

Nama : Anna Sanya Khazanah

NIM : K3521009

Pendidikan Teknik Informatika dan Komputer 2021

## **MODUL 6**

### **RESPONSIVE WEBSITE**

#### **A. TUJUAN PRAKTIKUM**

- a. Mahasiswa mampu menerapkan konsep desain web responsif ke dalam aplikasi
- b. Mahasiswa mampu menghasilkan desain aplikasi web responsif.
- c. Mahasiswa dapat menerapkan HTML dan CSS tingkat lanjut.

#### **B. ALOKASI WAKTU 1 x 50 menit**

#### **C. DASAR TEORI**

##### **a. Desain Web Responsif**

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.

##### **b. Langkah pengembangan desain website responsive**

Dalam proses pengembangan desain website responsive, terdapat 3 langkah yang harus diperhatikan :

###### **1) Mendefinisikan Meta Tag Viewport**

Mobile browser biasanya akan mengatur skala halaman HTML sesuai lebar viewport, yang akhirnya website dapat tampil pada layar mobile. Meta tag viewport ini digunakan untuk mereset ulang dan untuk memberitahukan kepada browser untuk menggunakan lebar perangkat sebagai acuan lebar viewport serta menonaktifkan skala awal. Anda bisa menyertakan meta tag seperti berikut ini dibagian <HEAD>

```

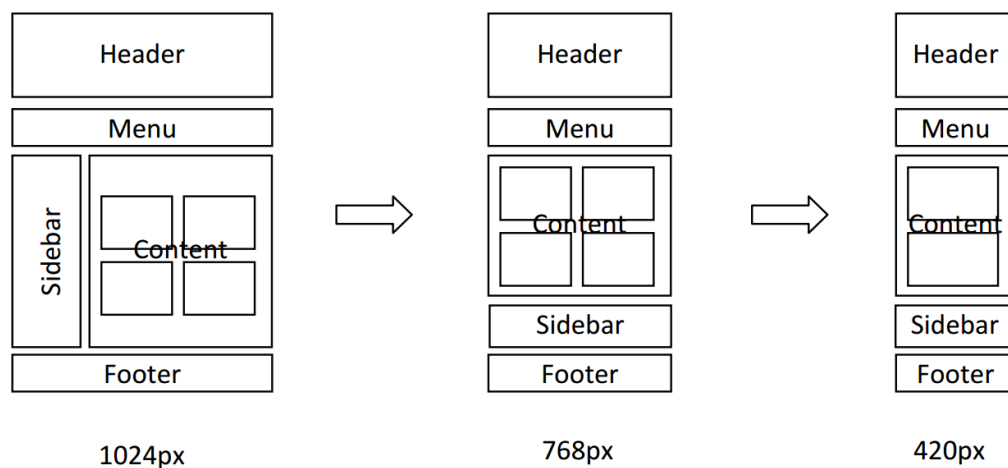
1 <html>
2   <head>
3     <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   </head>
5
6   </body>
7 </body>
8 </html>

```

## 2) Menentukan Layout dan Struktur HTML

Sebuah website biasanya terdiri dari elemen header, menu, sidebar, content, dan footer. Tinggi dan lebar masing-masing elemen sebaiknya direncanakan dulu sesuai kebutuhan sebelum melakukan coding script.

Berikut beberapa layout dan struktur yang biasa digunakan saat proses responsive diukur dari maksimal lebar layar perangkat :



## 3) Membuat Media Query di CSS

Media Query merupakan perintah di CSS3 untuk memerintahkan browser untuk proses rendering agar mengikuti ukuran sesuai script yang dituliskan. Contoh script media query adalah sebagai berikut :

```

1  /* Jika ukuran layar 680px atau kurang dari itu */
2  @media screen and (max-width:768px) {
3    #content {
4      width: auto;
5      float: none;
6    }
7
8    #sidebar{
9      width: auto;
10     float: none;
11   }
12 }

```

### c. Resolusi layar

Berikut beberapa nama atau istilah serta ukuran layar yang sudah umum.

Nama	Rasio	Panjang (piksel)	Lebar (piksel)
VGA	4:3	640	480
SVGA	4:3	800	600
XGA	4:2	1024	768
WXGA / HD	16:9	1280	720
HD	~16:9	1360	768
WXGA+	16:10	1440	900
FHD	16:9	1920	1080
WUXGA	16:10	1920	1200
WQHD	16:9	2560	1440
WQXGA	16:10	2560	1600
4K UHD	16:9	3840	2160
8K UHD	16:9	7680	4320

### 4) Flex Box CSS

Flex Box adalah model layout 1 dimensi yang dapat mengatur jarak dan alignment antar item dalam sebuah container. 1 dimensi artinya teknik pengaturan baris saja atau kolom saja. Flexbox menawarkan cara yang efektif untuk Menyusun, mensejajarkan dan mendistribusikan jarak antar item di dalam sebuah container, meskipun ukurannya tidak kita ketahui dan bersifat dinamis.

```
.container {  
  display: flex; /* atau inline-flex */  
}
```

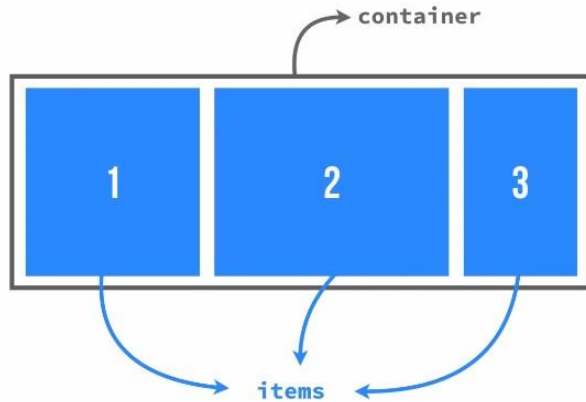
Berikut beberapa elemen yang perlu diketahui :

#### a. Container

Tempat meletakkan elemen yang bersifat sebagai pembungkus / parent

#### b. Items

Elemen yang akan dikenai konsep fleksibel box / child

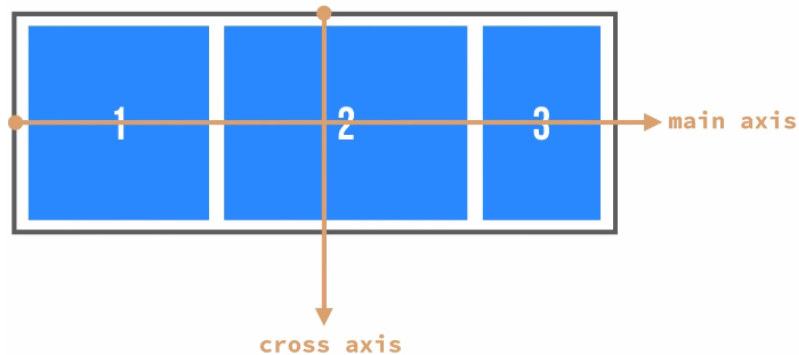


**c. Main axis**

Sumbu utama dari sebuah container yang menentukan urutan dari penempatan items secara horizontal

**d. Cross axis**

e. Sumbu pendukung dari sebuah container yang menentukan urutan dari penempatan items secara vertical

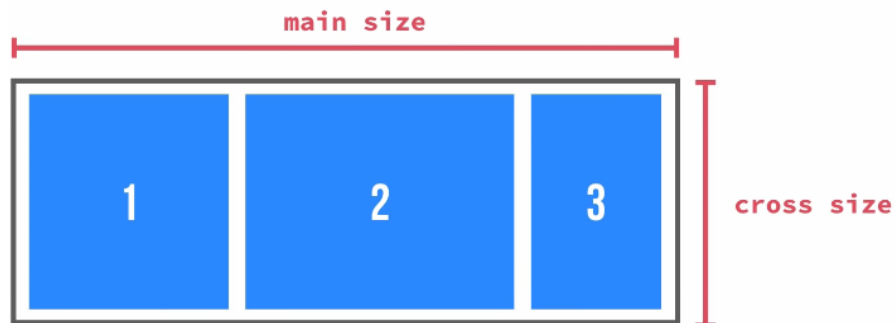


**f. Main size**

Ukuran panjang dari container yang menjadikan items bersifat relative secara horizontal

**g. Cross size**

Ukuran lebar dari container yang menjadikan items bersifat relative secara vertical

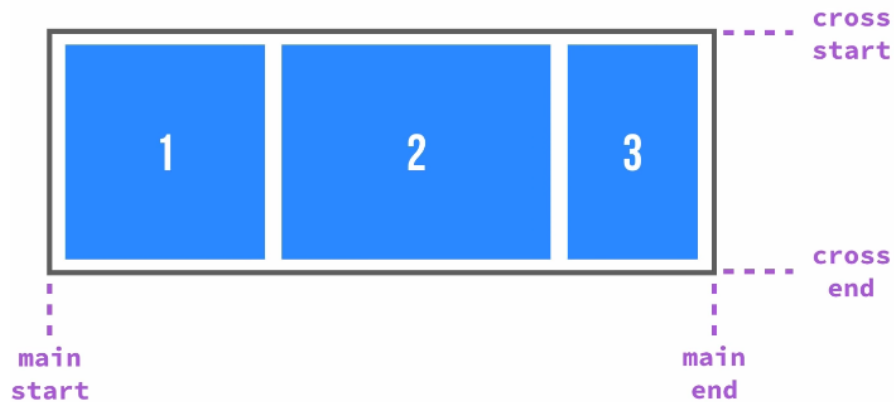


**h. Main start / end**

Mulai dan berakhirnya items yang disimpan dalam container secara horizontal

**i. Cross start / end**

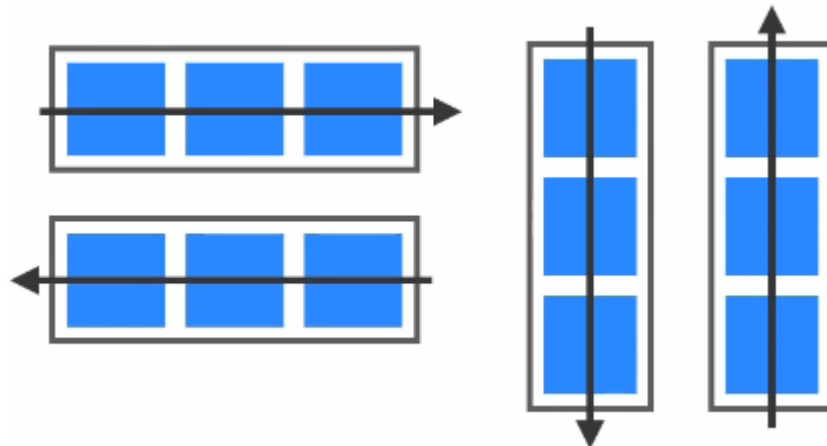
Mulai dan berakhirnya items yang disimpan dalam container secara vertical



### Flex-direction

Flex Box sebagai sarana alignment 1 dimensi memiliki flex-direction untuk menentukan ke arah mana items tersebut akan diurutkan di dalam container. Adapun value dari property flex-direction adalah sebagai berikut :

```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```



### Flex-wrap

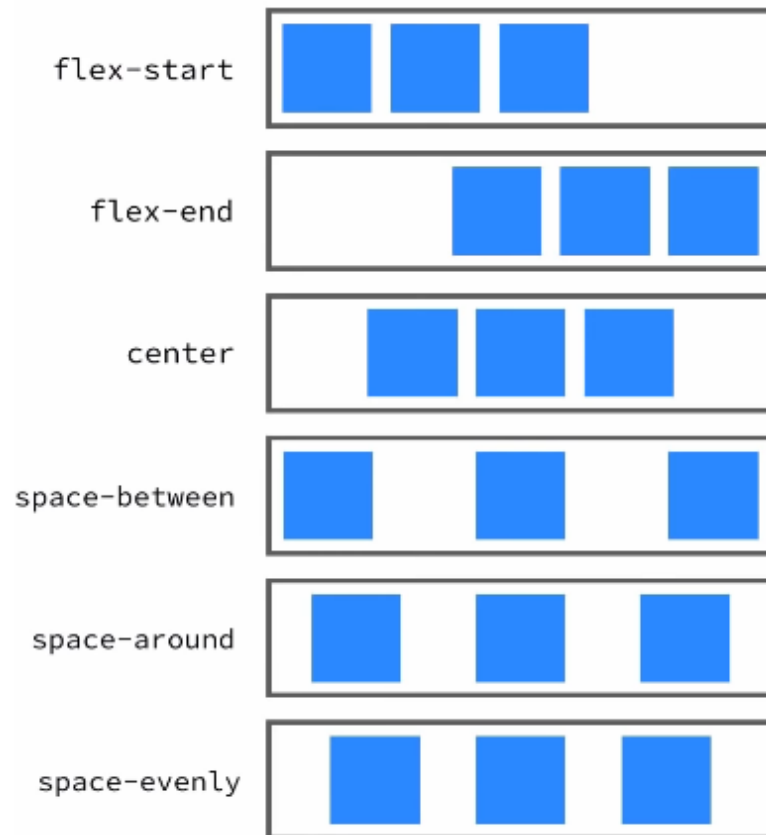
Flex-wrap akan membuat items yang berada di dalam satu baris pada container menjadi pindah ke baris selanjutnya. Pada umumnya semua items yang ada di dalam container flex box akan berada pada satu baris meskipun ukurannya sudah tidak cukup.

```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```



### Justify-content

Justify-content merupakan property yang membantu penataan jarak antar items yang ada di dalam container terhadap main axis.



```
.container {
  justify-content: flex-start | flex-end | center
    space-between | space-around | space-evenly;
}
```

## Align-items

Align-items merupakan property yang membantu pengaturan kesejajaran antar items yang ada di dalam container terhadap cross axis.

```
.container {
  align-items: flex-start | flex-end | center
    stretch | baseline;
}
```

flex-start



flex-end



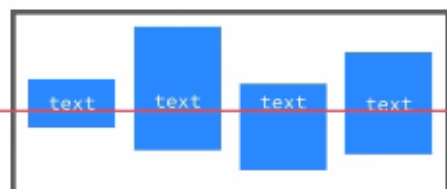
center



stretch



baseline



## D. PRAKTIKUM

### 1. Menggunakan Media Query

Konsep media query ini mirip dengan media type yang diperkenalkan oleh CSS 2, yakni ketika kita menyediakan tampilan khusus untuk dokumen tertentu. Kita bisa merasakan bagaimana mudahnya memisahkan style untuk halaman utama dan halaman khusus lainnya.

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 428 piksel (contoh iPhone 13). Aturan yang kita terapkan sangat sederhana, yakni memberi background warna kuning jika layar teridentifikasi berukuran maksimal 428 piksel.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Menggunakan media Query CSS3</title>
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7    <style>
8      .smartphone{
9        display: none;
10     }
11
12     @media only screen and (max-width: 428px){
13       body{
14         background-color: yellow;
15       }
16
17       .smartphone{
18         display: block;
19       }
20     }
21   </style>
22
23 </head>
24 <body>
25   <h3>Menggunakan Media Query CSS 3</h3>
26   <div class="smartphone">
27     <h2>Terdeteksi ukuran smartphone</h2>
28   </div>
29
30 </body>
31 </html>
```

Simpan kode tersebut dengan nama **latihan1a.html**. Buka kode HTML di browser desktop dan hasilnya akan terlihat seperti pada gambar di bawah ini.





Selanjutnya, resize lebar browser hingga merepresentasikan ukuran layar smartphone. Seharusnya hasilnya akan terlihat seperti pada Gambar di bawah ini.



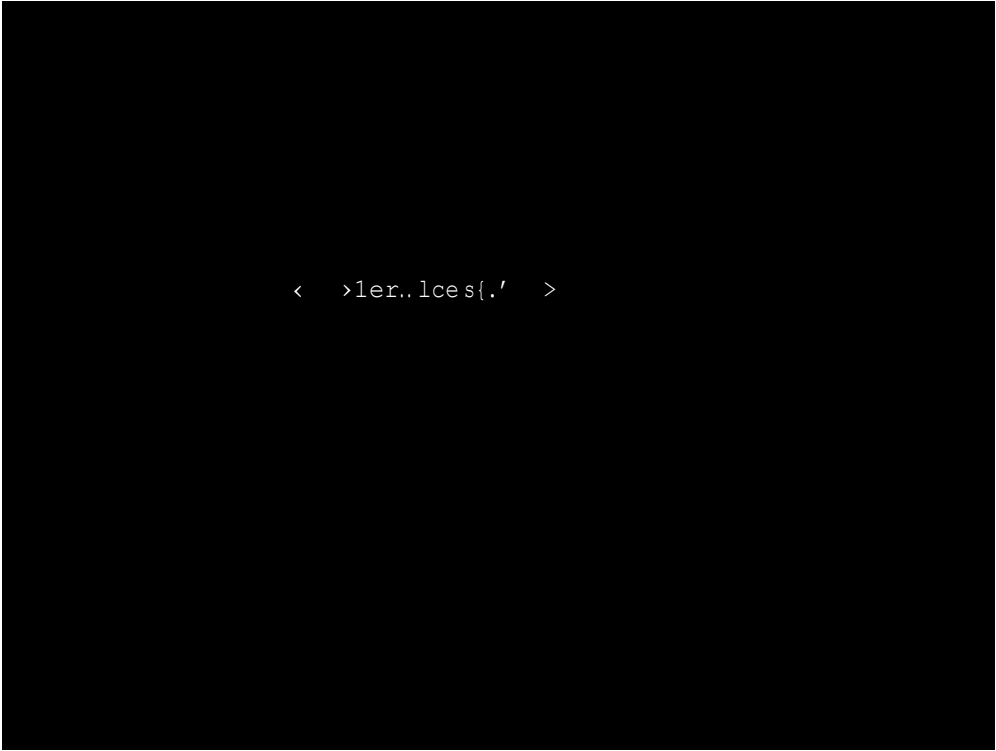
Terlihat bahwa aturan CSS media query yang kita tetapkan sudah bekerja dan teridentifikasi dengan baik.

**Dengan cara serupa, munculkan logo UNS (file cari sendiri) yang hanya nampak ketika ukuran layar lebih dari 1024 piksel dan simpan dengan nama latihan1b.html!**

## 2. Desain Halaman Web Responsif

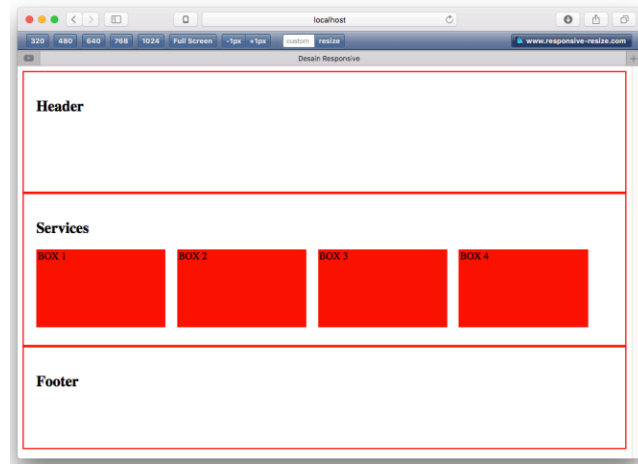
Setelah memahami konsep desain web responsif, selanjutnya kita bisa menerapkan di dalam pembuatan aplikasi web. Ada tiga sasaran perangkat yang akan kita identifikasi, yaitu komputer desktop, tablet, dan smartphone.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Desain Responsif</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <style type="text/css">
8          #wrapper{
9              margin: auto;
10             width: 100%;
11         }
12
13         #header{
14             height: 150px;
15             padding: 20px;
16             border: 2px solid red;
17         }
18
19         #nav{
20             margin: auto;
21             padding: 20px;
22             height: 200px;
23             border: 2px solid red;
24         }
25
26         .box{
27             float: left;
28             width: 22%;
29             margin: 0 2% 0 0;
30             height: 120px;
31             background: red;
32             border: 2px solid red;
33         }
34
35         #footer{
36             padding: 20px;
37             height: 120px;
38             border: 2px solid red;
39         }
40     </style>
41 </head>
42 </html>
```

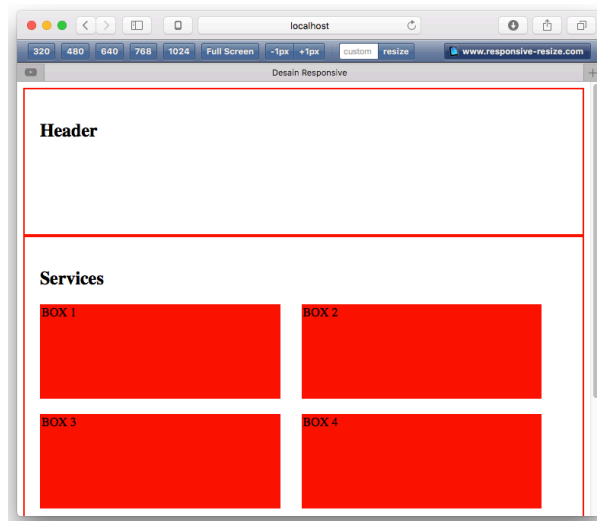


< >1er.,lces{.' >

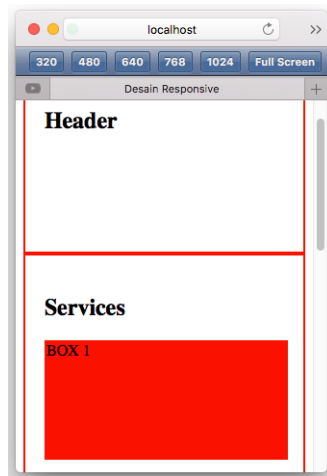
Simpan kode tersebut dengan nama **latihan2a.html**. Buka kode HTML di browser desktop dan hasilnya akan terlihat seperti pada gambar di bawah ini.



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



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



**Ubah media query di atas menjadi bentuk responsive menggunakan flex lalu simpan dengan nama latihan2b.html!**

### 3. Animasi Transisi

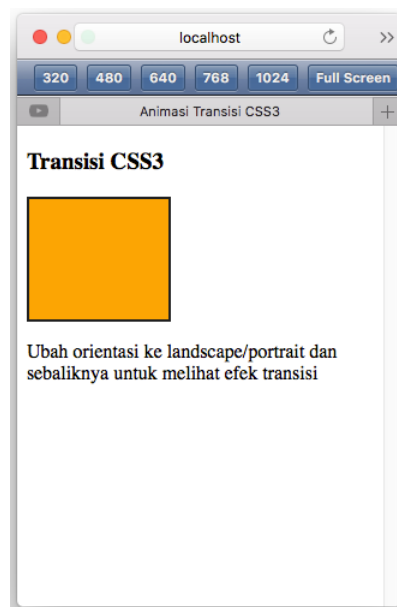
Kemampuan lain dari CSS3 adalah menghasilkan objek-objek bergerak atau animasi. Fitur animasi transisi merepresentasikan perpindahan dari satu status visual ke status visual lainnya atau dari satu aturan ke aturan lainnya. Sederhananya, ini merupakan bentuk transisi titik ke titik. Transisi CSS merupakan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

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Animasi Transisi CSS3</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <style type="text/css">
8          .box{
9              width: 40%;
10             height: 100px;
11             background: orange;
12             border: 2px solid #212121;
13
14             -webkit-transition: width 1s ease-in-out;
15             -o-transition: width 1s ease-in-out;
16             -moz-transition: width 1s ease-in-out;
17             transition: width 1s ease-in-out;
18         }
19
20         /*landscape*/
21         @media (orientation:landscape){
22             .box{
23                 width: 100%;
24             }
25         }
26     </style>
27
28 </head>
29 <body>
30     <h3>Transisi CSS3</h3>
31
32     <div class="box"></div>
33     <p>
34         ubah orientasi ke landscape/potrait dan sebaliknya untuk melihat efek transisi
35     </p>
36 </body>
37 </html>
```

Simpan kode tersebut dengan nama **latihan3a.html**. Buka kode HTML di browser desktop dan hasilnya akan terlihat seperti pada gambar di bawah ini.



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



**Dengan transition, ubah bentuk objek menjadi oval dan warna dari objek menjadi biru pada saat landscape!**

**Masih pada file yang sama, berikan animasi (animation dan keyframes) objek berputar 360 derajat infinite linier dengan tempo 2 detik pada saat landscape sehingga nampak seperti baling-baling pesawat terbang!**

**Simpan dengan nama latihan3b.html**

### Transisi CSS3



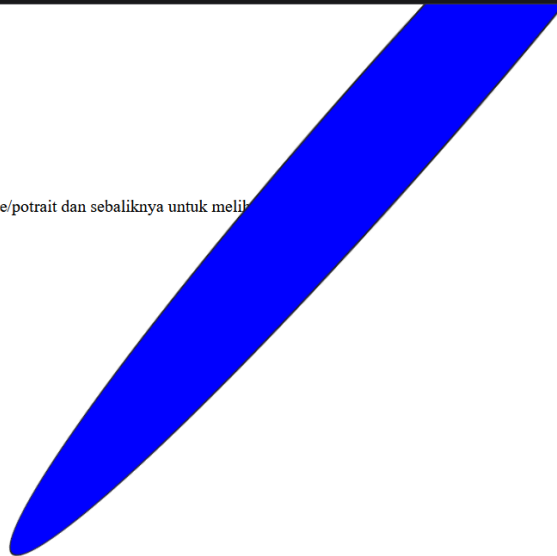
ubah orientasi ke landscape/potrait dan sebaliknya untuk melihat efek transisi

### Transisi CSS3



ubah orientasi ke landscape/potrait dan sebaliknya untuk melihat efek transisi

### Transisi CSS3



ubah orientasi ke landscape/potrait dan sebaliknya untuk melihat efek transisi