MODUL 10
ANGULAR I

DESKRIPSI TEMA

Pada mulai week ini kita akan belajar membuat web dengan sebuah framework Angular. Angular sendiri memiliki pondasi dasar yaitu Typescript, yang merupakan superset dari ES6. Yang berarti Typescript sendiri bukan bahasa yang baru, melainkan pengembangan dari Javascript juga, hanya saja dengan syntax yang lebih terstruktur dan rapi.

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Setelah mempelajari ini, mahasiswa diharapkan:

1. Mengenal dan dapat menggunakan Angular-CLI dengan baik
2. Dapat menggunakan Angular-CLI untuk membuat component dalam project sederhana
3. Dapat memahami konvensi penulisan Angular dengan baik.

PENUNJANG PRAKTIKUM

1. Node.js
2. Browser (Google Chrome, etc.)
3. Text Editor (Visual Studio Code, Sublime, etc.)

LANGKAH-LANGKAH PRAKTIKUM

Pada Instalasi Angular-CLI ditawarkan 2 langkah:

a. Instalasi Mudah

b. Instalasi Manual

Instalasi Mudah Angular-CLI

Langkah yang digunakan untuk instalasi mudah Angular-CLI:

1. Extract file pendukung
2. Jalankan program runMeFirst.bat

Instalasi Manual Angular-CLI

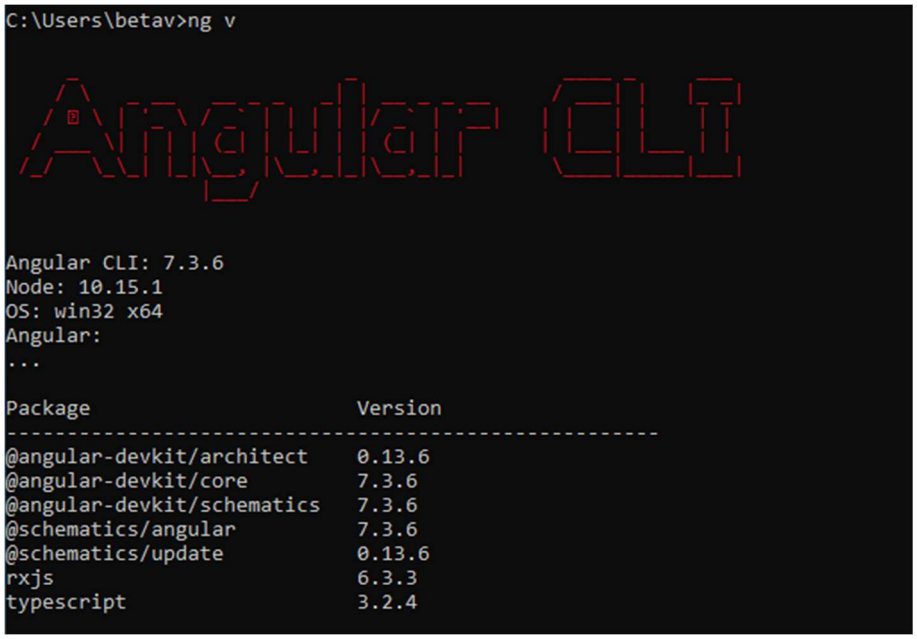
Langkah yang digunakan untuk instalasi manual Angular-CLI:

1. Bukalah https://nodejs.org
2. Download versi LTS (versi saat ini 10.15.3)
3. Jalankan Installer Node yang telah di download.
4. Bacalah instruksi yang telah disediakan pada Wizard Installer NodeJS, serta pastikan alamat instalasi sesuai yang diinginkan.
5. Bukalah Command Prompt / cmd.

6. Jalankan perintah berikut yang digunakan untuk melakukan instalasi Angular pada komputer anda.
   - $ npm install –g @angular/cli

Mulai Menggunakan Angular-CLI

1. Bukalah Command Prompt / cmd.
2. Setelah itu jalankan perintah berikut:
   i. Perintah berikut digunakan untuk melakukan instalasi Angular pada komputer anda.
   $ npm install –g @angular/cli
   ii. Berikutnya, untuk dapat memastikan apakah Angular sudah terinstall dengan baik lakukan pengecekan dengan menggunakan salah satu perintah berikut.
   $ ng version atau $ ng v



Untuk dapat melihat perintah – perintah yand dapat digunakan pada Angular CLI, dapat dilihat di *https://angular.io/cli*

First Angular Project

1. Untuk memulai program Angular pertama anda, bukalah Command Prompt. Lalu arahkan pada folder dimana anda ingin melakukan pengerjaan anda, lalu jalankanlah perintah berikut:
   $ ng new first-angular-project
   Lalu kita akan ditampilkan dengan beberapa pilihan
   Would you like to add Angular routing? No
   Which stylesheet format would you like to use? CSS
2. Bukalah index.html untuk pengerjaan kita pada folder berikut:

first-angular-project/src/index.html

Isi index.html kita hanya berisi berikut, dikarenakan Angular merupakan pohon bagi component – componentnya, yang berarti index adalah root nya.

```
<body>
  <app-root></app-root>
</body>
```

3. Bukalah Command Prompt dan arahkan pada folder pengerjaan tadi lalu, untuk dapat menjalankan project Angular yang telah kita buat tadi, dengan menggunakan command:

   $ ng serve

   Maka nantinya pada secara default akan menggunakan localhost:4200 agar dapat berjalan dengan baik.

Component

1. Pada tahap ini, kita akan mempelajari tentang bagaimana Angular berjalan dengan component – componentnya. Bukalah Command Prompt dan jalankan perintah berikut untuk membuat sebuah component pada folder pengerjaanmu.

   $ ng generate component hello-pti

2. Bukalah file component dari app yang bisa didapatkan dari file berikut.

   first-angular-project/src/app/app.component.html

   Rubahlah isi dari app.component.html menjadi seperti berikut, untuk menambahkan directive app-hello-pti

```
<h1>
  {{title}}

  <app-hello-pti></app-hello-pti>
</h1>
```

Maka component hello-pti akan terpanggil dan akan menampilkan sebagai tampilan berikut:

first-angular-project

hello-pti works!

Adding Data to the Component

1. Bukalah cmd dan buatlah component baru.

   $ ng generate component user-item

2. Setelah itu bukalah file pada app.component.html kita panggil '*app-user-item'*, pemanggilan disini digunakan untuk memanggil yang terdapat pada user-item. Lalu kita masuk ke file berikut. **first-angular-project/src/app/user-item/user-item.component.ts**

   Ubahlah isi dari UserItemComponent buat seperti contoh berikut:

```
export class UserItemComponent implements OnInit {
  name: string;
  nim: string;

  constructor() {
    this.name = 'Benedictus Betavian'; // masukkan nama kamu.
    this.nim = '00000016862'; // masukkan nim kamu.
  }
```

Adding Array of Data to the Component

1. Bukalah Command Prompt, lalu jalankan perintah berikut

   $ ng generate component mhs

   Maka akan membuat komponen dengan nama mhs, setelah itu, panggilah component tersebut pada app.component.html,

2. Bukalah file berikut:

   mhs/src/app/mhs/mhs.component.ts

3. Siapkan data array yang kita deklarasikan dan pada class dan menyiapkan isinya pada bagian constructor.

```
constructor() {
  this.nama = ['Mika', 'Rafa', 'Yosh', 'Gaby'];
}
```

4. Panggillah semua data tersebut menggunakan directive *ngFor, seperti pada langkah berikut.

```
<ul>
  <li *ngFor="let n of nama">Hello {{ n }} </li>
</ul>
```

Maka nantinya akan menampilkan tampilan seperti berikut:

Two-Way Data Binding

Pada Angular memungkinkan adanya Two-way Data Binding, yang berarti apabila ada perubahan pada data yang pertama, maka data variable dari file lainnya akan masuk.

1. Bukalah cmd dan buatlah component dengan menjalankan perintah berikut:

   $ ng generate component test-ngmodel

2. Bukalah ke dalam component test-ngmodel.component.html, dan buat seperti berikut

```
1   <h1> Two-Way Data Binding</h1>
2   <input type="text" value="msi" [(ngModel)]="product" />
3   <p>{{ product }}</p>
```

3. Tambahkanlah variable "product" yang nantinya dapat kita atur pada export di test-ngmodel.component.ts

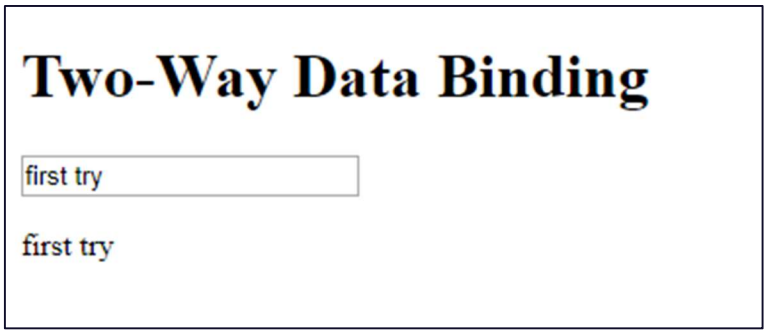
```
export class TestNgmodelComponent implements OnInit {
  product: string;

  constructor() {

  }

  ngOnInit() {
  }

}
```

4. Untuk dapat menggunakan ngModel, tambahkan module Forms, dengan import pada decorator (NgModule), seperti contoh berikut

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
```

```
@NgModule({
  declarations: [
    AppComponent,
    HelloPtiComponent,
    UserItemComponent,
    MhsComponent,
    TestNgmodelComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Setelah itu maka hasilnya yang didapat kan akan sebagai berikut



Single Page Application

Pada tahap ini, kita akan mencoba untuk membuat aplikasi ToDo sederhana yang sudah menerapkan Single Page Application (SPA).

1. Buatlah sebuah project baru dengan nama *project-todos*, dengan tidak menggunakan Routing dan styling dengan CSS.

   $ ng new project-todos

2. Buatlah component *todos* nya dengan

   $ ng generate component todos

3. Pindahkan isi dari file pendukung ke dalam file:

   project-todos/src/app/todos/todos.component.css

4. Buatlah class *todos* dengan

   $ ng generate class todos

5. Bukalah file berikut:

   project-todos/src/app/app.module.ts

6. Lakukan import FormsModule, dan CommonModule, seperti berikut:

```
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';
import { NgModule } from '@angular/core';
```

7. Pada decorator tambahkan FormsModule dan CommonModule pada imports:

```
@NgModule({
  declarations: [
    AppComponent,
    TodosComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    CommonModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

8. Tambahkan directive app-todos pada app.component.html, yang terdapat pada

   project-todos/src/app/app.component.html

```
app.component.html ×
1   <app-todos></app-todos>
2
```

9. Tambahkan variable 'name', dan 'done' pada class todos.ts, yang terdapat pada

   project-todos/src/app/todos.ts

```
TS todos.ts ×
1   export class Todos {
2       name: string;
3       done: boolean;
4   }
5
```

10. Bukalah **todos.component.ts** dan ubahlah agar menjadi seperti berikut:

    project-todos/src/app/todos/todos.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Todos } from '../todos';

@Component({
  selector: 'app-todos',
  templateUrl: './todos.component.html',
  styleUrls: ['./todos.component.css']
})
export class TodosComponent implements OnInit {
  newTodo: string = '';
  todoList: Array<Todos> = [];
  constructor() { }
  ngOnInit() { }
  addTodo = function () {
    if (this.newTodo == '') {
      alert("Todo must not be empty!");
      return false;
    }
    for (let x in this.todoList) {
      if (this.newTodo == this.todoList[x].name) {
        alert("Task already in the list!");
        return false;
      }
    }
```

```
    this.todoList.push({
      name: this.newTodo,
      done: false
    })
    this.newTodo = ''
    console.log(this.todoList);
  }
  isEmpty = function () {
    if (this.todoList.length > 0) {
      return false;
    }
    return true;
  }
  changeStatus = function (x: Todos) {
    x.done = !(x.done);
  }
  delete = function (x: Todos) {
    this.todoList.splice(this.todoList.indexOf(x), 1);
  }
}
```

11. Ubahlah isi **todos.component.html** seperti berikut, yang terdapat pada:

project-todos/src/app/todos/todos.component.html

```
<div class="w3-container w3-half w3-display-topmiddle w3-margin-top">
  <div class="w3-container w3-blue">
    <h2>New Todo</h2>
  </div>
  <form class="w3-container w3-border">
    <p>
      <input [(ngModel)]="newTodo" name="newTodo" placeholder="Todo..." class="w3-input" type="text">
    </p>
    <p>
      <button (click)="addTodo()" class="w3-btn w3-blue w3-large">
        Register
      </button>
    </p>
  </form>
  <div class="w3-container w3-margin-top w3-blue">
    <h2>Todo List</h2>
  </div>
  <div class="w3-container w3-border w3-padding">
    <div [hidden]="!isEmpty()">
      <i>Nothing to do?</i>
    </div>
    <div *ngFor="let todo of todoList">
      <p [ngClass]="todo.done ? 'done':''" style="width:95%;float:left" (click)="changeStatus(todo)">
        {{todo.name}}
      </p>
      <p style="width:5%;float:left" (click)="delete(todo)">x</p>
    </div>
  </div>
</div>
```

12. Jalankan aplikasi dan lihatlah hasilnya.

REFERENSI