



**PROGRAM STUDI
SISTEM INFORMASI**
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
Pengantar Pemrograman Web



CSS #4: Layouting

Tim Pengajar
2021

Capaian Pembelajaran

Mahasiswa memahami proses layouting dengan css

Kemampuan Akhir yang Diharapkan

Mahasiswa dapat melakukan layouting situs web dengan css



Position

Properti position berperan penting dalam layouting css. Berfungsi untuk menentukan jenis posisi yang digunakan pada elemen. Ada 5 jenis posisi:

- Static
- Relative
- Fixed
- Absolute
- sticky

Untuk memposisikan elemen menggunakan property top, bottom, left, dan right. Tetapi property ini tidak akan bekerja sebelum ditentukan dulu property position. Properti top, bottom, left, dan right juga akan bekerja berbeda tergantung jenis posisinya

Position: Static

Secara default elemen html diposisikan secara static. Posisi static tidak akan terpengaruh oleh property top, bottom, left, dan right. Elemen dengan posisi static tidak akan diposisikan secara khusus. Mereka diposisikan menurut aliran normal halaman (aliran normal dari atas kebawah, kiri ke kanan). Contoh:

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```

Position: Relative

Elemen dengan posisi relative akan diposisikan relative terhadap posisi normalnya. Properti top, bottom, left, dan right akan membuat posisi elemen diatur berdasarkan posisi normalnya. Konten yang lain tidak akan disesuaikan posisinya untuk mengisi kekosongan yang dibuat oleh elemen relative. Contoh:

```
div.relative {  
  position: relative;  
  left: 30px;  
  border: 3px solid #73AD21;  
}
```

position: relative;

An element with position: relative; is positioned relative to its normal position:

This div element has position: relative;

Elemen div bergeser kekanan kena perintah left 30px dengan posisi relative sehingga meninggalkan ruang kosong. Ruang ini tidak akan diisi oleh konten lain.

Position: Fixed

Elemen dengan posisi fixed akan diposisikan relative terhadap viewport (browser) sehingga dia akan tetap ditempatnya meski halaman diskroll. Properti top, bottom, left, dan right digunakan untuk mengatur posisinya. Elemen fixed tidak meninggalkan ruang kosong dimana dia normalnya berada. Contoh:

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}
```

Elemen div diposisikan dipojok kanan bawah dengan bottom dan right 0. Dan akan tetap ditempatnya meski halaman diskroll

position: fixed;

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled:



This div element has position: fixed;

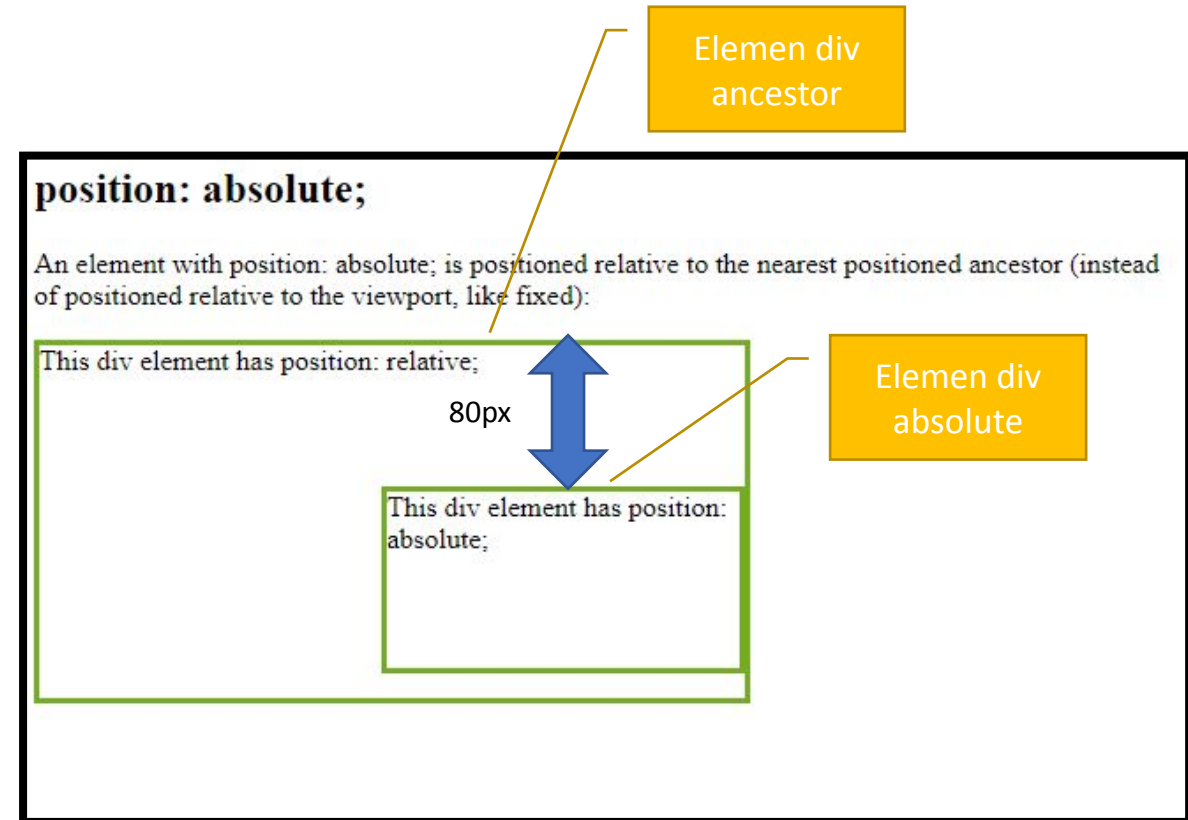
Position: Absolute

Elemen dengan posisi absolute akan diposisikan relative terhadap posisi ancestor(induk/container) terdekat.
Contoh:

```
div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

Elemen div yang didalam berposisi absolute dan diposisikan relative terhadap elemen div satunya (ancestor terdekatnya)



Position: Sticky

Elemen dengan posisi sticky akan diposisikan relative terhadap posisi skrol pengguna. Elemen dengan posisi sticky akan bergantian jenis posisinya antara relative dan fixed tergantung posisi skrol. Elemen diposisikan relative sampai posisi offsetnya sama dengan viewport lalu elemen akan di 'tancap' ditempat itu:

```
div.sticky {  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```

Posisi offsetnya adalah top 0. Selama belum berada di posisi tersebut elemen div akan bergulir mengikuti skrol pengguna, jika sudah di posisi top 0 maka elemen akan berhenti bergerak mengikuti skrol.

Sticky element

Try to **scroll** inside this frame to understand how sticky positioning works.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Scroll back up to remove the stickyness.

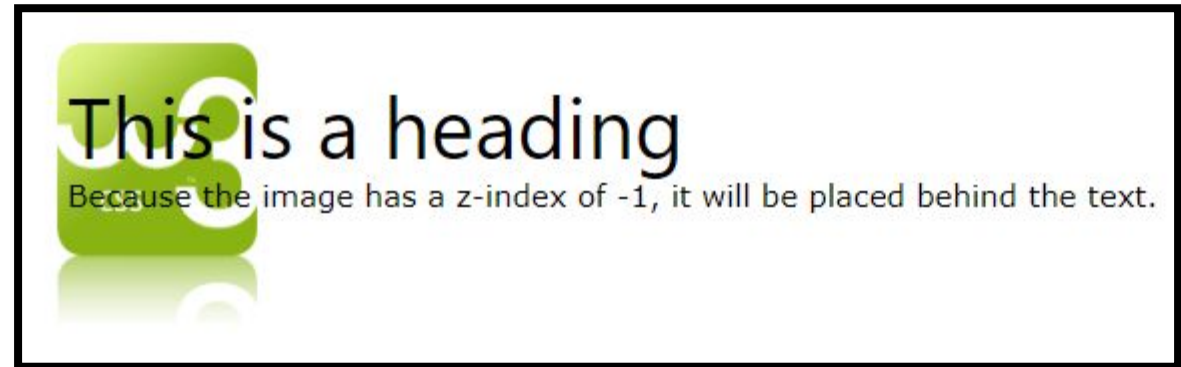
Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisse concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.

Overlapping Element

Ketika elemen-elemen diposisikan mereka dapat bertumpang tindih. Properti z-index menentukan urutan tumpukan sebuah elemen. Nilai property dapat negative. Contoh:

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

```
<h1>This is a heading</h1>  
  
<p>Because the image has a z-index of -1, it will be placed  
behind the text.</p>
```



Layouting CSS

Layouting atau pengaturan tata letak situs web dilakukan dengan CSS. Sebuah situs web biasanya dibagi menjadi bagian header, menu navigasi, konten, dan footer:



Layouting CSS

Untuk mudah melakukan layouting, semua property yang telah dijelaskan dalam slide ini diperlukan. Maka perlu dipahami dan banyak latihan agar dapat mengatur tata letak situs web dengan css. Berikut adalah contoh tata letak halaman yang diatur menggunakan semua property diatas.

Contoh tersebut menerapkan Responsive Web Design yang sudah umum dilakukan dalam mengembangkan sebuah situs web. Dengan RWD ini maka situs web akan disesuaikan tampilannya mengikuti ukuran layar perangkat dimana situs web tersebut ditampilkan

My Website

Resize the browser window to see the effect.

Link Link Link Link

TITLE HEADING

Title description, Dec 7, 2017

image

Some text...

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

TITLE HEADING

Title description, May 2, 2017

image

Some text...

Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

About Me

image

Some text about me in culpa qui officia deserunt mollit anim.

Popular Post

image

image

image

Follow Me

Some text...

Footer

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

body {
  font-family: Arial;
  padding: 10px;
  background: #f1f1f1;
}

/* Header/Blog Title */
.header {
  padding: 30px;
  text-align: center;
  background: white;
}

.header h1 {
  font-size: 50px;
}

/* Style the top navigation bar */
.topnav {
  overflow: hidden;
  background-color: #333;
}
```

```
/* Style the topnav links */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
  float: left;
  width: 75%;
}

/* Right column */
.rightcolumn {
  float: left;
  width: 25%;
  background-color: #f1f1f1;
  padding-left: 20px;
}
```

```
/* Fake image */
.fakeimg {
  background-color: #aaa;
  width: 100%;
  padding: 20px;
}

/* Add a card effect for articles */
.card {
  background-color: white;
  padding: 20px;
  margin-top: 20px;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Footer */
.footer {
  padding: 20px;
  text-align: center;
  background: #ddd;
  margin-top: 20px;
}
```

```
/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of each other instead of next to each other */
@media screen and (max-width: 800px) {
  .leftcolumn, .rightcolumn {
    width: 100%;
    padding: 0;
  }
}

/* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top of each other instead of next to each other */
@media screen and (max-width: 400px) {
  .topnav a {
    float: none;
    width: 100%;
  }
}
</style>
</head>
```

```
<body>

<div class="header">
  <h1>My Website</h1>
  <p>Resize the browser window to see the effect.</p>
</div>

<div class="topnav">
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#">Link</a>
  <a href="#" style="float:right">Link</a>
</div>

<div class="row">
  <div class="leftcolumn">
    <div class="card">
      <h2>TITLE HEADING</h2>
      <h5>Title description, Dec 7, 2017</h5>
      <div class="fakeimg" style="height:200px;">Image</div>
      <p>Some text..</p>
      <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod temp
or incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
    </div>
  </div>
</div>
```

```
<div class="card">
  <h2>TITLE HEADING</h2>
  <h5>Title description, Sep 2, 2017</h5>
  <div class="fakeimg" style="height:200px;">Image</div>
  <p>Some text..</p>
  <p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipiscing elit, sed do eiusmod te
mpor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
</div>
</div>
<div class="rightcolumn">
  <div class="card">
    <h2>About Me</h2>
    <div class="fakeimg" style="height:100px;">Image</div>
    <p>Some text about me in culpa qui officia deserunt mollit anim..</p>
  </div>
  <div class="card">
    <h3>Popular Post</h3>
    <div class="fakeimg"><p>Image</p></div>
    <div class="fakeimg"><p>Image</p></div>
    <div class="fakeimg"><p>Image</p></div>
  </div>
  <div class="card">
    <h3>Follow Me</h3>
    <p>Some text..</p>
  </div>
</div>
</div>
```



```
<div class="footer">  
  <h2>Footer</h2>  
</div>  
  
</body>  
</html>
```

```
* {  
  box-sizing: border-box;  
}
```

Maksudnya agar semua padding dan border dimasukkan sebagai tinggi dan lebar semua elemen yang digunakan. Ini untuk memudahkan pengaturan.

```
.header h1 {  
  font-size: 50px;  
}
```

Mengatur elemen h1 dalam kelas header

```
.topnav a {  
  float: left;  
  display: block;  
  color: #f2f2f2;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}
```

Mengatur elemen a dalam kelas topnav. Ini untuk mengatur link pada menu navigasi

```
.topnav a:hover {  
  background-color: #ddd;  
  color: black;  
}
```

Mengatur tampilan link menu navigasi ketika dilewati mouse

```
/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of each other instead of next to each other */
```

```
@media screen and (max-width: 800px) {  
  .leftcolumn, .rightcolumn {  
    width: 100%;  
    padding: 0;  
  }  
}
```

```
/* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top of each other instead of next to each other */
```

```
@media screen and (max-width: 400px) {  
  .topnav a {  
    float: none;  
    width: 100%;  
  }  
}
```

Pengaturan tampilan menu navigasi yang berbeda di perangkat yang berukuran kurang dari 800 piksel dan kurang dari 400 piksel

Tugas 3

Deskripsi Tugas

Membuat sebuah halaman homepage. Layouting halaman diatur dengan external style sheet. Layouting dapat menggunakan kode-kode dalam slide atau berkreasi sendiri. Halaman homepage harus terdiri minimal bagian header, menu navigasi, konten utama, dan footer. Pada bagian konten utama tempatkan data biografi beserta foto dari tugas sebelumnya.

Tujuan tugas

Melatih kemampuan layouting dengan css

Waktu

Seminggu

Pengumpulan

Kompres file html dan css jadi file zip beri nama layouting-nim-nama lalu kumpulkan ke kulino



THANKS

ANY QUESTIONS?