

**B. BIAS A. CH.**



**ANGULAR**

**WITH BOOTSTRAP**

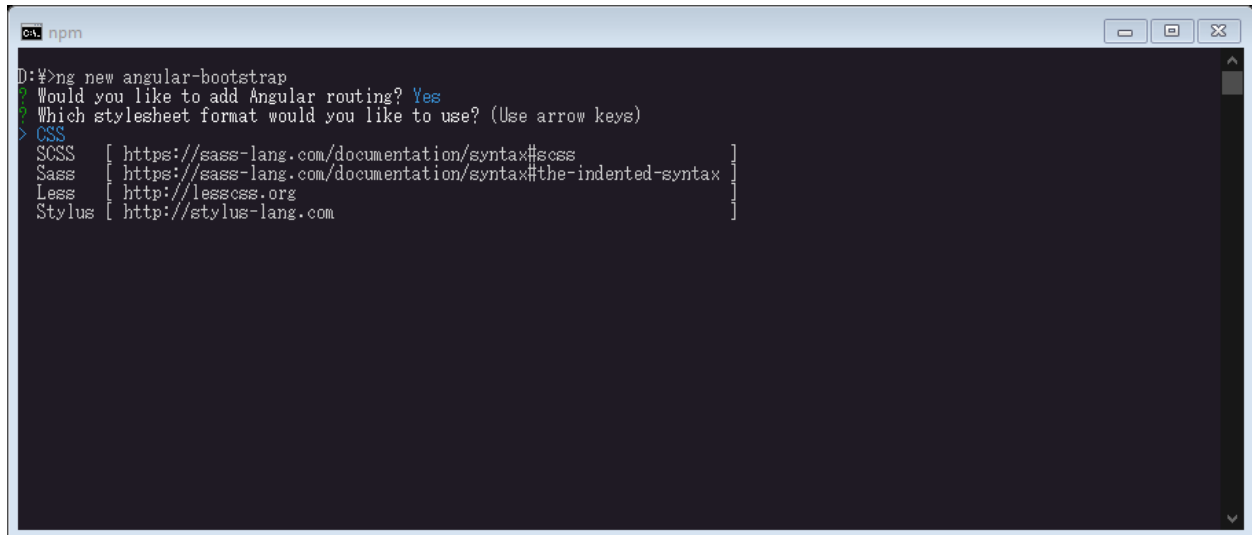
# Persiapan

1. Internet Stabil
2. Instalasi Angular
3. Instalasi jQuery
4. Instalasi Bootstrap
5. Setting-Up Workspace



# Persiapan `2-4` :

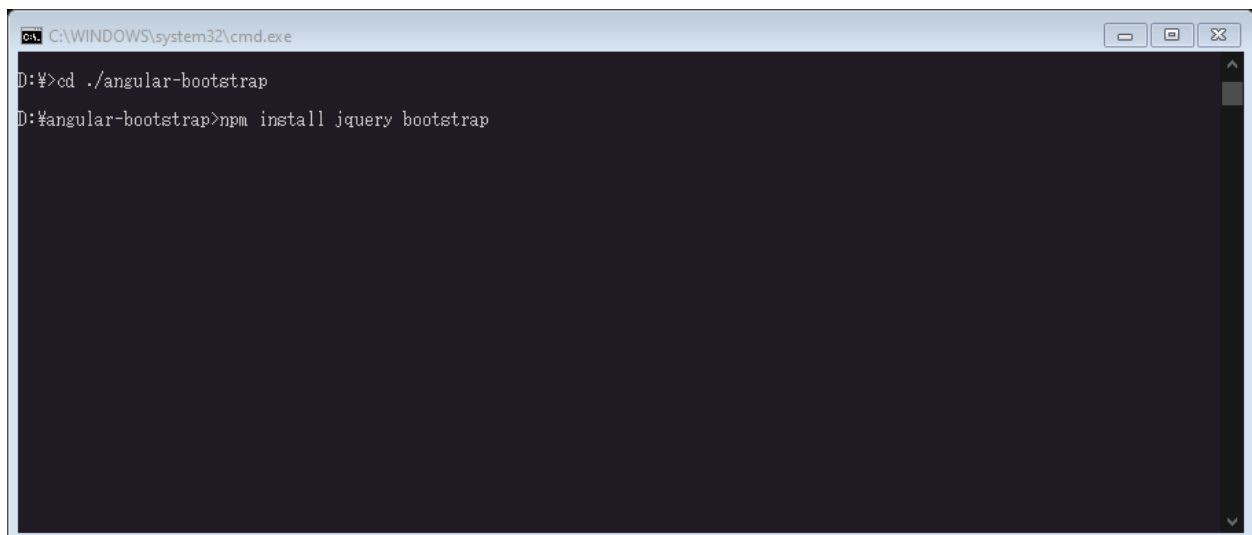
\$ ng new angular-bootstrap



```
cmd npm
D:\>ng new angular-bootstrap
? 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 ]
Stylus [ http://stylus-lang.com ]
```

\$ cd ./angular-bootstrap

\$ npm install jquery bootstrap

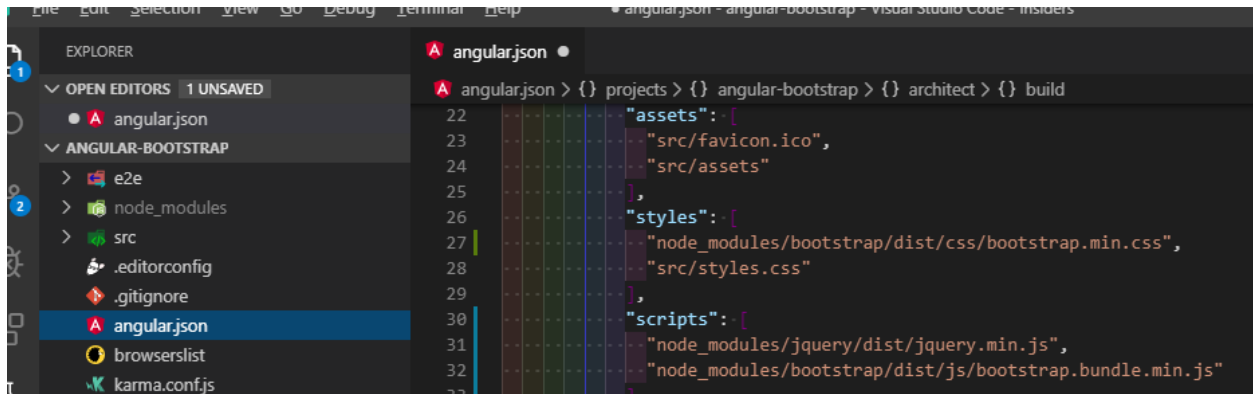


```
cmd C:\WINDOWS\system32\cmd.exe
D:\>cd ./angular-bootstrap
D:\angular-bootstrap>npm install jquery bootstrap
```

Delay Itu Bebas ~ Drop Itu Pilihan !

## Persiapan `5` :

Tambahkan `styles` & `scripts` Pada  
`angular.json`, Selesai~ Yeay^^



```
angular.json
{
  "projects": {},
  "angular-bootstrap": {},
  "architect": {},
  "build": {
    "assets": [
      "src/favicon.ico",
      "src/assets"
    ],
    "styles": [
      "node_modules/bootstrap/dist/css/bootstrap.min.css",
      "src/styles.css"
    ],
    "scripts": [
      "node_modules/jquery/dist/jquery.min.js",
      "node_modules/bootstrap/dist/js/bootstrap.bundle.min.js"
    ]
  }
}
```



Delay Itu Bebas ~ Drop Itu Pilihan !

# Routing

Ah~ Apa Ini ? Jadi Routing Itu Memiliki Fungsi Untuk Mengatur Navigasi Antar Halaman Gitu Deh^^ Kalau Kalian Sering Lihat Di URL Browser Kalian Pasti Gak Asing Sama Misalnya `http://web.site/user/1`, Nah Disini Yang Mengatur Halaman Apa Yang Akan Ditampilkan Ya Si `/user/:id`. Langsung Saja Kita Coba~



Pertama Kita Buat Dulu `Halaman` Home, User, & Detailnya~

\$ ng generate component ...

```
C:\WINDOWS\system32\cmd.exe

D:\angular-bootstrap>ng generate component home
CREATE src/app/home/home.component.html (19 bytes)
CREATE src/app/home/home.component.spec.ts (614 bytes)
CREATE src/app/home/home.component.ts (261 bytes)
CREATE src/app/home/home.component.css (0 bytes)
UPDATE src/app/app.module.ts (635 bytes)

D:\angular-bootstrap>ng generate component user-list
CREATE src/app/user-list/user-list.component.html (24 bytes)
CREATE src/app/user-list/user-list.component.spec.ts (643 bytes)
CREATE src/app/user-list/user-list.component.ts (280 bytes)
CREATE src/app/user-list/user-list.component.css (0 bytes)
UPDATE src/app/app.module.ts (727 bytes)

D:\angular-bootstrap>ng generate component user-detail
CREATE src/app/user-detail/user-detail.component.html (26 bytes)
CREATE src/app/user-detail/user-detail.component.spec.ts (657 bytes)
CREATE src/app/user-detail/user-detail.component.ts (288 bytes)
CREATE src/app/user-detail/user-detail.component.css (0 bytes)
UPDATE src/app/app.module.ts (827 bytes)

D:\angular-bootstrap>
```

Oh Iya .. Karena Kita Tidak Menginginkan Tampilan Default Dari Angular, Yuk Kita Hapus Isi File `app.component.html` Dan Kita Ganti Isinya Menjadi Tempat Routing Saja~

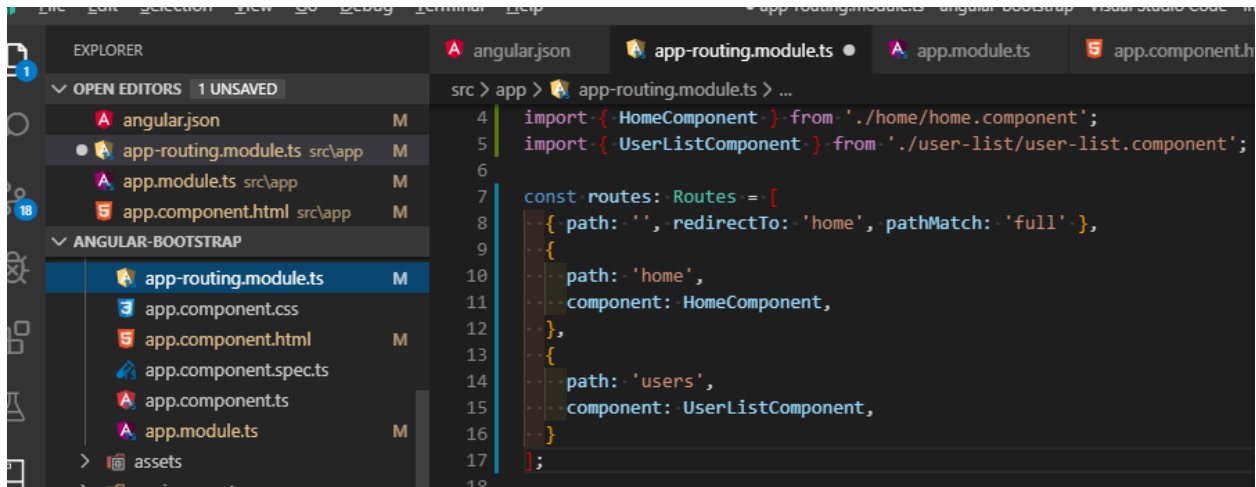
A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows the project structure. The main editor area displays the file `app.component.html` with the following content:

```
src > app > app.component.html > ...
1 | <router-outlet></router-outlet>
2 |
```

Delay Itu Bebas ~ Drop Itu Pilihan !

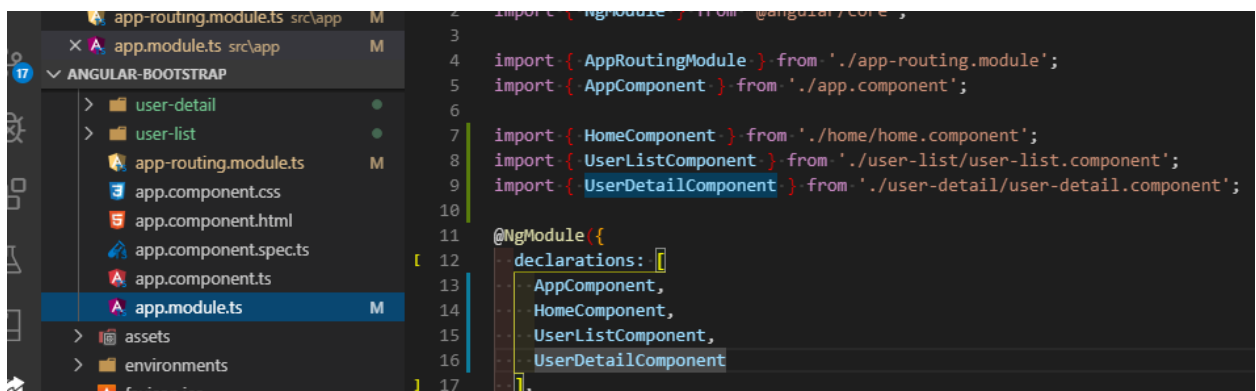
Manis Di Luar ~ Maso Di Dalam ! ^\_^

Kemudian Tambahkan Alamat Untuk Routing Pada `app-routing.module.ts`



```
src > app > app-routing.module.ts > ...
4 import { HomeComponent } from './home/home.component';
5 import { UserListComponent } from './user-list/user-list.component';
6
7 const routes: Routes = [
8   { path: '', redirectTo: 'home', pathMatch: 'full' },
9   {
10     path: 'home',
11     component: HomeComponent,
12   },
13   {
14     path: 'users',
15     component: UserListComponent,
16   },
17 ];
```

Modifikasi Juga Isi File `app.module.ts`

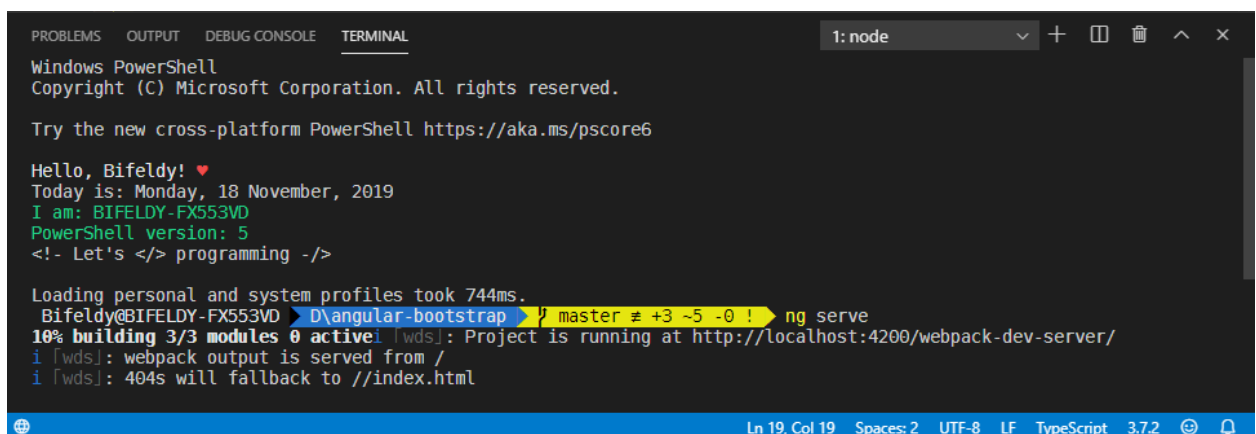


```

3 import { NgModule } from '@angular/core';
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6
7 import { HomeComponent } from './home/home.component';
8 import { UserListComponent } from './user-list/user-list.component';
9 import { UserDetailComponent } from './user-detail/user-detail.component';
10
11 @NgModule({
12   declarations: [
13     AppComponent,
14     HomeComponent,
15     UserListComponent,
16     UserDetailComponent
17 ],
```

Jalankan Servernya Untuk Uji Coba~

\$ ng serve -o



```
1: node
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

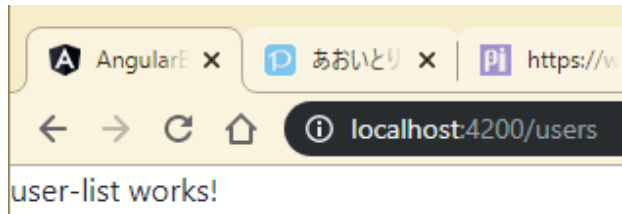
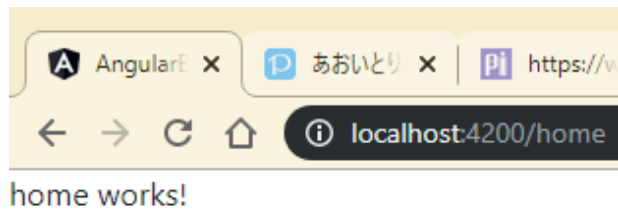
Try the new cross-platform PowerShell https://aka.ms/pscore6

Hello, Bifeldy! ❤
Today is: Monday, 18 November, 2019
I am: BIFELDY-FX553VD
PowerShell version: 5
<!-- Let's </> programming -/>

Loading personal and system profiles took 744ms.
Bifeldy@BIFELDY-FX553VD: D:\angular-bootstrap > master # +3 ~5 -0 ! ng serve
10% building 3/3 modules 0 active | wds: Project is running at http://localhost:4200/webpack-dev-server/
i [wds]: webpack output is served from /
i [wds]: 404s will fallback to //index.html
```

Delay Itu Bebas ~ Drop Itu Pilihan !

Tadaa~~ Routing URL Berhasil^^ Yeay!



Ta.. Tapi Kok Cuma Tulisan Doank Sih?

Kan Tadi Sudah Install Bootstrap. Mana ?

Okay-Okay .. Sabar ..

Setelah Ini Kita Akan Buat Yang Namanya `Templating`



# Templating

Yuhu~ Kita Lanjut Ke Bagian Templating ..  
Buat Kalian Yang Suka Nge-Desain Biasanya Pasti  
Bakal Menghabiskan Waktu Disini Nih!

Tapi Repot 'Kan Kalau Kita Buat Desain Yang  
Berulang Untuk Tiap Halamannya Kalau Isinya Ada  
Yang Sama .. Kayak Misalnya Menu Yang Ada Diatas  
(Header) Atau Yang Dibawah (Footer). Pasti Capek  
Donk~ Inilah Gunanya Templating, Langsung Aja Yuk!!



Pertama Kita Buat Dulu `Component` Header & Footer-nya~

\$ ng generate component ...

```
C:\WINDOWS\system32\cmd.exe
D:\angular-bootstrap>ng g c _shared/component/header
CREATE src/app/_shared/component/header/header.component.html (21 bytes)
CREATE src/app/_shared/component/header/header.component.spec.ts (628 bytes)
CREATE src/app/_shared/component/header/header.component.ts (269 bytes)
CREATE src/app/_shared/component/header/header.component.css (0 bytes)
UPDATE src/app/app.module.ts (760 bytes)
D:\angular-bootstrap>ng g c _shared/component/footer
CREATE src/app/_shared/component/footer/footer.component.html (21 bytes)
CREATE src/app/_shared/component/footer/footer.component.spec.ts (628 bytes)
CREATE src/app/_shared/component/footer/footer.component.ts (269 bytes)
CREATE src/app/_shared/component/footer/footer.component.css (0 bytes)
UPDATE src/app/app.module.ts (860 bytes)
D:\angular-bootstrap>
```

Oh Iya, Perintah Terminal Angular Juga Bisa Disingkat Lho~ Haha^^

Delay Itu Bebas ~ Drop Itu Pilihan !

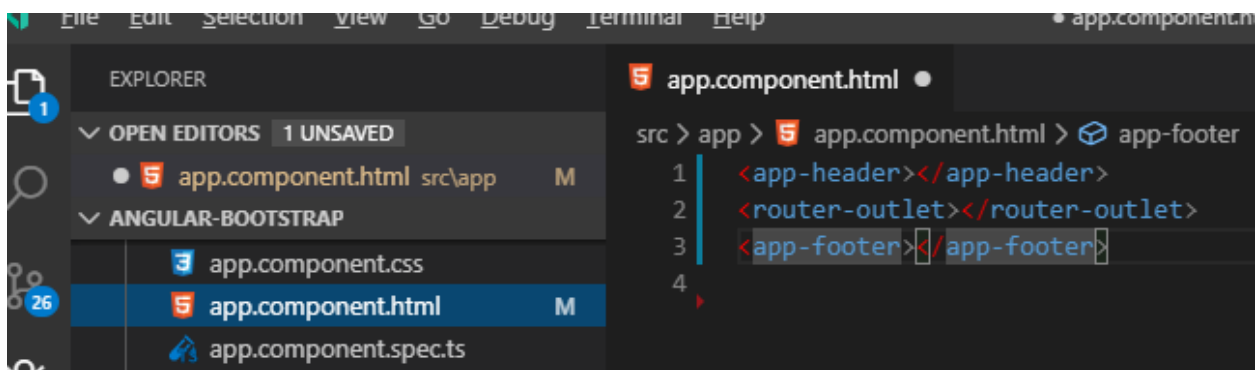


Manis Di Luar ~ Maso Di Dalam ! ^\_^

Ah Iya, Disini Aku Bikin Komponen-nya Di Dalam Folder ``_shared/component`` Kenapa? Karena Di Dalam Folder Inilah Yang Nantinya File-File Biasanya Akan Di Panggil Secara Berulang Di Seluruh Aplikasi Angular Kalian ..

Tentunya Kalian Boleh Untuk Tidak Mengikuti Kok .. Ini Hanya Kebiasaan Yang Aku Sering Lakukan Untuk Mempermudah Mencari File .. Tentunya Ini Juga Bisa Di Sebut Gaya Koding `Modular` 'Lho~ Hehehe~ ^\_^

Pertama-tama Kita Atur Dulu Tata Letak Urutan Tampilnya~



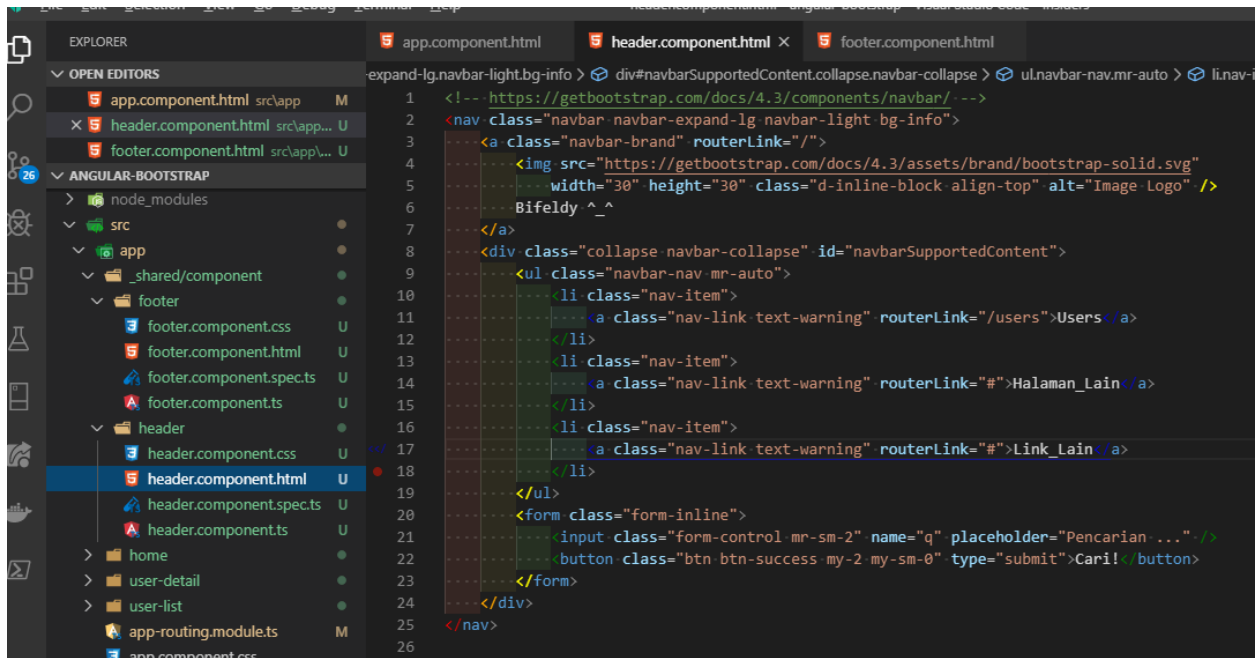
Kenapa Ditaruh Di Atas & Bawahnya ``router-outlet`` ??

-> Karena Halaman Ini Akan Di Load Sebanyak 1x Saja (Kecuali Di Re-Fresh Browsernya) Dan Karena Isi Halaman Yang Berubah Akibat Routing URL Berada Di Dalam ``router-outlet`` Saat Berpindah Halaman, Kita Tidak Ingin Header & Footer Melakukan Loading Ulang.

Delay Itu Bebas ~ Drop Itu Pilihan !

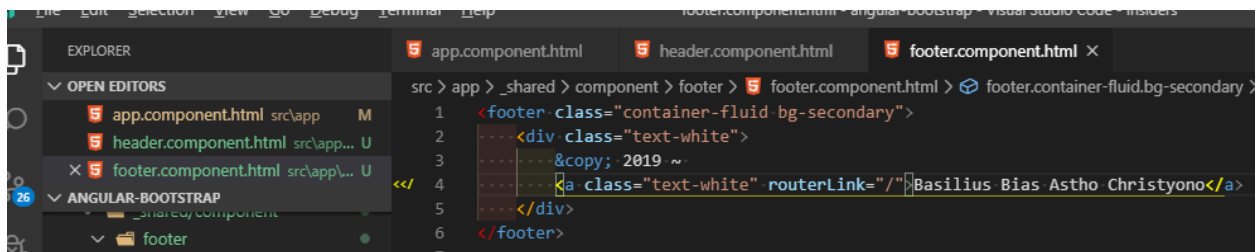
Manis Di Luar ~ Maso Di Dalam ! ^\_^

Header Enaknya Di Isi Apa Ya .. Hmn .. 'Eh Menu Navigasi Aja Deh'



```
1 <!-- https://getbootstrap.com/docs/4.3/components/navbar/ -->
2 <nav class="navbar navbar-expand-lg navbar-light bg-info">
3   <a class="navbar-brand" routerLink="/">
4     
6     Bifeldy ^_^
7   </a>
8   <div class="collapse navbar-collapse" id="navbarSupportedContent">
9     <ul class="navbar-nav mr-auto">
10      <li class="nav-item">
11        <a class="nav-link text-warning" routerLink="/users">Users</a>
12      </li>
13      <li class="nav-item">
14        <a class="nav-link text-warning" routerLink="#">Halaman Lain</a>
15      </li>
16      <li class="nav-item">
17        <a class="nav-link text-warning" routerLink="#">Link Lain</a>
18      </li>
19    </ul>
20    <form class="form-inline">
21      <input class="form-control mr-sm-2" name="q" placeholder="Pencarian ..." />
22      <button class="btn btn-success my-2 my-sm-0" type="submit">Cari! </button>
23    </form>
24  </div>
25 </nav>
26
```

Footernyaa .. Emn .. Ku Isi Nama Sendiri Aja Lah Yaa~ xD

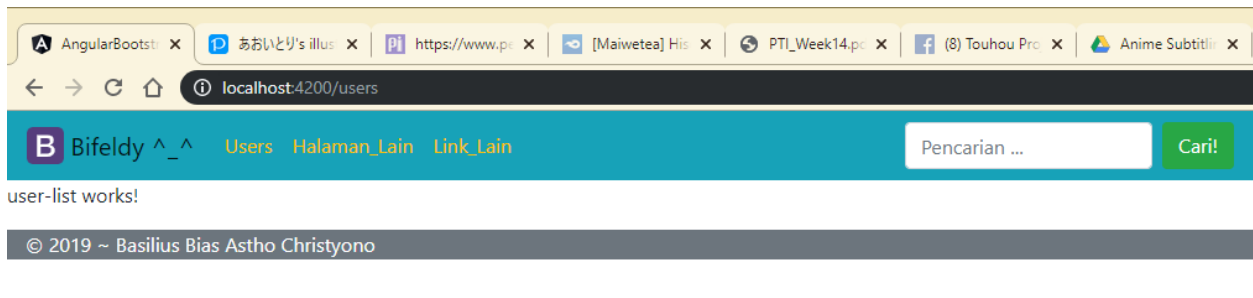
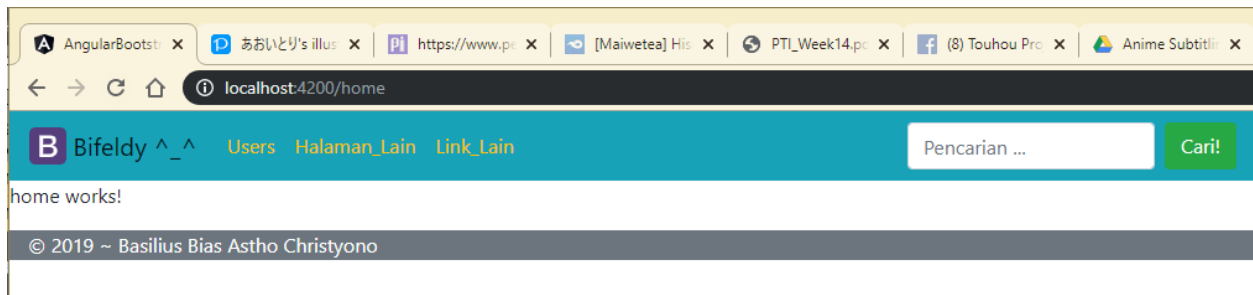


```
1 <footer class="container-fluid bg-secondary">
2   <div class="text-white">
3     &copy; 2019 ~
4     <a class="text-white" routerLink="/">Basilus Bias Astho Christyono</a>
5   </div>
6 </footer>
7
```

Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_^

Hasilnya~ Tadha—



Rehat Dulu Ah Sejenak—



Delay Itu Bebas ~ Drop Itu Pilihan !

# Services

Ih! Apaan Lagi Ini .. Service ? Pelayan ? Hemn ..  
Menarik .. ~~Fan-Service?~~ (๖\_๖)

Eits .. Ini Dia Yang Ditunggu-tunggu .. Sang  
Makhluk Penyelamat Programmer .. Si Service Ini Nih ..  
Bisa Ngurangin Jumlah Koding Yang Harus Kamu Tulis  
Lho ..



Gak Percaya .. Coba Banyangin .. Kalau Kamu Mau Nampilin Data Yang Sama ..  
Misalnya Data Tabel Dari Sumber Data Yang Sama .. Masa Iya Kamu Mau Ngoding 2x  
Di File `\*.component.ts` Sih ..

Dia Juga Bisa Lo Ngebantu Kamu Buat Kirim-kirim Data / Nilai Suatu Variabel  
Dari Halaman Yang Satu Dengan Halaman Lainnya ? Kuy Lah Cobain~

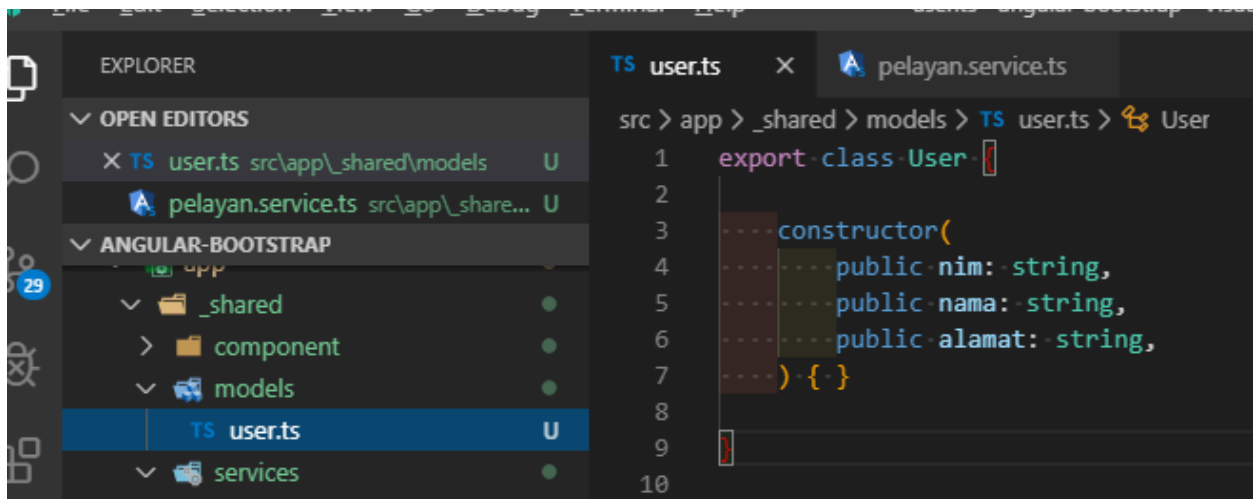
Pertama Kita Buat Dulu `Service` Dengan Nama Bebas Deh~

```
C:\WINDOWS\system32\cmd.exe
D:\angular-bootstrap>ng g service _shared/services/pelayan
CREATE src/app/_shared/services/pelayan.service.spec.ts (338 bytes)
CREATE src/app/_shared/services/pelayan.service.ts (136 bytes)
D:\angular-bootstrap>
```

Manis Di Luar ~ Maso Di Dalam ! ^\_^

Kita Buat Juga Model Data Yang Mau Kita Tampilkan—. Nah Tapi Kali Ini Kita Akan Buat Manual Karena Tidak Ada Dalam Fitur Terminalnya .. Ribet ? Enggak Kok ..

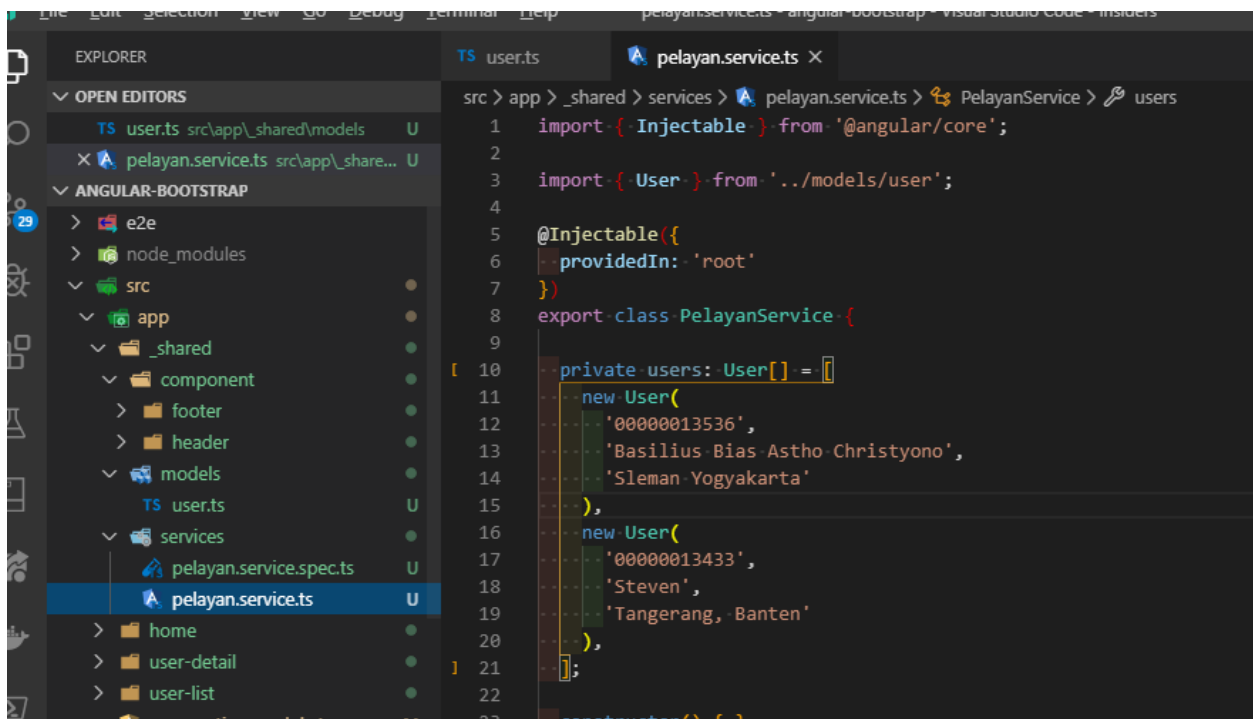
`Klik Kanan Folder '\_shared' > 'New Folder' Beri Nama 'models'` Lalu `Klik Kanan Folder 'models' > 'New File' Beri Nama 'user.ts'`



The screenshot shows the VS Code interface. In the Explorer on the left, the 'models' folder is selected under the '\_shared' directory. A new file 'user.ts' has been created. The Editor on the right shows the content of 'user.ts' with the following code:

```
src > app > _shared > models > TS user.ts > User
1 export class User {
2
3     constructor(
4         public nim: string,
5         public nama: string,
6         public alamat: string,
7     ) {}
8
9
10
```

Kita Isi Data User Di Pelayannya ..



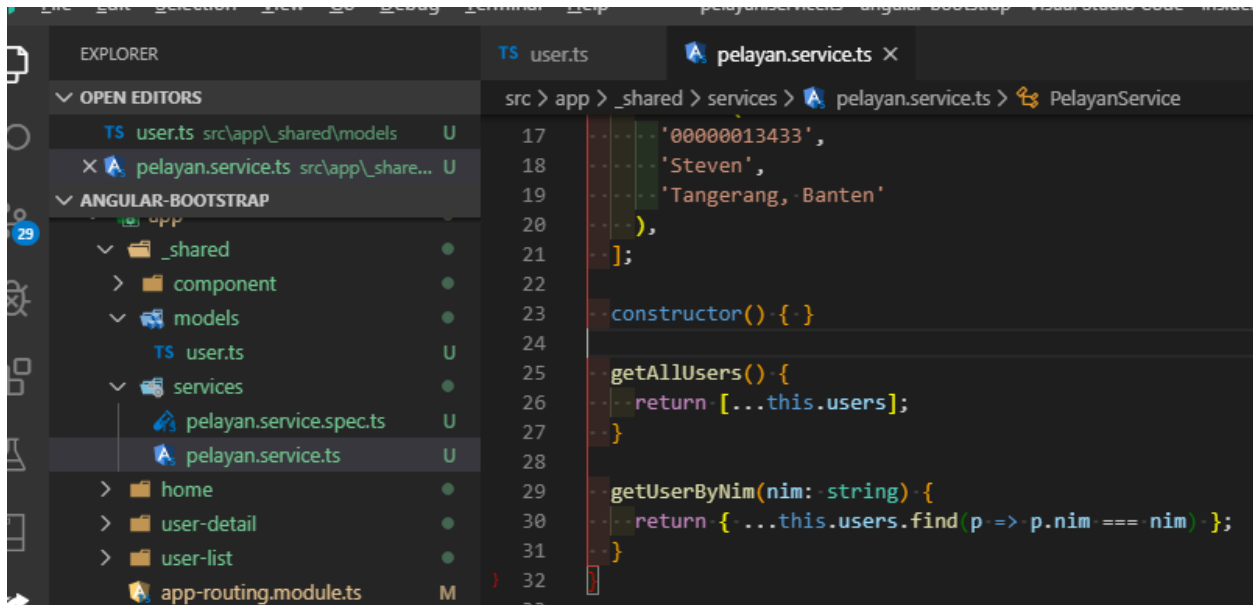
The screenshot shows the VS Code interface. In the Explorer on the left, the 'pelayan.service.ts' file is selected under the 'services' directory. The Editor on the right shows the content of 'pelayan.service.ts' with the following code:

```
src > app > _shared > services > pelayan.service.ts > PelayanService > users
1 import { Injectable } from '@angular/core';
2
3 import { User } from '../models/user';
4
5 @Injectable({
6     providedIn: 'root'
7 })
8 export class PelayanService {
9
10     private users: User[] = [
11         new User(
12             '00000013536',
13             'Basilius Bias Astho Christyono',
14             'Sleman Yogyakarta'
15         ),
16         new User(
17             '00000013433',
18             'Steven',
19             'Tangerang, Banten'
20         ),
21     ];
22
23     constructor() {}
24 }
```

Delay Itu Bebas ~ Drop Itu Pilihan !

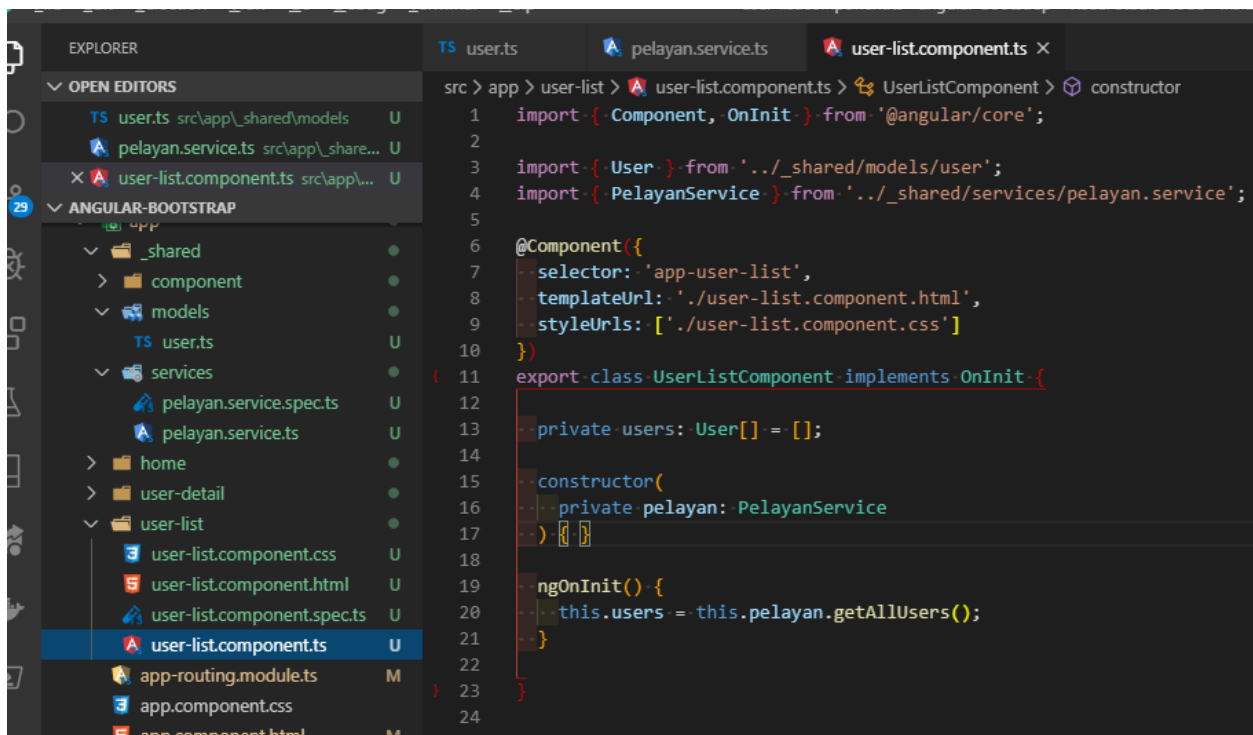
Manis Di Luar ~ Maso Di Dalam ! ^\_^

## Buat Juga Fungsi Untuk Mengambil Data Users



```
src > app > _shared > services > pelayan.service.ts > PelayanService
17  .... '00000013433',
18  .... 'Steven',
19  .... 'Tangerang, Banten'
20  ...),
21  ...];
22
23  constructor() {}
24
25  getAllUsers() {
26    return [...this.users];
27  }
28
29  getUserByNim(nim: string) {
30    return {...this.users.find(p => p.nim === nim)};
31  }
32
```

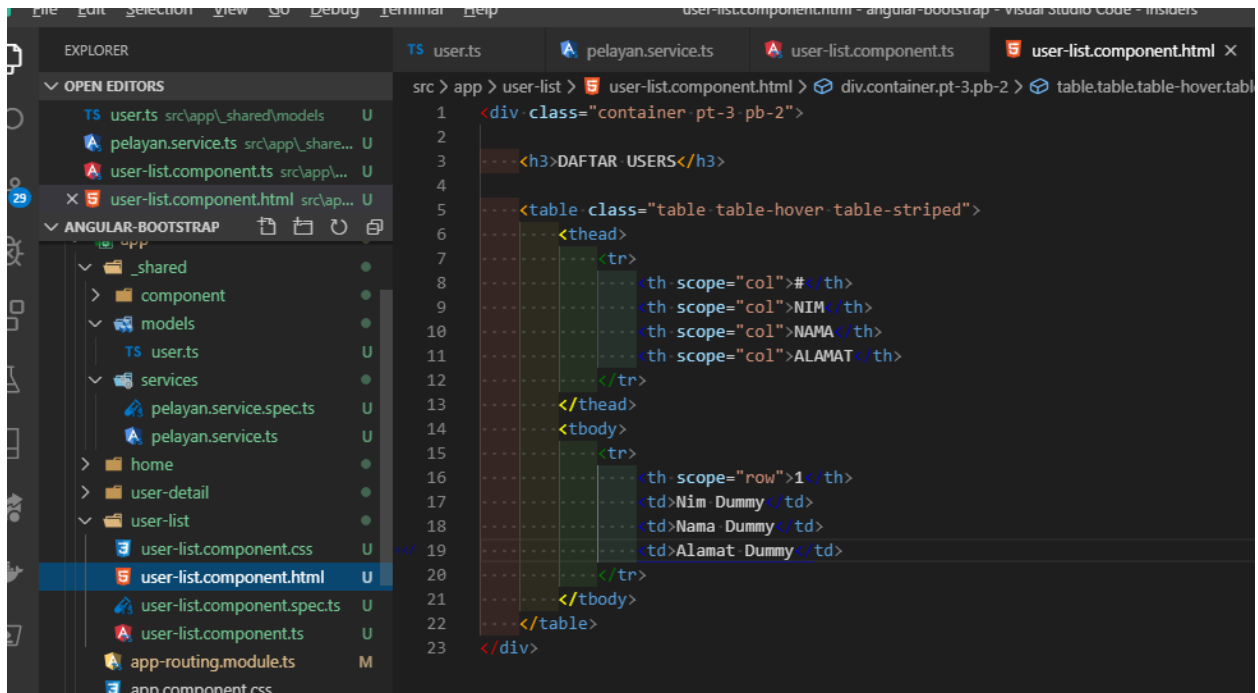
## Saatnya Injeksi Service Ke Dalam Halaman Yang Diperlukan .. YuuHuuu~



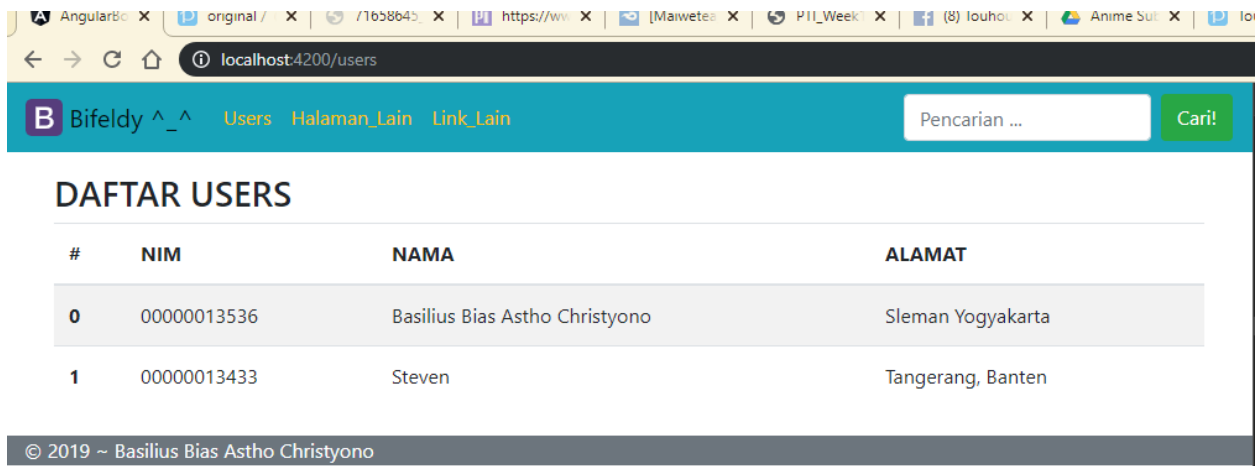
```
src > app > user-list > user-list.component.ts > UserListComponent > constructor
1  import { Component, OnInit } from '@angular/core';
2
3  import { User } from '../_shared/models/user';
4  import { PelayanService } from '../_shared/services/pelayan.service';
5
6  @Component({
7    selector: 'app-user-list',
8    templateUrl: './user-list.component.html',
9    styleUrls: ['./user-list.component.css']
10 })
11 export class UserListComponent implements OnInit {
12
13   private users: User[] = [];
14
15   constructor(
16     private pelayan: PelayanService
17   ) {}
18
19   ngOnInit() {
20     this.users = this.pelayan.getAllUsers();
21   }
22
23 }
24
```

Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_^



```
src > app > user-list > user-list.component.html > div.container.pt-3.pb-2 > table.table.table-hover.tbl
1 <div class="container pt-3 pb-2">
2
3 <h3>DAFTAR USERS</h3>
4
5 <table class="table table-hover table-striped">
6 <thead>
7 <tr>
8 <th scope="col">#</th>
9 <th scope="col">NIM</th>
10 <th scope="col">NAMA</th>
11 <th scope="col">ALAMAT</th>
12 </tr>
13 </thead>
14 <tbody>
15 <tr>
16 <th scope="row">1</th>
17 <td>Nim Dummy</td>
18 <td>Nama Dummy</td>
19 <td>Alamat Dummy</td>
20 </tr>
21 </tbody>
22 </table>
23 </div>
```



Aahh~ Aku Lupa Ngasih Potongan Code Untuk Nampilin Data Ke Table ..

Kalian Coba Sendiri Ya .. Clue :

- [-] Karena `users` Merupakan Variable `private` Maka Butuh Fungsi Getter
- [-] Bisa Juga Langsung Dijadikan `public`
- [-] Cara Akses Dari HTML Dengan Menggunakan Looping

Delay Itu Bebas ~ Drop Itu Pilihan !



# EXTRA

Wuih .. Udah Banyak Nih .. Jauh Pula .. Hahaha~

Eh Iya .. Masih inget Kan Awal” Tadi Kita Buat Halaman `user-detail` ?? Itu Lo .. Buat Lihat User Detailnya Kalau Misalkan URL Website-nya `http://web.site/user/:nim` Berarti Itu Halaman Detail Khusus Untuk Menampilkan Detail Tentang Si Dia .. Iya Dia Yang User Dengan Nimnya Tertentu ..



Oke .. Kita Langsung Saja .. Karena URL .. Berarti Ada Di Routing ..

A screenshot of the Visual Studio Code editor. The Explorer panel on the left shows the project structure with files like user-list.component.css, user-list.component.html, user-list.component.spec.ts, user-list.component.ts, app-routing.module.ts, app.component.css, app.component.html, and app.component.spec.ts. The main editor area shows the app-routing.module.ts file with the following code:

```
12 component: HomeComponent,  
13 },  
14 {  
15   path: 'users',  
16   component: UserListComponent,  
17 },  
18 {  
19   path: 'users/:nim',  
20   component: UserDetailComponent,  
21 },  
22 ],  
23 ];
```

Buat Link Untuk Navigasi Ke Halaman Detail Jika Tabel Di Klik~

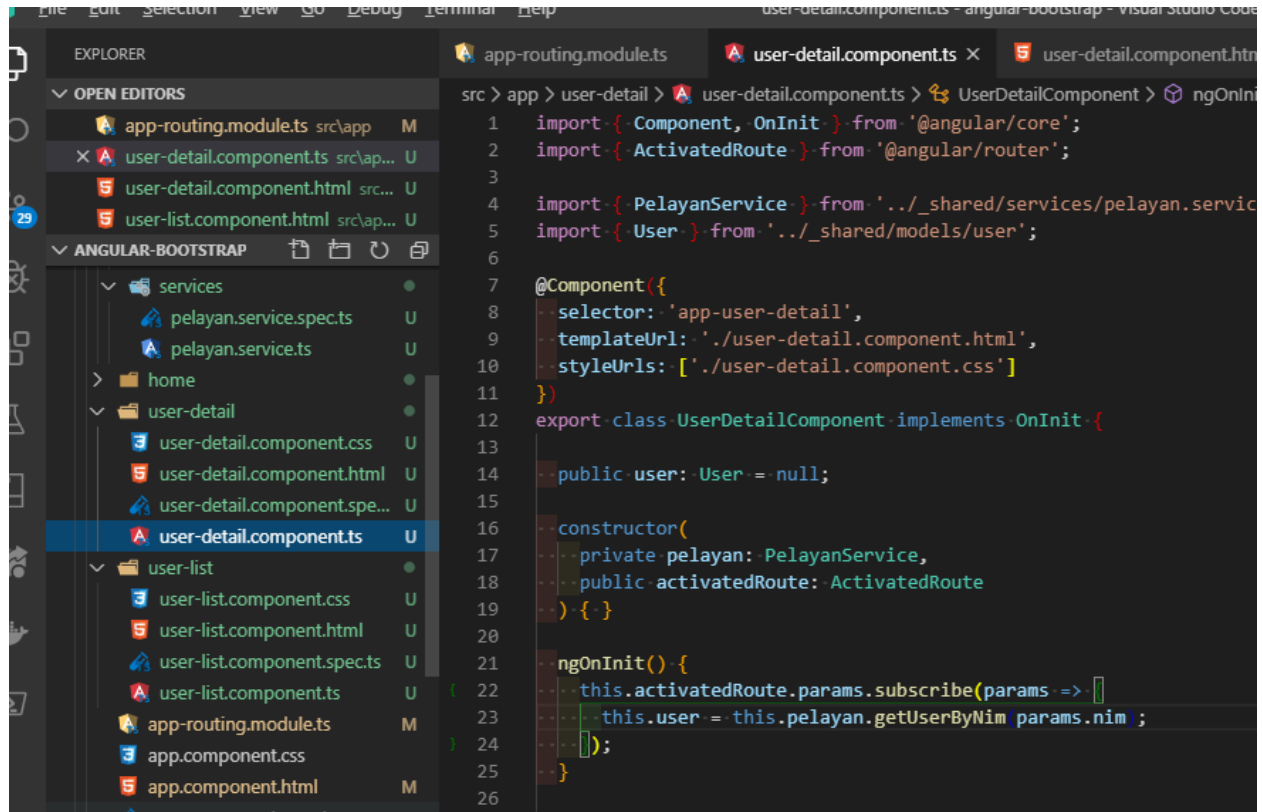
A screenshot of the Visual Studio Code editor showing the user-list.component.html file. The Explorer panel on the left shows the project structure with files like app-routing.module.ts, user-detail.component.ts, user-list.component.html, user-list.component.css, user-list.component.spec.ts, user-list.component.ts, and app-routing.module.ts. The main editor area shows the HTML code for the table:

```
13 </thead>  
14 <tbody>  
15 <tr *ngFor="let user of getUsers(); let i = index" routerLink="/{{user.nim}}">  
16 <th scope="row">{{i}}</th>  
17 <td>{{user.nim}}</td>  
18 <td>{{user.nama}}</td>  
19 <td>{{user.alamat}}</td>  
20 </tr>  
21 </tbody>  
22 </table>  
23 </div>
```

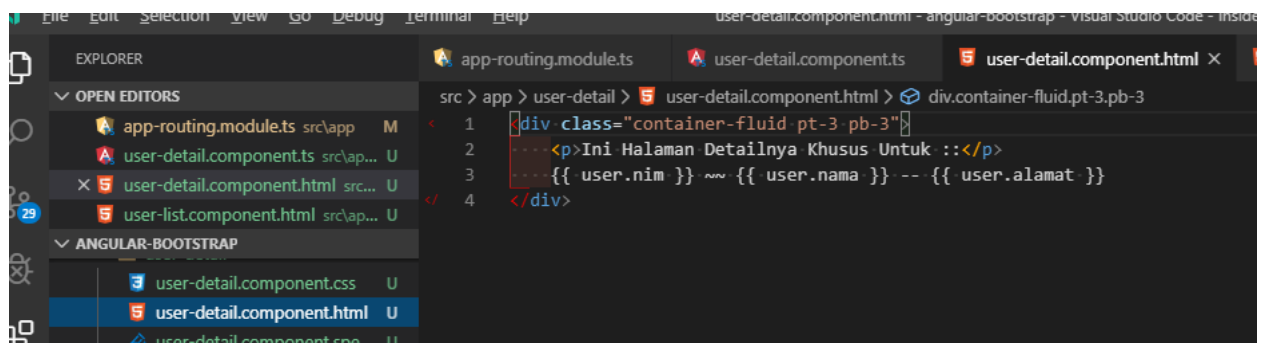
Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_^

Tambahkan Pada Halaman Detail Untuk Ambil NIM Berdasarkan URL /:nim ..



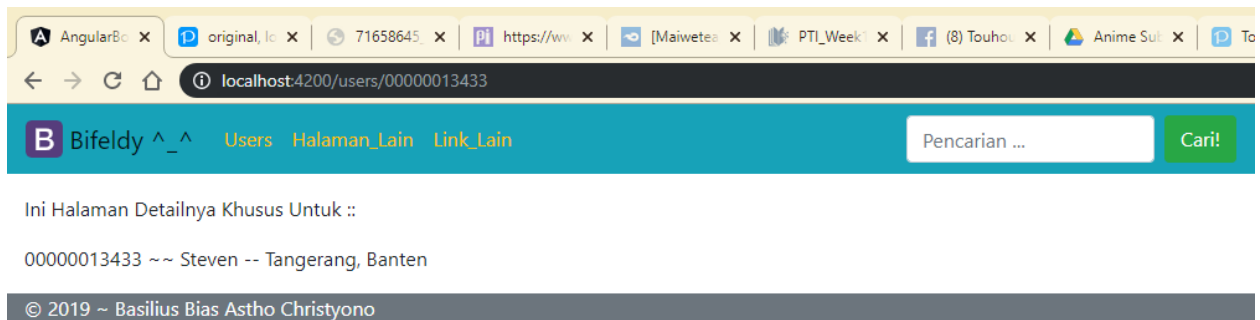
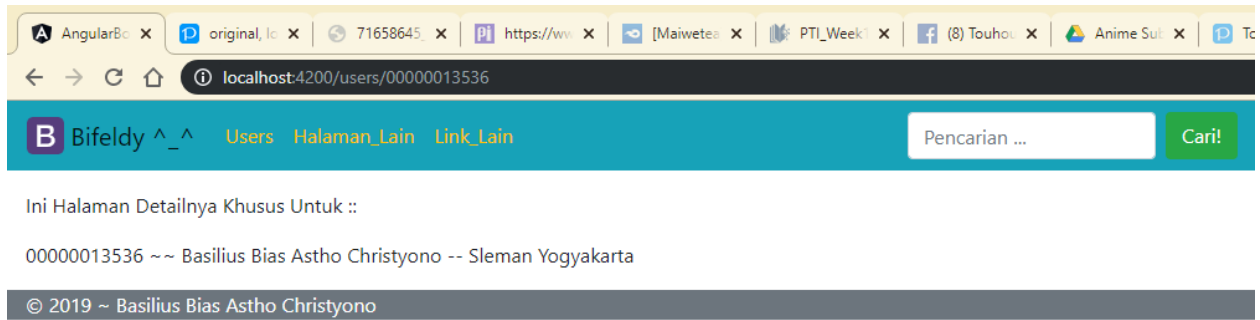
```
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3
4 import { PelayanService } from '../_shared/services/pelayan.service';
5 import { User } from '../_shared/models/user';
6
7 @Component({
8   selector: 'app-user-detail',
9   templateUrl: './user-detail.component.html',
10  styleUrls: ['./user-detail.component.css']
11 })
12 export class UserDetailComponent implements OnInit {
13
14   public user: User = null;
15
16   constructor(
17     private pelayan: PelayanService,
18     public activatedRoute: ActivatedRoute
19   ) {}
20
21   ngOnInit() {
22     this.activatedRoute.params.subscribe(params => {
23       this.user = this.pelayan.getUserByNim(params.nim);
24     });
25   }
26 }
```



```
1 <div class="container-fluid pt-3 pb-3">
2   <p>Ini Halaman Detailnya Khusus Untuk :</p>
3   <div>{{ user.nim }} ~ {{ user.nama }} -- {{ user.alamat }}
4 </div>
```

Delay Itu Bebas ~ Drop Itu Pilihan !

Dan Hasilnya ..



Selesai Sudah~

Gimana ? Sudah Cukup Capek ? ~~ Masih Ada I Lagi Lho ..

Tapi Ya .. Istirahatlah Dulu Sebentar ..

Rehat Sejenak ..



Delay Itu Bebas ~ Drop Itu Pilihan !

# API

Hikz .. Aku Kira Sudah Habis Materinya .. hUwUu~  
Ternyata Masih Ada .. Chi—Yare Yare! ..

Kalian Tau API Kan Ya Pastinya .. Bukan Si Jago Merah Lho~ Tapi `Application Programming Interface` Kalo Wikipedia Sih ..

API Itu *a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.*

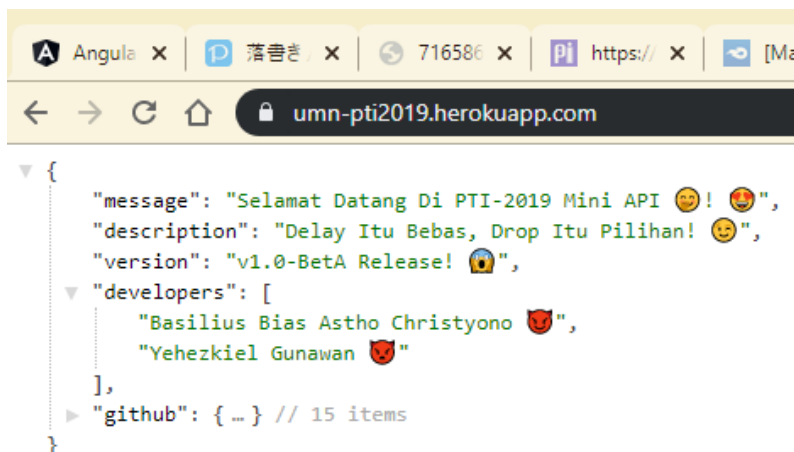
Dah-dah .. intinya API Digunakan Untuk Komunikasi Surat-Menyurat Antara Server Dengan Client .. Kirim Kirim Data Gitu Deh ..



buat service baru `_shared/services/pelayan-api`

buat models baru `_shared/models/mahasiswa`

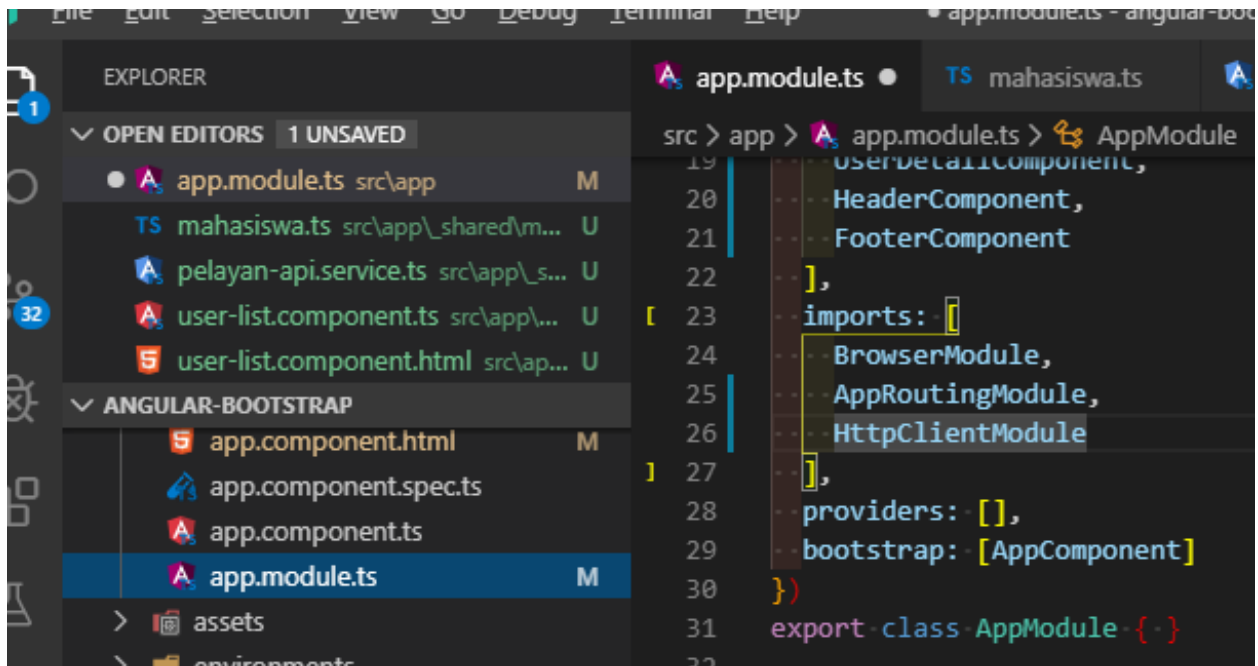
Target Kita Akan mengambil Data Dari <http://umn-pti2019.herokuapp.com/mahasiswa>



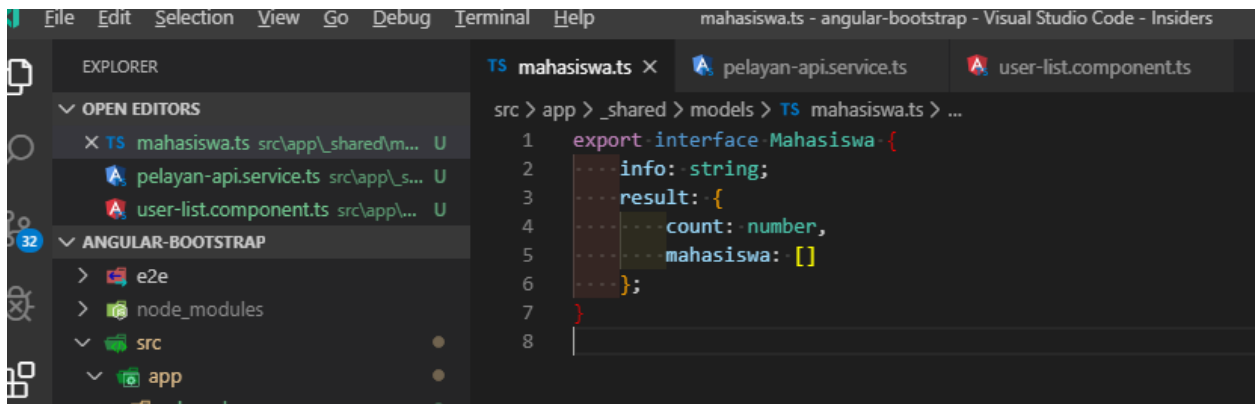
Seperti Inilah Kira Kira Data Yang Akan Kita Daparkan Saat Kita Meminta Ke API Server Dengan URL.

Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_^



Karena Kita Akan Menggunakan Protokol HTTP Untuk Mengakses API (Data Dari Luar Website) Maka Kita Perlu Menambahkan `HttpClientModule`



Kali Ini Kita Tidak Menggunakan `Class` Lagi Karena Data Yang Akan Kita Terima Bentuknya Langsung Objek Dengan Tampilan {}

Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_^

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'services' folder under 'app', with 'pelayan-api.service.ts' selected. The main editor displays the code for 'pelayan-api.service.ts'. The code defines the 'PelayanApiService' class, which is injectable and provides a root provider. It includes a constructor that takes an 'HttpClient' and two private properties: 'apiDocumentation' and 'urlApi'. The class has two methods: 'getAllMahasiswa' and 'getMahasiswaById'. The 'getAllMahasiswa' method returns an observable of 'Mahasiswa' objects by making a GET request to the 'urlApi/api/mahasiswa' endpoint. The 'getMahasiswaById' method returns an observable of 'Mahasiswa' objects by making a GET request to the 'urlApi/api/mahasiswa/{id}' endpoint.

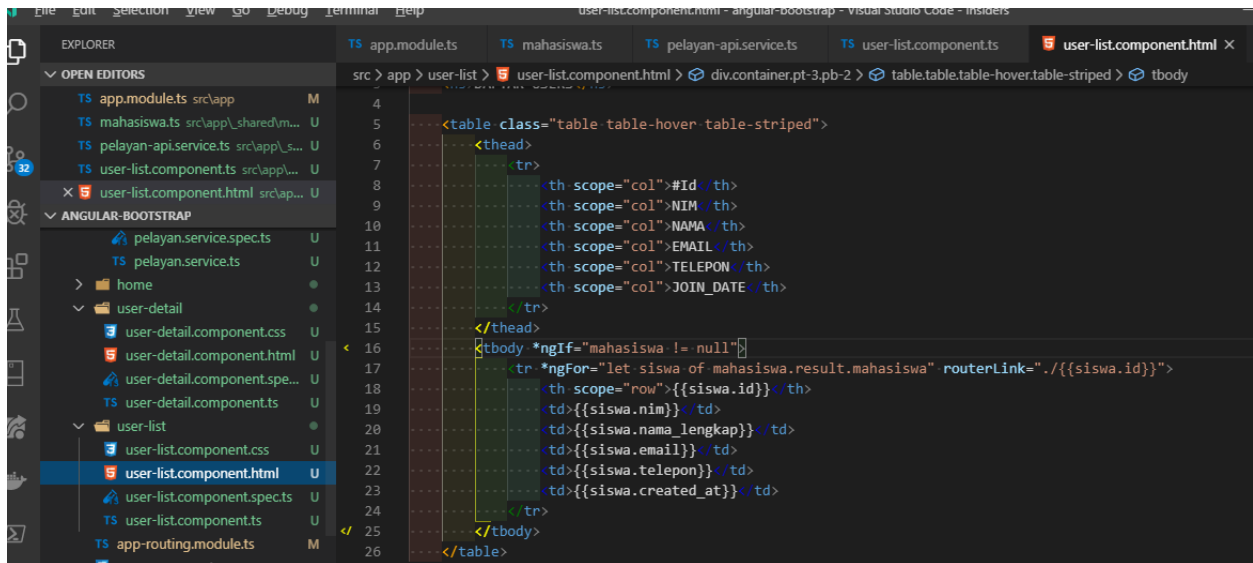
```
src > app > _shared > services > pelayan-api.service.ts > PelayanApiService > getAllMahasiswa
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  import { Mahasiswa } from '../models/mahasiswa';
6
7  @Injectable({
8    providedIn: 'root'
9  })
10 export class PelayanApiService {
11
12   private apiDocumentation = 'https://documenter.getpostman.com/view/5658787/Sw7WSpjd';
13   private urlApi = 'https://umn-pti2019.herokuapp.com';
14
15   constructor(
16     private http: HttpClient
17   ) {}
18
19   getAllMahasiswa(): Observable<Mahasiswa> {
20     return this.http.get<Mahasiswa>(`${this.urlApi}/api/mahasiswa`);
21   }
22
23   getMahasiswaById(id: number): Observable<Mahasiswa> {
24     return this.http.get<Mahasiswa>(`${this.urlApi}/api/mahasiswa/${id}`);
25   }
26
27 }
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar is expanded to show the 'user-list' folder under 'app', with 'user-list.component.ts' selected. The main editor displays the code for 'user-list.component.ts'. The code defines the 'UserListComponent' class, which is a component that implements the 'OnInit' interface. It has a selector of 'app-user-list', a templateUrl of './user-list.component.html', and styleUrls of ['./user-list.component.css']. The class has a constructor that takes a 'PelayanService' and a 'PelayanApiService' and initializes a private 'users' property to an empty array and a public 'mahasiswa' property to null. The 'ngOnInit' method subscribes to the 'getAllMahasiswa' method of the 'PelayanApiService' and updates the 'mahasiswa' property with the result.

```
src > app > user-list > user-list.component.ts > UserListComponent > ngOnInit
4  import { PelayanService } from '../_shared/services/pelayan.service';
5  import { PelayanApiService } from '../_shared/services/pelayan-api.service';
6  import { Mahasiswa } from '../_shared/models/mahasiswa';
7
8  @Component({
9    selector: 'app-user-list',
10   templateUrl: './user-list.component.html',
11   styleUrls: ['./user-list.component.css']
12 })
13 export class UserListComponent implements OnInit {
14
15   // private users: User[] = [];
16   public mahasiswa: Mahasiswa = null;
17
18   constructor(
19     // private pelayan: PelayanService
20     private pelayanApi: PelayanApiService
21   ) {}
22
23   ngOnInit() {
24     this.pelayanApi.getAllMahasiswa().subscribe(
25       result => { this.mahasiswa = result; },
26       error => { console.log(error); }
27     );
28   }
29 }
```

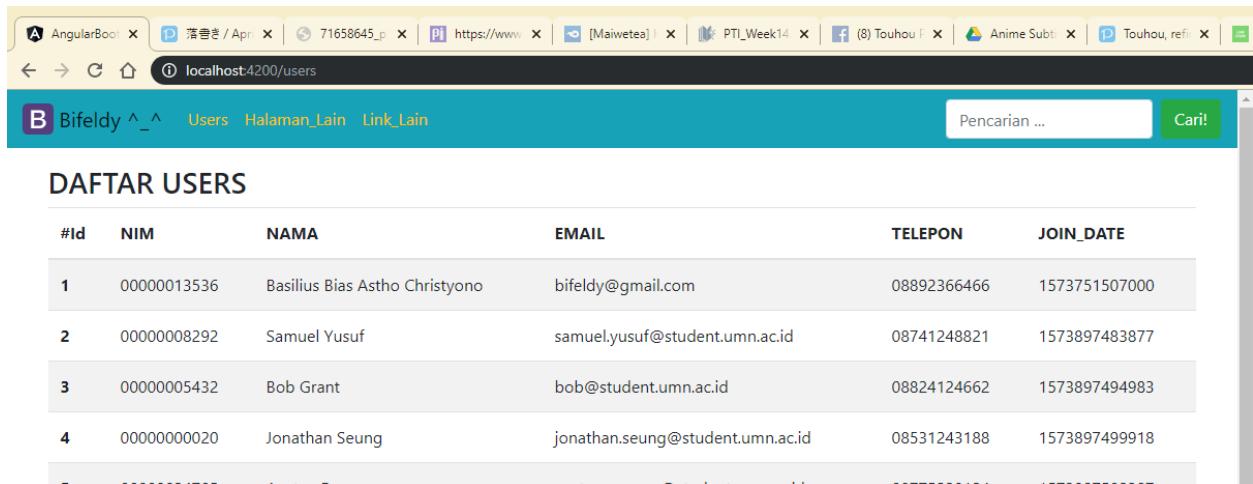
Delay Itu Bebas ~ Drop Itu Pilihan !

Manis Di Luar ~ Maso Di Dalam ! ^\_~



```
4 <table class="table table-hover table-striped">
5 <thead>
6 <tr>
7 <th scope="col">#Id</th>
8 <th scope="col">NIM</th>
9 <th scope="col">NAMA</th>
10 <th scope="col">EMAIL</th>
11 <th scope="col">TELEPON</th>
12 <th scope="col">JOIN_DATE</th>
13 </tr>
14 </thead>
15 <tbody *ngIf="mahasiswa != null">
16 <tr *ngFor="let siswa of mahasiswa.result.mahasiswa" routerLink="/{{siswa.id}}">
17 <td>{{siswa.id}}</td>
18 <td>{{siswa.nim}}</td>
19 <td>{{siswa.nama_lengkap}}</td>
20 <td>{{siswa.email}}</td>
21 <td>{{siswa.telepon}}</td>
22 <td>{{siswa.created_at}}</td>
23 </tr>
24 </tbody>
25 </table>
```

Jya Jya ~~ Jadi Deh ..



#Id	NIM	NAMA	EMAIL	TELEPON	JOIN_DATE
1	00000013536	Basilius Bias Astho Christyono	bifeldy@gmail.com	08892366466	1573751507000
2	00000008292	Samuel Yusuf	samuel.yusuf@student.um.ac.id	08741248821	1573897483877
3	00000005432	Bob Grant	bob@student.um.ac.id	08824124662	1573897494983
4	00000000020	Jonathan Seung	jonathan.seung@student.um.ac.id	08531243188	1573897499918
5	00000001700	Andika Supriatna	andika.supriatna@student.um.ac.id	08776338134	1573897503387

Ah Iya .. Lagi Dan Lagi ..

Untuk Mahasiswa Detailnya Silahkan Cobalah Untuk Membuat Jika Tabel Di Klik Akan Menarik Data Dari API Lagi Dan Menampilkannya Di Halaman Detail Berdasarkan Id Mahasiswa ..

Delay Itu Bebas ~ Drop Itu Pilihan !



Manis Di Luar ~ Maso Di Dalam ! ^\_^



Delay Itu Bebas ~ Drop Itu Pilihan !