

Pemrograman Web

Sebuah Pengantar

Modul pemrograman web berisi dasar-dasar penggunaan HTML, CSS, Javascript dan PHP dasar. Tujuan pembuatan modul ini adalah memberikan gambaran global tentang bagaimana memanfaatkan script-script yang ada untuk membuat web.

Modul ini bukan merupakan referensi lengkap, dan tidak dimaksudkan sebagai pengganti buku. Modul ini hanyalah alat bantu dalam penyampaian materi. Oleh sebab itu, dalam penggunaan modul ini, disarankan ada seorang instruktur/tutor yang dapat memberikan penjelasan secara lebih mendalam.

Penulis,



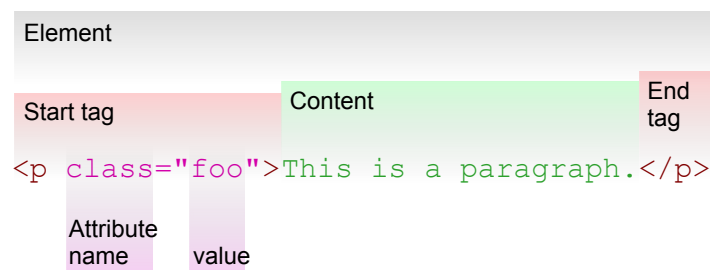
Go Frendi Gunawan

HTML

HTML (Hypertext Markup Language) adalah bahasa dasar dalam setiap halaman web. HTML terdiri atas tag-tag dan atribut. Pada dasarnya, seseorang dimungkinkan membuat halaman web statis hanya dengan menggunakan HTML. Untuk membuat sebuah file HTML, dapat digunakan program teks editor semisal notepad, notepad++, sublime text atau text-editor lain.

Tag & Atribut

Tag merupakan bagian terkecil dalam halaman HTML. Dalam sebuah tag, kita dapat mendefinisikan atribut (attribute) dan isi (content). Secara umum, tag diawali dengan awal-tag (start tag) dan akhir-tag (end tag). Berikut adalah contoh sebuah tag:



Tag p (seperti gambar di atas) berfungsi untuk membuat format paragraf. Atribut class digunakan sebagai penanda untuk mempermudah manipulasi menggunakan CSS atau javascript.

Selain tag p, juga terdapat tag-tag lain. Di antaranya ada tag-tag yang tidak berpasangan dan tidak memiliki content, semisal img dan br.

```

<br />
```

Tag img berfungsi untuk menampilkan gambar sesuai dengan atribut src nya, sedangkan tag br berfungsi untuk pindah baris.

Tag-tag umum

Dalam HTML ada banyak sekali tag yang masing-masing memiliki fungsi berbeda satu sama lain. Di sini akan dibahas beberapa tag yang sering dipakai dalam membuat halaman web.

HTML, head, title, meta, dan body

Tag HTML, head, title, meta dan body merupakan tag-tag yang membentuk kerangka dari sebuah halaman web. Walaupun keberadaannya tidak wajib, namun diperlukan untuk standarisasi dan SEO (Search Engine Optimization) sehingga halaman akan lebih mudah terindeks oleh search engine

semacam google. Berikut adalah contoh sebuah kerangka halaman web sederhana:

```
<!DOCTYPE HTML>
<HTML lang="en">
  <head>
    <title>Home</title>
    <link rel="icon" href="/favicon.ico" type="image/x-icon">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="Tutorial web gratis">
    <meta name="keywords" content="HTML,CSS,XML,JavaScript">
    <meta name="author" content="Go Frendi Gunawan">
    <meta charset="UTF-8">
  </head>
  <body>
    <!-- Letakkan isi halaman web di sini -->
  </body>
</HTML>
```

Tag doctype berfungsi untuk mendefinisikan tipe halaman. Jika tidak ada, maka browser secara default akan menganggap bahwa halaman tersebut memiliki tipe HTML. Atribut lang dalam tag HTML berfungsi untuk mendefinisikan bahasa.

Tag head berfungsi untuk mendefinisikan hal-hal yang perlu di-load oleh browser sebelum me-load halaman lain, antara lain judul halaman (isi dari tag title), icon halaman (menggunakan tag link), deskripsi, kata kunci, dan penulis halaman (masing-masing menggunakan tag meta, berfungsi untuk SEO). Tag meta dengan name viewport berfungsi agar halaman web terbuka dengan lebar penuh di device-device kecil seperti handphone atau tablet. Tag meta dengan name charset berfungsi untuk mendefinisikan karakter yang digunakan.

Isi halaman web umumnya diletakkan di tag body. Bagian yang berawalan <!-- dan berakhiran --> berfungsi untuk penanda komentar.

br

Tag br berfungsi untuk memindah baris. Berikut adalah contoh penggunaan tag br:

```
Ini adalah baris pertama.
Inipun adalah baris pertama.<br />
Ini baru merupakan baris kedua
```

p (paragraph)

Tag p (paragraph) merupakan tag yang berperan dalam membagi halaman web menjadi paragraf-paragraf. Penggunaan tag p adalah sebagai berikut:

```
<p>
  Ini adalah paragraf pertama, baris pertama<br />
  Ini adalah paragraf pertama, baris kedua
</p>
<p>
  Ini adalah paragraf kedua baris pertama<br />
  Ini adalah paragraf kedua baris kedua
</p>
```

div

Tag div merupakan tag yang berperan dalam membagi halaman web menjadi bagian-bagian terpisah. Content tag div bisa berupa tag-tag HTML lain. Tag ini biasanya berguna untuk styling (dikombinasikan dengan CSS).

```
<div>
  <p>Sebuah paragraf</p>
  <div>Sebuah div di dalam div</div>
</div>
<div>
  div yang berbeda
</div>
```

a (anchor)

Tag a (anchor) merupakan tag yang berperan sangat penting dalam sebuah website. Tag ini berfungsi untuk menciptakan link ke halaman website lain atau ke bagian lain dari website. Berikut adalah contoh penggunaan tag anchor:

```
<a href="http://www.google.com" target="blank">Menuju ke google</a>
<a href="mailto:gofrendiasgard@gmail.com">Kirim email</a>
<a href="#bawah">Pindah ke bawah</a>
<p>.... </p>
<a name="bawah"></a>
```

Atribut href pada tag a menentukan lokasi yang akan dituju jika link di-klik. Atribut target menentukan di mana lokasi tersebut akan terbuka. Jika target bernilai self, maka lokasi akan terbuka menggantikan halaman yang sedang terbuka. Jika target bernilai blank, maka lokasi akan terbuka di tab/window baru.

img (image)

Tag img (image) berfungsi untuk menampilkan gambar di halaman web. Contoh penggunaannya adalah sebagai berikut:

```

```

ul (unordered list)

Tag ul adalah tag yang berfungsi untuk menampilkan list dengan menggunakan bullet. Sebuah list terdiri atas beberapa list item (dibuat dengan menggunakan tag li). Sebuah list dapat bersarang (nested) pada list induk. Sifat ini menyebabkan tag ul sering digunakan untuk membuat menu halaman. Berikut adalah penggunaan tag ul:

```
<ul>
  <li>Nasi</li>
  <li>Susu</li>
  <li>Sayuran
    <ul>
      <li>Bayam</li>
      <li>Kangkung</li>
    </ul>
  </li>
  <li>Buah-buahan</li>
</ul>
```

ol (ordered list)

Tag ol adalah tag yang berfungsi untuk menampilkan list dengan menggunakan angka. Sebuah list terdiri dari beberapa list item, sama halnya dengan tag ul. Berikut adalah contoh penggunaan tag ol

```
<ol>
  <li>Nasi</li>
  <li>Susu</li>
  <li>Sayuran
    <ul>
      <li>Bayam</li>
      <li>Kangkung</li>
    </ul>
  </li>
  <li>Buah-buahan</li>
</ol>
```

table

Tag table merupakan tag yang sangat berguna untuk menampilkan data. Dalam sebuah tag table, terdapat 2 bagian, masing-masing thead untuk meletakkan judul tabel, dan tbody untuk meletakkan data. Dalam tag thead dan tbody, bisa terdapat beberapa tag tr untuk merepresentasikan baris. Jika tag tr terletak di dalam thead, umumnya setiap sel dalam baris diwakili oleh tag th, sedangkan jika tag tr terletak di dalam tbody, maka setiap sel dalam baris diwakili oleh tag td. Berikut adalah contoh penggunaan tag table:

```
<table>
  <thead>
    <tr>
      <th>Nama</th>
      <th>Alamat</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Anton</td>
      <td>Jalan Ananas no 1. Malang, Jawa Timur</td>
    </tr>
    <tr>
      <td>Budi</td>
      <td>Jalan Borobudur no 2. Tangerang, Jawa Barat</td>
    </tr>
    <tr>
      <td>Caca</td>
      <td>Jalan Cikampek no 3. Denpasar, Bali</td>
    </tr>
  </tbody>
</table>
```

Terkadang dibutuhkan untuk menggabungkan beberapa baris atau kolom menjadi satu. Dalam hal ini, maka dapat digunakan atribut rowspan (untuk menggabungkan baris) dan colspan (untuk menggabungkan kolom).

Atribut rowspan dan colspan harus diletakkan di tag th atau td. Berikut adalah contoh penggabungan baris dan kolom:

```
<table>
  <thead>
    <tr>
      <th rowspan="2">Nama</th>
      <th colspan="2">Lokasi Rumah</th>
    </tr>
    <tr>
      <th>Alamat</th>
      <th>Provinsi</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Anton</td>
      <td>Jalan Ananas no 1. Malang</td>
      <td>Jawa Timur</td>
    </tr>
    <tr>
      <td>Jalan Anoa no 5. Semarang</td>
      <td>Jawa Tengah</td>
    </tr>
    <tr>
      <td>Budi</td>
      <td colspan="2">Jalan Borobudur no 2. Tangerang, Jawa Barat</td>
    </tr>
    <tr>
      <td>Caca</td>
      <td colspan="2">Jalan Cikampek no 3. Denpasar, Bali</td>
    </tr>
  </tbody>
</table>
```

form, input, select, dan textarea

Form, input, select, dan textarea merupakan tag-tag yang berfungsi untuk menyediakan fasilitas bagi pengguna untuk memasukkan data. Data tersebut selanjutnya dapat diolah oleh server. Untuk mengolah data di server, dapat digunakan bahasa pemrograman server-side semisal PHP, java, python, dan ruby. Berikut adalah contoh sebuah form sederhana:

```
<form method="get" action="https://www.google.com/search">
  <input type="hidden" name="kode" value="abc01" />

  <label>Nama</label>
  <input type="text" name="nama" placeholder="Nama pengguna" value="Go
Frendi" /><br />

  <label>Password</label>
  <input type="password" name="password" placeholder="Password" />
  <br />

  <label>Hobi</label>
  <input type="checkbox" name="hobi[]" value="1" />Memancing<br />
  <input type="checkbox" name="hobi[]" value="2" />Membaca<br />
  <input type="checkbox" name="hobi[]" value="3" checked />Main<br />

  <label>Jenis kelamin</label>
  <input type="radio" name="jenis_kelamin" value="male">Male <br />
  <input type="radio" name="jenis_kelamin" value="female">Female<br />

  <label>Jabatan</label>
  <select name="jabatan">
    <option value="dir" selected>Direktur</option>
    <option value="man" selected>Manager</option>
    <option value="asman" selected>Asisten Manager</option>
  </select><br />

  <input type="submit" name="save" value="Simpan" />

</form>
```


style

Tag style berfungsi untuk meletakkan script CSS secara internal dalam halaman web. Tag ini seyogyanya diletakkan di bagian head dari sebuah dokumen HTML. Lebih lanjut mengenai CSS akan dibahas pada bagian lain. Berikut adalah contoh penggunaan tag style:

```
<style type="text/CSS">
  p{
    font-size : large;
    border-top: 1px solid gray;
  }
</style>
```

link

Selain berfungsi untuk meletakkan favicon, tag link juga berguna untuk menyertakan file CSS yang berasal dari file eksternal. Sebisa mungkin disarankan menggunakan CSS dari file eksternal, karena hal tersebut akan mempermudah proses caching pada browser. Tag link seyogyanya diletakkan di bagian head dari dokumen HTML. Berikut adalah contoh penggunaan tag link untuk memanggil file CSS eksternal:

```
<link rel="stylesheet" type="text/CSS" href="lokasi/file.CSS" />
```

script

Tag script berfungsi untuk meletakkan javascript dalam sebuah halaman web, baik secara internal maupun eksternal. Tag ini sebaiknya diletakkan di bagian akhir dari bagian body dalam dokumen HTML.

Berikut adalah contoh penggunaan tag script:

```
<script type="text/javascript" src="lokasi/file.js"></script>

<script type="text/javascript">
  var nama = prompt("Masukkan nama anda", "Farhat Lucu");
  alert("Halo, " + nama );
</script>
```

CSS

CSS (Cascading Stylesheet) merupakan bahasa untuk melakukan styling pada halaman web. Element CSS terdiri dari selector, diikuti kurung kurawal buka, pasangan-pasangan nilai yang dipisahkan titik dua dan diakhiri titik koma, serta kurung kurawal tutup. Berikut adalah contoh script CSS:

```
p{  
    color : red;  
    border-top: 1px solid black;  
}
```

p merupakan selector. Dalam hal ini, segala sesuatu yang terletak mulai dari kurung kurawal buka sampai kurung kurawal tutup akan mempengaruhi semua tag p yang ada dalam halaman. Bagian dalam kurung kurawal terdiri atas pasangan kunci dan nilai. Bagian color : red berfungsi untuk mengubah warna font pada semua tag p menjadi berwarna merah, sedangkan bagian border-top : 1px solid black; berfungsi untuk memberikan border setebal 1 pixel, berwarna hitam, dengan style solid (garis utuh) di bagian atas dari setiap tag p.

Selector

Ada berbagai macam selector yang dikenal dalam CSS. Secara umum semua nama tag dapat menjadi selector. Selain itu, terdapat pula tanda titik dan pagar yang masing-masing digunakan untuk menyeleksi tag berdasarkan atribut class dan id nya. Atribut id seharusnya bersifat unik (tidak ada tag yang memiliki id yang sama), sedangkan class boleh sama.

Selector	Contoh	Keterangan
Tag	<pre>p{ color: red; }</pre>	Jika digunakan tag, maka semua elemen HTML dengan tag tersebut akan terpengaruh.
#	<pre>#header{ color: red; } p#first{ color: blue; }</pre>	<p>Tanda # menyatakan id.</p> <p>Maka, jika digunakan #header, semua tag dengan id header akan terpengaruh, entah itu p, div, atau tag yang lain.</p> <p>Jika digunakan p#header, maka hanya tag p dengan id header saja yang terpengaruh.</p>

.	<pre> .active{ color: red; } li.active{ font-size: large; } </pre>	<p>Tanda . menyatakan class.</p> <p>Maka, jika digunakan .active, semua tag dengan class active akan terpengaruh, entah itu p, div, atau tag yang lain.</p> <p>Jika digunakan li.active, maka hanya tag li dengan class active saja yang terpengaruh.</p>
Bersarang	<pre> div#list li{ font-style: italic; } </pre>	<p>Pemisah spasi menandakan anak dari tag sebelumnya. Sehingga div#list li berarti semua tag li yang terletak dalam div dengan id list.</p>
a:hover	<pre> a:hover{ color: green; } </pre>	<p>Pseudo selector a:hover menandakan tag a yang dilalui kursor.</p>
a:visited	<pre> a:visited{ color: blue; } </pre>	<p>Pseudo selector a:visited menandakan tag a yang pernah di klik (dikunjungi)</p>

Masih banyak jenis selector yang lain, semisal a>img untuk memilih semua img yang merupakan anak langsung dari a. Untuk semua list selector yang mungkin, silahkan kunjungi <http://www.w3.org/TR/selectors/>

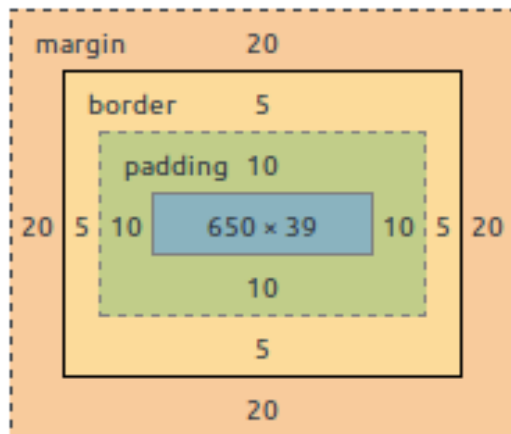
Box model

Setiap element tag dalam HTML bisa dipandang sebagai kotak-kotak yang memiliki ukuran lebar dan tinggi tertentu, serta border (bingkai). Jarak antara isi kotak dengan bingkai dikenal dengan sebutan padding, sedangkan jarak antar sebuah elemen dengan elemen lain dikenal dengan sebutan margin.

Untuk memanipulasi ukuran border, margin, dan padding, lebar dan tinggi dari elemen dapat digunakan script berikut:

```
selector{
    height: 39px;
    width: 650px;
    border-top: 5px solid black;
    border-bottom: 5px solid black;
    border-left: 5px solid black;
    border-right: 5px solid black;
    margin-top: 20px;
    margin-bottom: 20px;
    margin-left: 20px;
    margin-right: 20px;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 10px;
    padding-right: 10px;
}
```

Untuk lebih jelasnya, perhatikan gambar berikut:



float dan clear

Float menunjukkan kecenderungan posisi elemen. Jika nilai float adalah left, maka elemen akan cenderung ke kiri, jika posisi elemen adalah right, maka elemen cenderung akan ke kanan.

Berikut adalah contoh penggunaan float dan clear:

Kode CSS:

```
.pertama{
    float: left;
}
.kedua{
    float: right;
}
.tanpa-float{
    clear: both;
}
div{
    width: 20%;
    border: 1px solid black;
    padding: 5px;
}
```

Kode HTML:

```
<div class="pertama">Pertama</div>
<div class="pertama">Pertama</div>
<div class="pertama">Pertama</div>
<div>Normal</div>
<div class="tanpa-float">Tanpa-float</div>
<div class="kedua">Kedua</div>
<div class="kedua">Kedua</div>
<div class="kedua">Kedua</div>
<div>Normal</div>
<div class="tanpa-float">Tanpa-float</div>
```

position, left, top, dan z-index

Dalam CSS ada 4 macam posisi. Masing-masing adalah:

- static : left dan top tidak bisa diubah
- fixed : left dan top dihitung relatif terhadap pojok kiri atas browser
- absolute: left dan top dihitung relatif terhadap parent pertama yang position nya bukan static
- relative: left dan top dihitung dari letak normalnya, namun ruang yang seharusnya dimiliki elemen tidak dihilangkan.

Jika ada 2 elemen yang saling overlap, maka dapat digunakan z-index untuk mengatur elemen mana yang ada di depan, dan elemen mana yang ada di belakang. Semakin kecil z-index, artinya elemen akan terletak semakin belakang.

Berikut adalah contoh penggunaan position, left, top, dan z-index:

Kode CSS:

```
p{
    background-color: white;
    border: 1px solid black;
}
#pertama{
    left: 50px;
    top: 50px;
    position: static; /* ubah menjadi fixed, relative atau absolute */
    z-index: 1; /* ubah menjadi 2 */
    background-color: cyan;
}
#kedua{
    left: 100px;
    top: 55px;
    position: static; /* ubah menjadi fixed, relative atau absolute */
    z-index: 1;
    background-color: yellow;
}
```

Kode HTML:

```
<p>paragraf</p>
<p>paragraf</p>
<p>paragraf</p>
<p id="pertama">pertama</p>
<p id="kedua">kedua</p>
<p>paragraf</p>
```

color dan background-color

Color digunakan untuk mengatur warna tulisan, sedangkan background-color digunakan untuk mengatur warna background. Warna dapat ditulis dalam bahasa inggris (red, green, blue, yellow, cyan, black, gray, dsb), atau dalam bentuk heksadesimal, semisal #FFFFFF untuk putih, #FF0000 untuk merah, #00FF00 untuk hijau, #0000FF untuk biru.

Angka heksadesimal untuk merepresentasikan warna dibagi menjadi 3 bagian. 2 digit pertama adalah intensitas warna merah, 2 digit berikutnya elemen warna hijau, dan 2 digit terakhir elemen warna biru.

Berikut adalah contoh penggunaannya:

```
p{
  background-color: yellow; /* atau #FF00FF */
  color: black; /* atau #000000 */
}
```

font-size, font-style, font-family, dan font-weight

Font-size digunakan untuk mengatur ukuran font. Font-style digunakan untuk mengatur style font, font-family digunakan untuk mengatur jenis font, sedangkan font-weight digunakan untuk mengatur ketebalan font. Berikut contoh penggunaannya:

```
p{
  font-size: large;
  font-style: italic;
  font-family: tahoma;
  font-weight: bold;
}
```

background-image, background-position, dan background-repeat

Background-image digunakan untuk memberikan background pada elemen HTML. Background-position berguna untuk menentukan posisi background, sedangkan background-repeat digunakan untuk menentukan apakah background akan berulang. Ada 4 pilihan alternatif untuk background-repeat:

- repeat (berulang horisontal dan vertikal)
- repeat-y (berulang vertikal)
- repeat-x (berulang horizontal)
- no-repeat (tidak berulang)

Berikut contoh penggunaannya:

```
div{
  background-image: url('lokasi/gambar.jpg');
  background-position: right bottom;
  background-repeat: no-repeat;
}
```

Adapun background-repeat: repeat-x biasanya berguna untuk membuat banner website (dengan image yang memiliki lebar 1 pixel)

responsive (media screen)

Dewasa ini, para web developer lebih suka membuat satu halaman web yang tampil cantik baik di komputer layar lebar, laptop, tablet, dan smartphone. Pendekatan ini dimungkinkan dengan menggunakan css media. Cara kerja umumnya adalah membagi css ke dalam beberapa kategori layar. Salah satu pengkategorian yang bisa dipakai adalah sebagai berikut:

```
@media (min-width:320px) {  
    /* smartphones, iPhone, portrait 480x320 phones */  
}  
@media (min-width:481px) {  
    /* portrait e-readers (Nook/Kindle), smaller tablets @ 600 or @ 640  
    wide. */  
}  
@media (min-width:641px) {  
    /* portrait tablets, portrait iPad, landscape e-readers, landscape  
    800x480 or 854x480 phones */  
}  
@media (min-width:961px) {  
    /* tablet, landscape iPad, lo-res laptops ands desktops */  
}  
@media (min-width:1025px) {  
    /* big landscape tablets, laptops, and desktops */  
}  
@media (min-width:1281px) {  
    /* hi-res laptops and desktops */  
}
```

Bootstrap

Ada beberapa framework css untuk memudahkan dalam proses development. Salah satu yang terkenal adalah bootstrap (<http://getbootstrap.com>)

Bootstrap terdiri atas sekumpulan script CSS dan javascript dengan berbagai definisi class. Dengan menggunakan bootstrap, seorang web developer hanya perlu menyertakan CSS dan javascript yang disediakan, serta mendefinisikan elemen sesuai dengan tag-tag yang ditentukan.

CSS dan javascript yang disediakan bootstrap bisa didapatkan dengan cara mendownload atau dengan menggunakan CDN yang tersedia di internet. Untuk penggunaan offline (development tanpa terhubung ke internet), sebaiknya menggunakan CSS dan javascript yang sudah didownload, sedangkan untuk deployment, sebaiknya menggunakan CDN.

Beberapa elemen javascript bootstrap bergantung pada jquery (library javascript yang lain). Oleh sebab itu, perlu disertakan peng-include an jquery sebelum menyertakan javascript milik bootstrap.

Berikut adalah contoh peng-include an bootstrap:

```
<link rel="stylesheet"
href="http://netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.cs
s">
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
<script
src="http://netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js
"></script>
```

Untuk membuat website yang responsive, diperlukan pemahaman tentang konsep grid dalam bootstrap. Dalam bootstrap, sebuah halaman web dibagi menjadi 12 kolom. Untuk resolusi layar yang berbeda, kita dapat membuat pembagian yang berbeda pula. Dalam bootstrap ada 4 pembagian resolusi layar:

- .col-xs- untuk layar sangat kecil
- .col-sm- untuk layar kecil
- .col-md- untuk layar medium
- .col-lg- untuk layar besar

Berikut adalah contoh pembagian halaman menggunakan div secara responsive:

```
<div class="row">
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">isi</div>
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">isi</div>
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">isi</div>
  <div class="col-xs-12 col-sm-6 col-md-4 col-lg-3">isi</div>
</div>
```

Jika layar sangat kecil, maka setiap div akan memiliki lebar sesuai dengan lebar parent (12/12). Jika layar kecil, maka setiap div akan memiliki lebar setengah dari lebar parent (6/12). Jika layar medium, maka setiap div akan memiliki lebar sepertiga dari lebar parent (4/12). Sedangkan jika layar besar, maka setiap div akan memiliki lebar seperempat dari lebar parent (3/12).

Untuk memahami lebih lanjut mengenai bootstrap, silahkan kunjungi situs resmi bootstrap. Bootstrap juga menyediakan berbagai styling untuk form, tabel, dan navigasi.

Javascript

Jika HTML dan CSS masing-masing berkaitan dengan struktur dan tampilan web, maka javascript berkaitan dengan event dan aturan. Javascript biasanya digunakan untuk menambah nilai interaktifitas sebuah web.

alert

Alert berfungsi untuk menampilkan dialog pemberitahuan kepada pengunjung web. Contoh penggunaannya adalah sebagai berikut:

```
alert("Halo pengunjung, selamat datang");
```

Adapun penggunaan alert sebaiknya didasarkan pada alasan yang kuat, karena pop-up yang muncul terkadang terkesan mengganggu.

Variabel

Variabel berfungsi sebagai penampung nilai dinamis. Berikut adalah contoh sederhana penggunaan variabel:

```
var nama = "Budi";  
var rata_rata = 0;  
var nilai = new Array(100, 80, 90, 95);  
rata_rata = (nilai[0] + nilai[1] + nilai[2] + nilai[3]) / 4;  
alert("Nama : " + nama + ", Nilai rata-rata : " + rata_rata);
```

Berikut adalah penjelasan dari kode di atas:

- baris 1 : mendeklarasikan variabel "nama" yang bernilai "Budi"
- baris 2: mendeklarasikan variabel "rata_rata" yang bernilai 0
- baris 3: mendeklarasikan variabel "nilai" yang memiliki 4 nilai sekaligus: 100, 80, 90 dan 95. Variabel yang memiliki banyak nilai seperti ini, dikenal pula dengan sebutan Array. Untuk mengakses nilai array satu-persatu, dapat digunakan format sebagai berikut: nama_variabel[indeks], di mana indeks dimulai dari angka 0.
- baris 4: mengubah nilai variabel "rata_rata" menjadi penjumlahan dari semua nilai dalam variabel "nilai" dibagi empat.
- baris 5: menampilkan isi dari variabel nama dan rata-rata. Tanda "+" berfungsi untuk menghubungkan satu nilai dengan nilai lain.

prompt

Prompt berguna untuk mendapatkan inputan dari user (gunakan hanya jika memiliki alasan yang kuat untuk ini). Contoh:

```
var nama = prompt("Masukkan nama anda", "Farhat Lucu");
alert("Halo, " + nama );
```

Parameter pertama prompt adalah kata-kata yang akan muncul dalam kotak dialog, sedangkan parameter kedua adalah nilai defaultnya.

console.log

Terkadang untuk mengetahui isi dari sebuah variabel, seorang web developer menggunakan fungsi alert. Adapun cara tersebut kurang baik. Akan lebih baik menggunakan console.log. Berikut contoh penggunaannya:

```
var nilai = new Array(100, 80, 90, 95);
console.log(nilai);
```

Untuk menampilkan isi console, dapat digunakan developer tools pada browser chrome atau plugin firebug pada browser firefox.

Event

Jalannya sebuah aksi kadang sangat tergantung pada apa yang dilakukan user. Dalam javascript ada yang disebut event. Berikut adalah contoh event:

```
<button onclick="alert('hello');">Click me</button>
```

Dalam prakteknya, penulisan event seperti ini kurang baik, terutama jika sudah menggunakan library seperti jquery.

Function

Beberapa aksi yang sering dilakukan dapat dirangkum dalam sebuah function Berikut contohnya:

HTML:

```
<button onclick="tanya_nama();">Click me</button>
```

Javascript:

```
function tanya_nama(){
    var nama = prompt("Masukkan nama anda");
    alert("Halo, " + nama );
}
```

Percabangan

Terkadang jalannya sebuah aksi sangat dipengaruhi oleh hal lain. Di sini dibutuhkan percabangan. Cara melakukan percabangan adalah sebagai berikut:

```
var nama = prompt("Masukkan nama anda");
if(nama == "SBY"){
    alert("Selamat datang, Bapak Presiden");
}else{
    alert("Selamat datang, " + nama);
}
```

Format percabangan dapat dibuat bersarang, semisal berikut ini:

```
var nama = prompt("Masukkan nama anda");
if(nama == "SBY"){
    alert("Selamat datang, Bapak Presiden");
}else{
    if(nama == "Jokowi"){
        alert("Selamat datang, Bapak Gubernur");
    }else{
        alert("Selamat datang, " + nama);
    }
}
```

Perhatikan bahwa tanda == berbeda dengan tanda =. Tanda == digunakan untuk membandingkan dua nilai, sedangkan tanda = berfungsi untuk memberikan nilai pada variabel. Berikut adalah operator-operator yang bisa digunakan untuk perbandingan:

Operator	Arti	Penggunaan
==	Sama dengan	5 == (4+1)
!=	Tidak sama dengan	5 != 4
<	Kurang dari	4 < 5
>	Lebih dari	5 > 4
<=	Kurang dari atau sama dengan	4 <= 5
>=	Lebih dari atau sama dengan	5 >= 4
!	Tidak	false == !true
&&	Dan	(5 == (4+1)) && (4 < 5)
	Atau	5 == 4 5 == 5

Perulangan

Dalam program terkadang perlu dilakukan sesuatu secara berulang-ulang. Di sini dibutuhkan perulangan. Ada tiga macam perulangan yang umum digunakan, yakni do-while, while, dan for.

Pada format do-while, aksi akan dilakukan terlebih dahulu, kemudian diikuti pengecekan, sedangkan pada while dan for, pengecekan akan dilakukan terlebih dahulu sebelum aksi. Berikut adalah contoh implementasi dari ketiga jenis perulangan tersebut:

Do-while:

```
var nilai = new Array(100, 20, 50, 90, 80);
var i = 0;
do{
    alert("Nilai ke " + (i+1) + " : " + nilai[i]);
    i = i+1;
}while(i<5);
```

While:

```
var nilai = new Array(100, 20, 50, 90, 80);
var i = 0;
while(i<5){
    alert("Nilai ke " + (i+1) + " : " + nilai[i]);
    i = i+1;
}
```

For:

```
var nilai = new Array(100, 20, 50, 90, 80);
for(var i=0; i<5; i++){
    alert("Nilai ke " + (i+1) + " : " + nilai[i]);
}
```

Sama halnya seperti percabangan, dalam perulangan juga bisa terdapat perulangan atau percabangan lain.

Jquery

Jquery merupakan library javascript yang paling banyak digunakan. Untuk menyertakan jquery dapat digunakan cara sebagai berikut:

```
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
```

Penyertaan jquery dari server CDN sangat disarankan untuk deployment, namun untuk development, ada baiknya menggunakan versi offline dengan mendownload dari situs <http://jquery.com>

Dalam jquery, dikenal konsep selector yang persis sama dengan CSS. Pada umumnya, tanda \$ secara default merupakan alias untuk JQuery.

Event

Penggunaan event di JQuery sangatlah sederhana:

HTML:

```
<button id="tombol">Klik tombol</button>
```

Javascript:

```
$(document).ready(function() {  
    $('#tombol').click(function() {  
        alert('Tombol di-klik');  
    });  
});
```

html, val, dan attr

Html, val, dan attr masing-masing digunakan untuk mendapatkan atau mengubah isi, value, dan nilai atribut tag. Berikut contoh penggunaannya:

HTML:

```
<input type="text" id="teks" value="abc" />  
<div id="div">def</div>  
<button id="tombol">Klik</button>
```

Javascript:

```
$(document).ready(function() {
    $('#tombol').click(function() {

        // dapatkan atribut "type"
        var tipe_teks = $('#teks').attr('type');

        if(tipe_teks == 'text'){
            tipe_teks = 'password';
        }else{
            tipe_teks = 'text';
        }

        // ubah atribut "type"
        $('#teks').attr('type', tipe_teks);

        // dapatkan "value"
        var nilai_teks = $('#teks').val();

        nilai_teks = nilai_teks + "*";

        // ubah "value"
        $('#tks').val(nilai_teks);

        // dapatkan "isi"
        var isi_div = $('#div').html();

        isi_div = isi_div + "*";

        // ubah "isi"
        $('#div').val(isi_div);

    });
});
```

AJAX

AJAX adalah suatu teknologi yang memungkinkan pengiriman data ke server dan pengelolaan response dari server tanpa perlu terjadinya proses pindah halaman. Adanya AJAX akan membuat sebuah website tampak memiliki respon cepat. Contoh penggunaan AJAX adalah saran pencarian di google search serta tanda notifikasi di facebook.

Berikut adalah contoh AJAX sederhana menggunakan jquery:

HTML:

```
<input type="text" id="kata" placeholder="masukkan kata kunci" />
<div id="hasil">Hasil Pencarian</div>
```

Javascript:

```
$(document).ready(function() {
    $('#kata').keyup(function() {
        $.ajax({
            url: 'http://localhost/proxy.php',
            type: 'get',
            data: {
                q: $('#kata').val()
            },
            success: function(response) {
                $('#hasil').html(response);
            }
        });
    });
});
```

PHP (proxy.php):

```
<?php
echo file_get_contents('http://google.com/search?q=' . $_GET['q']);
```

Letakkan file PHP di server komputer lokal. Jika menggunakan windows dan xampp, letakkan di C:\xampp\htdocs\, jika menggunakan linux, letakkan di /var/www

PHP

PHP adalah bahasa pemrograman yang berjalan di sisi server. Sehingga script PHP harus diletakkan di web server. Script PHP diawali dengan <?php dan diakhiri dengan ?>

Echo

Untuk menampilkan teks di PHP, bisa digunakan fungsi echo dengan penggunaan sebagai berikut:

```
<?php
echo "Halo";
?>
```

Variabel

Variabel adalah penampung nilai dinamis. Dalam PHP, sebuah variabel dideklarasikan dengan prefix \$. Berikut contoh penggunaan variabel:

```
<?php
$nama = "Budi";
$rata_rata = 0;
$nilai = array(100, 80, 90, 95);
$rata_rata = ($nilai[0] + $nilai[1] + $nilai[2] + $nilai[3]) / 4;
echo "Nama : " . $nama + ", Nilai rata-rata : " . $rata_rata;
?>
```

- baris 2 : mendeklarasikan variabel "nama" yang bernilai "Budi"
- baris 3: mendeklarasikan variabel "rata_rata" yang bernilai 0
- baris 4: mendeklarasikan variabel "nilai" yang memiliki 4 nilai sekaligus: 100, 80, 90 dan 95. Variabel yang memiliki banyak nilai seperti ini, dikenal pula dengan sebutan Array. Untuk mengakses nilai array satu-persatu, dapat digunakan format sebagai berikut: nama_variabel[indeks], di mana indeks dimulai dari angka 0.
- baris 5: mengubah nilai variabel "rata_rata" menjadi penjumlahan dari semua nilai dalam variabel "nilai" dibagi empat.
- baris 6: menampilkan isi dari variabel nama dan rata-rata. Tanda "." berfungsi untuk menghubungkan nilai-nilai yang ada.

GET & POST

Get dan Post adalah dua cara yang paling umum untuk mengirimkan data ke web server. Data yang dikirim dapat berasal dari form ataupun URL. Berikut adalah contoh sederhana pengiriman dan

penerimaan data dengan menggunakan POST dan GET:

PHP (tangkap_data.php):

```
<?php
echo 'isi $_GET : ' ;
echo '<pre>';
var_dump($_GET);
echo '</pre>';
echo 'isi $_POST : ' ;
echo '<pre>';
var_dump($_POST);
echo '</pre>';
?>
```

HTML:

```
<!-- ubah method post dengan get -->
<form method="post" action="tangkap_data.php">
    <input type="text" name="nama" placeholder="Nama" />
    <input type="password" name="password" placeholder="Password" />
    <input type="submit" name="kirim" value="Kirim" />
</form>
```

tangkap_data.php akan menampilkan semua data \$_POST dan \$_GET. Untuk mengakses data secara individual, bisa digunakan \$_POST['nama'] atau \$_GET['password']

Percabangan

Cara paling sederhana untuk melakukan percabangan adalah dengan menggunakan if:

```
<?php
$nama = $_POST['nama'];
if($nama == "SBY"){
    echo "Selamat datang, Bapak Presiden";
}else{
    if($nama == "Jokowi"){
        echo "Selamat datang, Bapak Gubernur";
    }else{
        echo "Selamat datang, " . $nama;
    }
}
?>
```

Perhatikan bahwa tanda == berbeda dengan tanda =. Tanda == digunakan untuk membandingkan dua nilai, sedangkan tanda = berfungsi untuk memberikan nilai pada variabel. Berikut adalah operator-operator yang bisa digunakan untuk perbandingan:

Operator	Arti	Penggunaan
==	Sama dengan	5 == (4+1)
!=	Tidak sama dengan	5 != 4
<	Kurang dari	4 < 5
>	Lebih dari	5 > 4
<=	Kurang dari atau sama dengan	4 <= 5
>=	Lebih dari atau sama dengan	5 >= 4
!	Tidak	false == !true
&&	Dan	(5 == (4+1)) && (4 < 5)
	Atau	5 == 4 5 == 5

Perulangan

Dalam program terkadang perlu dilakukan sesuatu secara berulang-ulang. Di sini dibutuhkan perulangan. Ada tiga macam perulangan yang umum digunakan, yakni do-while, while, dan for.

Pada format do-while, aksi akan dilakukan terlebih dahulu, kemudian diikuti pengecekan, sedangkan pada while dan for, pengecekan akan dilakukan terlebih dahulu sebelum aksi. Berikut adalah contoh implementasi dari ketiga jenis perulangan tersebut:

Do-while:

```
$nilai = array(100, 20, 50, 90, 80);
$i = 0;
do{
    echo "Nilai ke " + ($i+1) + " : " + $nilai[$i] . '<br />';
    $i = $i+1;
}while($i<5);
```

While:

```
$nilai = array(100, 20, 50, 90, 80);
$i = 0;
while($i<5){
    echo "Nilai ke " + ($i+1) + " : " + $nilai[$i] . '<br />';
    $i = $i+1;
}
```

For:

```
var nilai = new Array(100, 20, 50, 90, 80);
for(var i=0; i<5; i++){
    echo "Nilai ke " + ($i+1) + " : " + $nilai[$i] . '<br />';
}
```

Sama halnya seperti percabangan, dalam perulangan juga bisa terdapat perulangan atau percabangan lain.

Akses Database

Untuk memanipulasi database, kita perlu menguasai script SQL. Salah satu database yang paling sering digunakan di web adalah MySQL. Berikut adalah contoh pengaksesan database dengan menggunakan kalimat SQL:

Select (Mengambil data):

```
SELECT * FROM nama_tabel;  
SELECT * FROM nama_tabel WHERE nama_kolom = 'suatu_nilai';  
SELECT nama_kolom_1, nama_kolom_2 FROM nama_tabel WHERE nama_kolom3 =  
'suatu_nilai';
```

Insert (Menambahkan data):

```
INSERT INTO nama_tabel(nama_kolom_1, nama_kolom_2) VALUES ('nilai_1',  
'nilai_2');
```

Delete (Menghapus data):

```
DELETE FROM nama_tabel WHERE nama_kolom_1 = 'suatu_nilai';
```

Update (Mengubah data):

```
UPDATE nama_tabel SET nama_kolom_1 = 'nilai_1', nama_kolom_2 = 'nilai_2'  
WHERE nama_kolom_3 = 'suatu_nilai';
```

Pada dasarnya, untuk mengakses database dari PHP, perlu dibuat koneksi ke database, dan dijalankan kalimat-kalimat SQL yang sesuai:

```
<?php  
// buka koneksi ke database  
$link = mysqli_connect('localhost', 'root', 'password', 'db');  
  
// masukkan data  
mysqli_query($link, "INSERT INTO nama_tabel(nama_kolom_1, nama_kolom_2)  
VALUES ('nilai_1', 'nilai_2')");  
  
// dapatkan data  
$query = mysqli_query($link, 'SELECT nama_kolom_1, nama_kolom_2 FROM  
nama_tabel');  
  
// tampilkan data  
while($row = mysqli_fetch_array($query)){  
    echo $row['nama_kolom_1'] . ' ' . $row['nama_kolom_2'] . '<br />';  
}
```

```
// tutup koneksi
mysql_close($link);
?>
```

MVC dengan laravel

Mengapa MVC penting?

Dalam membuat halaman web menggunakan gabungan HTML, css, javascript dan PHP, seringkali akan ditemui kendala karena tidak ada keterpisahan kode. Selain kode program menjadi sulit dibaca, juga akan muncul kendala untuk mengubah satu bagian dalam program.

Sebagai contoh terdapat sebuah program sebagai berikut:

```
<style type="text/css">
    table th{
        font-family: arial;
    }
</style>
<h1>Selamat Datang di halaman web kami</h1>
<?php
// buka koneksi ke database
$link = mysqli_connect('localhost', 'root', 'password', 'db');

// dapatkan data
$query = mysqli_query($link, 'SELECT nama_kolom_1, nama_kolom_2 FROM
nama_tabel');

// tampilkan data
?>
<ul>
<?php
while($row = mysqli_fetch_array($query)){
    echo '<li>'.$row['nama_kolom_1'] . ' ' . $row['nama_kolom_2'].
'</li>';
}
?>
</ul>

// tutup koneksi
mysql_close($link);
?>
```

Dalam script tersebut terdapat CSS, HTML dan PHP. Pada kenyataannya, terkadang masih ada tambahan javascript, tampilan banner dan sebagainya.

Semisal terdapat 10 halaman web untuk menampilkan data dari tabel-tabel yang berbeda, dan kesemuanya memiliki struktur pola yang sama. Kerumitan akan muncul taktala kata-kata "Selamat

Datang di halaman web kami" hendak diubah menjadi "Welcome to Our Fabulous Web Page". Perubahan sederhana ini akan menyebabkan programmer/designer harus membuka kembali 10 halaman web yang berbeda tersebut dan melakukan editing satu-persatu.

Sekarang bayangkan, seandainya terdapat kebijakan di mana data yang tadinya ditampilkan dalam bentuk list harus diubah ke dalam bentuk tabel. Tentu saja ini akan menimbulkan masalah baru, karena ada kemungkinan salah edit yang sangat besar.

Dewasa ini, para praktisi web mengembangkan suatu pola pengembangan yang disebut MVC. Tujuannya adalah untuk memisahkan sebuah program menjadi bagian data, bagian logic, dan bagian tampilan, sehingga kemungkinan salah edit menjadi berkurang. Selain itu, MVC juga bertujuan untuk mengurangi redundansi kode (kode yang sama, namun ditulis berulang kali).

M – Model

Model adalah bagian program yang berkaitan dengan data. Di dalam model biasanya didefinisikan bagaimana sebuah tabel dapat diakses secara sederhana. Biasanya sebuah model akan terdiri dari sebuah kelas yang memiliki banyak fungsi. Fungsi-fungsi tersebut mendefinisikan bagaimana melakukan select, insert, update, dan delete. Dengan fungsi-fungsi yang sudah terdefinisi tadi, maka seorang programmer hanya perlu memanggil fungsi terkait tanpa perlu menuliskan kalimat SQL secara berulang ulang. Sebuah model yang sama bisa dipakai di banyak controller yang berbeda.

V – View

View adalah bagian program yang berfungsi untuk menunjang tampilan. Dalam framework-framework modern, umumnya View memiliki template engine khusus (dalam laravel disebut blade), sehingga bagian-bagian seperti banner, footer dan sebagainya cukup didefinisikan sekali saja.

C – Controller

Controller adalah bagian program yang berfungsi untuk menangani request dari user. Secara umum dapat dikatakan bahwa controller bertugas menghubungkan model dan view dan menyediakan cara untuk berinteraksi dengan pengguna.

Menginstall laravel

Ada berbagai cara untuk menginstall laravel, misalnya dengan menggunakan composer atau phar. Dari berbagai alternatif cara yang ada, cara termudah adalah dengan langsung mendownload source-code laravel (<https://github.com/laravel/laravel>) dan meletakkannya di htdocs atau var/www.

Berbeda dengan banyak framework PHP yang lain, dalam laravel, hanya directory public saja yang wajib diletakkan di htdocs atau /var/www. Directory yang lain sebaiknya diletakkan di tempat lain. Hal

ini dimaksudkan untuk alasan keamanan. Untuk melakukan ini, maka dua baris di file `/public/index.php` perlu disesuaikan dengan lokasi folder bootstrap:

```
require __DIR__.'/../bootstrap/autoload.php';
$app = require_once __DIR__.'/../bootstrap/start.php';
```

Selain itu, lokasi public untuk file `/bootstrap/path.php` juga perlu diubah:

```
'public' => __DIR__.'/../public',
```

Hal lain yang perlu disesuaikan adalah konfigurasi database pada `/app/config/database.php`:

```
return array(
    ...
    'default' => 'mysql',
    'connections' => array(
        ...
        'mysql' => array(
            'driver' => 'mysql',
            'host' => 'localhost',
            'database' => 'laravel',
            'username' => 'root',
            'password' => 'password',
            'charset' => 'utf8',
            'collation' => 'utf8_unicode_ci',
            'prefix' => '',
        ),
    ),
);
```

route

Route didefinisikan di `/app/routes.php` dan berfungsi untuk memasang request dan response. Semisal diinginkan request <http://localhost/halo> mengembalikan response 'Halo', maka dapat didefinisikan route sebagai berikut:

```
<?php
Route::get('halo', function()
{
    return 'Halo';
});
?>
```

view (blade)

View terletak di dalam `/app/views`. Di dalamnya dapat didefinisikan template yang memudahkan urusan desain. Berikut adalah contoh penggunaan view:

app/views/**layout**.blade.php:

```
<html>
  <body>
    <h1>Selamat datang</h1>
    @yield('content')
    <hr />
    Powered by Laravel, developed by &copy; Go Frendi
  </body>
</html>
```

app/views/**halo**.blade.php:

```
@extends('layout')
@section('content')
    Halo !!!!
@stop
```

routes.php:

```
<?php
Route::get('halo', function()
{
    return View::make('halo');
});
?>
```

migration

Migration dibutuhkan karena terkadang diperlukan perubahan pada struktur database. Pada laravel terdapat program artisan yang dapat digunakan untuk menambahkan migrasi baru. Berikut adalah perintah yang digunakan:

```
php artisan migrate:make create_pokemon_table
```

Setelah perintah tersebut dijalankan, maka akan tercipta sebuah file di direktor /app/database/migrations. Di dalam file tersebut terdapat 2 buah fungsi, yakni up dan down. Fungsi up berguna untuk menciptakan/mengubah skema ke versi yang lebih baru. Fungsi down berguna untuk menghapus/mengubah skema ke versi yang lebih lama.

Berikut adalah contoh file migrasi:

```
<?php
public function up() {
    Schema::create('pokemons', function($table) {
        $table->increments('id');
        $table->string('name')->unique();
        $table->string('color');
        $table->timestamps();
    });
}
public function down() {
    Schema::drop('pokemons');
}
?>
```

Untuk mengaplikasikan migrasi, dapat digunakan perintah berikut:

```
php artisan migrate
```

model (eloquent)

Model adalah bagian program yang terkait dengan database. Model terletak di dalam /app/models/.

Berikut adalah contoh model untuk mengakses tabel pokemon:

app/models/Pokemon.php:

```
<?php
class Pokemon extends Eloquent {}
?>
```

Untuk menampilkan isi dari tabel pokemon, dapat digunakan route (di /app/routes.php) sebagai berikut:

```
<?php
Route::get('pokemon', function()
{
    $pokemons = Pokemon::all();
    return View::make('pokemon')->with('pokemons', $pokemons);
});
?>
```

Dan view sebagai berikut (app/views/pokemon.blade.php):

```
<?php
@extends('layout')
@section('content')
    @foreach($pokemons as $pokemon)
        <p>{{ $pokemon->name }}</p>
    @endforeach
@stop
?>
```

controller

Controller berfungsi untuk menghubungkan model & view.

app/models/Pokemon.php:

```
<?php
class Pokemon extends Eloquent {}
?>
```

Untuk menampilkan isi dari tabel pokemon, dapat digunakan route (di /app/routes.php) sebagai berikut:

```
<?php
class PokemonController extends BaseController{
    function get(){
        $pokemons = Pokemon::all();
        return View::make('pokemon')->with('pokemons', $pokemons);
    }
};
?>
```

Karena sebagian logic telah diletakkan di controller, maka di /app/routes.php, cukup didefinisikan:

```
<?php
Route::get('pokemon', 'PokemonController@get');
?>
```

Dan view sebagai berikut (app/views/pokemon.blade.php):

```
<?php
@extends('layout')
@section('content')
    @foreach($pokemons as $pokemon)
        <p>{{ $pokemon->name }}</p>
    @endforeach
@stop
?>
```

Tips:

1. Beberapa hal yang bisa dilakukan dengan Eloquent:

```
// dapatkan semua pokemon
Pokemon::all();

// dapatkan pokemon dengan id 1, ubah namanya menjadi pikachu, dan simpan
$pokemon = Pokemon::find(1);
$pokemon->name = 'pikachu';
$pokemon->save();
```

```
// buat pokemon baru
$pokemon = new Pokemon()
$pokemon->name='charmender';
$pokemon.save();

// hapus pokemon
$pokemon->delete();
```

2. Walaupun Laravel menyediakan Eloquent untuk manipulasi data di database, namun Laravel juga masih menyediakan cara untuk mengeksekusi kalimat SQL secara sederhana.

```
$results = DB::select('select * from users where id = ?',
array('value'));
DB::insert('insert into users (id, name) values (?, ?)', array(1,
'Dayle'));
DB::update('update users set votes = 100 where name = ?', array('John'));
DB::delete('delete from users');
```

3. Sebisa mungkin gunakan controller agar isi file routes.php tidak terlalu banyak.

4. Cara menggunakan session:

```
Session::get('key');
Session::get('key', 'default');
Session::get('key', function(){ return 'default'; });
Session::put('key', 'value');
```

5. Validasi input:

```
$rules = array(
    'name' => 'required',
    'email' => 'required|email',
    'nerd_level' => 'required|numeric'
);
$validator = Validator::make(Input::all(), $rules);
if($validator->fails()){
    ...
}else{
    ...
}
```

Android web-view

Untuk membuat aplikasi android, dapat digunakan android-studio atau eclipse dengan plugin android.

Setelah membuat aplikasi, AndroidManifest.xml perlu diberi tambahan permission untuk mengakses internet:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Dalam aplikasi yang sudah dibuat, harus ditambahkan komponen web-view. Kemudian pada event onCreate, diberi script sebagai berikut:

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.webview);  
    webView = (WebView) findViewById(R.id.webView1);  
    webView.getSettings().setJavaScriptEnabled(true);  
    webView.loadUrl("http://www.google.com");  
}
```