

WARNING :

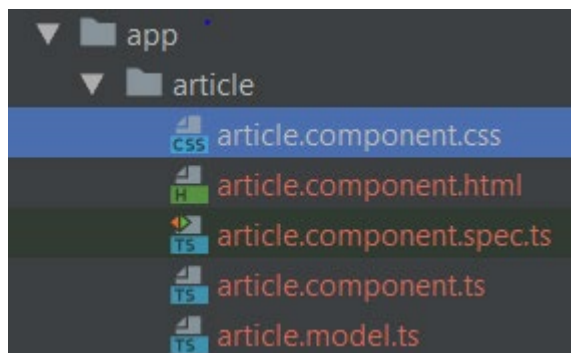
Mohon Langkah pada tutorial ini dibaca secara perlahan dan Mohon tidak melewati langkah-langkah yang ada.

Tutorial minggu ini cukup berat sehingga tidak bisa dimengerti apabila Anda hanya melakukan copy-paste saja. Mohon diperhatikan *comment* yang ada juga. Apabila tidak mengerti, mohon ditanyakan ke Asisten Lab atau Dosen Pengampu Pengantar teknologi Internet.

Week 12 IF 300 Tutorial 1 – Angular model with objects

Menggunakan angular model dengan object kita bisa lakukan inject class model yang dibuat dengan object dengan fungsi import. Agar kita lebih terbayang bagaimana mari kita sembari mengcode contoh program angular reddit. Sebelumnya kita ingat minggu sebelumnya kita sudah mempelajari beberapa jenis directive pada week sebelumnya seperti ngFor, dsb . Pada week ini kita bakal memakai angular directive tersebut.

Langkah pertama diawali dengan kita buat project terlebih dahulu menggunakan ng new. (nama project bebas untuk tutorial ini). Selanjutnya kita akan membuat component bernama article menggunakan ng generate component, akan terbentuk dalam folder :



Selanjutnya karena nantinya kita akan menggunakan bootstrap, mari kita import cdn bootstrap dan taruhlah pada index.html:

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.3.1.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-U02eT0CpHqdSJK6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEeFF/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
```

```

4 <meta charset="utf-8">
5 <title>AngularObjectModel</title>
6 <base href="/">
7
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="icon" type="image/x-icon" href="favicon.ico">
10 <!--Bootstrap CDN-->
11 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-384-384-384" crossorigin="anonymous">
12 <script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha384-F8EsH2636IJDz80b70plBzfKAP1oYb+gkfmw3kfh263216t0E5Qk6jPclA+dCx" crossorigin="anonymous"></script>
13 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-Uo2eT0CpHqdSqQAShQ2Njy6677718" crossorigin="anonymous"></script>
14 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1pLxp64" crossorigin="anonymous"></script>
15 </head>
16 <body>
17 <nav class="navbar navbar-default navbar-dark bg-dark navbar-fixed-top">
18 <a class="navbar-brand" href="#">Angular Object Model Tutorial</a>
19 </nav>
20 <br/>
21 <app-root>
22 </body>
23 </html>

```

Bootstrap sudah kita import, langkah yang selanjutnya harus kita lakukan bisa mengisi html terlebih dahulu atau bisa langsung saja mengisi logic pada **app.component.ts**. (sesuai preferensi)

Ketiklah potongan kodingan html ini pada **app.component.html**

```

<div class="form-group">
  <label for="title">Title: </label>
  <input type="text" class="form-control" name="title" placeholder="Title" #title>
</div>
<div class="form-group">
  <label for="link">Link: </label>
  <input type="text" class="form-control" placeholder="Link" #link>
</div>
<button class="btn btn-primary" >Submit link</button>

```

Lihat Hasilnya :

Angular Object Model Tutorial

Title:

Link:

Tampilan sudah kita bentuk pada html, kita akan menambahkan atau menyisipkan container article pada article.component. Sebelum kita bisa melakukan menyisipkan mari terlebih dahulu kita bangun model untuk dijadikan object nantinya pada artcilenya. Untuk membangun model kalian bisa menambahkan file typescript baru buatan kita. Beri saja nama article.model :



Selanjutnya kita bangun modelnya :

```
export class Article {
  title: string; // untuk title article
  link: string; // untuk link pada article nantinya
  votes: number; // untuk jumlah vote pada article

  // kita bangun constructor pembentuk model, karena nantinya
  // article akan bisa bertambah ketika ditekan submit link
  constructor(title: string, link: string, votes?: number) {
    this.title = title;
    this.link = link;
    this.votes = votes || 0;
  }

  // article nantinya akan memiliki vote up dan vote down
  // seperti forum pada reddit
  voteUp(): void {
    this.votes += 1;
  }

  voteDown(): void {
    this.votes -= 1;
  }
}
```

Dan selesai kita telah berhasil membangun model dari article. Kalau ada yang bingung kenapa bentuknya begini, karena kita akan membentuk model dengan object, mau tidak mau hal ini berhubungan dengan ObjectOrientedProgramming. Bagi yang tidak mengerti dan penasaran ingin tahu lebih lanjut, pada pdf Week4 telah diberikan link untuk pengenalan dasar OOP pada javascript. Bedanya kita menerapkannya dalam bentuk typescript.

Model yang telah kita buat akan kita implementasikan pada article component. Untuk itu bukalah `article.component.ts` lalu ketiklah :

```

import {Component, HostBinding, Input, OnInit} from '@angular/core';
import { Article } from './article.model'; // panggil model yang telah kita buat

@Component({
  selector: 'app-article',
  templateUrl: './article.component.html',
  styleUrls: ['./article.component.css']
})
export class ArticleComponent implements OnInit {
  @HostBinding('attr.class') cssClass = 'row';
  @Input() article: Article;
  votes: number;
  title: string;
  link: string;

  constructor() {
  }

  ngOnInit() {
  }

  voteUp(): boolean { // untuk memanggil method voteUp pada model
    this.article.voteUp();
    return false;
  }

```

```

    voteDown(): boolean { // untuk memanggil method voteDown pada model
      this.article.voteDown();
      return false;
    }
  }
}

```

Selanjutnya ganti isi article component.html, untuk dijadikan component article yang nantinya akan kita panggil app component.

```

1 <div class="card" style="width: 18rem;">
2   <div class="card-body">
3     <!-- {} -> digunakan untuk binding data pada angular kali ini binding kita lakukan -->
4     <!-- Untuk mengeluarkan data dari model yang telah dibangun -->
5     <h5 class="card-title">{{article.votes}} Points</h5>
6     <h6 class="card-subtitle mb-2 text-muted" >{{article.domain()}}</h6>
7     <p class="card-text" >{{article.title}}</p>
8     <!-- href (click) salah satu metode pada angular7 -->
9     <!-- agar kita bisa melakukan click pada tag -->
10    <a href="#" class="card-link" href (click)="voteUp()" >UpVote</a>
11    <a href="#" class="card-link" href (click)="voteDown()" >DownVote</a>
12  </div>
13 </div>
14

```

Article sudah selesai kita bangun, sekarang kita tinggal mengisi app component saja, ketiklah potongan code ini pada app.component.ts

```
1  import { Component } from '@angular/core';
2  import { Article } from '../article/article.model'; // import juga model article
3  // yang telah dibangun,
4  // oh ya isi from pada import merupakan path karena article model ada di folder article maka
5  // kita harus masuk ke folder article dulu, baru ambil article.model ts
6
7  @Component({
8    selector: 'app-root',
9    templateUrl: './app.component.html',
10   styleUrls: ['./app.component.css']
11 })
12
13 export class AppComponent {
14   title = 'angular-object-model';
15   articles: Article[];
16
17   constructor() {
18     // kita deklarasi saja data awal pada constructor dengan membuat object article
19     this.articles = [ new Article( title: 'Angular 2',
20                               link: 'https://angular.io',
21                               votes: 10),
22                       new Article( title: 'FullStack', link: 'https://fullstack.io', votes: 2),
23                       new Article( title: 'Angular Homepage', link: 'https://angular.io', votes: 1),
24     ];
25   }
26
27   // untuk tambah article
28   addArticle(title: HTMLInputElement, link: HTMLInputElement ): boolean {
29     console.log(`Adding article title: ${title.value} and link: ${link.value}`);
30     this.articles.push(new Article(title.value, link.value, votes: 0));
31     title.value = '';
32     link.value = '';
33     return false;
34   }
35   // untuk mensort article
36   sortedArticle(): Article[] {
37     return this.articles.sort( compareFn: (a: Article, b: Article) => b.votes - a.votes);
38   }
```

Langkah terakhir jangan lupa kita update app.component.html agar data bisa ditambah dan di sort, sekaligus memanggil component article :

```

1      <!--The content below is only a placeholder and can be replaced.-->
2
3      <div class="form-group">
4          <label for="title">Title: </label>
5          <input type="text" class="form-control" name="title" placeholder="Title" #title>
6      </div>
7      <div class="form-group">
8          <label for="link">Link: </label>
9          <input type="text" class="form-control" placeholder="Link" #link>
10     </div>
11     <button class="btn btn-primary" (click)="addArticle(title, link)">Submit link</button>
12
13     |
14     <app-article *ngFor="let article of sortedArticle()" [article]="article"></app-article>
15

```

Dan pada akhirnya web app kalian jadi, silahkan dicoba :

Title:

Link:

Title:

Link:

Hasil :

10 Points

angular.io

Angular 2

[UpVote](#) [DownVote](#)

2 Points

fullstack.io

FullStack

[UpVote](#) [DownVote](#)

1 Points

angular.io

Angular Homepage

[UpVote](#) [DownVote](#)

0 Points

www.google.com

Justin Susanto

[UpVote](#) [DownVote](#)

Coba kalian melakukan upvote sampai 2, sort akan berjalan otomatis sehingga posisi menjadi :

10 Points

angular.io

Angular 2

[UpVote](#) [DownVote](#)

3 Points

www.google.com

Justin Susanto

[UpVote](#) [DownVote](#)

2 Points

fullstack.io

FullStack

[UpVote](#) [DownVote](#)

Sebelum kita melanjutkan ke untuk mempelajari materi utama kita yaitu angular form ada yang perlu kalian ketahui, yaitu codingan yang kita lakukan pada tutorial angular model with object adalah cara mengcode secara benar dan sesuai dengan kaedah Teknik model view dan controller (MVC), kita tidak akan membahas ini pada mata perkuliahan IF300 tetapi hal yang perlu diingat ini adalah cara coding yang benar dan di code secara *clean code* pada penerapan type scriptnya. Sulit dipahami ? ya memang tetapi code tersebut

akan reusable dan memiliki cohesion yang baik (code yang sesuai *proper use*). Sekarang untuk tutorial angular form selanjutnya, kita akan melakukan code secara simple dan memang merupakan code object secara simple dan bukan memang bukan penggunaan secara *Best Practice*.

Week 12 IF 300 Tutorial 2 – Angular Form

Easier than tutorial 1

Di tutorial 2 kali ini kita akan menggunakan cara mengimport bootstrap yang lebih keren, tidak manual seperti tutorial 1. Pertama kita buat project angular baru menggunakan ng new. Dan masuk ke foldernya lalu ketik. Npm install bootstrap –save

\$ npm install bootstrap --save

Jika sukses akan muncul tampilan yang kurang lebih sebagai berikut:

```
+ bootstrap@4.3.1
added 1 package from 2 contributors and audited 42609 packages in 32.71s
found 0 vulnerabilities
```

Jika telah sukses selanjutnya kita akan panggil bootstrap yang telah terinstall ke node_module dengan mengconfigurasi file angular.json. Pada bagian styles, tambahkan / kita import bootstrapnya :

```
"styles": [
  "src/styles.css",
  "node_modules/bootstrap/dist/css/bootstrap.css"
],
```

Sekarang kita akan mempelajari form, pada angular terdiri dari 3 jenis form reactive forms, template driven forms, dan dynamic forms. Yang populer adalah reactive form, jika template driven form merupakan form dimana kita membuat logic, validasi, control, dll secara template bagian dari code html. Kita akan pakai reactive forms. Untuk itu silahkan import reactive forms dari app.module.ts dan common module yang nantinya akan kita pakai.

```

1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { ReactiveFormsModule } from '@angular/forms'; // import module reactive form
6 import { DataComponent } from './data/data.component';
7 import { CommonModule } from '@angular/common'; // taruhlah juga common module agar angular directive
8 // bisa dibaca oleh tag normal html seperti div, dll
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     DataComponent
14   ],
15   imports: [
16     BrowserModule,
17     ReactiveFormsModule, // taruh reactive form yang mau diimport
18     CommonModule // taruh commonmodule yang mau diimport
19   ],
20   providers: [],
21   bootstrap: [AppComponent]
22 })

```

Tulis html layout pada app.component.html

```

<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link" href="#" (click)="tab=1">Forms</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#" (click)="tab=2">Data</a>
  </li>
</ul>

<div *ngIf="tab==1" label="Tab 1">
<div class="container">
  <div class="form-group">
    <label>First Name: </label>
    <input type="text" [formControl]="fname" class="form-control" />
  </div>
  <div class="form-group">
    <label>Last Name: </label>
    <input type="text" [formControl]="lname" class="form-control" />
  </div>
  <div class="form-group">
    <label>NIM : </label>
    <input type="text" [formControl]="nim" class="form-control" />
  </div>
  <div class="form-group">
    <label>Email:</label>
    <input type="email" [formControl]="email" class="form-control" />
  </div>
</div>

```

```

<div class="form-group">
  <label>Password:</label>
  <input type="password" [formControl]="password" class="form-control" />
</div>
<div class="form-group">
  <label>Confirm Password:</label>
  <input type="password" [formControl]="cpassword" class="form-control" />
</div>
<div class="form-group">
  <button (click)="setData()" class="btn btn-dark form-control">SetData</button>
</div>
<p>
  Halo nama saya {{fname.value}} {{lname.value}} nim saya {{nim.value}}.
  Anda bisa contact saya dengan email {{email.value}}
</p>
</div>
</div>
<div *ngIf="tab==2" label="Tab 2">
  <app-data></app-data>
</div>

```

Abaikan kolom password terlebih dahulu.

Isi logic pada app.component.ts.

```

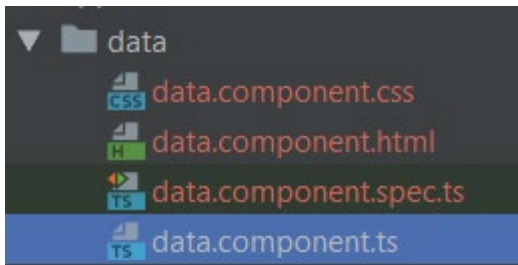
import { Component } from '@angular/core';
import { FormControl } from '@angular/forms';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'angular-forms';
  tab = 1;
  fname = new FormControl( formState: '' ); // buat ambil value dari field pada form
  email = new FormControl( formState: '' );
  lname = new FormControl( formState: '' );
  nim = new FormControl( formState: '' );

  setData() {
    alert('Data has been set');
    localStorage.setItem('fname', this.fname.value);
    localStorage.setItem('lname', this.lname.value);
    localStorage.setItem('email', this.email.value);
    localStorage.setItem('nim', this.nim.value);
  }
}

```

Sekarang generate komponen data



Ketiklah isi layout data.component.html

```
<p>
  First Name : {{fname}} <br/>
  Last Name  : {{lname}} <br/>
  NIM        : {{nim}} <br/>
  Email      : {{email}} <br/>
</p>
```

Logic dari data component :

```
import { Component, OnInit } from '@angular/core'

@Component({
  selector: 'app-data',
  templateUrl: './data.component.html',
  styleUrls: ['./data.component.css']
})
export class DataComponent implements OnInit {

  constructor() { }

  fname = localStorage.getItem( key: 'fname' );
  lname = localStorage.getItem( key: 'lname' );
  nim = localStorage.getItem( key: 'nim' );
  email = localStorage.getItem( key: 'email' );

  ngOnInit() {
  }
}
```

Dan form bagian awal sudah jadi.., dan silahkan dicoba pakai ng serve !

Forms Data

First Name:

asd

Last Name:

bcd

NIM :

asd.bcd@student.umn.ac.id

Email:

asd.bcd.student.umn.ac.id

Password:

Confirm Password:

SetData

Halo nama saya asd bcd nim saya asd.bcd@student.umn.ac.id. Anda bisa contact saya dengan email asd.bcd.student.umn.ac.id

Pada tab data :

Forms Data

First Name : asd

Last Name : bcd

NIM : asd.bcd@student.umn.ac.id

Email : asd.bcd.student.umn.ac.id

Namun form ini belum kita beri validasi untuk itu, kita harus beri validasi. Dari codingan yang tadi kita tulis, kita akan banyak memodifikasi. (modifikasi hanya pada app component, data component, dibiarkan saja)

```
<div *ngIf="tab==1" label="Tab 1">
  <div class="container">
    <form [formGroup]="dataForm" (ngSubmit)="onSubmit()">
      <div class="form-group">
        <label>First Name: </label>
        <input type="text" formControlName="fname" class="form-control"
          [ngClass]="{ 'is-invalid': submitted && f.fname.errors }" />
        <div *ngIf="submitted && f.fname.errors" class="invalid-feedback">
          <div *ngIf="f.fname.errors.required">
            First name field cannot be empty
          </div>
        </div>
      </div>
      <div class="form-group">
        <label>Last Name: </label>
        <input type="text" formControlName="lname" class="form-control"
          [ngClass]="{ 'is-invalid': submitted && f.lname.errors }" />
        <div *ngIf="submitted && f.lname.errors" class="invalid-feedback">
          <div *ngIf="f.lname.errors.required">
            Last name field cannot be empty
          </div>
        </div>
      </div>
      <div class="form-group">
        <label>NIM : </label>
        <input type="text" formControlName="nim" class="form-control"
          [ngClass]="{ 'is-invalid': submitted && f.nim.errors }" />
      </div>
    </form>
  </div>
</div>
```

```

        [ngClass]="{ 'is-invalid': submitted && f.email.errors }" />
      <div *ngIf="submitted && f.email.errors" class="invalid-feedback">
        <div *ngIf="f.email.errors.required">
          Email field cannot be empty
        </div>
        <div *ngIf="f.email.errors.email">
          Must Enter valid email
        </div>
      </div>
    </div>
    <div class="form-group">
      <label>Password:</label>
      <input type="password" formControlName="password" class="form-control"
        [ngClass]="{ 'is-invalid': submitted && f.password.errors }"/>
      <div *ngIf="submitted && f.password.errors" class="invalid-feedback">
        <div *ngIf="f.password.errors.required">
          Password field cannot be empty
        </div>
        <div *ngIf="f.password.errors.minlength">
          The minimum passwords length is 8
        </div>
      </div>
    </div>
    <div class="form-group">
      <label>Confirm Password:</label>
      <input type="password" formControlName="cpassword" class="form-control"
        [ngClass]="{ 'is-invalid': submitted && f.cpassword.errors }"/>

```

```

      <div *ngIf="submitted && f.cpassword.errors" class="invalid-feedback">
        <div *ngIf="f.cpassword.errors.required">
          Confirm Password field cannot be empty
        </div>
        <div *ngIf="f.cpassword.errors.mustMatch">
          Confirm Password must match Password
        </div>
      </div>
    </div>
    <div class="form-group">
      <button class="btn btn-dark form-control">SetData</button>
    </div>
  </form>
<p>
  Halo nama saya {{f.fname.value}} {{f.lname.value}} nim saya {{f.nim.value}}.
  Anda bisa contact saya dengan email {{f.email.value}}
</p>
</div>
</div>
<div *ngIf="tab==2" label="Tab 2">
  <app-data></app-data>
</div>

```



```

import {Component, OnInit} from '@angular/core';
import {Form, FormBuilder, FormGroup, Validators} from '@angular/forms';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  title = 'angular-forms';
  dataForm: FormGroup; // deklarasi form group
  submitted = false;
  check = false;
  tab = 1;

  constructor(private formBuilder: FormBuilder) { } // parameter constructor ini akan digunakan untuk get

  ngOnInit(): void { // disini kita akan deklarasi data form, kita taruh input field di html di sini
    this.dataForm = this.formBuilder.group( controlsConfig: {
      fname: ['', Validators.required],
      lname: ['', Validators.required],
      nim: ['', Validators.required],
      email: ['', [Validators.required, Validators.email]],
      password: ['', [Validators.required, Validators.minLength( 8)]],
      cpassword: ['', [Validators.required]]
    }, options: {validator: this.checkPasswords} );
  }

```

```

checkPasswords(fg: FormGroup) { // untuk check password dan confirm password, sama atau tidak
  const pass = fg.controls.password.value;
  const cpass = fg.controls.cpassword.value;
  // console.log()
  const control = fg.controls.password;
  const matchcontrol = fg.controls.cpassword;
  if (pass === cpass) {
    matchcontrol.setErrors( errors: null);
    return true;
  } else {
    matchcontrol.setErrors( errors: {mustMatch: true});
    return false;
  }
}

get f() { // getter untuk ambil value dari data form
  return this.dataForm.controls;
}

onSubmit() {
  this.submitted = true;
  if (this.dataForm.invalid) {
    alert('Input data error');
    return;
  } else {
    alert('Data has been set');
    localStorage.setItem('fname', this.f.fname.value);
  }
}

```

Keterangan : pada get f(), ini merupakan metode getter, sebenarnya ini adalah bagian dari OOP, bagi kalian yang belum dapat materi OOP getter ini harus digunakan karena kita akan mengambil value dari object formGroup, hal ini menyebabkan jika kita coba akses secara langsung pakai, this.dataForm.constrols.namafield akan error, sehingga kita terpaksa pakai metode getter, untuk mempermudah mendapatkan valuenya. Jika yang kalian tulis berhasil hasilnya akan seperti ini.

First Name:

justin

Last Name:

susanto

NIM :

00000013613

Email:

asdf@ghjkl.com

Password:

●●●●●●●●●●

Confirm Password:

●●●●●●●●



Confirm Password must match Password

SetData

Halo nama saya justin susanto nim saya 00000013613. Anda bisa contact saya dengan email asdf@ghjkl.com

Dan jika set data ditekan ketika ada yang salah, maka akan muncul alert berupa invalid data input !

Reference

1. Nate Murray, Felipe Coury, Ari Lerner, and Carlos Taborda(2018). ng-book The Complete Guide to Angular. Fullstack.io
2. Angular.io
3. <https://appdividend.com>
4. <https://www.codingame.com/>