



Front End Development 1

Sesi 25



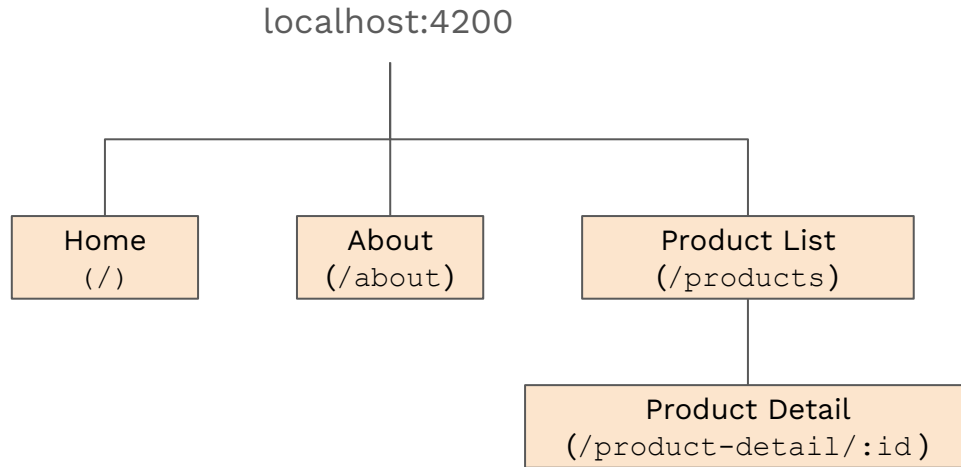


Angular Routing

+

Angular Routing

Berikut diagram navigasi untuk routing-routing yang akan dibuat dalam contoh sesi ini:



Angular Routing: Instalasi & Konfigurasi

Buat project baru dan Install Angular Routing didalamnya dengan memilih **Yes**, di pertanyaan pertama.

Pengaktifan routing ini, akan mengimport module **AppRoutingModule** di app.module dan men-generate file app-routing.module yang didalamnya mengimport dan mendaftarkan **RouterModule** dan **Routes** agar kita bisa memakai angular router.

```
l-> ng new routing-app
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
```

```
imports: [
  BrowserModule,
  AppRoutingModule
],
```

app.module.ts

```
app-routing.module.ts x
src > app > app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [];
5
6 @NgModule({
7   imports: [RouterModule.forRoot(routes)],
8   exports: [RouterModule]
9 })
10 export class AppRoutingModule { }
```

Angular Routing: Instalasi & Konfigurasi

Install bootstrap, dan masukkan konfigurasi ke file angular.json

```
npm i bootstrap
```

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

Angular Routing: Instalasi & Konfigurasi

Buatlah komponen-komponen berdasar kebutuhan routing (Diagram slide sebelumnya). Dan tambahkan 1 komponen **NoPageFound**.

```
ng generate component home
```

```
ng generate component about
```

```
ng generate component products
```

```
ng generate component product-detail
```

```
ng generate component NoPageFound
```

Angular Routing: Pendaftaran Komponen & Path

Pada file **app-routing.module**, Import komponen-komponen yang telah dibuat, dan define path serta komponen di dalam variable routes. Di dalam variable routes ini, Angular akan mengarahkan ke komponen yang sesuai dengan path terdaftar.

```
app-routing.module.ts M X
src > app > app-routing.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { HomeComponent } from './home/home.component';
4  import { AboutComponent } from './about/about.component';
5  import { ProductsComponent } from './products/products.component';
6  import { ProductDetailComponent } from './product-detail/product-detail.component';
7  import { NoPageFoundComponent } from './no-page-found/no-page-found.component';
8
9  const routes: Routes = [
10   { path: '', redirectTo: '/home', pathMatch: 'full' },
11   { path: 'home', component: HomeComponent },
12   { path: 'about', component: AboutComponent },
13   { path: 'products', component: ProductsComponent },
14   { path: 'product-detail/:id', component: ProductDetailComponent },
15   { path: '**', component: NoPageFoundComponent }
16 ];
17
18 @NgModule({
19   imports: [RouterModule.forRoot(routes)],
20   exports: [RouterModule]
21 })
22 export class AppRoutingModule { }
```

path ****** akan mengalihkan path yang tidak terdaftar, ke komponen NoPageFound.

Angular Routing: Penjelasan pathMatch

Untuk pathMatch ini, Angular menyediakan 2 mekanisme yaitu **full** dan **prefix**.

Sebagai contoh pathMatch

full:

```
{ path: '', redirectTo: '/home', pathMatch: 'full' },
```

Untuk setiap path kosong (localhost:4200 atau localhost:4200/) akan diarahkan ke path /home.

prefix:

```
{ path: 'coba', redirectTo: '/home', pathMatch: 'prefix' },
```

Untuk setiap path yang dimulai dengan prefix path **/coba** (localhost:4200/coba atau localhost:4200/coba/asal) akan diarahkan ke path /home.

Angular Routing: Parameter Path

Karena halaman detail product membutuhkan **id** dari product tersebut, maka perlu mendaftarkan routing ber-params nama **id**. Penamaan parameter di routing selalu dimulai dengan tanda :

```
{ path: 'product-detail/:id', component: ProductDetailComponent }
```

Berdasar pendaftaran route diatas, nantinya contoh path yang bisa diakses:

```
localhost:4200/product/123
```

```
localhost:4200/product/id_ke_1
```

123 dan id_ke_1 akan menjadi parameter **id**, yang sebelumnya sudah didefinisikan.

Angular Routing: Passing Parameter

Di komponen product-detail, import class **ActivatedRoute**, agar kita dapat mengakses parameter ke komponen. Gunakan properti **.snapshot.params** dalam activatedroute untuk mengakses parameter router.

```
product-detail.component.ts U x
src > app > product-detail > product-detail.component.ts > ...
1  import { Component } from '@angular/core';
2  import { ActivatedRoute } from '@angular/router';
3
4  @Component({
5    selector: 'app-product-detail',
6    templateUrl: './product-detail.component.html',
7    styleUrls: ['./product-detail.component.css']
8  })
9
10 export class ProductDetailComponent {
11   product_id: string;
12
13   constructor(private actRoute: ActivatedRoute) {
14     this.product_id = this.actRoute.snapshot.params.id;
15   }
16 }
```

```
product-detail.component.html U x
src > app > product-detail > product-detail.component.html > ...
1  <h5 class="card-title">Product Id is: {{product_id}}</h5>
```

Angular Routing: Product List

Tambahkan array **products** di product.component dan di dalam template lakukan iterasi terhadap products tersebut. Dalam setiap product, sediakan 1 link menuju halaman product-detail.

```
export class ProductsComponent implements OnInit {  
  products = [  
    {  
      id: "ABC8441189035",  
      name: "Tshirt",  
      description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit."  
    },  
    {  
      id: "DEF6510463347",  
      name: "Shoes",  
      description: "Proin ac metus in diam porttitor viverra eu sit amet ligula."  
    },  
    {  
      id: "GHI0831819467",  
      name: "Handbags",  
      description: "Duis sodales dui vitae urna varius, at ullamcorper purus tempor."  
    }  
  ]  
}
```

```
products.component.html U x  
src > app > products > products.component.html > ...  
1 <div class="row">  
2   <div class="col-sm-4" *ngFor="let product of products">  
3     <div class="card mb-3">  
4       <div class="card-body">  
5         <h5 class="card-title">{{product.name}}</h5>  
6         <p class="card-text">{{product.description}}</p>  
7         <a [routerLink]="['/product-detail/', product.id]" class="btn btn-danger">  
8           Check Product  
9         </a>  
10      </div>  
11    </div>  
12  </div>  
13 </div>
```

Angular Routing: Navbar

Generate 1 komponen navbar, dan masukkan template dibawah ini:

```
src > app > navbar > navbar.component.html > ...
1 <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
2   <ul class="navbar-nav">
3     <li class="nav-item">
4       <a routerLink="/home" routerLinkActive="active" class="nav-link">
5         Home
6       </a>
7     </li>
8     <li class="nav-item">
9       <a routerLink="/about" routerLinkActive="active" class="nav-link">
10        About
11      </a>
12    </li>
13    <li class="nav-item">
14      <a routerLink="/products" routerLinkActive="active" class="nav-link">
15        Products
16      </a>
17    </li>
18  </ul>
19 </nav>
```

Angular Routing: router-outlet

Ganti template file **app.component.html**, dengan:

```
5 app.component.html M ×  
src > app > 5 app.component.html > ...  
1 | <app-navbar></app-navbar>  
2 | <div class="container text-center mt-5">  
3 | | <router-outlet></router-outlet>  
4 | </div>  
5 |
```

router-outlet berfungsi untuk menampilkan komponen yang sesuai dengan path di url browser. Contoh router-outlet diatas, dibungkus dengan class bootstrap.

Silakan coba jalankan aplikasi dan arahkan url ke path-path yang sudah didaftarkan!

Angular Routing: No Page Found

Ganti template file **no-page-found.component**, dengan:

```
no-page-found.component.html U x
src > app > no-page-found > no-page-found.component.html > ...
1 <div class="col-md-12">
2   <div class="error-template">
3     <h3>Oops!</h3>
4     <h2>404 Not Found</h2>
5     <div class="error-details mb-3">
6       Sorry, Requested page not found!
7     </div>
8     <div class="error-actions">
9       <a routerLink="/home" class="btn btn-danger btn-lg">
10         Go Back to Home
11       </a>
12     </div>
13   </div>
14 </div>
```

localhost:4200/routeasal

Oops!
404 Not Found

Sorry, Requested page not found!

Go Back to Home

Jalankan aplikasi, dan arahkan ke route sembarang, maka akan nampak komponen NoPageFound ini.

Angular Routing: Penjelasan RouterLink

Routerlink, berfungsi sebagaimana properti href dalam tag <a> di html, berfungsi untuk menavigasi dari 1 halaman ke halaman lain.

Contoh (1):

```
<a routerLink="/home" routerLinkActive="active" class="nav-link">  
  Home  
</a>
```

properti **routerLink**, akan menavigasikan path ke /home setelah link bertekstan Home di-klik.
properti **routerLinkActive** berfungsi untuk mengaktifkan css class 'active', apabila link terpilih.

Angular Routing: Penjelasan RouterLink (Link Dinamis)

Apabila url memiliki link atau params dinamis, maka kita bisa mem-bind **[routerLink]** dan mengisinya dengan value **array** dan akan di-generate oleh Angular menjadi satu kesatuan link atau path.

Contoh (2) url dinamis:

```
<a [routerLink]="['/product-detail/', product.id]" class="btn btn-danger">  
  Check Product  
</a>
```

Sebagai contoh lain, apabila value berisi `['/team', 'user', 3]` nantinya akan digenerate menjadi: `localhost:4200/team/user/3`.