

## **BAB II**

### ***CASCADING STYLE SHEET (CSS)***

#### **A. TUJUAN**

- Memahami jenis dan struktur dasar dokumen CSS
- Mampu memanfaatkan CSS untuk memformat dokumen HTML.
- Mampu memanfaatkan pendekatan CSS untuk menghasilkan halaman web yang elegan dan menarik..

#### **B. ALOKASI WAKTU**

4 JS (4 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**

##### **1. Cascading Style Sheet**

*Cascading style sheet (CSS)* merupakan sekumpulan aturan yang menyatakan bagaimana style diaplikasikan ke tag-tag HTML di dalam dokumen. Rekomendasi CSS menguraikan tiga jenis style:

- **Embedded:** properti style diletakkan di dalam satu blok di dokumen HTML.
- **Inline:** properti style diterapkan secara langsung per baris atau per elemen HTML.
- **Linked:** properti style diletakkan di file berekstensi css dan dikaitkan dengan dokumen HTML.

Selain mengulas tentang pemanfaatan CSS, bagian ini juga menekankan pada pembuatan desain dengan menggunakan division. Bagaimanapun, untuk menghasilkan desain halaman web yang baik, penggunaan tabel tidak disarankan dan sebagai gantinya adalah dengan memanfaatkan division.

##### **2. Pengenalan CSS**

CSS (Cascading Style Sheet) dapat menerapkan suatu format ketika lebih dari satu style berlaku. Misalkan kita menginginkan semua paragraph harus

memiliki font biru, namun secara khusus menginginkan satu kata agar berwarna merah. CSS bisa melakukan itu.

#### File index.html

```
<!DOCTYPE html>
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="stylesheet.css"/>
    <title>Fancy Fonts</title>
  </head>
  <body>
    <p>I'm a paragraph written in red font, but one of my words is
<span>blue</span>!/</p>
  </body>
</html>
```

#### File stylesheet.css

```
p {
    color: red;
}

span {
    /*Write your CSS here!*/
}
```

## E. LATIHAN

### 1. Menggunakan Style Sheet

Sebagaimana diketahui, ada tiga pendekatan yang bisa digunakan untuk mengimplementasikan style sheet. Bagian ini akan menjelaskan langkah penerapan ketiga pendekatan tersebut.

- **Inline**

Pada pendekatan ini, kita menerapkan style per baris atau per tag melalui atribut style.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Inline Style</title>
</head>

<body>

  <!-- menerapkan embedded style pada paragraf -->
  <p style="color:red; font-style:italic">
    Ini paragraf pertama

  <!-- menerapkan embedded style pada paragraf -->
  <p style="color:blue; font-weight:bold">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

  <p>
    Ini paragraf ketiga

</body>

</html>
```

Pendekatan inline sangat cocok digunakan untuk pemberian style khusus pada beberapa tag saja.

- ***Embedded***

Pada pendekatan ini, keseluruhan aturan style didefinisikan ke dalam satu blok baru kemudian digunakan di elemen-elemen HTML.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Embedded Style</title>

  <style type="text/css">
    <!--
    p {
      color: red;
      font-style: italic;
    }
    -->
  </style>

</head>

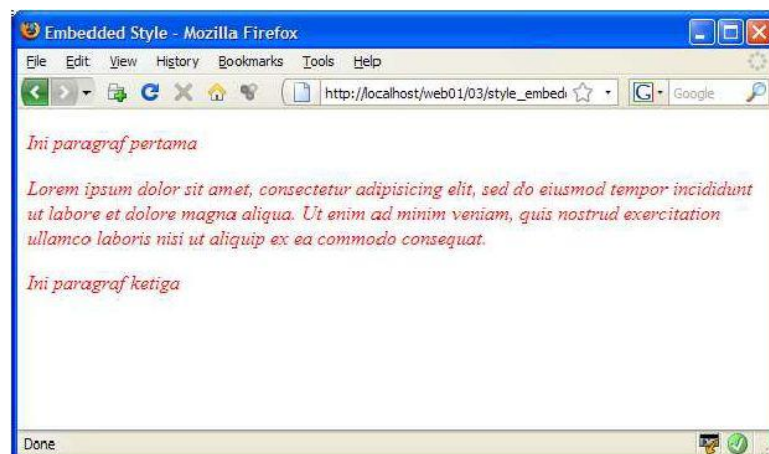
<body>
  <p>
    Ini paragraf pertama

  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  <p>
    Ini paragraf ketiga

</body>

</html>
```



Gambar 1. Tampilan penerapan embedded style

Terlihat bahwa pendefinisian style berdampak pada seluruh elemen paragraf. Bagaimana jika hanya ingin memberikan style pada elemen tertentu. Untuk lebih menspesifikasikan pemberian style, kita bisa menggunakan atribut class atau id. Tekniknya, class dinotasikan dengan titik (.); id dinotasikan dengan pagar (#).

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Embedded Style</title>

  <style type="text/css">
    <!--
    .style1 {
      color: red;
      font-style: italic;
    }

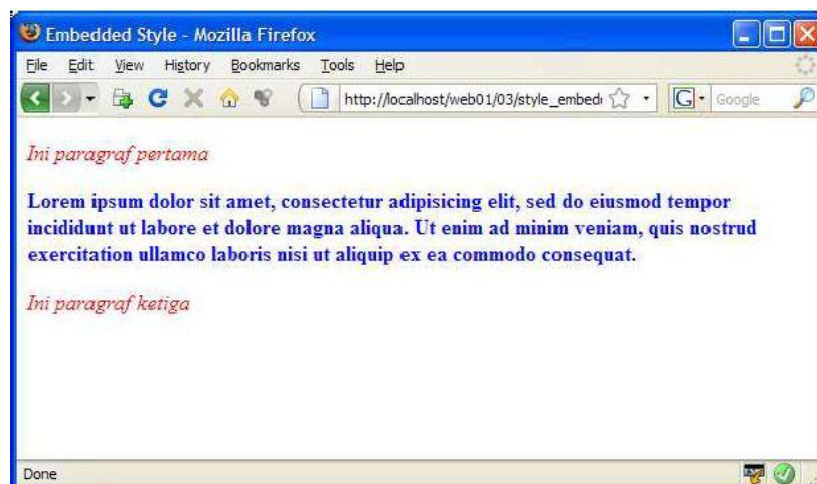
    #style2 {
      color: blue;
      font-weight: bold;
    }
    -->
  </style>
</head>

<body>
  <p class="style1">
    Ini paragraf pertama

  <p id="style2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  <p class="style1">
    Ini paragraf ketiga

</body>
</html>
```



Gambar 2. Embedded style spesifik

Sebagai tambahan, kita juga bisa mendefinisikan sebuah aturan untuk lebih dari satu tag. Perhatikan contoh berikut:

```
b, p, a {  
  color: red;  
  font-style: italic;  
}
```

Style di atas akan berlaku untuk semua tag <b>, <p>, dan <a> yang ada di halaman web.

- **Linked**

Pendekatan ini mirip dengan embedded, kecuali style-nya diletakkan di file terpisah dan berekstensi css. Pemisahan style ini juga menjadikan pendekatan ini sangat efektif dan efisien, khususnya ketika akan digunakan oleh lebih dari satu dokumen.

Langkah-langkah pembuatan file css diperlihatkan sebagai berikut:

1. Buka editor teks.
2. Ketikkan rule style seperti berikut:

```
.style1 {  
  color: red;  
  font-style: italic;  
}  
  
#style2 {  
  color: blue;  
  font-weight: bold;  
}
```

3. Simpan di satu folder dengan ekstensi css, misalnya **style.css**.

Setelah selesai mendefinisikan file style, kita bisa menggunakannya di dokumen HTML melalui suatu link.

```
<!DOCTYPE html  
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
  
  <head>  
    <title>Linked Style</title>  
  
    <link rel="stylesheet" href="style.css" type="text/css" />  
  
  </head>  
  
  <body>  
    <p class="style1">  
      Ini paragraf pertama  
  
    <p id="style2">  
      Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
      sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
  
    <p class="style1">  
      Ini paragraf ketiga  
  
  </body>  
</html>
```

## 2. Tipografi

Berkaitan dengan visualisasi teks, ada beragam style yang bisa kita berikan untuk menghasilkan bentuk sesuai keinginan. Heading



Perlu diperhatikan, dalam pembahasan ini akan digunakan pendekatan embedded. Langkah ini dimaksud untuk memfokuskan perhatian pada satu topik yang dijelaskan dalam satu dokumen. Bagaimanapun, untuk implementasi nyata disarankan menggunakan pendekatan linked style.

- **Style Font**

Ada beragam style yang bisa diterapkan pada teks, misalnya bentuk, ukuran, dan warna.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Style Font</title>

  <style type="text/css">
    <!--
    .style_font {
      font-family: Georgia;
      font-size: 18px;
      font-style: italic;
      font-weight: bold;
      text-decoration: underline;
      color: red;
    }
    -->
  </style>

</head>

<body>
  <p class="style_font">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

    <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

  </body>
</html>
```

- **Mengatur Spasi**

Properti style line-height memungkinkan kita untuk mengubah spasi standar dari suatu teks.

```
<style type="text/css">
<!--
.style_font {
  line-height: 36px;
}
-->
</style>
```

- **Initial Cap**

Jika diperlukan, kita bisa mengatur huruf pertama dari paragraph merepresentasikan huruf besar (initial cap) seperti layaknya di majalah.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Style Font</title>

  <style type="text/css">
    <!--

    p:first-letter {
      font-size: 3em;
      background-color: black;
      color: white;
    }
    -->
  </style>

</head>

<body>
  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  </body>

</html>
```



Gambar 3. Initial Cap

### 3. Elemen-Elemen Halaman

CSS dapat digunakan untuk memformat elemen-elemen HTML apa pun, misalnya border dan padding.

- **Border**

Properti border merepresentasikan batas dari suatu bidang area, misalnya paragraf. Properti ini menyediakan beberapa style yang bisa dimanfaatkan untuk membuat variasi border.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Style Sheet</title>

  <style type="text/css">
    <!--
    .border1 {
      border: 1px solid red;
    }

    .border2 {
      border: 1px dashed red;
    }

    .border3 {
      border: 1px dotted red;
    }
    -->
  </style>
</head>

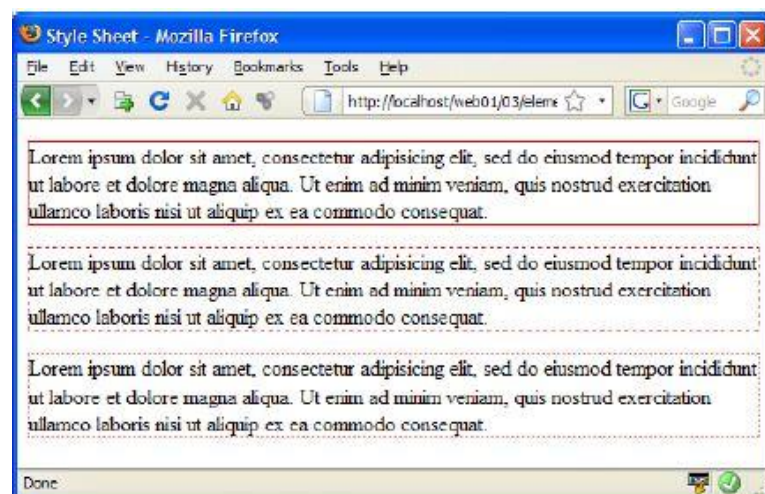
<body>

  <p class="border1">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  <p class="border2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  <p class="border3">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

</body>
</html>
```



Gambar 4. Mengatur posisi heading



- **Padding**

Seperti halnya di tabel, padding berfungsi untuk menetapkan jarak antara border dengan konten. Sintaks padding:

```
padding: nilai_semua_sisi
padding: nilai_atas nilai_kanan nilai_bawah nilai_kiri
padding-top: nilai_atas
padding-right: nilai_kanan
```

Contoh penggunaan padding:

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Style Sheet</title>

  <style type="text/css">
    <!--
    .border1 {
      border: 1px solid red;
      padding: 10px;
    }

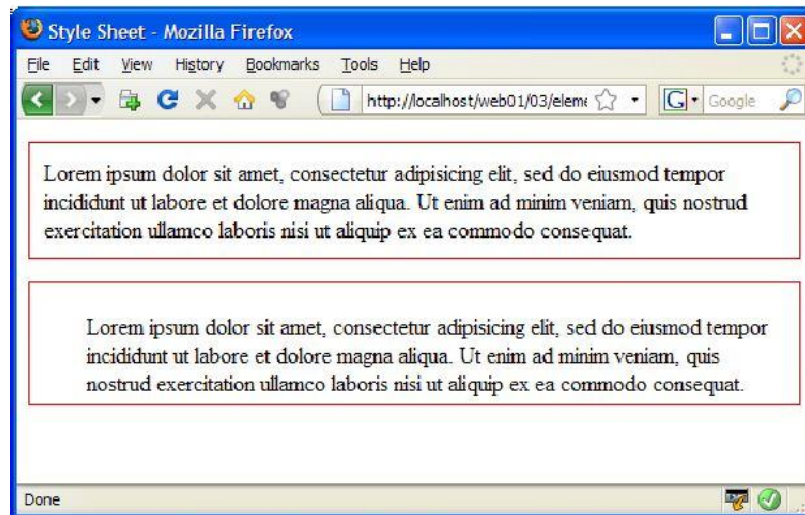
    .border2 {
      border: 1px solid red;
      padding: 20px 10px 5px 40px;
    }

    -->
  </style>
</head>

<body>
  <p class="border1">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

  <p class="border2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.

</body>
</html>
```



Gambar 5. Menggunakan paragraf

#### 4. Link

CSS juga memungkinkan kita untuk mengatur style maupun perilaku dari suatu link. Sebagai contoh, kita bisa menghilangkan garis bawah yang normalnya ada di setiap link.

```
<head>
  <title>Style Link</title>

  <style type="text/css">
    <!--
    .link_none {
      text-decoration: none;
    }
    -->
  </style>
</head>

<body>

  <a href="link2.html">Link Normal</a> <br />
  <a href="link2.html" class="link_none">Link tanpa garis bawah</a>

</body>
</html>
```

Pengaturan link yang paling menarik adalah berkaitan dengan diarahkannya kursor di atasnya.

```
<head>
  <title>Style Link</title>

  <style type="text/css">
    <!--
    .link1 a:hover {
      text-decoration: none;
    }

    .link2 a:hover {
      font-style: italic;
      font-weight: bold;
    }

    .link3 a:hover {
      cursor: wait;
    }
    -->
  </style>
</head>

<body>
  <a href="link2.html">Link Normal</a> <br />
  <p class="link1">
    <a href="link2.html">Link tanpa garis bawah</a>
  </p>

  <p class="link2">
    <a href="link2.html">Ubah style font</a>
  </p>

  <p class="link3">
    <a href="link2.html">Ubah kursor</a>
  </p>
</body>
</html>
```

## 5. List

Kemampuan menarik lainnya dari CSS adalah memformat list menjadi struktur menu yang elegan.

```
<head>
  <title>Style List</title>

  <style type="text/css">
    <!--
    #leftmenu ul {
      width: 200px;
      list-style-type: none;
      padding: 0; margin: 0;
    }
  </style>
</head>
```

```
}
#leftmenu a:link, #leftmenu a:visited, #leftmenu a:active {
  padding-left: 15px;
  text-decoration: none;
}
#leftmenu a {
  padding: 5px 0px 5px 15px; display: block;
  background: #6cae15 no-repeat left center;
  margin: 0px 0px 1px; color: #ffffff;
}
#leftmenu a:hover {
  background: #5e9711 no-repeat left center; color: #ffffff
}
-->
</style>

</head>

<body>

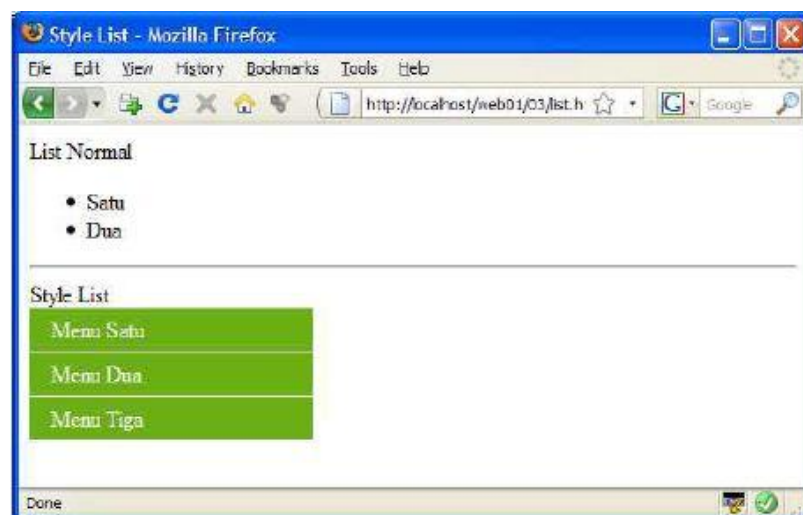
  List Normal
  <ul>
    <li>Satu</li>
    <li>Dua</li>
  </ul>

  <hr />

  Style List

  <div id="leftmenu">
  <ul>
    <li><a href="#">Menu Satu</a></li>
    <li><a href="#">Menu Dua</a></li>
    <li><a href="#">Menu Tiga</a></li>
  </ul>
  </div>

</body>
```



Gambar 6. Menu berbasis list

## 6. Menggunakan Division

Pendekatan yang efektif untuk memformat elemen-elemen HTML termasuk juga division—adalah dengan menggunakan CSS.

- **Ukuran Area**

Seperti di kebanyakan elemen, kita bisa memanfaatkan atribut width dan height untuk menspesifikasikan ukuran area. Selain itu, atribut-atribut seperti padding dan margin juga bisa dimanfaatkan di sini.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

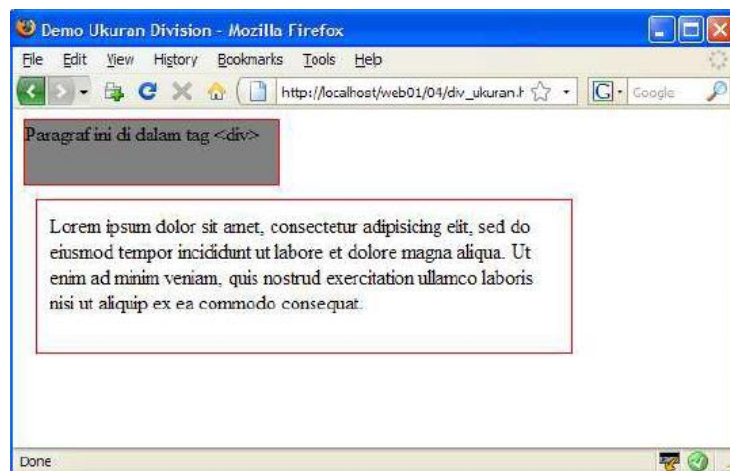
<head>
  <title>Demo Ukuran Division</title>
  <style type="text/css">
    <!--
    .box1 {
      width: 200px;
      height: 50px;
      background: grey;
      border: 1px solid red;
    }

    .box2 {
      margin: 10px;
      padding: 10px;
      width: 400px;
      height: 100px;
      border: 1px solid red;
    }
    -->
  </style>
</head>

<body>

  <div class="box1">
    Paragraf ini di dalam tag <div>
  </div>

  <div class="box2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.
  </div>
```



Gambar 7. Mengatur ukuran area



- **Format Font**

Di dalam suatu area, kita juga bisa menerapkan style spesifik yang tentunya akan terisolasi (unik) dengan keseluruhan area.

```
.box1 {  
  width: 200px;  
  height: 50px;  
  background: grey;  
  text-transform: uppercase;  
  font-weight: bold;  
  border: 1px solid red;  
}
```



Gambar 8. Mengatur font di area spesifik

- **Posisi Area**

Seperti halnya paragraf, elemen-elemen div akan menghasilkan area secara berurutan dari atas ke bawah. Adapun jika dikehendaki, kita bisa mengatur posisi area secara fleksibel dengan memanfaatkan atribut float CSS.

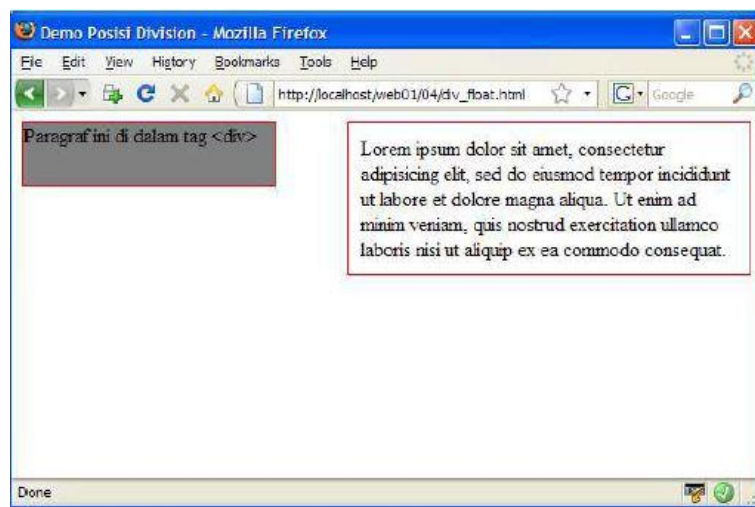
```
<!DOCTYPE html  
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  
  
  <head>  
    <title>Demo Posisi Division</title>  
    <style type="text/css">  
      <!--  
      .box1 {  
        float: left;  
        width: 200px;  
        height: 50px;  
        background: grey;  
        border: 1px solid red;  
      }  
  
      .box2 {  
        float: right;  
        padding: 10px;  
        width: 300px;  
        height: 100px;  
        border: 1px solid red;  
      }  
      -->  
    </style>  
  </head>
```

```
<body>

  <div class="box1">
    Paragraf ini di dalam tag <div>
  </div>

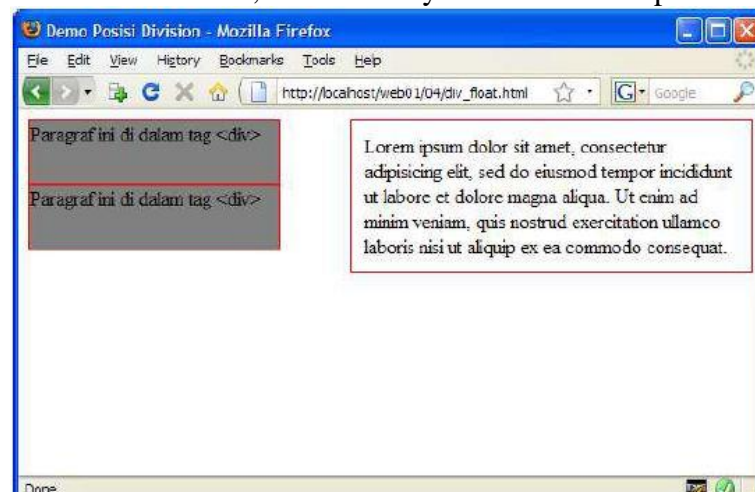
  <div class="box2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.
  </div>

</body>
</html>
```



Gambar 9. Mengatur posisi area

Perlu diperhatikan, atribut float normalnya akan mencoba memampatkan area sepanjang masih bisa dilakukan. Misalkan di dokumen sebelumnya kita tambahkan sebuah area kecil, maka hasilnya akan terlihat seperti Gambar 10.



Gambar 10. Perilaku normal atribut float

Bergantung kebutuhan, perilaku normal dari float bisa sesuai keinginan atau sebaliknya. Apabila dua kotak pertama dikehendaki sejajar sehingga

penambahan baru akan dilakukan setelah batas kota terluas (kotak kanan), kita bisa me-reset atribut float dengan menggunakan atribut clear.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Posisi Division</title>
  <style type="text/css">
    <!--
    .box1 {
      float: left;
      width: 200px;
      height: 50px;
      background: grey;
      border: 1px solid red;
    }

    .box2 {
      float: right;
      padding: 10px;
      width: 300px;
      height: 100px;
      border: 1px solid red;
    }

    .box3 {
      /* me-reset pengaturan float left maupun right */
      clear: both;
      float: left;
      width: 200px;
      height: 50px;
      background: grey;
      border: 1px solid red;
    }
    -->
  </style>
</head>

<body>

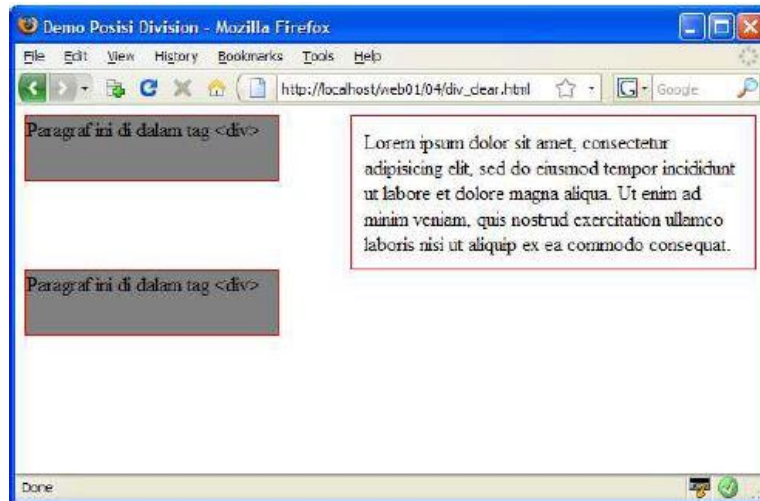
  <div class="box1">
    Paragraf ini di dalam tag &lt;div&gt;
  </div>

  <div class="box2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
    nisi ut aliquip ex ea commodo consequat.
  </div>

  <div class="box3">
    Paragraf ini di dalam tag &lt;div&gt;
  </div>

</body>

</html>
```



Gambar 11. Me-reset atribut float

## 7. Membuat Kerangka

Kerangka dasar merupakan kulit luar atau pembungkus (*wrapper*) yang umumnya digunakan untuk menampung keseluruhan desain. Sederhananya, karakteristik utama dari kerangka ini adalah terletak di tengah *web browser*. Untuk menempatkan area kerangka berada di tengah secara horizontal *browser*, kita dapat memanfaatkan atribut margin. Tekniknya, tetapkan nilai auto pada margin, di mana akan mengakibatkan posisi secara otomatis diatur di tengah.

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Demo Wrapper</title>
  <style type="text/css">
    <!--
    #wrapper {
      margin: auto;
      width: 700px;
      border: 1px solid red;
    }
    -->
  </style>
</head>

<body>

  <div id="wrapper">
    <p>
      Paragraf ini di dalam tag &lt;div&gt;

    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit,
      sed do eiusmod tempor incididunt ut labore et dolore magna
      aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.
    </p>
  </div>

</body>
```





Gambar 12. Menciptakan wrapper

## 8. Desain Layout

Setelah mempersiapkan *wrapper*, langkah selanjutnya adalah membuat desain layout. Sebagai contoh, kita akan membuat layout standar (*header*, *menu*, *content*, dan *footer*) dengan style 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 seperti berikut:

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

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

<body>

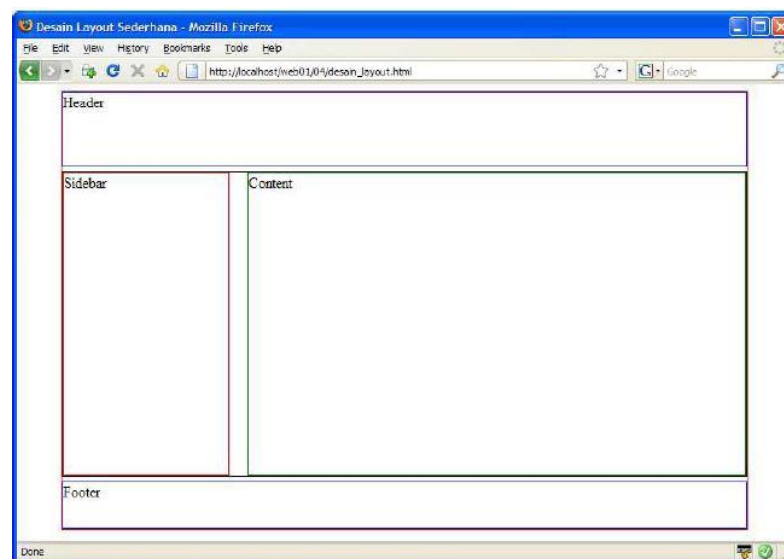
  <div id="wrapper">
    <div id="header">
      Header
    </div>

    <div id="inner">
      <div id="sidebar">
        Sidebar
      </div>
      <div id="content">
        Content
      </div>
    </div>

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

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

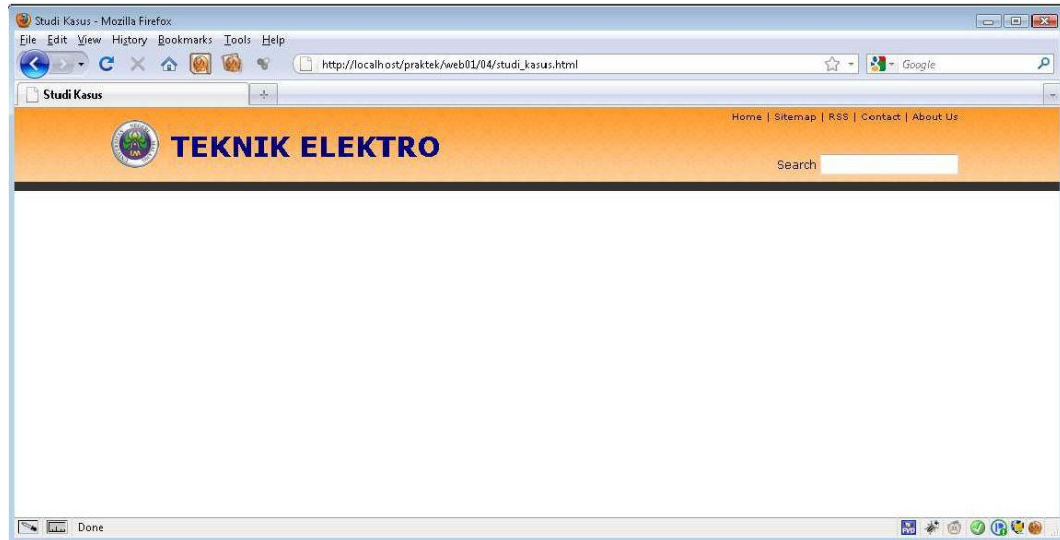
Hasilnya akan terlihat seperti Gambar 13.



Gambar 13. Desain layout sederhana

## **F. STUDI KASUS**

1. Buat desain header web memanfaatkan CSS dan background gambar terserah seperti terlihat pada gambar 14.



*Gambar 14. Desain header web*