

TEKNIK *RESPONSIVE WEB DESIGN (RWD)* SERTA PENERAPANNYA DALAM RANCANG BANGUN LAYOUT WEB

Miftah Faroq Santoso

Program Studi Magister Ilmu Komputer
Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri
<http://www.nusamandiri.ac.id>
miftah_fs@yahoo.com



Ciptaan disebarluaskan di bawah Lisensi Creative Commons Atribusi-NonKomersial 4.0 Internasional.

Abstract— *Technology development make influences the way to access the internet, which generally uses Desktop and Notebook computers, giving to other alternatives, namely accessing the internet using Tablets and Smartphones. The problem is that the majority of web layout designs cannot adjust the layout size automatically for these devices, because the resolution size of each device is different. To overcome the existing problems, we need a responsive web design method and technique in order to produce web layouts that can adjust the size of screen resolution automatically. The method used by applying the concept of Responsive Web Design (RWD), so that web layouts can adapt to screen size. Responsive web design techniques can be done in several ways, including using manual techniques using CSS Media Query and implementing CSS Frameworks such as Bootstrap CSS. The test results showed that the implementation of responsive web layouts using Bootstrap was very effective and efficient, because web designers only needed the main library from Bootstrap, which was then combined by modifying according to the wishes of the user.*

Keywords: *responsive, web, css, framework*

Intisari—Perkembangan teknologi ikut mempengaruhi cara mengakses *internet* yang pada umumnya menggunakan komputer *Desktop* maupun *Notebook*, memunculkan alternatif lain yaitu mengakses internet dengan menggunakan *Tablet* dan *Smartphone*. Yang menjadi kendala adalah mayoritas desain *layout web* tidak dapat menyesuaikan ukuran *layout* secara otomatis terhadap perangkat tersebut, dikarenakan ukuran resolusi setiap perangkat berbeda. Untuk mengatasi permasalahan yang ada, diperlukan sebuah metode dan teknik desain web responsif agar dapat menghasilkan *layout web* yang dapat menyesuaikan ukuran resolusi layar secara otomatis. Metode yang digunakan dengan

menerapkan konsep *Responsive Web Design (RWD)*, agar *layout web* dapat beradaptasi terhadap ukuran layar. Teknik desain web responsif dapat dilakukan dengan beberapa cara, diantaranya dapat menggunakan teknik manual dengan menggunakan *Media Query CSS* dan pengimplementasian *Framework CSS* seperti *Bootstrap CSS*. Hasil pengujian didapatkan implementasi *layout web* responsif dengan menggunakan *Bootstrap* sangat efektif dan efisien, karena desainer *web* hanya memerlukan *library* utama dari *Bootstrap*, yang kemudian dipadukan dengan memodifikasi sesuai dengan keinginan pengguna.

Kata Kunci: *responsif, web, css, framework.*

PENDAHULUAN

Pada era modern sekarang ini perkembangan dalam mendesain sebuah website begitu cepat dan pesat mengalami perubahan. Mulai dari mendesain *layout* dengan hanya menggunakan tabel, hingga saat ini menggunakan *CSS (Cascading Style Sheet)* yang berfungsi sebagai penghias sekaligus pengatur gaya tampilan *layout* supaya dapat terlihat menarik dan elegan. Tahun 2007 perusahaan Apple memperkenalkan ponsel pintar (*smartphone*) iPhone, yang merupakan tonggak sejarah baru, mereka memberikan pengalaman baru bagi pengguna internet bagaimana berselancar didunia maya dengan cara yang berbeda, yaitu dengan menggunakan *smartphone* dengan ukuran layar yang tidak lebih dari 5 inci.

Mengakses halaman web melalui perangkat *mobile* akan berbeda secara tampilan, jika dibandingkan pada saat mengakses halaman web dengan menggunakan komputer *Desktop* dan *Notebook*, karena jenis perangkat tersebut memiliki resolusi ukuran yang berbeda. Ukuran resolusi pada layar *Desktop*, *Notebook* dan *Tablet*

memiliki ukuran yang lebih besar jika dibandingkan resolusi layar yang terdapat pada *Smartphone*. Yang menjadi kendala adalah mayoritas halaman web yang ada masih menerapkan konsep *fix-width design*, atau *layout web* dengan ukuran lebar tetap, dapat dikatakan *layout web* belum dapat menyesuaikan ukuran layar secara otomatis berdasarkan perangkat yang digunakan. Hal ini menyebabkan pengguna harus mengatur ukuran halaman web, agar tampil secara maksimal. Hal ini menuntut para desainer *web* untuk mendesain tampilan sebuah *web* agar dapat menyesuaikan di beberapa layar sekaligus atau lebih dikenal dengan *Responsive Web Design (RWD)*. Sebuah *website* harus dapat merespon alat pengaksesnya dari mulai layar lebar, hingga layar kecil. Tuntutan responsif pun tidak mengada-ada, mengingat saat ini telah beredar perangkat bergerak (*mobile devices*) seperti *smartphone*, *tablet*, *netbook*, dan produk-produk sejenis di seluruh penjuru dunia (Anugerah, 2013).

Adapun manfaat dan tujuan dari penggunaan teknik responsif pada saat membangun sebuah *layout web* adalah agar *website* tersebut dapat secara otomatis mendeteksi dan mengetahui, perangkat jenis apa yang sedang digunakan, serta dapat mengetahui segala jenis ukuran layar yang bervariasi. Maka akan didapatkan tampilan *layout web* yang secara otomatis dapat menyesuaikan ukuran layar, dengan tetap memperhatikan fitur-fitur dari sebuah halaman web, tanpa merubah isi dari web tersebut.

Dalam penelitian ini digunakan metode teknik *Responsive Web Design (RWD)*, agar *layout web* dapat menyesuaikan terhadap ukuran layar secara otomatis, diantaranya dapat menggunakan teknik manual dengan menggunakan *Media Query CSS* atau dapat menggunakan *Framework CSS* seperti *Bootstrap* dan *Foundation Framework* (Anugerah, 2013). Namun pada penelitian ini dibatasi hanya menggunakan *Framework CSS Bootstrap* versi 4 di dalam pembuatan perancangan *layout web* responsif. Serta sebagai *tools* pendukung digunakan *Sublime Text* sebagai editor dan *Mozilla, Chrome* yang digunakan sebagai *browser*, untuk mensimulasikan tampilan *website*.

Hasil yang diharapkan tentunya dengan menggunakan teknik *Responsive Web Design (RWD)*, kenyamanan dalam berselancar di dunia maya memberikan pengalaman yang berbeda bagi pengguna, ketika mengakses sebuah *web*, baik di komputer, *smartphone* dan *tablet*, namun akan tetap mendapatkan tampilan yang sama, meskipun diakses pada perangkat yang berbeda, yang memiliki ukuran layar yang bervariasi, dengan

tidak menghilangkan kualitas tampilan dan estetika dari desain sebuah *web*.

BAHAN DAN METODE

Pada bagian paper ini akan dijelaskan seputar metode penelitian dan perancangan desain.

1. Alat dan Bahan

Adapun sebagai bahan (*tools*) pendukung yang digunakan dalam perancangan desain *layout web* responsif ini, terbagi menjadi dua, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*).

Tabel 1. Spesifikasi *Software* dan *Hardware*

No	Alat (<i>tools</i>)	Keterangan
1	Perangkat Lunak (<i>Software</i>)	Editor
	a. Sublime Text	Pengujian dan simulasi
	b. Mozilla/ Chrome	
2	Perangkat Keras (<i>Hardware</i>)	Spesifikasi PC Standar
	a. 1 Unit PC/ Laptop	minimal: RAM 2G, HDD 80G
	b. Mouse, monitor, keyboard	Ujicoba pada ponsel Lenovo
	c. Ponsel	dengan OS Android

Sumber: (Santoso, 2019)

2. Metode

Metodologi penelitian ini dilakukan dengan beberapa tahapan, diantaranya:

A. Studi literatur

Pada tahap awal dilakukan studi terhadap beberapa artikel dan buku, yang berkaitan dengan perancangan desain web responsif, *Bootstrap CSS Framework*. Pada tahap ini juga dilakukan untuk menemukan fokus persoalan serta perumusan solusi yang akan ditawarkan pada penelitian ini.

B. Analisis permasalahan dan perumusan hipotesis.

Setelah didapatkan bahan literatur, pada tahap ini dilakukan analisis untuk merumuskan sebuah solusi yang akan ditawarkan.

C. Perancangan solusi

Pada tahap ini, akan diawali dengan perancangan sebuah desain yang akan dijadikan sebuah solusi dari permasalahan yang diangkat.

D. Implementasi solusi dan uji coba

Penerapan solusi dari perancangan yang telah dilakukan, akan diujikan dengan mengimplementasikan ke dalam kode pemrograman.

E. Penarikan kesimpulan

Hasil pengujian kemudian dianalisis untuk mendapatkan rumusan beberapa kesimpulan yang dapat diambil dari penelitian.

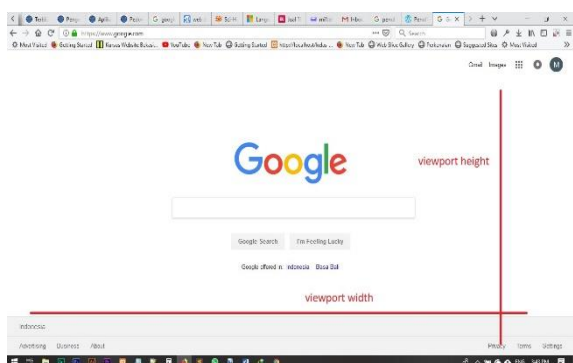
3. Perancangan Desain

Di dalam penelitian ini pembuatan layout web menggunakan teknik dan konsep Responsive Web Design. Dalam menggunakan teknik responsive web, caranya sangat bervariasi diantaranya dengan menggunakan teknik manual seperti *Media Query CSS* atau dengan menggunakan *CSS Framework* seperti Bootstrap dan Foundation.

A. Responsive Web Design (RWD)

Terminologi *Responsive Web Design (RWD)*, kali pertama dipekenalkan oleh Ethan Marcotte ditahun 2010, di dalam artikel webnya yang berjudul "A book apart" (M. SOEKARNO PUTRA, 2017). Responsif desain *web* memungkinkan pengguna berselancar di *internet* dengan menggunakan berbagai perangkat (*multi device*) seperti *smartphone*, *tablet* atau *laptop* selain komputer *desktop*. Karena desainya yang dapat menyesuaikan serta dapat beradaptasi dengan berbagai ukuran layar, maka dikenal juga dengan istilah *fluid design*, *elastic layout*, *rubber layout*, *liquid design*, *adaptive layout*, *cross-device design*, dan *flexible design* (Bryant & Jones, 2012).

Viewport merupakan salah satu fungsi paling penting yang terdapat pada desain responsif, *viewport* digunakan untuk menset ukuran layar sesuai dengan piranti yang sedang digunakan (Winarno & Zaki, 2015). Ukuran akan disesuaikan dengan ukuran layar, artinya pengungjung akan men-zoom konten untuk melihat lebih detail. Berikut ini tampilan *viewport* pada browser, akan terlihat seperti gambar 1 dibawah ini:



Sumber: (Santoso, 2019)

Gambar1. Width dan Height dari Viewport

Untuk sintaks penulisan meta tag name viewport, seperti berikut ini:

```
<meta name="viewport" content="width=device-width">
```

Terdapat tiga konsep yang sangat penting di dalam teknik penggunaan responsif web, diantaranya: a) Media Query dan Resolusi Layar b) Fluid Grid Layout c) Flexible Images dan Media (Baturay & Birtane, 2013).

B. Media Query CSS

Media Query sangat erat kaitannya dengan resolusi layar dan merupakan salah satu komponen utama dari desain responsif, karena memungkinkan desainer *web* menargetkan *style CSS* secara spesifik berdasarkan ukuran piranti. Jadi desainer web memiliki kontrol penuh terhadap *CSS* yang memungkinkan mendesain satu *layout website* untuk digunakan di beberapa platform sekaligus (Winarno & Zaki, 2015). Terdapat beberapa tipe query salah satu diantaranya adalah: *width* | *min-width* | *max-width*, perintah *query* ini memungkinkan menargetkan *style* pada *width* dari *viewport browser*, juga dapat menargetkan *width* minimum dan maksimum. Untuk sintaks penulisan *media query*, akan terlihat seperti berikut ini:

```
@media only screen and (min-width:768px) and (max-width:1023px){
  /* style yang akan diisikan*/
}
```

C. Bootstrap Twitter (CSS Framework)

Fluid Grid Layout atau *layout relatif* konsep *responsive web design* dengan menjaga ukuran lebar (*width*) agar tetap relatif terhadap *viewport*, hal ini berdampak pada *layout* yang memungkinkan untuk beradaptasi serta menyesuaikan terhadap ukuran layar (Zakir, 2016). Prinsip desain *fluid* adalah: 1) Jangan menggunakan *height* yang tetap (*fix*). 2) Jangan menggunakan *scrollbar horizontal*. 3) Pikirkan bagaimana agar media, seperti gambar dapat tampil sepenuhnya diberbagai perangkat. Teknik *fluid* ini merupakan teknik yang juga digunakan oleh *CSS Framework* Bootstrap dan Foundation untuk penerapannya di dalam desain *layout web* responsif.

Bootstrap Twitter merupakan kerangka *front-end* yang berfungsi untuk mengembangkan responsif *web layout* agar lebih cepat dan mudah digunakan (Zakir, 2016). Bootstrap kali pertama diperkenalkan oleh dua orang karyawan Twitter yaitu Mark Otto dan Jacob Thornton, hal ini juga yang membuat *framework CSS* Bootstrap lebih cepat dikenal. Dengan menggunakan konsep *grid system*, sehingga didalam penggunaannya mudah

untuk dipahami. Yang harus diketahui tentang penggunaan Bootstrap adalah:

1. *Mobile First Approach*

Fokus utama Bootstrap *layout* didesain untuk dapat mengikuti ukuran layar ponsel.

2. *Browser Support*

Dukungan dari beberapa browser terkenal seperti Internet Explorer, Safari, Mozilla, Chrome dan Opera.

3. *Knowledge to get started*

Syarat dalam mempelajari Bootstrap harus terlebih dahulu memahami HTML dan CSS

4. *Responsive Design*

Penggunaan Bootstrap bertujuan agar *layout* yang dibangun dapat menyesuaikan, serta beradaptasi terhadap tampilan layar *desktop*, *tablet* dan *ponsel*.

Beberapa paket yang sudah *built-in* dari Bootstrap, diantaranya adalah, sebagai berikut:

1. *Scaffolding*

Struktur Bootstrap menyediakan struktur dasar dengan *Grid System Model*

2. *CSS (Cascading Style Sheets)*

Menyediakan *CSS library* tersendiri, yang dibuat dengan konsep *media query*.

3. *Component*

Bootstrap menyediakan fitur-fitur yang interaktif seperti menu *dropdown*, *navigasi*, *alert*, *panel*, dll.

4. *Javascript Plugin*

Fitur yang berfungsi agar *layout* lebih interaktif, Bootstrap sudah menyediakan Bootstrap *Javascript Plugin*, yang dipadukan

dengan *Library jQuery*. Contoh dalam penggunaan *slider* untuk *slideshow* gambar.

Struktur *library file* yang terdapat didalam Bootstrap *framework*, diantaranya: a) folder *css* terdapat *file* yang berisi *file-file* *css*, seperti *file* *bootstrap.css*. b) folder *js* yang berisi *file-file* *javascript* seperti *file* *bootstrap.js*. Pada Bootstrap versi 4 ini, ada beberapa file dari luar Bootstrap yang harus disertakan agar Bootstrap dapat berjalan sebagaimana mestinya. File *jquery.js*, yang didapatkan dari *Libray jQuery* serta *Popper* yang memuat *library file* *popper.js*

Bootstrap mengadopsi teknik *grid system*, untuk mengatur ukuran tampilan layar. *Grid system* yang berfungsi untuk membuat pengaturan dari lebar masing-masing komponen, sehingga dapat dengan mudah dan fleksibel dalam pengaplikasiannya didalam *layout* web. Terdapat 12 *grid* yang dapat diatur sesuai dengan kebutuhan *layout*. Bootstrap memiliki *class-prefix* yang dapat digunakan, dan merespon ketika ukuran atau *breakpoint pixel* terpenuhi. *Class prefix* bootstrap terdapat empat *class*, diantaranya: 1) *col-lg-**, akan berjalan ketika *layout* berada di ukuran layar monitor besar. 2) *col-md**, akan berjalan ketika *layout* berada di ukuran layar sedang (*default*). 3) *col-sm**, *class prefix* ini akan berjalan pada layar ukuran tablet. 4) *col-xs*, *class extra small* akan berjalan pada ukuran layar ponsel.

Berikut ini tampilan *grid system* yang terdapat pada *css framework* Bootstrap dapat dilihat pada Tabel 2 dibawah ini:

Tabel 2. Spesifikasi Bootstrap *Grid System*

	Extra Small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra Large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of column)				
Nestable	Yes				
Column ordering	Yes				

(Santoso, 2019)

Struktur *templete* yang Bootstrap telah sediakan, sintaksnya akan terlihat sebagai berikut:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width,
shrink-to-fit=no"> 1
  <!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstr
ap/4.2.1/css/bootstrap.min.css"
integrity="sha384-
GJzZqFGwb1QTTN6wy59ff11BuGJpLSa9DkKMP0DgiMDm4iYM
j70gZWkYbI706tWS" crossorigin="anonymous"> 2
  <title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then
Bootstrap JS -->
```

```

<script
src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRv
H+8abtTE1Pi6jizo"
crossorigin="anonymous"></script> 3
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popp
er.js/1.14.6/umd/popper.min.js"
integrity="sha384-
wHAiFfR1MFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5W
HdZYU1hA6ZOb1gUt"
crossorigin="anonymous"></script> 4
<script
src="https://stackpath.bootstrapcdn.com/bootstra
p/4.2.1/js/bootstrap.min.js" integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I413gm9GU6Hc1og6Ls7i6U/
mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script> 5
</body>
</html>

```

Penjelasan pada skrip diatas dibaris dengan label angka 1 menjelaskan bahwa, meta tag *name viewport* memerintahkan kepada *browser* untuk menampilkan layar sesuai dengan perangkat, lebih tepatnya *browser* dapat beradaptasi secara otomatis, kemudian *shrink-to-fit=no*, yang berfungsi bahwa *layout web* tidak bisa diperbesar (*zoom*) dengan melakukan *pinch* atau ketuk pada layar, karena fitur dimatikan. Pada label dengan angka 2, menjelaskan lokasi *file bootstrap.css*, label dengan angka 3 sumber lokasi *file jquery.js*, label dengan angka 4 *popper.js* dan label dengan angka 5 merupakan *bootstrap.js* plugin. *File jquery.js*, *popper.js* dan *bootstrap.js* merupakan *file* yang berfungsi menjalankan fitur-fitur Bootstrap seperti *navigasi*, *slider*, *dropdown* agar *layout* lebih terlihat interaktif.

D. Desain Model

Desain mengacu kepada struktur standar sebuah *website* yang terdiri atas beberapa bagian, seperti *header*, *navigasi*, *content*, *sidebar* dan *footer*.

1. Research/ Scoping

Penelitian ini dillakukan dengan melakukan pengamatan pada *layout web* yang belum menerapkan teknik *layout* responsif. Perancangan desain responsif ini bertujuan agar *layout* tradisional dapat mengakomodir dan menyesuaikan disegala perangkat baik ponsel (*Hidayat, Utomo, & Djohan, n.d.*), *tablet* dan *desktop* dengan menggunakan *Framework CSS* Bootstrap versi 4 (empat).

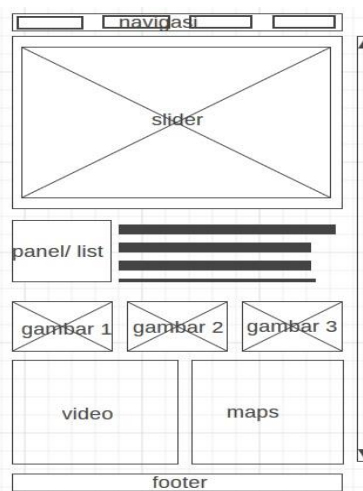
2. Wireframing

Pada tahapan *wireframing*, mulai mendefinisikan serta melakukan perancangan terhadap pengaturan tata letak *layout* terhadap lebar layar (*Hidayat et al., n.d.*). Secara umum *layout* akan didesain mengikuti ukuran layar sedang (*default*) yaitu *desktop*, kemudian *tablet*

dan *ponsel*. Selanjutnya menjelaskan alur pemrograman dibelakang layar bagaimana *CSS* bekerja membuat *layout* dapat menyesuaikan ukuran layar.

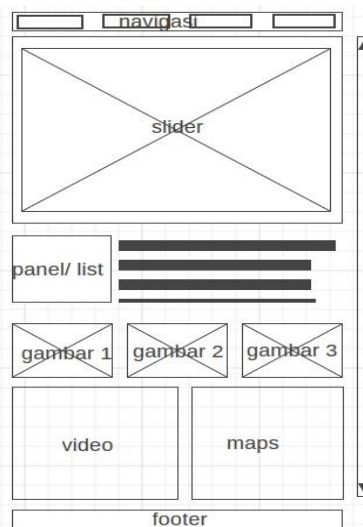
Adapun yang perlu diperhatikan disaat melakukan *wireframing*, diantaranya: a) Menentukan *grid* untuk setiap *layout*. b) Menyiapkan *file* yang berkaitan dengan Bootstrap. c) Perancangan desain *layout* untuk setiap perangkat d) Pengujian *layout*.

Berikut rancangan tampilan pada *desktop* dengan melakukan teknik *wireframing* terlebih dahulu, akan terlihat pada gambar dibawah 3 ini:



Sumber: (Santoso, 2019)
Gambar 3. Rancangan tampilan Desktop

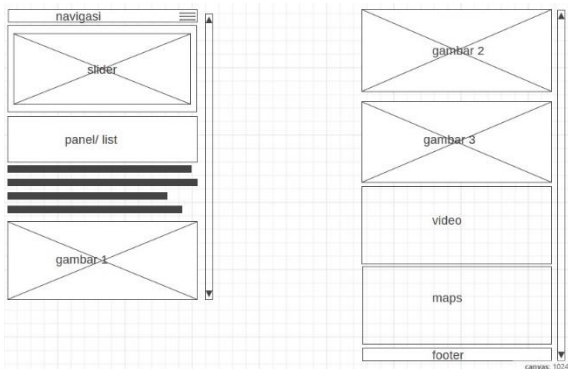
Desain *layout* tersebut yang akan tampil pada ukuran layar sedang (*default*) yang dapat menyesuaikan ukuran perangkat seperti *tablet* dan *ponsel*, untuk tampilan *tablet* akan terlihat pada gambar 4 dibawah ini:



Sumber: (Santoso, 2019)
Gambar 4. Rancangan tampilan Tablet

Disini hasil tampilan *layout* jika dibuka pada piranti seperti *tablet*, maka menu akan menyusut digantikan dengan simbol 3 bar, atau lebih dikenal dengan sebutan *hamburger icon*.

Hal ini akan terlihat sangat berbeda pada saat *layout* diakses pada layar dengan ukuran lebih kecil seperti ponsel. *Layout* akan ditampilkan secara keseluruhan menjadi satu kolom, jika *layout* semula terdapat dua kolom, disaat mode ponsel *layout* akan disatukan hanya satu kolom. Untuk tampilannya akan terlihat seperti pada gambar 5 dibawah ini:



Sumber: (Santoso, 2019)
Gambar 5. Rancangan tampilan Ponsel

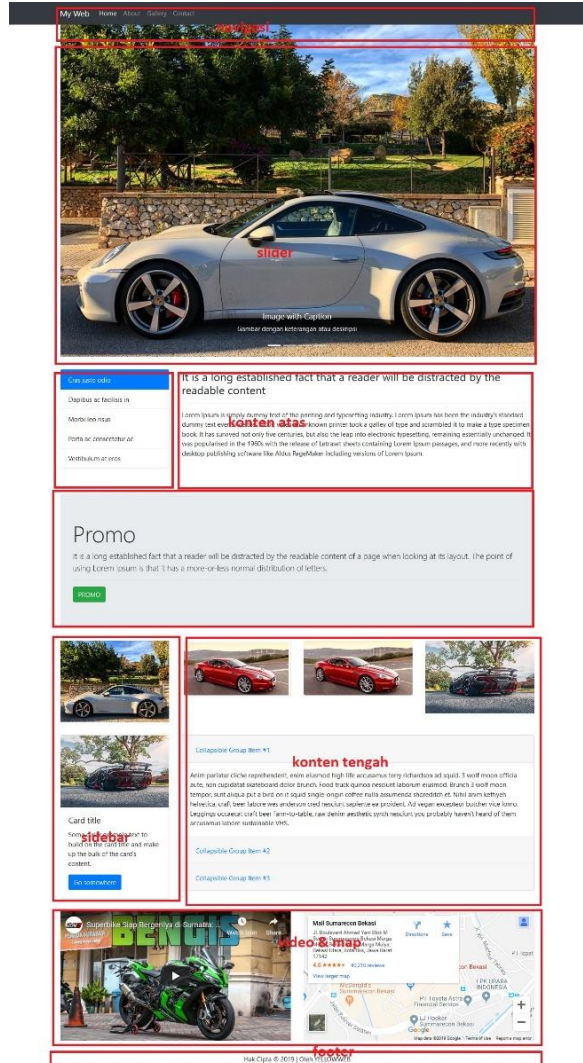
HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan seputar implementasi model dan pengujian.

A. Implementasi Model

Hasil dari perancangan desain *layout web* responsif dapat dilihat secara online di <https://www.kursusbuatweb.com/rwd> dan dapat diakses pada perangkat *desktop*, *tablet* dan ponsel. Terdapat bagian navigasi, bagian konten terdiri dari *slider*, beberapa baris gambar, *panel* pada sisi kiri dan google maps, serta video pada bagian konten bawah, sebelum *footer*. *Layout* secara otomatis beradaptasi terhadap layar pengguna, karena menggunakan *CSS Framework Bootstrap*. Hal ini tentunya membuat pengguna merasa nyaman, disini lain *layout Bootstrap* sangat *user friendly*, warna tidak terlalu kontras dan tidak mencolok. *Grid system* yang dimiliki oleh *Bootstrap* membuat blok-blok *div* yang digunakan dalam membangun *layout*, akan terus beradaptasi sesuai dengan ukuran yakni 12 *grid*, secara keseluruhan dan *grid* tersebut dapat berjalan aktif pada saat *class prefix Bootstrap* berada pada *breakpoint* yang sudah ditetapkan.

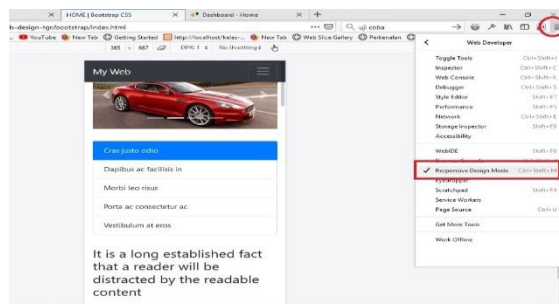
Implementasi *layout* yang menggunakan *Bootstrap Framework* secara menyeluruh akan terlihat seperti pada gambar 6 berikut:



Sumber: (Santoso, 2019)
Gambar 6. *Layout Web* dengan *Bootstrap CSS*

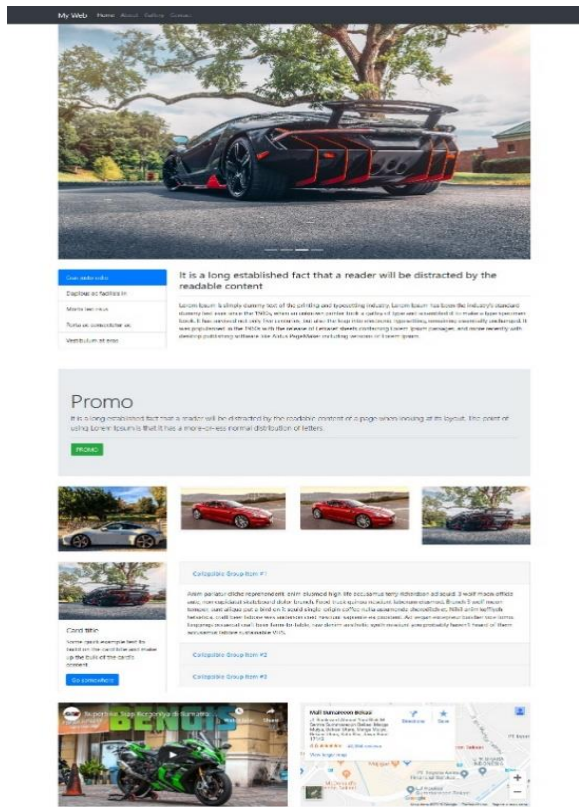
B. Pengujian

Pada bagian pengujian dipaper ini akan diperlihatkan hasil implementasi berupa tampilan *layout* yang mengimplementasikan *Bootstrap CSS Framework*. Untuk simulasi dan ujicoba digunakan *tools* pendukung seperti perambahan (*browser*) Mozilla dan Chrome. Simulasi pada browser akan terlihat seperti pada gambar 7 dibawah ini:



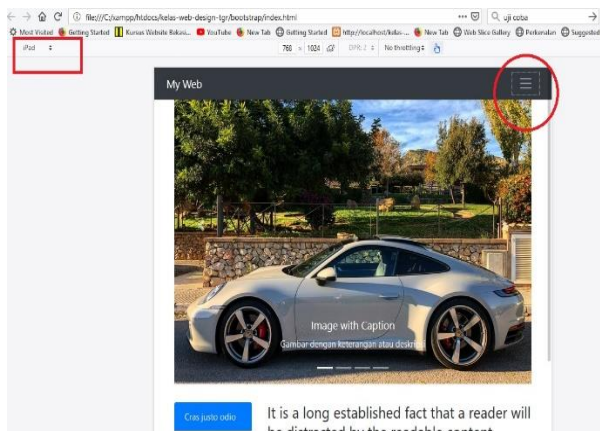
Sumber: (Santoso, 2019)
Gambar 7. Fitur *Responsive Design Mode* pada Mozilla

Untuk hasil ujicoba pada perangkat *desktop* atau *laptop* akan terlihat pada gambar 8 dibawah ini:



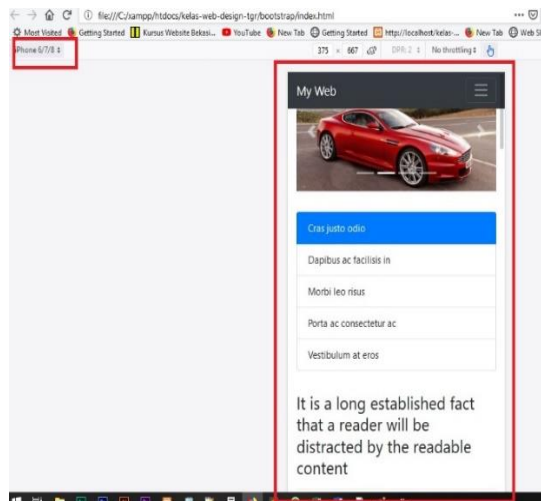
Sumber: (Santoso, 2019)
Gambar 8. Tampilan pada layar *Desktop* atau *Laptop*

Tampilan pada *tablet*, pada bagian (*section*) menu akan menyusut dan digantikan dengan ikon *humberger icon* tiga baris *horizontal*. Akan terlihat pada gambar dibawah 9 ini:



Sumber: (Santoso, 2019)
Gambar 9. Tampilan pada layar *Tablet*

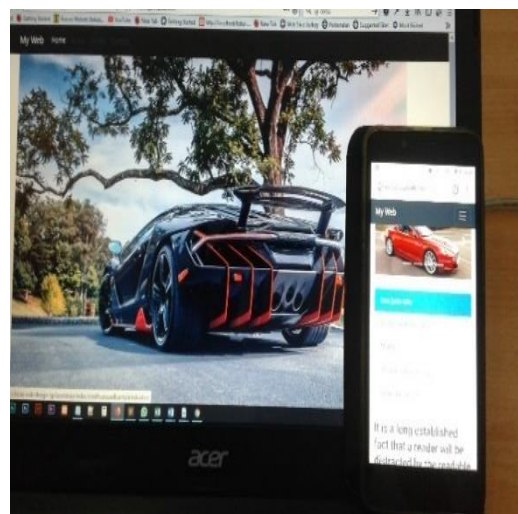
Sedangkan tampilan yang akan terlihat pada ponsel, akan terlihat pada gambar 10 dibawah ini:



Sumber: (Santoso, 2019)
Gambar 10. Tampilan pada layar Ponsel

Pada gambar tersebut diatas terlihat bahwa *browser* Mozilla sedang menjalankan fitur *responsive mode*. Dengan adanya fitur tambahan tersebut, dapat menampilkan secara langsung saat dilakukan simulasi terhadap tampilan web responsif di beberapa perangkat sekaligus.

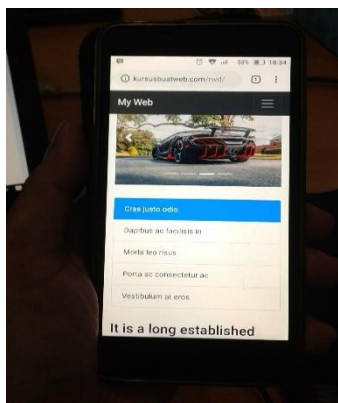
Sedangkan tampilan yang akan terlihat pada ponsel, akan terlihat pada gambar dibawah 11 ini:



Sumber: (Santoso, 2019)
Gambar 11. Perbandingan Tampilan pada layar Ponsel dan *Desktop*

Terlihat perbandingan yang cukup signifikan antara *layout* tampilan *desktop* dan ponsel, namun secara estetika, *layout* akan tetap terjaga, serta dapat beradaptasi terhadap ukuran perangkat secara otomatis.

Hasil ujicoba dilakukan secara langsung dengan menggunakan perangkat ponsel Lenovo dengan OS Android, akan terlihat pada gambar 12 dibawah ini:



Sumber: (Santoso, 2019)

Gambar 12. Tampilan pada layar Ponsel#2

Dengan dilakukannya ujicoba dapat dilihat bahwa dengan menggunakan Bootstrap CSS Framework hasilnya *layout* dapat berjalan dan bekerja secara otomatis menyesuaikan ukuran layar sesuai dengan perangkat yang digunakan. Hal ini dipengaruhi oleh *media query* dan *viewport*. Dalam implementasinya Bootstrap menggunakan *Grid System*, yang terdiri dari 12 kolom. Jika *layout* sedang dibuka pada layar ukuran ponsel, maka *layout* akan ditampilkan 1 kolom. Perlu diketahui pada saat membuat *layout web* responsif, *scroll horizontal* sangat dihindari, karena akan mengganggu pengguna disaat melakukan akses pada *web* tersebut dan meninggalkan kesan serta pengalaman (*user experience*) yang tidak menyenangkan. Halaman web hanya menerapkan *scroll vertical* pada bar sisi sebelah kanan.

Beberapa keuntungan pada saat menerapkan *layout web* responsif, diantaranya: 1) Dapat diakses secara mudah oleh pengguna, karena *layout* dapat menyesuaikan di beberapa perangkat (*multi device*) sekaligus. 2) Dapat menampilkan dan memunculkan bagian tertentu secara berbeda pada perangkat. 3) Tidak memerlukan *zoom*, karena teks dapat terlihat dengan jelas. 4) Mudah dalam bernavigasi. 5) Efisien dan efektif, karena pengembang tidak harus membuat *web* dibanyak perangkat secara terpisah.

Adapun jika dilihat dari sisi kerugian dalam penerapan *layout* responsif ini, diantaranya: 1) *Screen reader* yang digunakan untuk menterjemahkan tulisan pada *web*, akan mengalami sedikit kendala. 2) Tidak semua *browser support* dengan CSS 3 seperti Internet Explorer, namun cara ini dapat disiasati dengan menggunakan JavaScript. 3) Dilihat dari sisi *performance*, dengan menggunakan fitur *layout* responsif, gambar dengan ukuran yang besar akan

sedikit memperlambat pada saat dilakukan proses *render* halaman.

KESIMPULAN

Hasil implementasi merancang *layout web* responsif selain dengan menggunakan cara manual seperti penggunaan *media query* pada CSS, hal ini dapat dilakukan dengan menggunakan CSS Framework seperti Bootstrap. Dengan teknik *grid system* yang menerapkan 12 kolom pada ukuran layar, hal ini mempengaruhi cara *layout* menyajikan tampilan sesuai dengan ukuran perangkat, karena *layout* secara otomatis beradaptasi terhadap ukuran layar. Tanpa mengesampingkan fungsi dan tetap mengedepankan tampilan serta fitur terlihat sangat elegan dan intuitif.

REFERENSI

- Anugerah, S. (2013). Pemodelan Responsive Web Menggunakan Foundation Framework Dalam Pengembangan Perangkat Lunak, 2013(semnasIF), 230–236.
- Baturay, M. H., & Birtane, M. (2013). Responsive Web Design: A New Type of Design for Web-based Instructional Content. *Procedia - Social and Behavioral Sciences*, 106, 2275–2279. <https://doi.org/10.1016/j.sbspro.2013.12.259>
- Bryant, J., & Jones, M. (2012). *Responsive Web Design. Pro HTML5 Performance*. https://doi.org/10.1007/978-1-4302-4525-4_4
- Hidayat, A., Utomo, V. G., & Djohan, H. A. (n.d.). PENERAPAN RESPONSIVE WEB DESIGN DALAM PERANCANGAN SISTEM MODUL ONLINE ADAPTIF, 44–48.
- M. SOEKARNO PUTRA. (2017). Perancangan Desain Website Digital Library Universitas Bina Darma Dengan Menerapkan Responsive Web Design. *Jurnal INFORMANIKA*, 3(1), 29–33.
- Winarno, E., & Zaki, A. (2015). *Desain Web Responsif dengan HTML5 dan CSS3*. PT Elex Media Komputindo.
- Zakir, A. (2016). Rancang Bangun Responsive Web Layout Dengan Menggunakan Bootstrap Framework. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 1(1), 7–10. <https://doi.org/10.30743/infotekjar.v1i1.31>