MODUL TEKNIK *LAYOUT CSS*

CHAPTER 5

Understanding Grids

Pada pembahasan ini, kita akan mempelajari mengenai teknik *layout* menggunakan “grid”. Sebuah *grid* digunakan untuk mengatur desain elemen dan konten, karena semua *alignment*, *positioning*, *floating* dan *clearfix* adalah bagian dari *grid* *system*. Dengan *web* yang *responsive*, kita dapat mengaplikasikannya pada banyak *device*.

Ketika kita hendak membuat sebuah *layout* *grid* *system* kita akan membutuhkan baris dan kolom. Baris kita gunakan untuk sebuah konten, dan konten tersebut kita posisikan sesuai dengan *grid* yang sudah dibuat. Sedangkan bagian kolom sering digunakan untuk mengatur *margin* dari setiap konten.

Hal-hal yang harus dipersiapkan untuk menggunakan teknik *grid* adalah menentukan jumlah kolom dan margin dari setiap kolom tersebut. Pada pembahasan ini kita akan menggunakan 12 kolom. Sedangkah untuk margin adalah 20 *pixel*.

Creating Fluid Grid Columns

Pertama kita akan membuat bagian kolom, mengatur panjang dan lebar serta jumlah maksimum kolom yang akan digunakan. Disini kita akan membuat dua belas kolom dengan ukuran 65pixel. Penjelasan lebih lanjut terdapat pada langkah-langkah berikut ini.

1. Buatlah dokumen CSS dengan nama “grid.css”, kemudian tambahkan kode berikut ini didalamnya.

/\* --- Global --- \*/

img{

width: 100%;

}

/\* --- Media Queries --- \*/

@media (min-width: 768px){

/\* - Columns are 65px wide, with 20px gutters -\*/

/\* - Clearfix - \*/

.group:after{

content: "";

display: table;

clear: both;

}

}

1. Buatlah link untuk file “grid.css”, dan hapus link untuk “modernizr.js”



1. Masih pada dokumen HTML, hapus class “group” pada *div* “main-header” dan kembalikan logo diluar elemen list serta hilangkan *attribute* “data-icon” pada tiap elemen list.



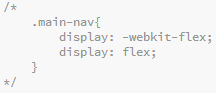
1. Pada dokumen CSS, *uncomment selector* “main-header” dan *comment property* “overflow” didalamnya.

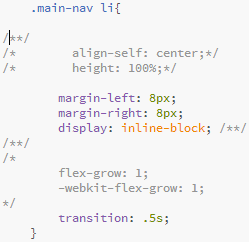


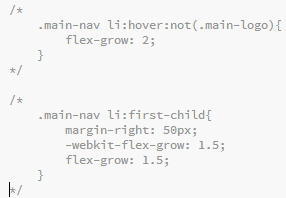
1. Kemudian, *uncomment selector* “main-nav li”.

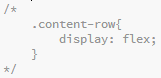


1. *Comment* seluruh *rule* yang berhubungan dengan teknik *flexbox*. Pada *selector* “main-nav li” *uncomment property* “display”.

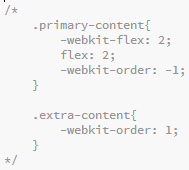




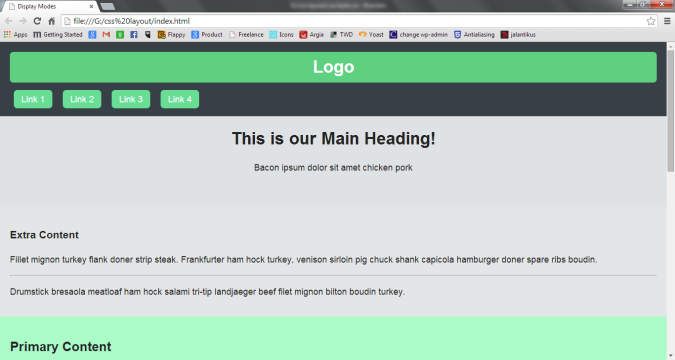








1. Setelah itu lakukan *refresh* pada *web browser*.



1. Seperti pada *comment* setelah *media-query* diatas, kita akan membuat 65px lebar kolom. Untuk membuatnya menjadi *fluid* kita harus mengubahnya menjadi persen. Caranya adalah, 65 dibagi 1000 sehingga menghasilkan 0.065. Kita kalikan dengan 100 untuk membuatnya menjadi persen, sehingga menjadi 6.5%. Untuk menggunakan *grid* ini, tambahkan kode berikut pada dokumen “grid.css”.

.grid-1{

width: 6.5%;

}

1. Untuk membuat presentase grid kedua kita gunakan perhitungan 65 ditambah 65 hasilnya sama dengan 130 kemudian 130 ditambah dengan 20, 20 adalah jarak antar grid. Sehingga hasilnya 150, 150 bagi dengan 1000, maka hasilnya 0.15 atau 15%. Jadi untuk dua kolom lebarnya 15%. Untuk menggunakan *grid* ini, tambahkan kode berikut pada dokumen “grid.css”.

.grid-2{

width: 15%;

}

1. Sedangkan untuk tiga kolom perhitungannya dimulai dari jumlah dua kolom yaitu 150 ditambah dengan 65 ditambah jarak antar grid yaitu 20 maka hasilnya 235 dibagi dengan 1000. Sehingga lebar tiga kolom adalah 0.235 atau 23.5%. Untuk menggunakan *grid* ini, tambahkan kode berikut pada dokumen “grid.css”.

.grid-3{

width: 23.5%;

}

1. Selanjutnya untuk grid empat kolom sampai dengan duabelas kolom gunakan perhitungan yang sama sehingga dihasilkan kode *grid* dibawah ini.

.grid-4 {

width: 32%;

}

.grid-5 {

width: 40.5%;

}

.grid-6 {

width: 49%;

}

.grid-7 {

width: 57.5%;

}

.grid-8 {

width: 66%;

}

.grid-9 {

width: 74.5%;

}

.grid-10 {

width: 83%;

}

.grid-11 {

width: 91.5%;

}

.grid-12 {

width: 100%;

}

Creating Grid Container

Pada materi sebelumnya kita telah belajar bagaimana menghitung untuk mengubah *grid* menjadi persentase tertentu sehingga menjadi *fluid*. Pada pembahasan kali ini kita akan belajar bagaimana membuat *grid container*. Penjelasan lebih lanjut terdapat pada langkah-langkah berikut.

1. Pada dokumen “grid.css”, tambahkan kode berikut pada bagian *comment “*global” untuk membuat *grid container*.

/\* --- Global --- \*/

.grid-container{

padding-left: 10px;

padding-right: 10px;

margin-left: auto;

margin-right: auto;

}

1. Kemudian, pada *media-query* “768px” tambahkan kode CSS berikut. Sebagai catatan margin-left disini bernilai 2%, dihasilkan dari 20 dibagi 1000, kemudian dikalikan 100%.

.grid-container > [class^="grid-"]{

float: left;

min-height: 1px;

padding-left: 15px;

padding-right: 15px;

margin-left: 2%;

}

1. Agar tidak terdapat *margin* pada *grid* paling awal dibawah kode pada langkah diatas, tambahkan kode berikut ini.

.grid-container > [class^="grid-"]:first-child{

margin-left: 0;

}

1. Kemudian, untuk membuat *grid* terakhir berada di sebelah kanan, tambahkan kode berikut ini dibawah kode pada langkah diatas.

.grid-container > [class^="grid-"]:last-child{

float: right;

}

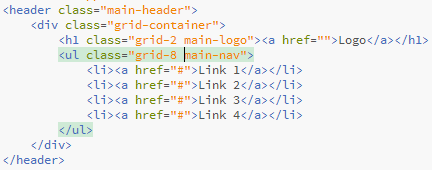
1. Agar *grid* tidak rusak oleh penggunaaan teknik *float*, tambahkan *selector* “grid-container:after” pada *media-query* “768px”.



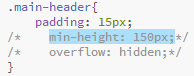
Testing Our Grid

Pada pembahasan kali ini, kita akan menguji *grid* yang telah kita buat pada dokumen HTML. Untuk itu, perhatikan dan ikutilah langkah-langkah berikut ini.

1. Pertama, kita akan menggunakan *grid* pada “header”.



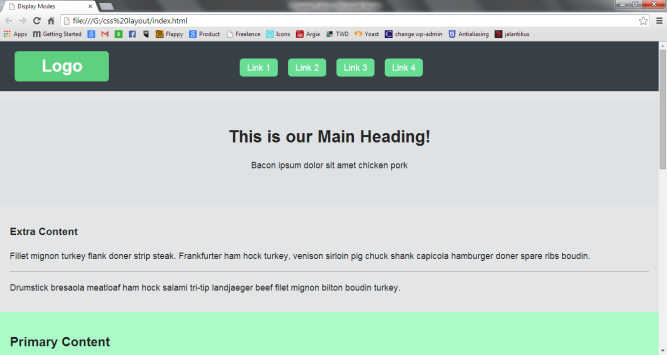
1. Kemudian, pada dokumen “style.css” *comment* *property* “min-height” pada *selector* “main-header”.



1. Masih pada dokumen yang sama, pada *media-query* “769px”, *comment property* “height” pada *selector* “main-header”.



1. Kemudian, lakukan *refresh* pada *web browser*.



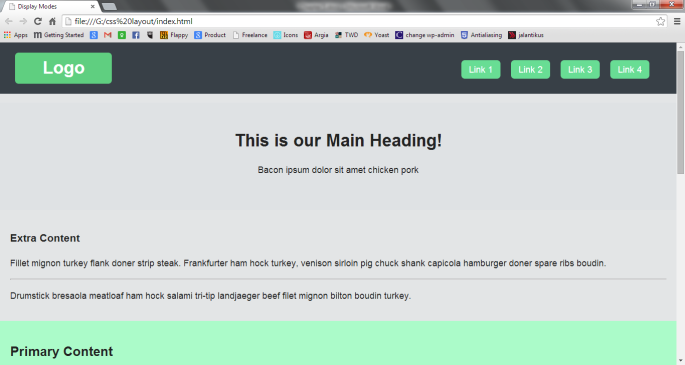
1. Kemudian pada dokumen “style.css”, untuk membuat elemen navigasi berada disebelah kanan, tambahkan kode berikut ini dibawah *selector* “main-header” pada *media-query* “769px”.

.main-nav {

text-align: right;

}

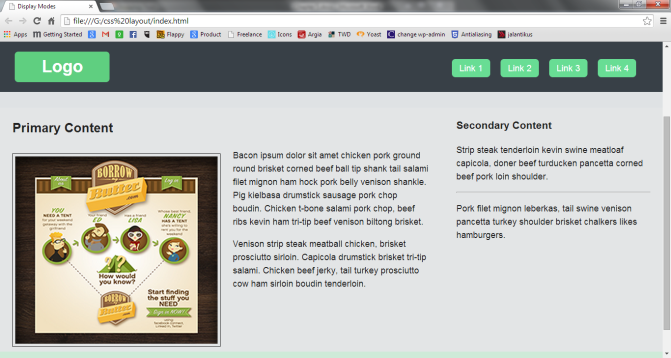
1. Setelah itu, lakukan *refresh* pada *web browser*.



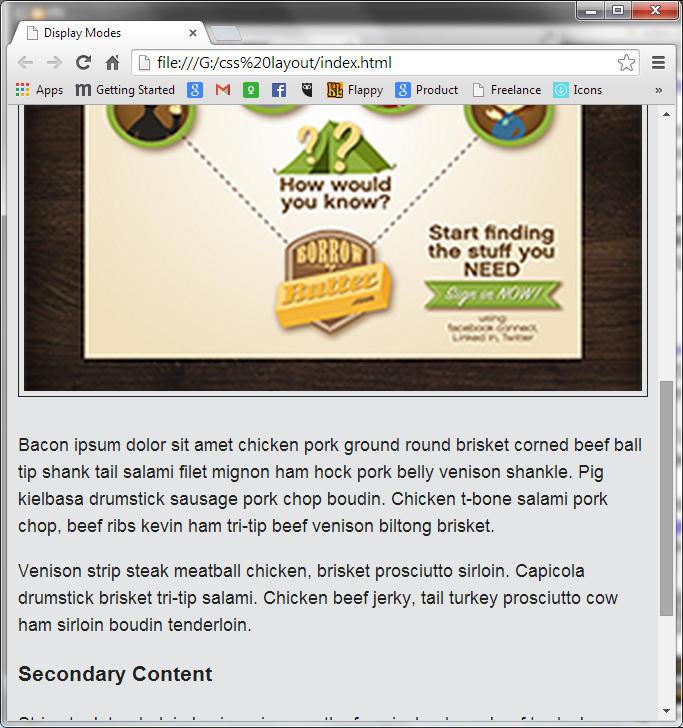
1. Lalu, pada dokumen HTML, ubahlah nilai dari *attribute class* menjadi seperti berikut ini.



1. Kemudian lakukan *refresh* pada *web browser*.



1. Kecilkan layar *browser* untuk mencoba *responsivitas* dokumen *web* .



1. Setelah itu, tambahkan kode berikut pada dokumen HTML, untuk membuat konten baru.

<div class="grid-container">

<div class="grid-4">

<h2>Content "Below the Fold"</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis facilisis, nisl id feugiat venenatis, massa nunc dignissim arcu, ut dictum arcu magna ac urna. Aliquam eget enim ac diam tincidunt tristique. In blandit ultricies tempor. Pellentesque habitant morbi tristique senectus et netus et malesuada.</p>

</div>

<div class="grid-4">

<h2>Content "Below the Fold"</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis facilisis, nisl id feugiat venenatis, massa nunc dignissim arcu, ut dictum arcu magna ac urna. Aliquam eget enim ac diam tincidunt tristique. In blandit ultricies tempor. Pellentesque habitant morbi tristique senectus et netus et malesuada.</p>

</div>

<div class="grid-4">

<h3>Extra Content</h3>

<p>Capicola doner turkey tail swine pork belly shank, t-bone prosciutto pastrami pork chop ground round.</p>

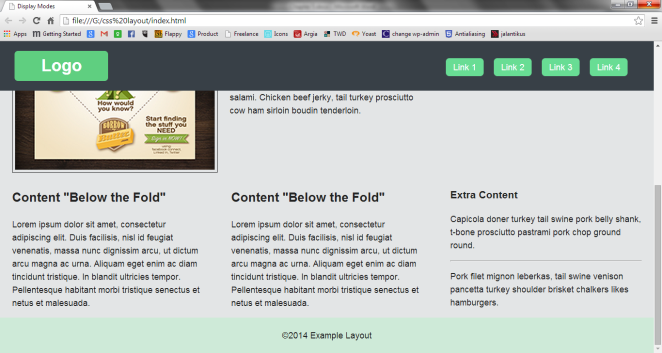
<hr>

<p>Pork filet mignon leberkas, tail swine venison pancetta turkey shoulder brisket chalkers likes hamburgers.</p>

</div>

</div>

1. Kemudian, lakukan *refresh* pada *web browser*.



1. Untuk membatasi lebar dari *container*, tambahkan kode berikut pada dokumen “grid.css”.

@media (min-width: 1200px){

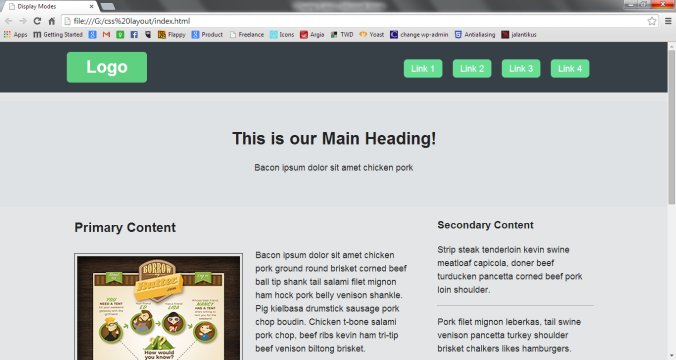
.grid-container{

max-width: 1100px;

}

}

1. Setelah itu, lakukan *refresh* pada *web browser*. Tampak bahwa elemen pada dokumen web menjadi lebih ke tengah.



1. Kemudian, untuk mengatur tampilan konten pada ukuran *mobile*, tambahkan kode berikut pada dokumen “grid.css” diatas *media-query* yang ada sebelumnya.

@media (min-width: 1px) and (max-width: 767px){

.grid-container > [class^="grid-"]{

padding-top: 5px;

padding-bottom: 5px;

}

}

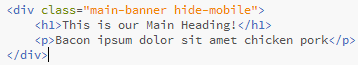
1. Tambahkan juga *selector* “hide-mobile” pada *media-query* pada langkah sebelumnya, untuk menyembunyikan elemen ketika layar browser diubah pada ukuran *mobile*.

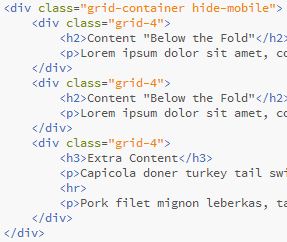
.hide-mobile{

display: none;

}

1. Pada dokumen HTML, letakkan *class* “hide-mobile” pada *div* “main-banner” dan “grid-container” yang kedua.





1. Lakukan *refresh* pada *web browser* kemudian kecilkan ukuran layarnya.

|  |  |
| --- | --- |
|  |  |
| Tampilan bagian atas, terlihat *main-banner* hilang | Tampilan bagian bawah, terlihat *grid-container* yang kedua juga hilang |

1. Untuk membuat tampilan dari dokumen menjadi lebih menarik. Tambahkan, ubahlah *property* “background-color” dan “font” menjadi seperti berikut ini.



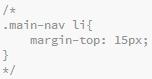
1. Kemudian pada dokumen “style.css”, dalam *media-query* “769px” *uncomment property* “height” di *selector* “main-header”.



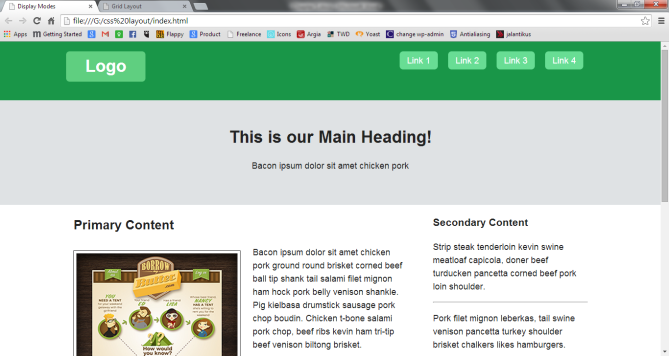
1. Lalu, pada *selector* “main-header” bagian *comment* “Layout Element Colors” ubahlah nilai dari *property* “background-color”.



1. *Comment selector* “main-nav li” berikut.



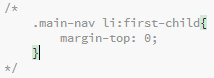
1. Lakukan *refresh* pada *web browser*.



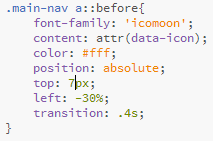
1. Setelah itu, *uncomment selector-selector* dibawah ini.



1. Kemudian, *comment selector* “main-nav li:first-child” pada media query “min-width:1px”



1. Setelah itu, ubahlah *property* “top” pada *selector* “main-nav a::before”.



1. Lakukan *refresh* pada *web browser*.

|  |  |
| --- | --- |
|  |  |
| Tampilan pada layar *browser* normal | Tampilan pada layar *browser* ketika dikecilkan |

1. Hapus *comment* yang ada pada seluruh dokumen HTML dan CSS yang telah anda buat. Hal ini dimaksudkan agar tidak banyak memori yang digunakan pada pembuatan halaman *web*.