

## BAB IV

### CASCADING STYLE SHEET (CSS)

#### A. KOMPETENSI DASAR

- Memahami jenis dan struktur dasar dokumen CSS dan CSS3.
- Mampu memanfaatkan CSS untuk memformat dokumen HTML.
- Mampu memanfaatkan pendekatan CSS untuk menghasilkan halaman web yang elegan 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

*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.

Penggunaan CSS dalam pembahasan ini juga menyinggung CSS3 yang merupakan aturan terbaru. Dengan demikian, hal ini diharapkan dapat memberikan referensi yang terkini. Selain itu, 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.

## 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>
<html 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 adipiscing 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>
<html lang="en">

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

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

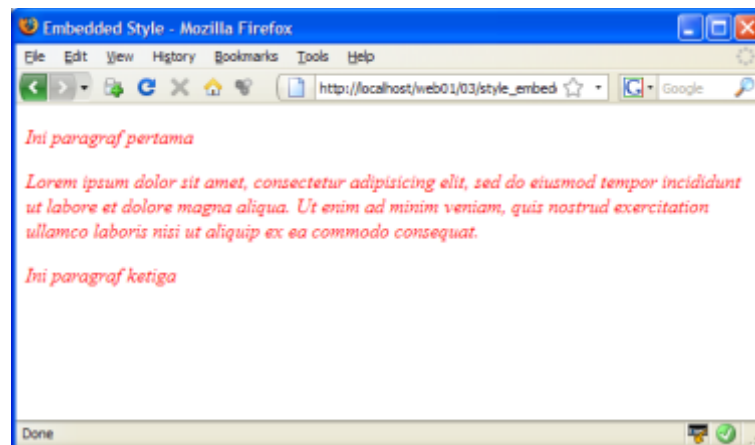
```
</head>

<body>
  <p>
    Ini paragraf pertama

  <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing 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>
<html lang="en">

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

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

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

```
-->
</style>

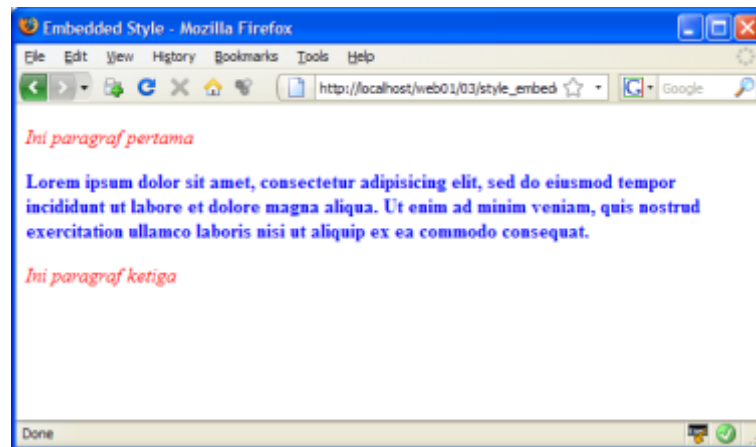
</head>

<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>
<html 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.

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>
<html 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>
```

### 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>
<html 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>

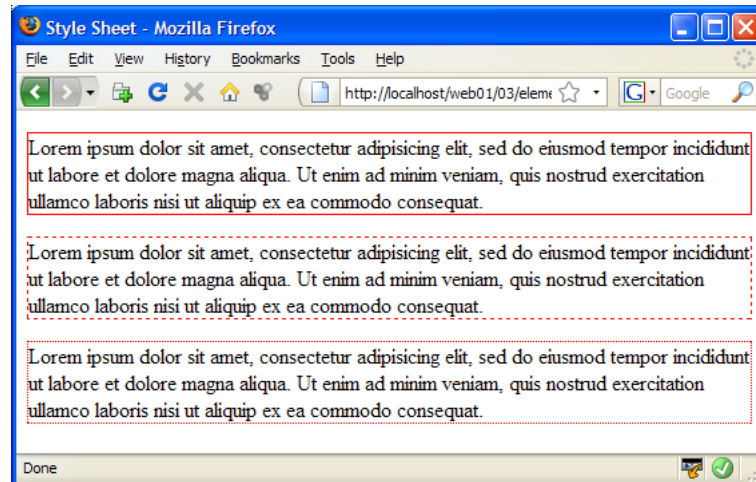
  <p class="border1">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
  Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
  nisi ut aliquip ex ea commodo consequat.

  <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 3. Menggunakan border

- 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>
<html 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>

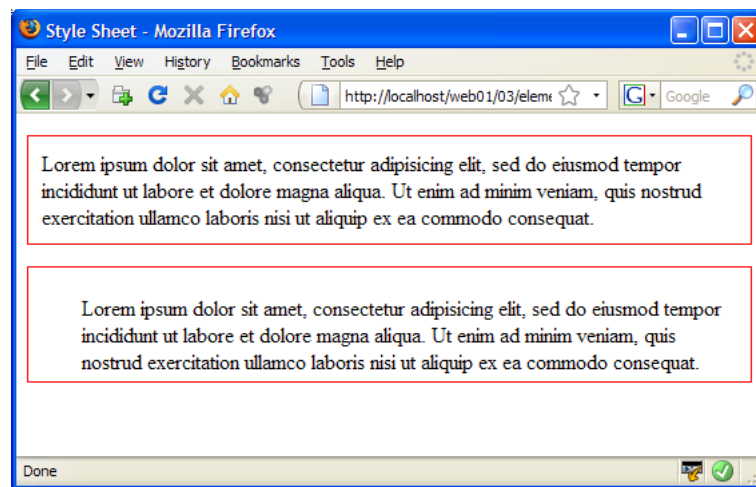
  <p class="border1">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
```



```
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 4. Menggunakan padding

#### 4. List

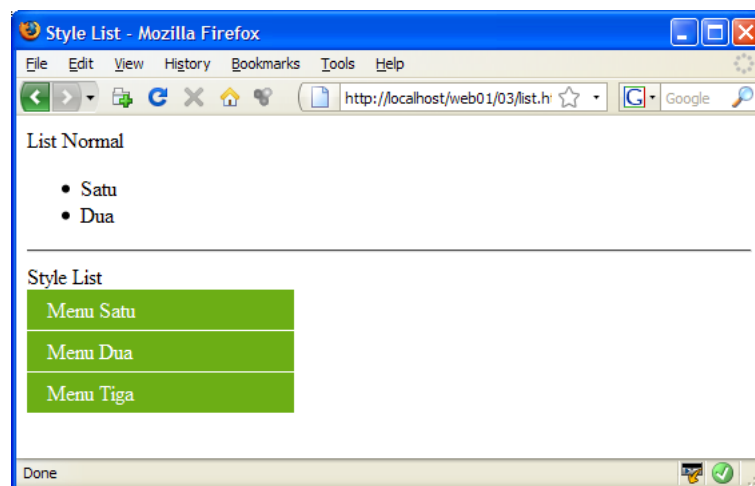
Kemampuan menarik lainnya dari CSS adalah memformat list menjadi struktur menu yang elegan.

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

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

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



Gambar 5. Menu berbasis list

## 5. 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>
<html 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>

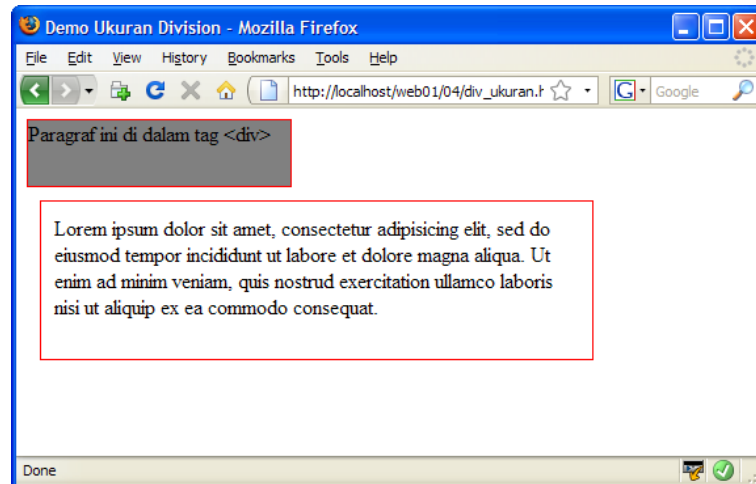
  <div class="box1">
    Paragraf ini di dalam tag &lt;div&gt;
  </div>

  <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>

</body>
</html>
```



Pada saat desain, kita bisa memanfaatkan border guna mengetahui batasan area elemen. Langkah ini juga akan digunakan dalam pembahasan-pembahasan selanjutnya



Gambar 6. Mengatur ukuran area

- 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>
<html 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;
    }

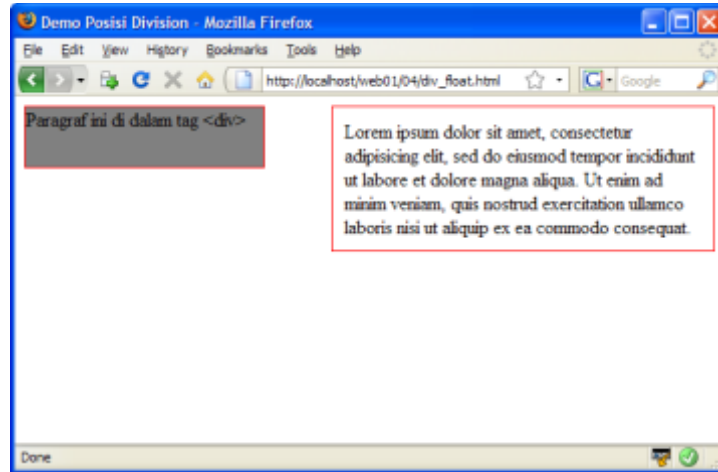
    .box2 {
      float: right;
      padding: 10px;
      width: 300px;
      height: 100px;
      border: 1px solid red;
    }
    -->
  </style>
</head>

<body>

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

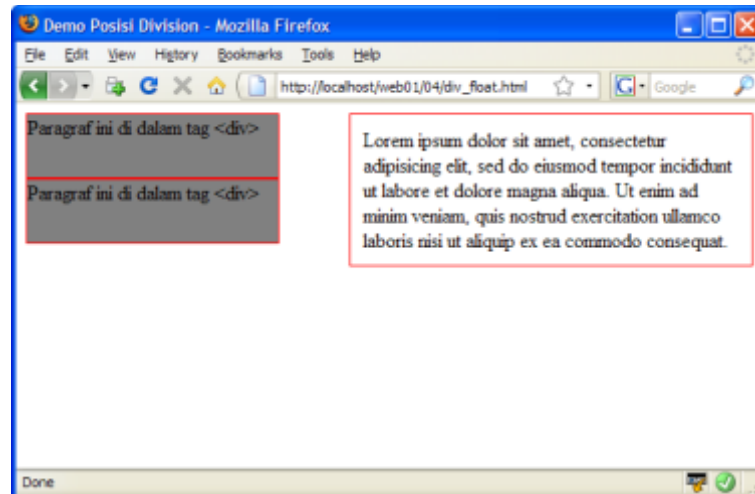
  <div class="box2">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
```

```
nisi ut aliquip ex ea commodo consequat.  
</div>  
  
</body>  
  
</html>
```



Gambar 7. 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.



Gambar 8. 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>  
<html 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>

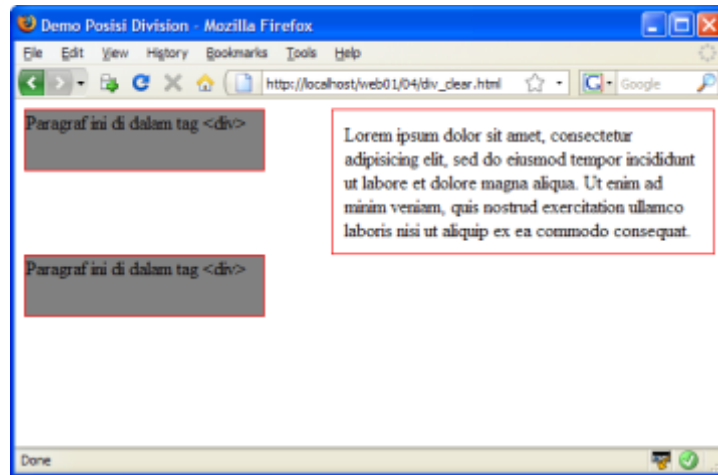
<body>

  <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 9. Me-reset atribut float

## 6. Membuat Border

Salah satu kemampuan menarik dari CSS3 adalah dalam pembuatan kotak bersudut bulat (rounded rectangle). Langkah ini dilakukan dengan memanfaatkan properti `moz` dan `webkit`.

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

<head>
  <title>Demo Rounded Border</title>
  <style type="text/css">
    <!--
    .round {
      background-color: #ddccb5;
      -moz-border-radius: 5px;
      -webkit-border-radius: 5px;
      border: 2px solid #897048;
      padding: 10px;
      width: 310px;
    }
    -->
  </style>
</head>

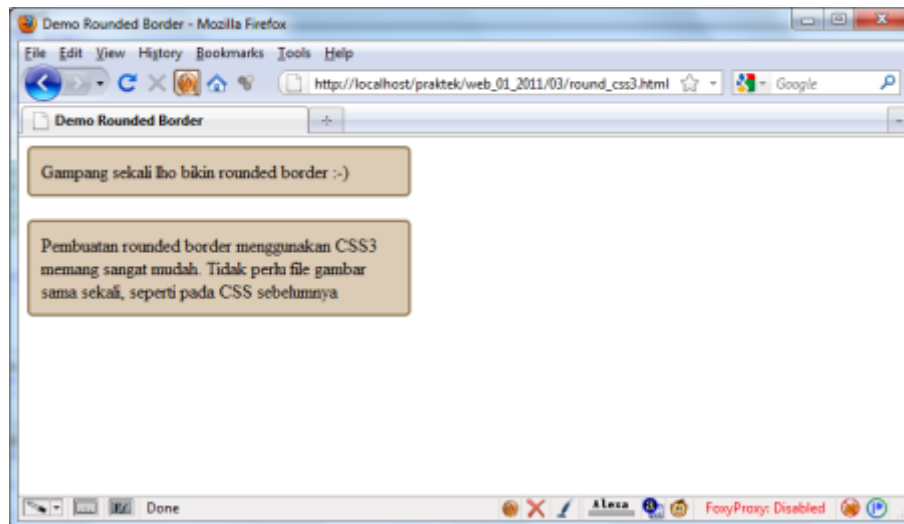
<body>

  <div class="round">
    Gampang sekali lho bikin rounded border :- )
  </div>
  <br />

  <div class="round">
    Pembuatan rounded border menggunakan CSS3 memang sangat mudah.
    Tidak perlu file gambar sama sekali, seperti pada CSS sebelumnya
  </div>

</body>

</html>
```

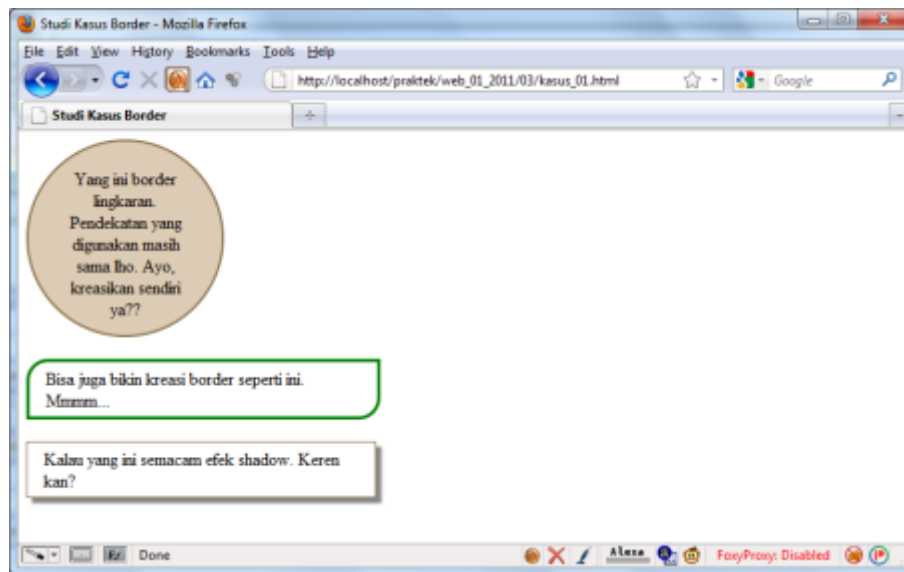


*Gambar 10. Membuat rounded border*



## F. STUDI KASUS

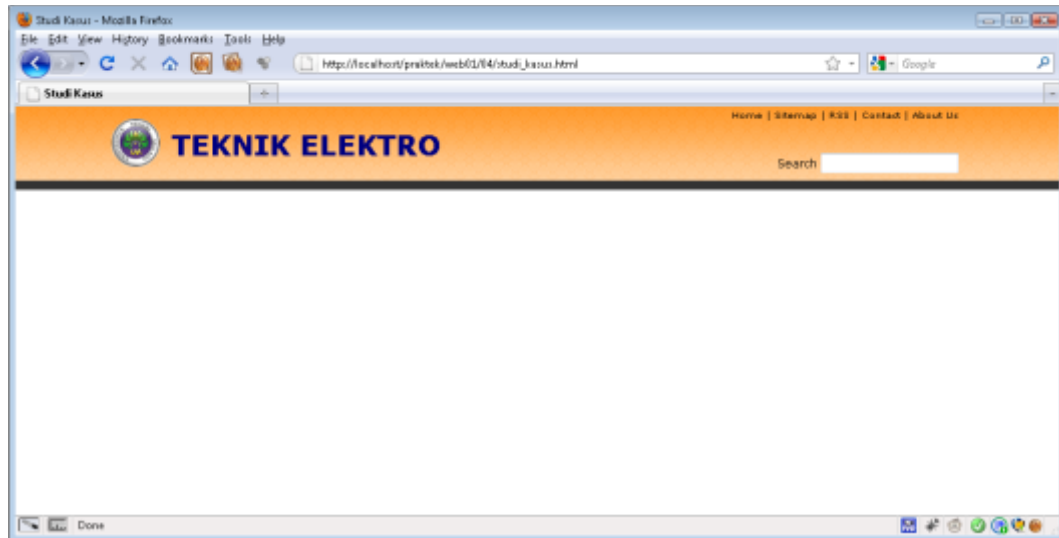
1. Buat kreasi-kreasi objek seperti terlihat pada Gambar 11.



Gambar 11. Kreasi border

## G. TUGAS PRAKTIKUM

1. Buat desain header web memanfaatkan CSS dan background gambar seperti terlihat pada Gambar 12.



Gambar 12. Desain header web

Kebutuhan gambar:

Logo:



Background:

