



Responsive Design

Silver - Chapter 3 - Topic 4

Selamat datang di **Chapter 3 Topik 4** online
course **Full Stack Web** dari Binar Academy!





Setelah kamu selesai dijelaskan tentang dasar-dasar membuat halaman web dan juga bahasa pemrograman Javascript, **Chapter 3 ini bakalan ngajak kamu buat belajar cara menjalankan kodingan dan cara berkolaborasi dengan developer lain dalam mengembangkan sebuah sistem.** Mulai dari penjelasan terkait terminal, IDE, GIT, membuat web layout dan responsive design.

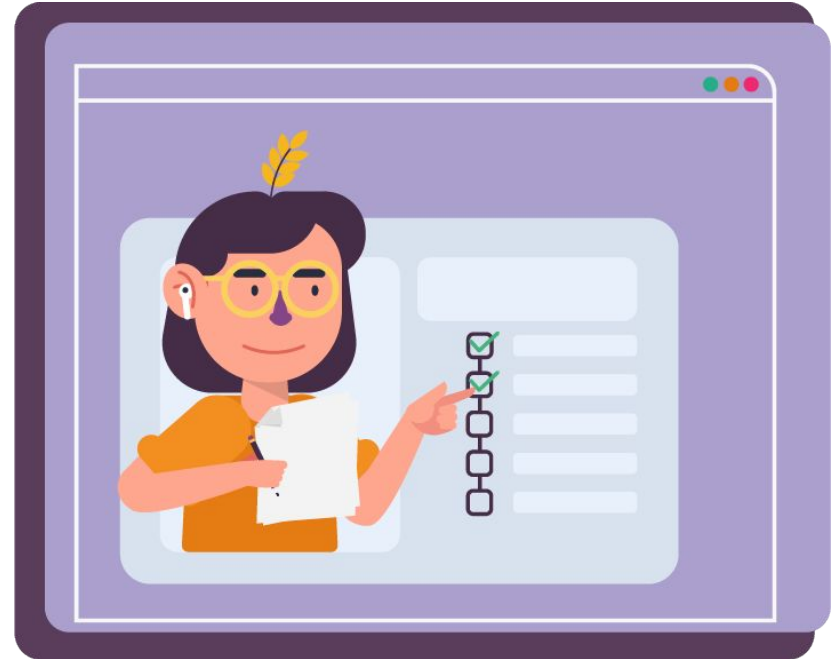
Pada topik pertama ini, kita bakal bahas tentang **responsive layout**. Cus langsung aja~





Detailnya, kita bakal bahas hal-hal berikut ini:

- Konsep Responsive Design
- Media Query
- Unit
- Teknik dalam Responsive Design



Kamu pasti pernah kan buka suatu website di komputer kamu, terus juga buka website yang sama di hp kamu?

Kalo kamu perhatiin, tampilannya hampir sama, lho! Wihh kok bisa gitu yaa~

Nah, kita bakal cari tahu jawabannya di topik tentang **Responsive Design** ini.

Letsgooo~

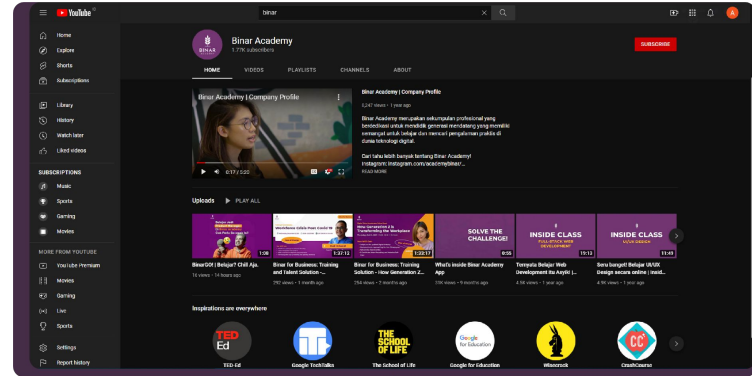


Biar lebih jelas, coba langsung kamu praktekin!

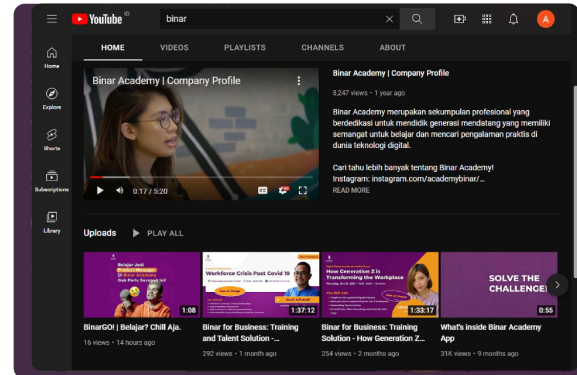
Buka laman [Youtube](#) dari browser kamu. Terus, coba kamu resize window menjadi lebih kecil.

Nah, kalo kamu perhatiin, desain web dari Youtube akan berubah dan menyesuaikan dengan ukuran window. Hal seperti inilah yang disebut **responsive design**.

Responsive design adalah **konsep tampilan web yang dapat merespon perubahan yang terjadi, dan secara otomatis, melakukan penyesuaian baik itu dari segi ruang maupun layout.**



Resize
browser





Sebelum masuk ke pembahasan yang lebih teknis, kamu perlu tau dulu tipe layar yang akan menampilkan sebuah website.

Yap~ Ada dua tipe layar secara umum, yaitu **Portrait** dan **Landscape**.

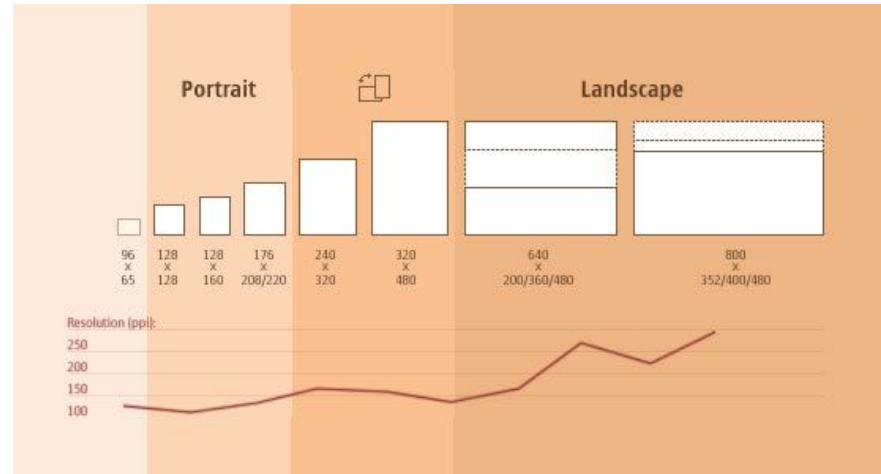




Selain tipe layar, kamu juga perlu tahu **display ratio**. Yaitu ukuran-ukuran dari sebuah layar. Display ratio ini menjadi acuan dalam pembagian layar berdasarkan resolusinya.

Display Ratio yang umum digunakan, yaitu:

- 4:3
- 3:2
- 16:9
- 16:10



Kalo kamu udah tahu tipe-tipe layar dan display ratio, sekarang saatnya kita masuk ke pembahasan yang lebih teknis, yaitu **media query**.





Apa itu media query?

Kalo ngomongin media query, maka yang kita maksud adalah CSS. Kita harus tahu seperti apa wujud media query di CSS.

Untuk itu, yuk kita pahami dulu maksud kode CSS disamping!

@media itu adalah sebuah operator yang memerlukan operan. Operannya adalah target dan kondisi. Pada contoh disamping, operannya adalah only screen dan max-width: 600px.

Jadi, maksud dari kode disamping adalah, **deklarasi CSS yang berada di dalam media query hanya akan diterapkan bila layarnya berukuran maksimal 600px.**

Terus, kalo layarnya lebih besar gimana? Kalo lebih dari 600px, maka deklarasi CSS di dalamnya tidak akan diterapkan pada browser.

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```



Mungkin kamu penasaran, apa maksud dari `max-width: 600px` dan kenapa kita memilih `600px`?

`Max-width: 600px` itu adalah yang biasa kita sebut **breakpoint**. Jadi, ketika memenuhi kondisi `max-width: 600px` (yang berarti lebar layar harus berada kurang dari `600px` agar memenuhi kondisi), maka deklarasi CSS di dalam media query akan diterapkan di browser.

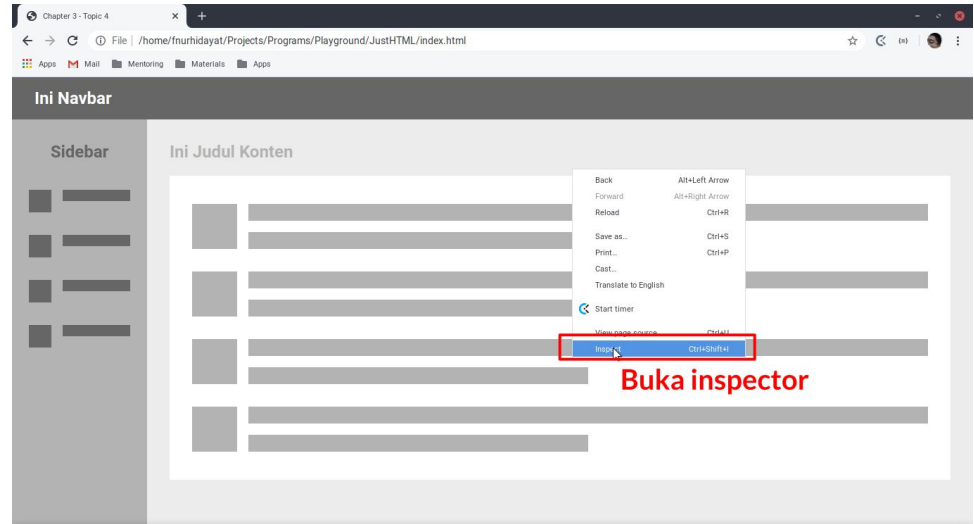
Dengan kata lain, **breakpoint itu adalah batas maksimal yang diterima.**

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```



Terus, gimana caranya nentuin breakpoint?

Pertama, buka browser kamu kayak gambar di samping.





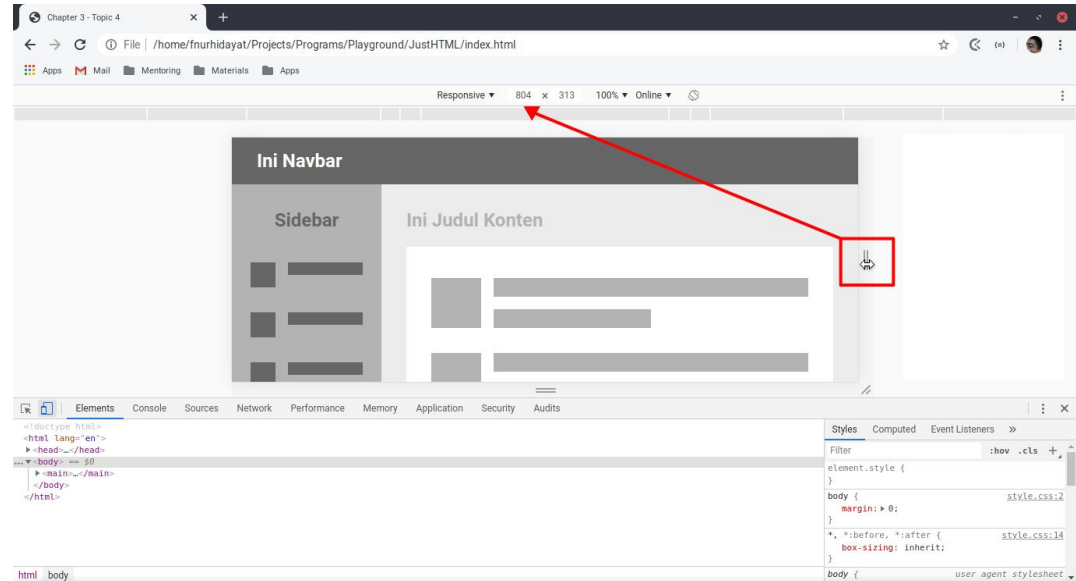
Lalu, buka device toolbar.





Abis itu, geser sampai sejauh mana tampilan website itu responsif (bisa dibesarkan dan dikecilkan). Geser sampai ukuran maksimum hingga tampilan 'hancur'.

Nah, inilah yang disebut **breakpoint**.





Berikut beberapa breakpoint yang umum dipakai~



```
/* Layar HP yang sangat kecil biasanya ukuran kurang dari 576px */  
  
/* Small devices (landscape phones, 576px and up) */  
@media (min-width: 576px) { ... }  
  
/* Medium devices (tablets, 768px and up) */  
@media (min-width: 768px) { ... }  
  
/* Large devices (desktops, 992px and up) */  
@media (min-width: 992px) { ... }  
  
/* Extra large devices (large desktops, 1200px and up) */  
@media (min-width: 1200px) { ... }
```

Kamu udah tahu apa itu media query, breakpoint, dan apa fungsinya.

Nah, hal lain yang nggak kalah penting buat kamu tahu adalah **Pemilihan Unit**.

Cus langsung kita bahas~





Unit dan pembagiannya~

Secara umum, unit dibagi dalam dua jenis, yaitu:

- **Absolute unit**, dimana nilainya akan tetap sama di semua ukuran layar.
Contoh absolute unit: **px**
- **Relative unit**, dimana nilainya akan menyesuaikan di setiap ukuran layar.
Contoh relative unit: **em, rem, %, vh, dan vw.**





Dari beberapa contoh relative unit sebelumnya, **em** dan **rem** paling membingungkan.

So, kita akan bahas secara khusus dua unit ini pada halaman selanjutnya~





Untuk ngebahas em, coba perhatikan contoh kode di samping ya!

Em adalah sebuah unit yang bergantung pada elemen pembungkusnya. Jadi, em ini ukurannya menyesuaikan dengan elemen pembungkusnya.

Kalo kamu perhatiin, nilai em dari kode di samping akan relatif terhadap 24px.

Jadi, 0.5 em akan sama dengan 12px. Sedangkan 2 em akan sama dengan 48px

```
<style>
  div {font-size: 24px}
  div > strong {font-size: .5em}
  div > small {font-size: 2em}
  div > strong, div > small {display: block}
</style>

<div>
  <strong>teks tebal</strong>
  <small>teks kecil</small>
</div>
```



Sekarang, kita bahas rem.

Kalo em tadi relatif dengan elemen pembungkus (induknya), maka rem ini relatif terhadap Root Element alias `<html>`.

Biar lebih jelas, coba perhatikan contoh kode di samping! Dari kode tersebut, terlibat bahwa 2 rem sama dengan 48px.

```
<style>
  html {
    font-size: 24px
  }

  div > strong {
    font-size: 2rem
  }
</style>

<div>
  <strong>teks tebal</strong>
</div>
```

Nah, kita udah belajar banyak nih tentang responsive design.

Terakhir, kita bakal belajar tentang **teknik yang digunakan dalam responsive design.**

Gaskuyy~





Ada dua teknik umum dalam responsive design.

- **Dekstop First**

Desktop first artinya, kita akan mengerjakan websitenya dari layar yang paling besar, terus layar sedang, dan terakhir layar kecil atau layar smartphone.

- **Mobile First**

Ini kebalikan dari dekstop first. Jadi, kita akan mengerjakan dari layar yang ukurannya paling kecil, lanjut ke layar sedang, dan terakhir ukuran besar atau layar dekstop.



Saatnya kita
Quiz!





1. Ketika dibuat kode seperti gambar disamping, maka `font-size: 2rem` akan menunjukkan font size dengan ukuran...

- A. 2 cm
- B. 20 px
- C. 48 px
- D. 26 cm

```
<style>
  html {
    font-size: 24px
  }

  div > strong {
    font-size: 2rem
  }
</style>

<div>
  <strong>teks tebal</strong>
</div>
```




1. Ketika dibuat kode seperti gambar disamping, maka `font-size: 2rem` akan menunjukkan font size dengan ukuran...

- A. 2 cm
- B. 20 px
- C. 48 px
- D. 26 cm

rem adalah unit relatif yang terikat terhadap Root Element alias `<html>`.

```
<style>
  html {
    font-size: 24px
  }

  div > strong {
    font-size: 2rem
  }
</style>

<div>
  <strong>teks tebal</strong>
</div>
```



2. Ketika dibuat kode seperti gambar di bawah, maka font-size: 2em akan menunjukkan font size dengan ukuran...

- A. 7 cm
- B. 12 px
- C. 24,5 px
- D. 31 cm

```
<style>
  div {font-size: 24px}
  div > strong {font-size: .5em}
  div > small {font-size: 2em}
  div > strong, div > small {display: block}
</style>

<div>
  <strong>teks tebal</strong>
  <small>teks kecil</small>
</div>
```



2. Ketika dibuat kode seperti gambar di bawah, maka font-size: .5em akan menunjukkan font size dengan ukuran...

- A. 7 cm
- B. 12 px
- C. 24,5 px
- D. 31 cm

em adalah unit relatif yang terikat terhadap elemen pembungkusnya.

```
<style>
  div {font-size: 24px}
  div > strong {font-size: .5em}
  div > small {font-size: 2em}
  div > strong, div > small {display: block}
</style>

<div>
  <strong>teks tebal</strong>
  <small>teks kecil</small>
</div>
```



3. Perhatikan contoh media query ini: @media (min-width: 576px).

Apa yang dimaksud min-width pada media query diatas?

- A. CSS akan di terapkan untuk layar yang lebarnya kurang dari 576px
- B. CSS akan di terapkan untuk layar yang lebarnya lebih dari 576px
- C. CSS akan di terapkan untuk layar yang lebarnya 576px



3. Perhatikan contoh media query ini : @media (min-width: 576px).

Apa yang dimaksud min-width pada media query diatas?

- A. CSS akan di terapkan untuk layar yang lebarnya kurang dari 576px
- B. CSS akan di terapkan untuk layar yang lebarnya lebih dari 576px
- C. CSS akan di terapkan untuk layar yang lebarnya 576px

Deklarasi CSS yang berada di dalam media query (min-width) ini hanya akan diterapkan bila layarnya berukuran minimal 576px.



4. Manakah di bawah ini yang bukan merupakan ratio yang umum digunakan...

- A. 4 : 3
- B. 3 : 1
- C. 16 : 9
- D. 16 : 10



4. Manakah di bawah ini yang bukan merupakan ratio yang umum digunakan...

- A. 4 : 3
- B. 3 : 1
- C. 16 : 9
- D. 16 : 10

Ratio yang umum digunakan adalah 4 : 3 , 3 : 2 , 16 : 9 , dan 16 : 10



5. Rita sedang mengembangkan front end dari sebuah aplikasi e-commerce. Ia akan mengerjakan layar dari ukuran yang paling kecil kemudian ke layar sedang dan berakhir dengan ukuran besar. Teknik yang digunakan Rita adalah...

- A. Mobile first
- B. Smartphone first
- C. Desktop first



5. Rita sedang mengembangkan front end dari sebuah aplikasi e-commerce. Ia akan mengerjakan layar dari ukuran yang paling kecil kemudian ke layar sedang dan berakhir dengan ukuran besar. Teknik yang digunakan Rita adalah...

- A. Mobile first
- B. Smartphone first
- C. Desktop first

Mobile First adalah teknik mengerjakan layar dari ukuran yang paling kecil, kemudian ke layar berukuran sedang, dan terakhir ukuran besar atau dekstop.



Nah, selesai sudah pembahasan kita di Chapter 3 Topic 4 ini.

Selanjutnya, kita bakal masuk ke chapter berikutnya ~



Terima Kasih!



Chapter ✓

completed