesin pencari.