Modul Praktikum Minggu 2

Pemrograman Mobile Cross Platform

Ionic Framework with Angular (1)

OVERVIEW

Finishing this module, students will be able to:

- Understand how to install and setup development environment for Ionic 4 with Angular
- Able to start an Ionic project
- Create and use pages and services component in Ionic & Angular

This module is divided into several parts. OVERVIEW explains the overall objective of this module and how to understand its contents. MATERIALS NEEDED and SOFTWARE REQUIREMENTS shows the requirements. ACTIVITIES section explains how to do various things, as explained in class. Finally, the TASK describes the tasks that students need to finish and get graded in the corresponding module.

MATERIALS NEEDED / REFERENCES

- This module
- Materials from the class (PowerPoint slides, etc.) – *dari kelas teori*
- Ionic Framework Official Documentation

SOFTWARE REQUIREMENTS

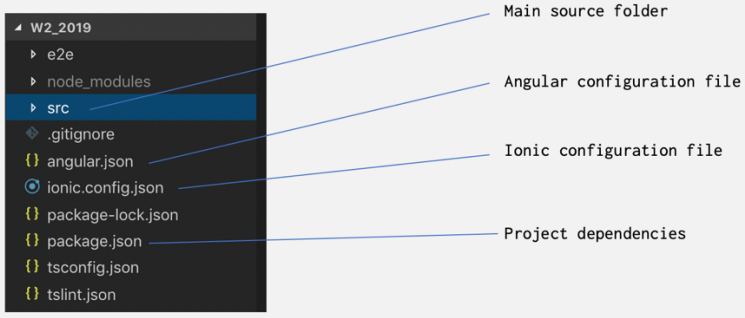
- Visual Studio Code
- Google Chrome

ACTIVITIES

Installing Ionic 4 with Angular & Starting a New Project

Installation

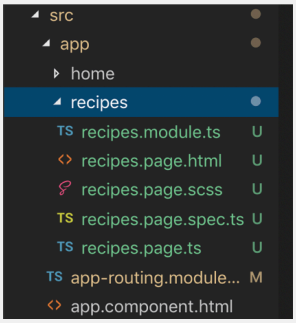
1. Download and install NodeJS LTS/Dubnium v10.xx
   → NPM, development server
2. Install Ionic CLI using NPM via terminal by running:
   npm install -g ionic (use **sudo** in mac/linux)
3. Navigate to your intended project directory, then run: **ionic start**, then type your project name, then choose the template (we'll use blank in this course), then type **n** when prompted with Ionic Appflow SDK to reject using it for now.
4. Open the project directory using VSCode, open the built-in terminal using shortcut CTRL+` (backtick), then run: **ionic serve**



Generating a Page

1. Run: **ionic generate**, then choose **page** then set the page name
2. Check the **src/app/app-routing.module.ts** and modify as needed

```
? What would you like to generate? page
? Name/path of page: recipes
> ng generate page recipes
CREATE src/app/recipes/recipes.module.ts (548 bytes)
CREATE src/app/recipes/recipes.page.scss (0 bytes)
CREATE src/app/recipes/recipes.page.html (126 bytes)
CREATE src/app/recipes/recipes.page.spec.ts (698 bytes)
CREATE src/app/recipes/recipes.page.ts (260 bytes)
UPDATE src/app/app-routing.module.ts (525 bytes)
[OK] Generated page!
```



app-routing.module.ts

src > app > app-routing.module.ts > AppRoutingModule

```ts
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  { path: '', redirectTo: 'recipes', pathMatch: 'full' },
  { path: 'recipes', loadChildren: './recipes/recipes.module#RecipesPageModule' },
];

@NgModule({
```

Creating and Using Interface

src/app/recipes/recipe.model.ts

```ts
export interface Recipe {
    id: string;
    title: string;
    imageUrl: string;
    ingredients: string[];
}
```

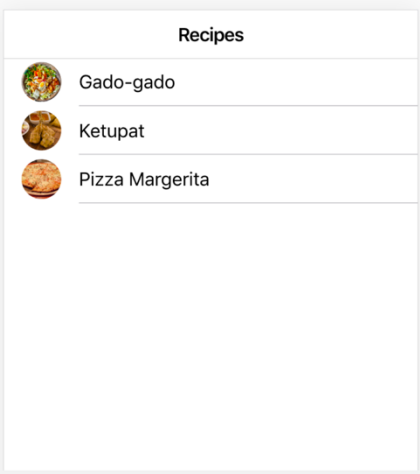
src/app/recipes/recipes.page.ts

```ts
import { Recipe } from './recipe.model';

...

export class RecipesPage implements OnInit {

recipes: Recipe[] = [
    {
        id: 'r1',
        title: 'Gado-gado',
        imageUrl:
        'https://www.bbcgoodfood.com/sites/default/files/styles/recipe/public/recipe/recipe-image/2016/05/gado-gado-salad.jpg?itok=MTTSriC8',
        ingredients: ['Lontong', 'Sawi', 'Bumbu Kecap', 'Tauge']
    },

    ...

];
```

Data Binding and <ion-avatar>

src/app/recipes/recipes.page.html

```html
<ion-content>
    <ion-list>
        <ion-item *ngFor="let recipe of recipes">
            <ion-avatar slot="start">
                <ion-img [src]="recipe.imageUrl"></ion-img>
            </ion-avatar>
            <ion-label>{{ recipe.title }}</ion-label>
        </ion-item>
    </ion-list>
</ion-content>
```

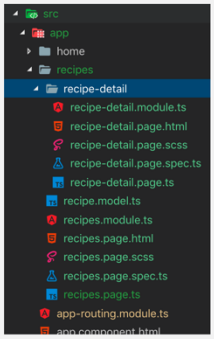


Create a New Page and Setup the Routing Module

- Generate a new page under **recipes**, name it **recipe-detail**
    - ionic generate page recipes/recipe-detail
- Setup the routing module as explained below

src/app/app-routing.module.ts

```
const routes: Routes = [
    { path: '', redirectTo: 'recipes', pathMatch: 'full' },
    { path: 'recipes', loadChildren:
    './recipes/recipes.module#RecipesPageModule' },
    { path: 'recipe-detail', loadChildren: './recipes/recipe-
    detail/recipe-detail.module#RecipeDetailPageModule' },
];
```



src/app/app-routing.module.ts

```
const routes: Routes = [
    { path: '', redirectTo: 'recipes', pathMatch: 'full' },
    {
        path: 'recipes',
        children: [
            {
                path: '',
                loadChildren: './recipes/recipes.module#RecipesPageModule'
            },
            {
                : './recipes/recipe-detail/recipe-detailpath: ':recipeId',
                loadChildren.module#RecipeDetailPageModule'
            }
]}];
```

Creating and Using Service for Managing States

```
> ionic generate service recipes/recipes
```

```
import { Injectable } from '@angular/core';
import { Recipe } from './recipe.model';

@Injectable({
    providedIn: 'root'
})

export class RecipesService {
    private recipes: Recipe[] = [
        {
            id: 'r1',
            title: 'Gado-gado',
            imageUrl:
            'https://www.bbcgoodfood.com/sites/default/files
            /styles/recipe/public/recipe/recipe-
            image/2016/05/gado-gado-
            salad.jpg?itok=MTTSriC8',
            ingredients: ['Lontong', 'Sawi', 'Bumbu Kecap',
            'Tauge']
        },
        ...
    ];
```

```
    constructor() { }

    getAllRecipes() {
        return [...this.recipes];}

    getRecipe(recipeId: string) {




    }
}
```

src/app/recipes/recipes.service.ts

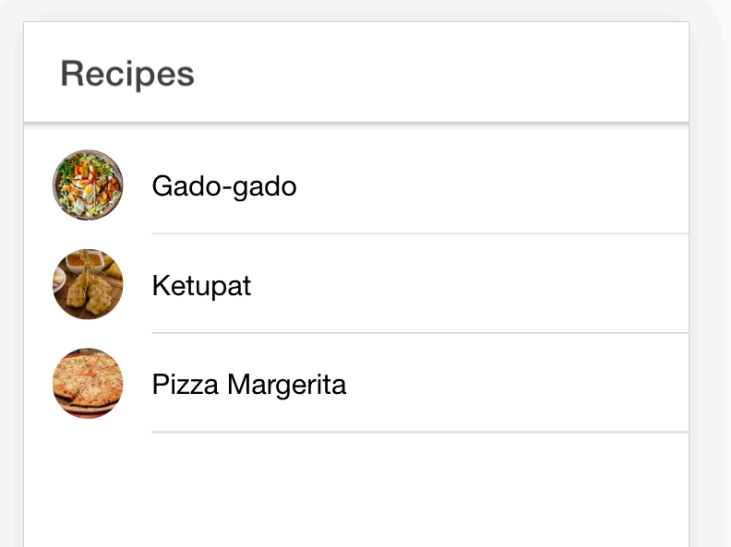
Then, use the service.

src/app/recipes/recipes.page.ts

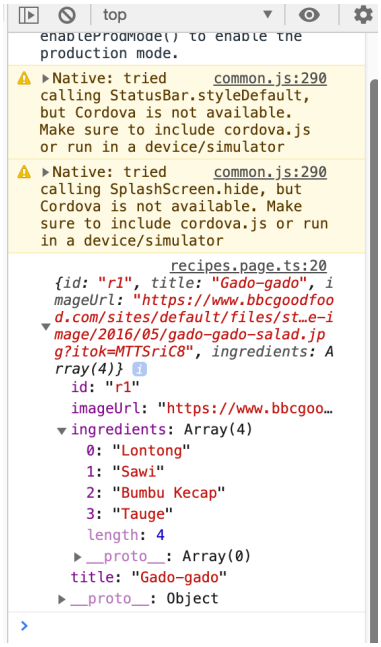
```
import { Component, OnInit } from '@angular/core';
import { Recipe } from './recipe.model';
import { RecipesService } from './recipes.service';

@Component({
    selector: 'app-recipes',
    templateUrl: './recipes.page.html',
    styleUrls: ['./recipes.page.scss'],
})

export class RecipesPage implements OnInit {
    recipes: Recipe[];

    constructor(private recipesService: RecipesService) { }

    ngOnInit() {
        this.recipes = this.recipesService.getAllRecipes();
    }
}
```

TASKS *(deadline today)*

1. Create the Recipe App using all the materials explained previously (page, interface, routing, details page, ion-(list, item, avatar), ngFor, service).



2. Modify the code so that if a recipe is clicked, your application will log the corresponding recipe detail in the console as shown below. Use the getRecipe() method from the RecipesService class.



3. Modify the code so the RecipesService will also have a function to delete a recipe, with the recipe id as the parameter
   ➔ **`deleteRecipe(recipeId: string){ … }`**