

BAB VI

DESAIN WEB RESPONSIF

A. KOMPETENSI DASAR

- Memahami konsep dasar desain halaman web responsif
- Mampu menerapkan konsep desain web responsif ke dalam aplikasi
- Mampu menghasilkan desain aplikasi web responsif.

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

Desain web responsif (*responsive web design* atau RWD) adalah sebuah konsep pengembangan aplikasi web yang memungkinkan layout untuk menyesuaikan dengan resolusi layar pengguna (*viewport*). Desain ini menjadikan aplikasi web mampu menyediakan tampilan yang optimal untuk semua ukuran layar pengguna. Jadi, tidak lagi pengguna yang harus menyesuaikan perangkat yang digunakan (seperti pendekatan lawas).

Konsep RWD di dalam pengembangan aplikasi web direalisasikan melalui fitur CSS3. Melalui fitur *media query*, pengembang aplikasi dapat menyediakan aturan-aturan CSS yang dapat diberlakukan untuk variasi layar pengguna. Secara sederhana, *media query* adalah filter yang akan diberlakukan terhadap halaman web sesuai dengan identifikasi *browser*.

Penerapan RWD sudah menjadi kebutuhan dalam pengembangan aplikasi web modern seperti saat ini. Bagaimanapun, pengguna aplikasi web saat ini tidak hanya memanfaatkan komputer desktop atau laptop ketika mengakses halaman web, namun sangat dinamis dan mencakup peralatan-peralatan bergerak seperti komputer tablet dan telepon pintar (*smartphone*). Bagian ini akan memberikan gambaran mengenai penerapan konsep RWD di dalam aplikasi web.

E. LATIHAN

1. Menggunakan Media Query

Konsep media query ini mirip dengan media type yang diperkenalkan oleh CSS2, yakni ketika kita menyediakan tampilan khusus untuk dokumen pencetakan. Kita bisa merasakan bagaimana mudahnya memisahkan style untuk halaman utama dan halaman khusus pencetakan.

Sebagai contoh awal untuk mengetahui cara kerja media query, kita akan menyediakan aturan khusus pada perangkat *smartphone*. Dalam hal ini kita tetapkan misalkan ukuran lebar layar *smartphone* adalah 375 piksel (contoh iPhone 6). Aturan yang kita terapkan sangat sederhana, yakni memberi *background* warna kuning jika layar teridentifikasi berukuran maksimal 375 piksel.

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

  <head>
    <title>Menggunakan Media Query CSS3</title>

    <style>

      /** Pengaturan default display none **/
      .smartphone {
        display: none;
      }

      /** Media query untuk smartphone 375px **/
      @media only screen and (max-width: 375px) {
        body {
          background-color: yellow;
        }
        .smartphone {
          display: block;
        }
      }

    </style>
  </head>

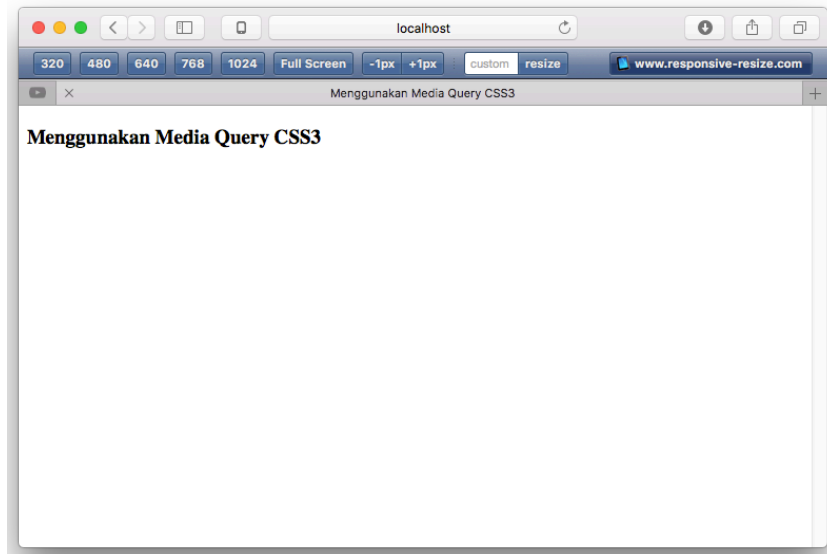
  <body>

    <h3>Menggunakan Media Query CSS3</h3>

    <div class="smartphone">
      <h2>Terdeteksi ukuran smartphone</h2>
    </div>
  </body>

</html>
```

Buka kode HTML di browser desktop dan hasilnya akan terlihat seperti pada Gambar 1.



Gambar 1. Tampilan halaman pada komputer desktop

Selanjutnya, *resize* lebar *browser* hingga merepresentasikan ukuran layar *smartphone*. Seharusnya hasilnya akan terlihat seperti pada Gambar 2.



Gambar 2. Tampilan halaman pada ukuran layar smartphone

Terlihat bahwa aturan CSS media query yang kita tetapkan sudah bekerja dan teridentifikasi dengan baik. Cukup sederhana bukan? 😊

2. Desain Halaman Web Responsif

Setelah memahami konsep desain web responsif, selanjutnya kita bisa menerapkan di dalam pembuatan aplikasi web. Di sini akan dijelaskan bagaimana membuat desain halaman web yang responsif. Ada tiga sasaran perangkat yang akan kita identifikasi, yaitu komputer desktop, tablet, dan smartphone.

```
<!DOCTYPE html>
<html lang="id">
  <head>
    <title>Desain Responsive</title>

    <style>
      #wrapper {
        margin: auto;
        width: 100%;
      }

      #header {
        height: 150px;
        padding: 20px;
        border: 2px solid red;
      }

      #nav {
        margin: auto;
        padding: 20px;
        height: 200px;
        border: 2px solid red;
      }

      .box {
        float: left;
        width: 22%;
        margin: 0 2% 0 0;
        height: 120px;
        background: red;
        border: 2px solid red;
      }

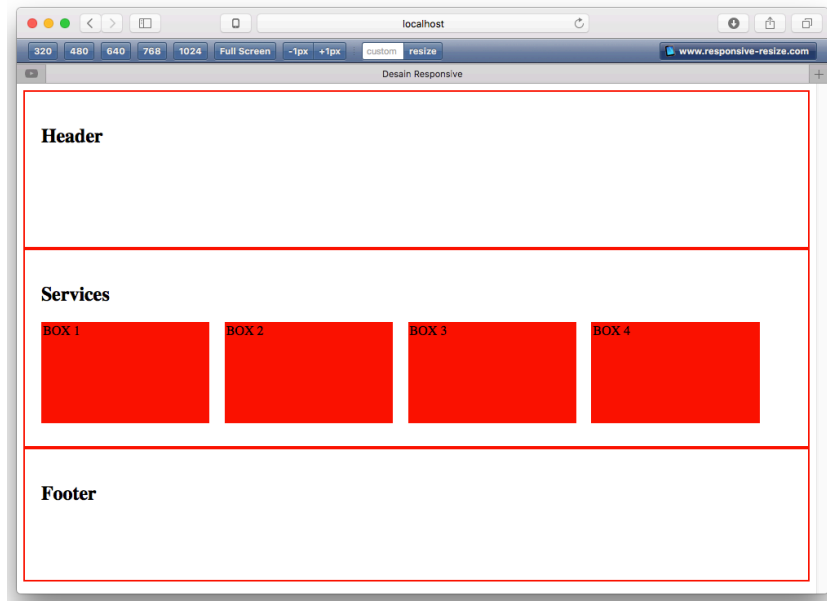
      #footer {
        padding: 20px;
        height: 120px;
        border: 2px solid red;
      }

      /** Responsif tablet **/
      @media only screen and (max-width: 768px) {
        #nav {
          height: 360px;
        }
        /** atur box 2 kolom **/
        .box {
          width: 45%;
          margin: 0 4% 20px 0;
        }
      }

      /** Responsif smartphone **/
      @media only screen and (max-width: 375px) {
        #nav {
          height: 640px;
        }
      }
    </style>
  </head>
</html>
```

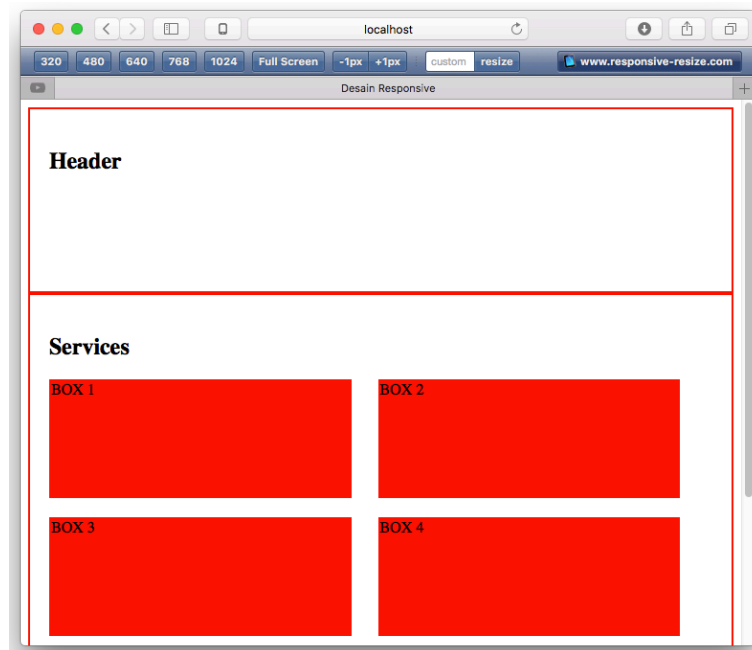
```
    /** atur box 1 kolom **/  
    .box {  
        width: 100%;  
        margin: 0 0 20px 0;  
    }  
}  
</style>  
</head>  
  
<body>  
  
    <div id="wrapper">  
  
        <div id="header">  
            <h2>Header</h2>  
        </div>  
  
        <div id="nav">  
            <h2>Services</h2>  
            <div class="box">  
                BOX 1  
            </div>  
  
            <div class="box">  
                BOX 2  
            </div>  
  
            <div class="box">  
                BOX 3  
            </div>  
  
            <div class="box">  
                BOX 4  
            </div>  
        </div>  
  
        <div id="footer">  
            <h2>Footer</h2>  
        </div>  
  
    </div>  
</body>  
</html>
```

Tampilan ketika diakses melalui komputer desktop terlihat seperti pada Gambar 3.



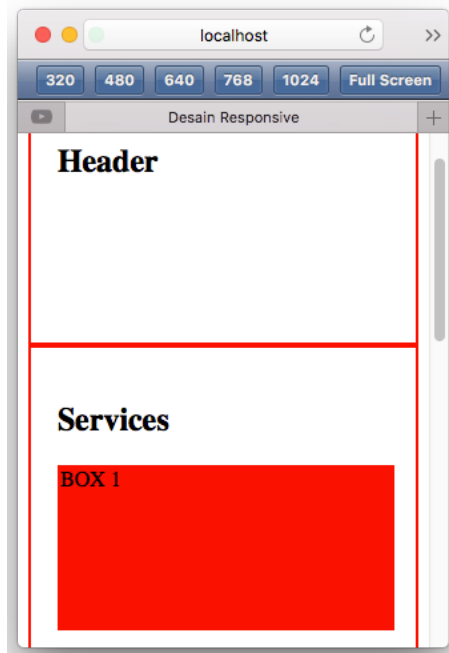
Gambar 3. Desain halaman web responsif

Aturan yang kita tetapkan untuk ukuran layar komputer tablet adalah posisi box menjadi dua kolom.



Gambar 4. Tampilan pada layar komputer tablet

Untuk ukuran layar smartphone, sebagaimana umumnya, kita buat posisi box menjadi satu kolom.



Gambar 5. Tampilan pada layar smartphone

3. Animasi Transisi

Kemampuan fenomenal lain dari CSS3 adalah menghasilkan objek-objek bergerak atau animasi. Fitur animasi transisi merepresentasikan perpindahan dari satu status visual ke status visual lainnya atau dari satu aturan ke aturan lainnya. Sederhananya, ini merupakan bentuk transisi titik ke titik. Transisi CSS merupakan jenis animasi yang paling mendasar. Realisasi konsep animasi transisi ini diimplementasikan melalui sebuah properti bernama – webkit-transition.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Animasi Transisi CSS3</title>
    <style>
      /* Kondisi box awal */
      .box {
        width: 40%;
        height: 100px;
        background: orange;
        border: 2px solid #212121;

        /* Animasi lebar box */
        -webkit-transition: width 1s ease-in-out;
        -o-transition: width 1s ease-in-out;
        -moz-transition: width 1s ease-in-out;
        transition: width 1s ease-in-out;
      }

      /* Jika landscape, perlebar 100% */
      @media (orientation:landscape) {
        .box {
          width: 100%;
        }
      }
    </style>
  </head>
</html>
```

```
    }
  </style>
</head>

<body>
  <h3>Transisi CSS3</h3>

  <div class="box"></div>

  <p>
    Ubah orientasi ke landscape/portrait dan sebaliknya
    untuk melihat efek transisi
  </p>

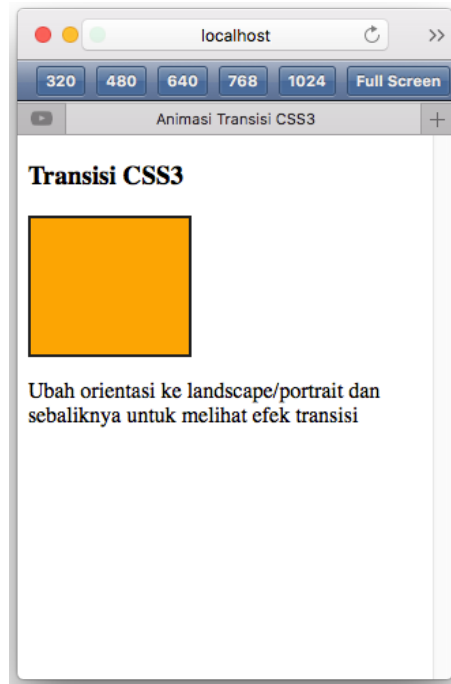
</body>
</html>
```

Tampilan ketika diakses melalui komputer desktop terlihat seperti pada Gambar 6.



Gambar 6. Efek transisi CSS3

Ketika layar diperkecil seukuran layar smartphone, objek akan mengikuti dengan diiringi efek animasi transisi.



Gambar 7. Efek transisi pada layar smartphone

F. STUDI KASUS

1. Tambahkan efek transisi pada desain halaman web responsif yang sudah Anda kerjakan.