



Front End Development 1

Sesi 23



Angular Modules

+

Angular Modules (NgModules)

NgModules berfungsi untuk mengkonfigurasi injector/ library dan compiler yang siap dipakai dalam Aplikasi Angular. Merupakan suatu class yang ditandai dengan decorator **@NgModule**.

Library-Library Angular termasuk dalam NgModules, seperti [FormsModule](#), [HttpClientModule](#), dan [RouterModule](#). Beberapa library 3rd party lainnya juga disediakan sebagai NgModules seperti [Material Design](#), [Ionic](#), and [AngularFire2](#).

Hal-hal yang dilakukan NgModule metadata antara lain:

- Mendeklarasikan mana yang component, directives dan pipes dalam suatu module
- Membuat component, directives dan pipes dapat diakses dan digunakan oleh module komponen lainnya
- Meng-import modules yang mengandung komponen, directives dan pipes di komponen module lain yang membutuhkan.
- Menyediakan service yang komponen lain dapat gunakan

Application Root module

```
src > app > app.module.ts > ...
1  /* JavaScript imports */
2  import { BrowserModule } from '@angular/platform-browser';
3  import { NgModule } from '@angular/core';
4
5  import { AppComponent } from './app.component';
6
7  /* the AppModule class with the @NgModule decorator */
8  @NgModule({
9    declarations: [
10     AppComponent
11   ],
12   imports: [
13     BrowserModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }
19
```

App.module.ts merupakan modul root dan di-buat ketika pertama kali men-generate aplikasi Angular dengan CLI ng new.

Dimulai dengan statement import untuk meng-import module-module Javascript. Kemudian dikonfigurasi dalam **@NgModules** dengan penjelasan:

1. **declarations:** berisi component, directives dan pipes. App.module pertama kali akan memiliki 1 komponen yang bernama AppComponent.
2. **imports:** Berisi NgModules lainnya. AppModule yang pertama kali di-generate akan memiliki BrowserModule yang dapat digunakan sebagai service browser seperti directives umum, DOM rendering, sanitasi, location, dll. BrowserModule hanya diimport sekali di root module.
3. **providers:** Berisikan service yang dapat digunakan oleh komponen. Tidak ada service yang disediakan pada app.module.ts yang baru di generate.
4. **bootstrap:** Entry component yang Angular buat dan masukkan kedalam index.html, untuk bootstrapping aplikasi. AppComponent juga masuk kedalam sini.

Frequently-used modules

Beberapa Angular Modules yang sering digunakan antara lain:

NgModule	Import it from	Why you use it
<code>BrowserModule</code>	<code>@angular/platform-browser</code>	When you want to run your application in a browser
<code>CommonModule</code>	<code>@angular/common</code>	When you want to use <code>NgIf</code> , <code>NgFor</code>
<code>FormsModule</code>	<code>@angular/forms</code>	When you want to build template driven forms (includes <code>NgModel</code>)
<code>ReactiveFormsModule</code>	<code>@angular/forms</code>	When you want to build reactive forms
<code>RouterModule</code>	<code>@angular/router</code>	When you want to use <code>RouterLink</code> , <code>.forRoot()</code> , and <code>.forChild()</code>
<code>HttpClientModule</code>	<code>@angular/common/http</code>	When you want to talk to a server

Feature modules: Generate Module

Ketika aplikasi membesar, butuh mengorganisir kode agar relevan dengan fitur yang spesifik. (Separation of Concern). Dengan feature modules, kita bisa meletakkan/mengategorikan kode berdasar fitur. Untuk mengenerate feature module, ketikkan CLI dibawah ini: `ng generate module <nama_module_baru>`

```
ng generate module CustomerDashboard
```

```
customer-dashboard.module.ts U x
src > app > customer-dashboard > customer-dashboard.mod
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3
4
5
6  @NgModule({
7    declarations: [],
8    imports: [
9      CommonModule
10   ]
11 })
12 export class CustomerDashboardModule { }
```

Struktur NgModule akan sama saja, baik sebagai root module atau feature module. Penjelasan Module:

- Gunakan decorator **@NgModule**
- Imports **CommonModule**, agar dapat menggunakan directives umum seperti ngIf atau ngFor.

Feature modules: Generate Module Component

Suatu module, juga bisa memiliki component, directives dan pipes. Dan semua ini hanya akan dimiliki oleh module yang meng-importnya.

Dibawah ini merupakan cara mengenerate komponen kedalam folder module yang sebelumnya sudah digenerate:

```
ng generate component customer-dashboard/CustomerDashboard
```

```
customer-dashboard.module.ts U x
src > app > customer-dashboard > customer-dashboard.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { CustomerDashboardComponent } from './customer-dashboard/customer-dashboard.component';
4
5
6
7  @NgModule({
8    declarations: [
9      CustomerDashboardComponent
10   ],
11   imports: [
12     CommonModule
13   ]
14 })
15 export class CustomerDashboardModule { }
```

Feature modules: Import to root

Agar feature module yang dibuat dapat dipakai dalam aplikasi Angular, maka kita perlu meng-import module ini kedalam root module (app.module.ts). Import module kedalam key “imports” dalam @NgModule.

```
app.module.ts X
1  import { HttpClientModule } from '@angular/common/http';
2  import { NgModule } from '@angular/core';
3  import { FormsModule } from '@angular/forms';
4  import { BrowserModule } from '@angular/platform-browser';
5
6  import { AppComponent } from './app.component';
7  // import the feature module here so you can add it to the imports array
   below
8  import { CustomerDashboardModule } from
   './customer-dashboard/customer-dashboard.module';
9
10 @NgModule({
11   declarations: [
12     AppComponent
13   ],
14   imports: [
15     BrowserModule,
16     FormsModule,
17     HttpClientModule,
18     CustomerDashboardModule // add the feature module here
19   ],
20   providers: [],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
```

