

# Panduan Lengkap AngularJS untuk Pemula dan Menengah

By Y.N. Pamungkas Jayuda at @2014 ThinkBuntu<sup>1</sup>

## Pendahuluan

Selamat datang di AngularJS Tutorial. AngularJS adalah aplikasi client-side javascript framework. AngularJS biasanya dipakai oleh programmer yang menguasai javascript dengan baik. Ada banyak tutorial mengenai AngularJS, jadi dalam tutorial ini saya akan berusaha menyajikan secara sederhana dan mendasar pada AngularJS. Misalkan kita akan belajar AngularJS untuk membuat satu halaman aplikasi sederhana. Selain itu saya juga menuliskan source code dengan lengkap, atau Anda bisa download di Github saya. Adapun pembahasan kita yaitu :

1. Pendahuluan
2. Membuat Aplikasi Sederhana
3. MVC Architecture
4. Modules di AngularJS
5. Kompleks Models di AngularJS
6. Membaca JSON di AngularJS
7. Filter data
8. Data Binding
9. Routing
10. Route Parameter
11. Navigasi
12. Animasi
13. Tabs
14. Form Validasi
15. Custom Directive.

Tutorial ini akan menjawab pertanyaan berikut :

1. Apa yang tercover dalam tutorial ini ?

Kita akan belajar bagaimana AngularJS bisa digunakan untuk membuat aplikasi dengan sederhana, cepat dan efisien. Kita akan membuat aplikasi AngularJS secara lengkap dan utuh.

---

<sup>1</sup> [www.thinkbuntu.com](http://www.thinkbuntu.com) an open community for developers

Kita akan banyak berbicara mengenai konsep MVC. Selain itu dibahas juga bagaimana kita bekerja dengan module dan AngularJS service (http, routes, provider dan animasi).

2. Apa yang perlu diketahui sebelum belajar AngularJS ?

Anda harus familier dan sudah mengenal HTML, Javascript dan perbedaan antara JS dan JSON. AngularJS bukan untuk pemula, karena AngularJS merupakan javascript tingkat atas. Untuk bekerja dengan AngularJS baiknya Anda mempunyai text-editor seperti Notepad, Sublime Text Editor.

3. Apa itu AngularJS APP ?

AngularJS adalah javascript framework untuk membuat SINGLE PAGE WEB APPLICATION, jadi hanya 1 page aja kawan. Aplikasi Anda akan lebih cepat, karena website Anda hanya akan mengambil atau load data content yang diperlukan saja.

4. Kenapa harus AngularJS framework ?

1. Lebih cepat coding dengan AngularJS

1. AngularJS tidak tertutup, jadi Anda bisa memakai javascript lain bersamaan dengan AngularJS.
2. Terdapat fungsi AJAX di AngularJS untuk load data ( Database, text, etc). Selain itu AngularJS sangat mudah handle JSON data.
3. AngularJS menggunakan arsitektur MVC.

5. Apa feature dasar dari AngularJS ?

1. directive

1. data-binding
2. filters
3. modules
4. routes
5. controllers

Ok, itu yang akan di pelajari dalam tutorial ini. Jika ada yang kurang jangan sungkan - sungkan untuk memberitahukan ke saya via email di [yulius.jayuda@gmail.com](mailto:yulius.jayuda@gmail.com) Source code yang dibahas dalam tutorial ini Anda bisa download di github saya

<https://github.com/Jayuda/Tutorial-AngularJS-HTML5>

## Membuat Aplikasi Sederhana

Dalam bab ini kita akan belajar hal paling dasar dari AngularJS. Jika Anda belum mengerti mengenai AngularJS silahkan baca di *Pendahuluan*. Berikut langkah-langkah awal mengenai AngularJS :

1. Download AngularJS

Pastikan anda mendownload versi terbaru dan Angular.min.js harus ada di dalamnya.

2. Buat folder dengan nama KaryawanApp.

3. Masuk ke dalam folder KaryawanApp, lalu buat folder baru dengan nama lib.
4. Masuk ke folder lib tersebut, lalu buat folder baru dengan nama Angular.
5. Copy Paste file Angular.min.js ke folder Angular tersebut (lihat tahap 4).
6. Buat file index.html di dalam folder KaryawanApp (lihat tahap 2) dan tuliskan baris coding berikut ini :

```
1  <!doctype html>
2  <html ng-app>
3  <head>
4      <meta charset="UTF-8">
5      <title>KaryawanAPP</title>
6      <script type="text/javascript"
7      src="lib/Angular/angular.min.js"></script>
8  </head>
9  <body>
10     <input type="text" ng-model="name">
11     <h2> Welcome {{name+1}} </h2>
12 </body>
13 </html>
```

7. Selesai, untuk mencoba silahkan buka dari browser. Kalo gagal, pastikan coding Anda benar dan Folder KaryawanApp berada di folder webserver.

#### *Penjelasan Syntax Program*

Ok,... Sekarang akan saya jelaskan apa maksud dari coding diatas, penjelasan lebih detail ada di Bagian 6.

1. ng-app  
Syntacs inilah yang memberitahukan ke browser bahwa coding dijalankan. Semacam trigger ke javascript yang mangacu dari ng-app ini.
2. ng-model :  
Ini ada lah control ke model. Jadi misal kita punya object input, select, atau textarea maka data dari object itu di binding ke variable. Misalkan nama dari textbox "NIP" maka dibinding ke {{sNIP}}. Penamaan tersebut terserah Anda, biasanya saya memberikan penamaan diawali dengan karakter huruf kecil yang menandakan tipe data dari variable tersebut. Misalkan sNIP maka tipe data String untuk variable NIP. Lebih mudah kan,.. IMHO :)

Dua syntax tersebut berasal dari AngularJS, untuk syntax lainnya itu HTML kawan.

#### **Kembangkan Kreatifitas Anda**

Jadi, dari coding sederhana diatas kita membuat textbox dimana setiap text yang kita tuliskan akan muncul dibawahnya. Anda bisa mengembangkan dengan operasi penjumlahan misalkan {{1+nama}}, etc.

## MVC Architecture

Dari struktur, AngularJS cukup berbeda dengan Javascript biasa. Salah satu struktur arsitek bahasa pemrograman yang terkenal yaitu MVC bisa di terapkan di AngularJS ini. MVC adalah struktur yang membagi object menjadi *model*, *view*, *controller*. Mari kita lihat seperti apa MVC di AngularJS.

Sebelum memulai AngularJS, Anda harus membaca bagian I : *Membuat Aplikasi Sederhana dg AngularJS*. Karena dalam bagian tersebut dijelaskan hal paling mendasar dari AngularJS.

### Model

Mudahnya, Model bisa dianggap sebagai DATA. Jadi data bisa diambil dari database, atau hard-code langsung atau dari JSON, etc. Biar anda bisa memahami hal tersebut, berikut contoh model :

```
<script type="text/javascript">
function controller($scope) {
    $scope.dataKaryawan = {
        'nama' : 'Nunung',
        'jabatan' : 'Jabat Tangan',
        'usia' : '1000 Tahun'
    }
}
</script>
```

*Penjelasan Syntax :*

*\$scope* adalah object dan *dataKaryawan* adalah variable yang ditambahkan di *\$scope*. Data yang disajikan berupa array jadi termasuk JSON.

### View

View berfungsi untuk menampilkan data dari model. Syntax yang dipakai untuk view adalah tanda kurung kurawal sebanyak 2 kali. Misalkan seperti ini :

```
{{dataKaryawan.nama}}
```

*Penjelasan Syntax :*

View mengambil data dari variabel *dataKaryawan*, dengan mengambil data JSON untuk nama.

### Controller

Controller berfungsi untuk mengontrol **model** melalui tampilan yang dihasilkan **view**. Contoh syntax dari controller misalkan seperti ini :

```
function namacotroller($scope)
```

```
}
```

dengan view seperti berikut :

```
<div ng-controller = "namacontroller"> </div>
```

*Penjelasan Syntax :*

Controller yang di tampilkan di view tersebut akan memanggil/mentrigger function food.

Sehingga hasil akhir dari coding MVC tersebut seperti berikut :

```

1  <!doctype html>
2  <html ng-app>
3  <head>
4      <meta charset="UTF-8">
5      <title>Guitar App</title>
6      <script type="text/javascript" src="lib/Angular/angular.min.js"></script>
7      <script type="text/javascript">
8          function controller($scope) {
9              $scope.dataKaryawan = {
10                  'nama' : 'Nunung',
11                  'jabatan' : 'Jabat Tangan',
12                  'usia' : '1000 Tahun'
13              }
14          }
15      </script>
16  </head>
17  <body>
18      <div ng-controller = "controller">
19          <h1> {{dataKaryawan.nama}} </h1>
20      </div>
21  </body>
22  </html>

```

## Module di AngularJS

Sejauh ini kita sudah membuat aplikasi sederhana dengan AngularJS. Aplikasi yang baru saja kita bangun itu sangat kuno dan tidak praktis, karena semua coding berada pada satu halaman yang sama. Hal ini akan sangat menyulitkan jika kita ketemu dengan aplikasi yang cukup besar. Jadi akan lebih baik kita mengorganisasi semua code tersebut ke dalam module. Ingat module ada bukan hanya di AngularJS. Karena hampir setiap bahasa pemrograman terutama javascript terdapat module.

Module merupakan salah satu cara untuk mengorganisir code berdasar fungsi. Coding yang berfungsi sama dikelompokkan menjadi satu. Berikut salah satu tampilan dari module :

```
Angular.module("myApp", []);
```

Dari controller pada bagian sebelumnya, Lakukan langkah-langkah berikut :

1. Buat folder js
2. buat file controllers.js
3. Cut model dan controller dari index.html

4. Paste ke controller.js
5. Tambahkan baris code berikut sebelum function controller.  
    `var myApp = angular.module("myApp", []);`  
    `myApp.controller("controller",`  
        `function controller($scope) {`  
            `$scope.dataKaryawan = {`  
                `'nama' : 'Nunung',`  
                `'jabatan' : 'Jabat Tangan',`  
                `'usia' : '1000 Tahun'`  
            `}`  
        `}`  
    `)`
6. Sehingga tampilan menjadi :

```
1  var myApp = angular.module("myApp", []);
2  myApp.controller("controller",
3      function controller($scope) {
4          $scope.dataKaryawan = {
5              'nama' : 'Nunung',
6              'jabatan' : 'Jabat Tangan',
7              'usia' : '1000 Tahun'
8          }
9      }
10 )
```

## Kompleks Models di AngularJS

Sejauh ini kita sudah berhasil membuat aplikasi sederhana dengan AngularJS. Lebih lanjut kita akan membuat MVC yang lebih kompleks. Kompleks disini berarti kita membagi project ke dalam beberapa JSON. tapi untuk tahap ini kita cukup memakai satu JSON saja. Tutorial pada bab ini akan berkaitan dengan bagian berikutnya. Ok,.. mari kita mulai, supaya penasaran Anda terobati. Kita akan mendapatkan data ( JSON ) dari controllers.js dan menampilkan data tersebut melalui looping.

Ok,... Dari project sebelumnya, sekarang anda ubah index.html dengan code berikut :

```

1 <!doctype html>
2 <html ng-app="myApp">
3 <head>
4   <meta charset="UTF-8">
5   <title>Guitar App</title>
6   <script type="text/javascript" src="lib/Angular/angular.min.js"></script>
7   <script type="text/javascript" src="js/controllers.js"></script>
8 </head>
9 <body>
10   <div class="main" ng-controller="controller">
11     <li class="itemHolder" ng-repeat="item in dataKaryawan">
12       <div>
13         <h3>Nama : {{item.name }}</h3>
14         <p>Jabatan : {{item.jabatan}}</p>
15         <p>Usia : {{item.usia}}</p>
16       </div>
17     </li>
18   </div>
19 </body>
20 </html>

```

Sedangkan untuk controllers.js anda tambahkan code item menjadi seperti ini :

```

1 var myApp = angular.module("myApp", []);
2 myApp.controller("controller",
3 function controller($scope) {
4   $scope.meal = [
5     {
6       "name" : "1. Acoustic Guitar",
7       "description": "Acoustic, electro-acoustic and classical guitars, from leading brands.",
8       "image" : "Acoustic"
9     },
10    {
11      "name" : "2. Electric Guitar",
12      "description": "Ever-popular solid-body guitars, to hollow-body guitars and archtops.",
13      "image" : "Electric"
14    },
15    {
16      "name" : "3. Bass Guitar",
17      "description": "Classic Fender Bass, Precision and Jazz basses, to Thunderbirds.",
18      "image" : "Bass"
19    }
20  ]
21 }
22 );

```

#### Penjelasan Syntax

1. `ng-app="myApp"`  
Syntax tersebut digunakan untuk deklarasi html untuk load module myApp di controllers.js
2. `ng-controller="controller"`  
Bagian inilah sebagai baris code yang menjembatani model dengan view
3. `ng-repeat="item in dataKaryawan"`  
*item* adalah referensi untuk variabel di controllers.js dan *ng-repeat* berfungsi sebagai loop atau perulangan dari JSON mulai dari item pertama sampai item terakhir.
4. `{{item.name}}` mengambil item dengan kategori 'name' dari variable *dataKaryawan*.

Bagaimana tidak banyak perubahan kan ? Pada tahap ini kita belajar bagaimana looping di view

untuk menampilkan data list items dari model. Hasil dari coding tersebut diatas akan seperti ini :



Setelah Anda mempelajari looping tersebut, bagaimana cukup mudah kan ? maka yang tak kalah penting adalah bagaimana membaca data dari JSON. Baca bab berikutnya :)

### Membaca JSON di AngularJS

Kita dapat membaca data eksternal dengan AngularJS. Dari latihan pertama sampai bab looping data, kita menuliskan data di variable *dataKaryawan* yang ditulis secara hardcoded. Anda mungkin berpikir, bagaimana membaca data dari luar sehingga dalam mengatur data akan lebih mudah. Pada teknik tradisional javascript anda mengambil data menggunakan XHR request dari JSON, atau sering disebut dengan AJAX.

Di AngularJS kita bisa membaca data eksternal dengan teknologi yang disebut sebagai *AngularJS Service* atau http service di AngularJS. AngularJS service tidak lebih dari cara membaca data dari suatu file. Dalam hal ini kita akan membaca data di http service di server.

Di tutorial sebelumnya anda memiliki controllers.js, dimana data dituliskan langsung di variabel *dataKaryawan*. Di controllers.js tersebut Anda ubah dengan mengambil data dari file JSON. Pertama kita buat *data.json* di folder js. Dengan code seperti ini :



```

1  [
2      {
3          "name"    : "1. Nunung",
4          "jabatan" : "Jabat Tangan.",
5          "usia"    : "1000 Tahun"
6      },
7      {
8          "name"    : "2. Pamungkas",
9          "jabatan" : "Jabat Kaki.",
10         "usia"    : "101 Dalmation"
11     },
12     {
13         "name"    : "3. Jayuda",
14         "jabatan" : "Jabat Hati.",
15         "usia"    : "1 Hari"
16     }
17 ]
18

```

Setelah itu kita ubah controllers.js agar membaca ke data.json. Sehingga tampilan controllers.js menjadi seperti ini :

```

1  var myApp = angular.module("myApp", []);
2  myApp.controller(
3      "controller",
4      [
5          '$scope',
6          '$http',
7          function($scope, $http) {
8              $http.get('js/data.json').success (function(data){$scope.dataKaryawan = data;});
9          }
10     ]
11 );|

```

Ok,.. pada tahap ini kita telah belajar membaca data dari file eksternal. Baris *js/data.json* bisa anda ubah ke data server misalkan *http://serveranda.com/data.json* sehingga aplikasi Anda mengambil data dari server.

## Membuat Filter dengan AngularJS

AngularJS memungkinkan Anda mengontrol data, misalkan Anda ingin menampilkan data tertentu yang memenuhi kriteria. Fungsi ini biasa disebut dengan filter. Filter di AngularJS hanya mengubah tampilan data dari scope ke viewer, tanpa mengubah sumber data. Agar lebih mudah di pahami kita langsung terapkan di coding saja.

Dari materi sebelumnya Anda tambahkan viewer untuk filter dan order seperti berikut :

```
<div>
  <h1>Data Karyawan</h1>
  <label>Search</label>
  <input type="text" ng-model="query" >

  <label>Sort by:
    <select ng-model="orderKaryawan">
      <option value="name" selected>name</option>
      <option value="reknown">Reknown</option>
    </select>
  </label>
  <br>
  <label>
    <input type="radio" ng-model="direction" name="direction" checked>
    Ascending
  </label>
  <label>
    <input type="radio" ng-model="direction" name="direction" value="reverse" >
    Descending
  </label>
</div>
<hr>
```

#### Penjelasan Syntax

1. *ng-model="query"* directive nama model yang dipakai buat input query text
2. *ng-model="orderKaryawan"* directive nama model buat order data
3. *ng-model="direction"* directive model buat mengubah tipe order

Lalu ubah *ng-repeat* menjadi seperti ini :

```
<li class="itemHolder" ng-repeat="item in dataKaryawan | filter:query | orderBy:orderKaryawan:direction ">
  <div>
    <h3>Nama : {{item.name | uppercase }}</h3>
    <p>Jabatan : {{item.jabatan}}</p>
    <p>Usia : {{item.usia}}</p>
  </div>
</li>
```

#### Penjelasan Syntax

*\$scope* yang handle model *orderKaryawan*

Pada model tambahkan object filter. Sehingga *controllers.js* menjadi :

```

1  var myApp = angular.module("myApp", []);
2  myApp.controller(
3      "controller",
4      [
5          '$scope',
6          '$http',
7          function($scope, $http) {
8              $http.get('js/data.json').success (function(data){
9                  $scope.dataKaryawan = data;
10                 $scope.orderKaryawan = 'name';
11             });
12         }
13     ]
14 );

```

Ok,.. Selesai,.. silahkan anda coba program Anda

## Routing

Saat aplikasi yang kita bangun terlalu rumit dan besar, maka jalan terbaik adalah mengorganisir dan mengelompokkan program berdasar kriteria tertentu. AngularJS adalah framework yang digunakan untuk membuat single aplikasi. Salah satu cara untuk membuat single aplikasi tersebut disebut sebagai *routing*.

Routing berarti mengakses sub menu berdasar url halaman yang di akses. Banyak orang menyebutnya dengan *Deep-Linking*

Ok Mari kita mulai membuat single aplikasi. Langkah pertama adalah mengcopy file angular-route.js ke folder `/lib/Angular`. Setelah itu buatlah folder pages, yang berlokasi di `/www/pages/`. Lalu masuk ke folder pages tersebut dan buatlah 3 halaman berbeda.

1. home.html

```

1  <div>
2      <h1>Home Page</h1>
3      <p>{{ message }}</p>
4  </div>

```

1. about.html

```

1 <div>
2   <h1>About Page</h1>
3   <p>{{ message }}</p>
4 </div>

```

1. contact.html

```

1 <div>
2   <h1>Contact Page</h1>
3   <p>{{ message }}</p>
4 </div>

```

#### *Penjelasan Syntax*

Dari 3 halaman diatas, akan dijadikan sebagai view untuk variable {{message}}. Kenapa harus 3 halaman agar kita bisa mengkondisikan bagaimana mengubah halaman dengan 3 page berbeda.

Setelah itu mari kita buat halaman index.html menjadi seperti ini :

```

1 <!DOCTYPE html>
2 <html ng-app="scotchApp">
3 <head>
4   <script src="lib/Angular/angular.min.js"></script>
5   <script src="lib/Angular/angular-route.js"></script>
6   <script src="js/script.js"></script>
7 </head>
8 <body ng-controller="mainController">
9   <ul>
10    <li><a href="#">Home</a></li>
11    <li><a href="#about"> About</a></li>
12    <li><a href="#contact">Contact</a></li>
13  </ul>
14
15  <div id="main">
16    <div ng-view></div>
17  </div>
18  <p>View the tutorial on Pamungkas Jayuda</p>
19 </body>
20 </html>

```

Lalu kita buat file script.js yang berfungsi sebagai controller. Berikut syntax yang saya pakai :

```

1  // create the module and name it scotchApp
2  var scotchApp = angular.module('scotchApp', ['ngRoute']);
3
4  // configure our routes
5  scotchApp.config(function($routeProvider) {
6      $routeProvider
7          // route for the home page
8          .when('/', {
9              templateUrl : 'pages/home.html',
10             controller : 'mainController'
11         })
12
13         // route for the about page
14         .when('/about', {
15             templateUrl : 'pages/about.html',
16             controller : 'aboutController'
17         })
18
19         // route for the contact page
20         .when('/contact', {
21             templateUrl : 'pages/contact.html',
22             controller : 'contactController'
23         });
24 });
25
26 // create the controller and inject Angular's $scope
27 scotchApp.controller('mainController', function($scope) {
28     // create a message to display in our view
29     $scope.message = 'Everyone come and see how good I look!';
30 });
31
32 scotchApp.controller('aboutController', function($scope) {
33     $scope.message = 'Look! I am an about page.';
34 });
35
36 scotchApp.controller('contactController', function($scope) {
37     $scope.message = 'Contact us! JK. This is just a demo.';
38 });
39 |

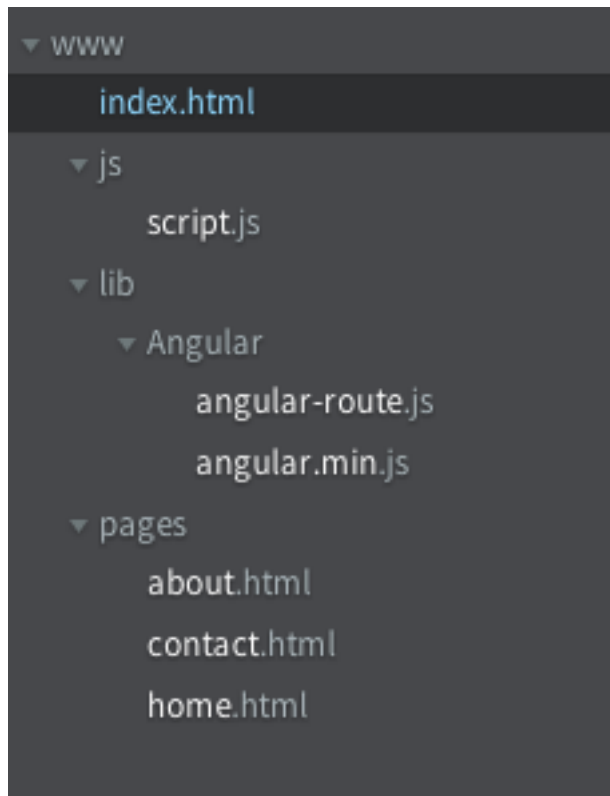
```

#### Penjelasan Syntax

Baris pertama menjelaskan bahwa saya kasih nama module dengan *scotchApp* Sehingga jika Anda lihat di index.html maka akan menemukan baris `<html ng-app="scotchApp">`.

Syntax [ 'ngRoute' ] disini menjelaskan bahwa module ini menggunakan service routing dari Angular-route.js. Sehingga pada baris selanjutnya terdapat 3 kondisi yang masing masing di lempar ke view dan controller berbeda. Cukup sederhana bukan :)

Sehingga tree struktur data menjadi seperti ini. Silahkan Anda cek apakah lokasi dan nama file sudah sesuai.



Ok, Selesai silahkan di coba