

## BAB V DESAIN WEB CSS

### A. KOMPETENSI DASAR

- Memahami konsep dan strategi desain web yang rapi berbasis HTML dan CSS/CSS3.
- Mampu mengoptimalkan fitur-fitur CSS dalam desain web.
- Mampu memanfaatkan pendekatan untuk menghasilkan halaman web yang rapi, standar, dan menarik.

### B. ALOKASI WAKTU

1 JS (2 x 50 menit)

### C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

### D. DASAR TEORI

Pada bab sebelumnya telah dibahas apa itu CSS (*Cascading Style Sheet*) dan bagaimana cara penggunaannya secara mendasar. Pada bab ini akan membahas lebih lanjut mengenai fungsionalitas CSS secara spesifik berkaitan dengan desain web.

Desain web merupakan bagian terpenting dalam pembangunan sebuah halaman website. Dalam desain web yang baik, unsur estetika yang harus diperhatikan adalah: warna, tata letak, dan bentuk. Ketiga unsur tersebut dapat diatur dengan menggunakan CSS, seperti mengatur warna *background*, warna huruf, bentuk menu, letak menu, dan lain sebagainya.

Berikut alasan mengapa menggunakan CSS dalam mendesain web:

- a. Kode HTML menjadi lebih sederhana dan lebih mudah diatur (*clean design*).
- b. Kode HTML lebih rapi dan mendukung SEO friendly
- c. Memungkinkan aspek guna ulang (*reusability*)
- d. Mudah untuk dikelola (*maintainability*)
- e. Ukuran file menjadi lebih kecil, sehingga *load* file lebih cepat.
- f. Digunakan dalam hampir semua web browser.

## E. LATIHAN

### 1. Desain *Layout*

Untuk mendesain sebuah *layout* website secara utuh, dibutuhkan sebuah kerangka untuk menampung seluruh isi website tersebut (misalnya mulai dari bagian *header*, *content* dan *footer*) agar *layout* bersifat ajeg atau presisi. Kerangka ini sering disebut *wrapper* atau *container*. Pada modul ini akan digunakan istilah *wrapper* untuk menyebut kerangka tersebut.

- Pendekatan Sebelum HTML5

Pertama, definisikan terlebih dahulu *style* untuk *wrapper*. Setelah mendefinisikan *wrapper*, langkah selanjutnya adalah membuat desain layout. Sebagai contoh, kita akan membuat layout standar (*header*, *menu*, *content*, dan *footer*) dengan *style id* terpisah.

Definisikan *style* seperti berikut dan simpan dengan nama **mystyle.css**.

```
#wrapper {  
    margin: auto;  
    width: 750px;  
    border: 1px solid red;  
}  
  
#header {  
    height: 80px;  
    border: 1px solid blue;  
}  
  
#inner {  
    float: left;  
    margin: 5px 0;  
    border: 1px solid black;  
}  
  
#sidebar {  
    float: left;  
    margin-right: 20px;  
    width: 180px;  
    height: 330px;  
    border: 1px solid red;  
}  
  
#content {  
    float: left;  
    width: 544px;  
    height: 330px;  
    border: 1px solid green;  
}  
  
#footer {  
    clear: both;  
    height: 50px;  
    border: 1px solid blue;  
}
```

Buat desain layout HTML seperti berikut:

```
<!DOCTYPE HTML>
<html lang="id">

<head>
  <title>Desain Layout Sederhana</title>
  <link rel="stylesheet" href="mystyle.css" type="text/css" />
</head>

<body>

  <div id="wrapper">
    <div id="header">
      Header
    </div>

    <div id="inner">

      <div id="sidebar">
        Sidebar
      </div>

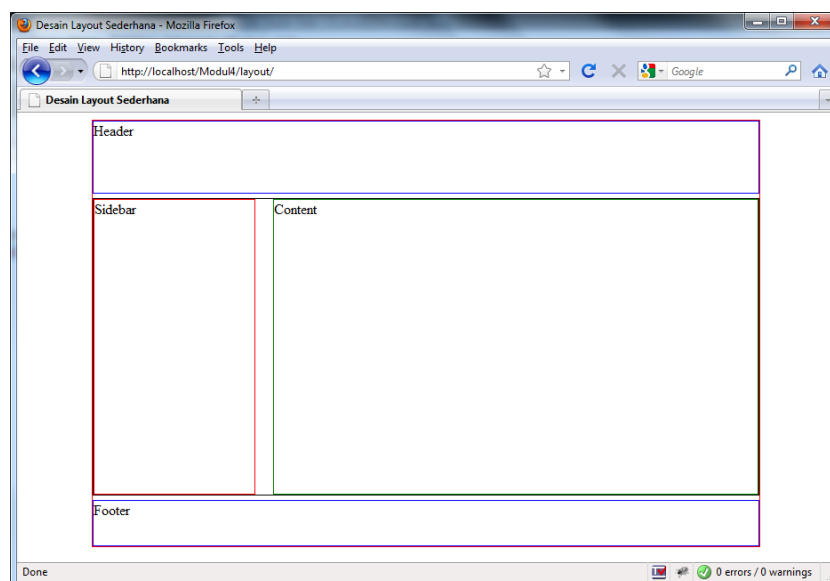
      <div id="content">
        Content
      </div>

    </div>

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

</body>
</html>
```

Hasil pembuatan desain umum sebuah halaman web akan terlihat seperti Gambar 1.



Gambar 1. Desain layout sederhana

#### ▪ Pendekatan Setelah HTML5

Di HTML5 diperkenalkan fitur-fitur baru, di antaranya adalah berkaitan dengan desain layout, misalnya elemen *header*, *nav*, *section* dan *footer*. Elemen-elemen baru tersebut membuat penyusunan *style* lebih mudah dan praktis sehingga tidak perlu lagi mendefinisikan elemen *division* sebagai pengganti elemen tersebut.

Pertama, definisikan *style* dan simpan dengan nama **mystyle2.css**.

```
body {
margin: 10px auto;
width: 750px;
}

header,nav,section,footer {
display:block;
border:1px solid blue;
}

header{
height:80px;
}

nav{
float:left;
width:200px;
height:370px;
}

section{
margin-left:200px;
width:500;
height:370px;
}

footer{
clear:both;
height:20px;
}
```

Kemudian definisikan HTML-nya, lalu simpan dengan nama **layouthtml5.html**.

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

  <head>
    <title>Desain Layout Sederhana HTML5</title>
    <link rel="stylesheet" href="mystyle2.css" type="text/css" />
  </head>

  <body>

    <header>
      header
    </header>

    <nav>
      nav
    </nav>

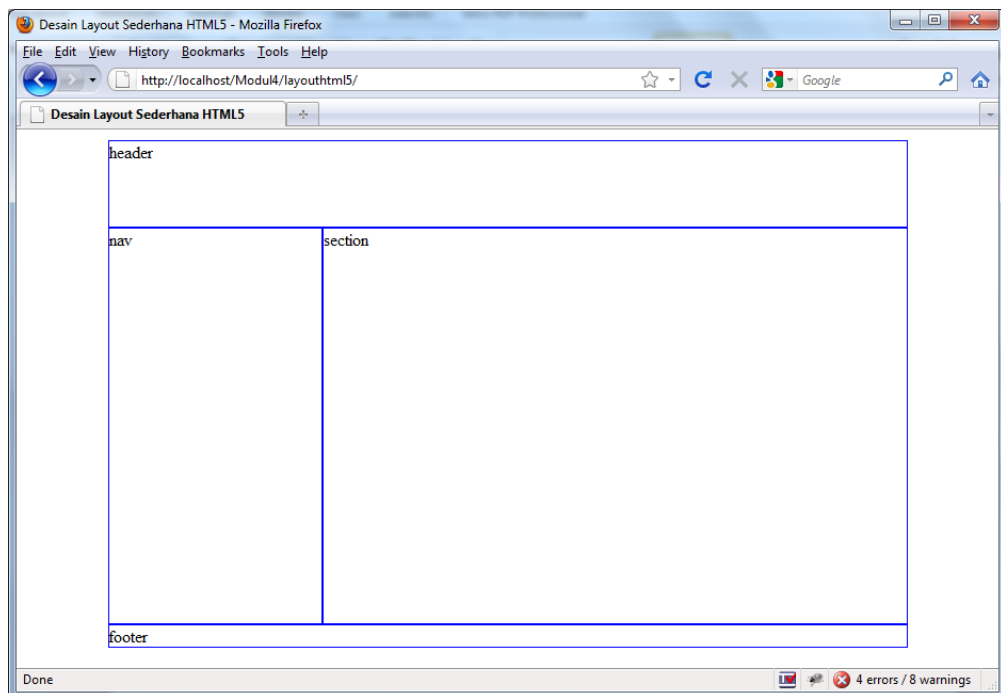
    <section>
```

```
        section
    </section>

    <footer>
        footer
    </footer>

</body>
</html>
```

Terlihat bahwa penggunaan komponen-komponen layout HTML5 menjadikan kode lebih sederhana dan natural. Hasil tampilannya akan terlihat seperti pada Gambar 2.



Gambar 2. Desain layout menggunakan HTML5

Tampilan pada web browser akan menghasilkan 4 errors/8 warnings. Hal ini disebabkan versi validator yang dipakai pada Mozilla Firefox belum mengenal elemen-elemen baru dalam HTML5 (4 error tersebut bersumber dari elemen: header, nav, section, dan footer).

## 2. Desain Background

Setelah menghasilkan desain layout, kita bisa menata background dengan memberikan sentuhan grafik yang lebih menarik. Pengaturan style yang bias diterapkan ke background bias berupa warna dan gambar.

- Background Warna

Untuk mengatur warna background, CSS menyediakan properti `background-color`. Nilai yang bisa diberikan untuk properti ini sama seperti properti warna lainnya, yaitu nama umum warna (red, green, blue, dsb), heksadesimal (#000000), atau kode RGB (255, 0, 255).

Contoh penerapan background dengan warna kuning pada halaman web adalah seperti berikut:

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

  <head>
    <title>Desain Background</title>

    <style>
      body {
        background-color: yellow;
        margin: 10px auto;
        width: 750px;
      }

      header {
        display: block;
        border: 1px solid blue;
      }

      header {
        height: 120px;
      }
    </style>
  </head>

  <body>

    <header>
      header
    </header>

  </body>
</html>
```

Hasilnya akan terlihat seperti Gambar 3.



*Gambar 3. Background Warna*

- Background Gambar

Pengaturan gambar pada halaman web menjadikan tampilan halaman lebih menarik dan hidup. Untuk melakukan pengaturan ini, terlebih dahulu kita siapkan file gambar yang akan digunakan. Langkah selanjutnya, kita bias menetapkan pengaturan melalui CSS.

Sebagai contoh, kita akan menggunakan gambar dengan ukuran 200 x 400 piksel.



*Gambar 4. Gambar untuk Background*

Langkah selanjutnya adalah memanggil dan mengatur posisi gambar di CSS.

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

  <head>
    <title>Desain Background</title>

    <style>
      body {
        background: yellow url("motif.png");
        margin: 10px auto;
        width: 750px;
      }

      header {
        display: block;
        border: 1px solid blue;
      }

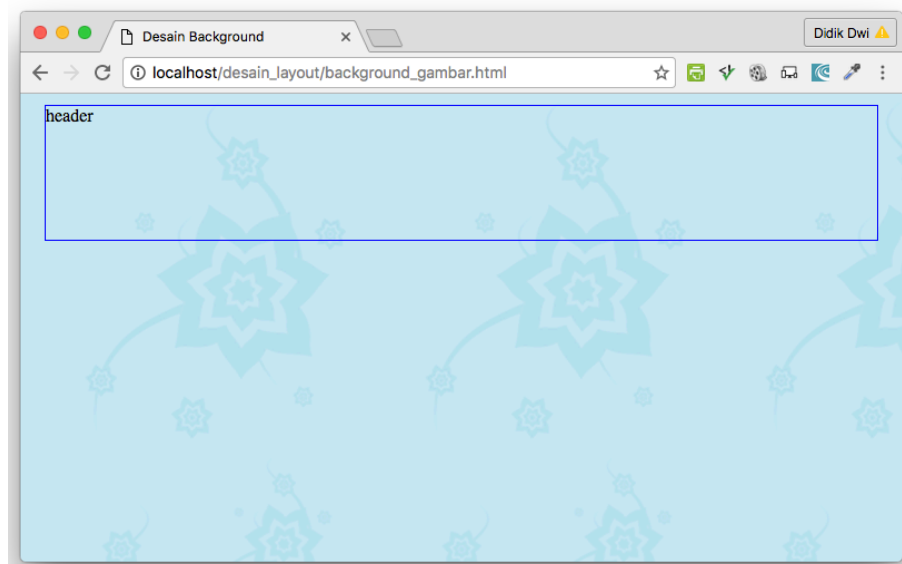
      header {
        height: 120px;
      }
    </style>
  </head>

  <body>

    <header>
      header
    </header>

  </body>
</html>
```

Hasilnya akan terlihat seperti Gambar 5.

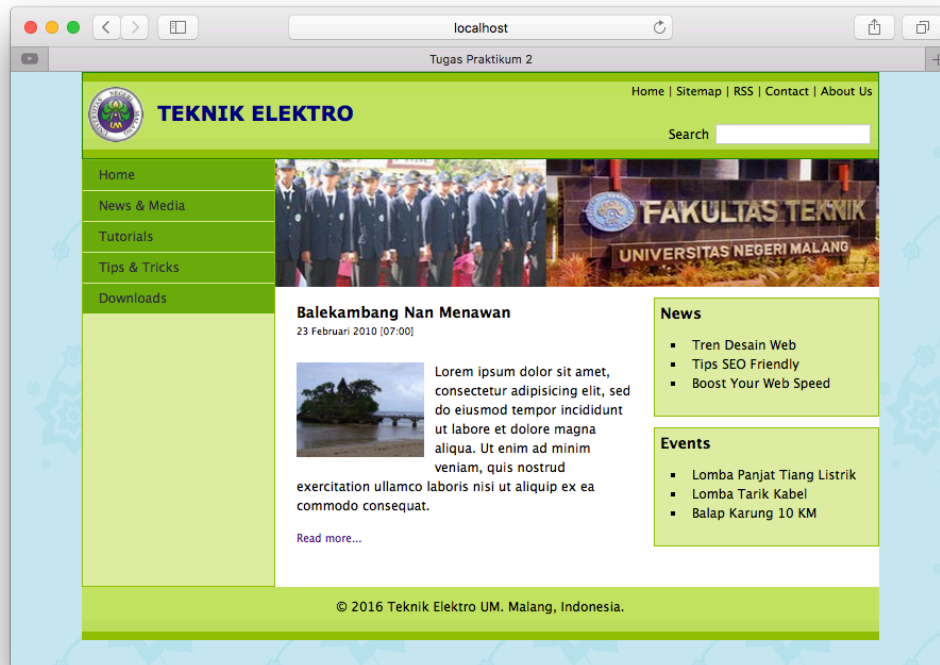


Gambar 5. Background Gambar



## F. STUDI KASUS

1. Lengkapi desain di studi kasus untuk menghasilkan contoh halaman web yang menarik. Tampilan hasilnya minimal seperti Gambar 6.



Gambar 6. Desain halaman web