

LAPORAN PRAKTIKUM

Javascript Framework : Angular



Nama : Fany Tania Lumban Siantar
NIM : 11S20041
Prodi : Sarjana Informatika
Kelas : 13 IF 2

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL



INSTITUT TEKNOLOGI DEL

MATERI PRAKTIKUM

11S3109 - Pengembangan Aplikasi Berbasis Web

Mata Kuliah	:	Pengembangan Aplikasi Berbasis Web
Tanggal	:	03 Oktober 2022
Semester	:	5
Topik	:	Javascript Framework: Angular
Minggu ke/Session ke	:	06/04
Aktivitas	:	Membuat aplikasi menggunakan angular.
Waktu pengerjaan	:	2x50 menit
Setoran	:	Softcopy (File)
Batas Penyerahan	:	Di akhir sesi praktikum
Tempat Penyerahan	:	Ecourse
Tujuan Praktikum	:	Mahasiswa mampu membuat aplikasi sederhana menggunakan angular pada sebuah website..

Petunjuk Praktikum:

1. Mahasiswa sudah meng-install node.js.
2. Mahasiswa bebas menggunakan tools editor seperti Visual Studio Code.
3. Mahasiswa belajar secara mandiri dengan mencari sumber-sumber lain baik cetak, *offline* maupun *online* (**catt:** w3school versi offline dapat diakses di **duktek.del.ac.id**).
4. Mahasiswa harus mengikuti semua prosedur yang diberikan pada praktikum.
5. Apabila ada pertanyaan pada setiap langkah pengerjaan praktikum, silahkan berkonsultasi pada TA.

Prosedur Praktikum:

1. Praktikum ini dikerjakan secara individu di ruangan laboratorium komputer yang telah ditentukan.
2. Tuliskan kode dibawah ini pada setiap halaman dokumen yang dikerjakan (paling atas). Ganti **xxx** dengan isian yang sesungguhnya.

```
<!--
    Nama File   : XXX.html
    NIM         : XXX
    Kelas       : XXX
-->
```

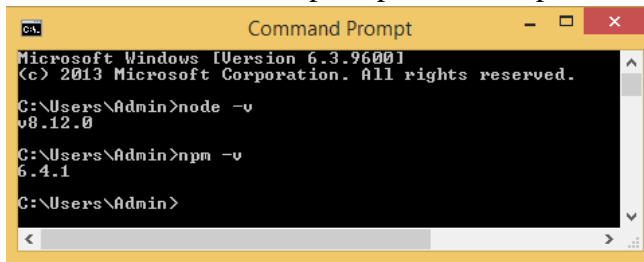
3. Buat sebuah laporan mengenai praktikum ini, beri nama **Laporan Praktikum PABWE W04S04.docx**. Isi laporan ini berupa *capture* koding anda dan tampilan hasilnya .

Langkah-Langkah Praktikum:

1. Instalasi Perangkat Lunak

Lakukan pemasangan semua perangkat lunak yang diperlukan sebagai berikut ini:

- Langkah 1: Pastikan nodeJS telah ter-install di komputer anda. Untuk melakukannya masuklah ke command prompt dan ketik perintah berikut ini.



```
C:\Users\Admin>node -v
v8.12.0

C:\Users\Admin>npm -v
6.4.1

C:\Users\Admin>
```

- Langkah 2: Install angular cli. Untuk melakukannya ketikkan perintah berikut.

```
npm install -g @angular/cli@6.0.5
```

Catt: pastikan jaringan anda terhubung dengan internet dan tidak menggunakan proxy.

- Langkah 3: Verifikasi instalasi anda dengan perintah berikut ini.

```
ng -v
```

Dengan tampilan hasil berikut ini:



```
Angular CLI: 6.0.5
Node: 8.12.0
OS: win32 x64
Angular:
...
Package      Version
-----
@angular-devkit/architect 0.6.5
@angular-devkit/core      0.6.5
@angular-devkit/schematics 0.6.5
@schematics/angular       0.6.5
@schematics/update        0.6.5
rxjs                 6.3.3
typescript           2.7.2
```

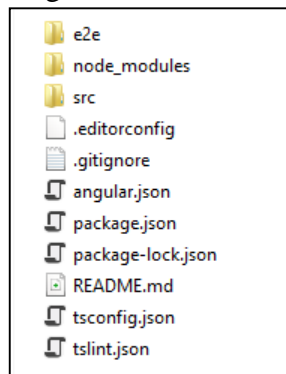
2. Welcome to Angular.

Untuk melakukannya ikuti langkah-langkah seperti berikut ini:

- Buka **konsol (command prompt –windows)** dan arahkan ke dalam *folder* w04s04. Kemudian ketikkan perintah berikut ini: (**jangan lupa mengganti kata nim dengan dua digit terakhir nim anda, dan ingat untuk menyalakan internet anda**)

```
ng new appl-nim
```

- Angular CLI akan meng-generate file proyek dengan nama di atas seperti terlihat berikut.



Silahkan matikan internet anda, dan SELAMAT anda baru saja mendapatkan file dengan ukuran ±200MB.

- c. Buka browser anda (disarankan chrome). Kembali ke konsol, masuk kedalam folder proyek yang baru saja anda ciptakan kemudian ketikkan perintah berikut.

```
ng serve --open
```

Aplikasi akan terbuka dengan port default 4200 (silahkan ganti port jika port 4200 sudah dipakai web server lain).



- d. Kembali ke folder proyek, sekarang masuk kedalam subfolder berikut:

```
app1-nim > src > app
```

Buka file *app.component.ts* perhatikan kelas berikut

```
export class AppComponent {  
  title = 'app1-nim';  
}
```

kemudian ubah teks **app1-nim** menjadi **nama** anda setelah itu perhatikan kembali browser anda. Jika browser tidak berubah berarti server anda tidak jalan, **jangan menutup konsol!**

- e. Lakukan juga perubahan pada file *app.component.html* dan file *app.component.css* seperti gambar berikut ini.

```
<!--The content below is only a placeholder and can be replaced.-->  
<div style="text-align:center">  
  <h1>  
    Selamat Datang {{ title }} di Situs ini!  
  </h1>  
    
</div>
```

```
h1{  
  color: blue;  
}
```

Kemudian lihat kembali browser anda. Jika langkah-langkah anda benar seharusnya muncul tampilan seperti berikut:



3. **Membuat aplikasi voting.** Untuk melakukannya ikuti langkah-langkah berikut ini:

- a. Dengan mengikuti langkah-langkah seperti contoh di atas, buatlah sebuah folder dengan nama **voting-npm** (ganti npm dengan 2 digit terakhir nim) di dalam folder w04s04.
- b. Masuk ke folder tersebut dan arahkan ke subfolder **src -> app** dan buka file *app.component.html*. Hapus isi file tersebut dan tuliskan tag berikut ini:

```
1 <form class="ui large form segment">
2   <h3 class="ui header">Add a Link</h3>
3   <div class="field">
4     <label for="title">Title:</label>
5     <input name="title" #newtitle> <!-- perhatikan ini -->
6   </div>
7   <div class="field">
8     <label for="link">Link:</label>
9     <input name="link" #newlink> <!-- perhatikan ini -->
10  </div>
11  <button (click)="addArticle(newtitle, newlink)" class="ui positive right floated button">
12    Submit link
13  </button>
14 </form>
```

Lupakan dulu cerita *class* pada css, untuk saat ini perhatikan tanda *hashtag* (#), karena kita akan menggunakannya untuk memberitahu **angular** bahwa tag tersebut akan dilewatkan sebagai **lokal variabel**.

- c. Sekarang beralih ke file *app.component.ts* dan tuliskan skrip berikut ini untuk membuat fungsi **addArticle**.

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   addArticle(title: HTMLInputElement, link: HTMLInputElement): boolean {
10     console.log(`Tambahkan article dengan judul: ${title.value} dan tautan: ${link.value}`);
11     return false;
12   }
13 }
```

- d. Langkah berikutnya adalah mem-*binding* fungsi **addArticle** dengan *event onClick*. Dikarenakan akan menerima parameter berupa **title** dan **link** maka ubahlah event yang memanggil fungsi tersebut pada file *app.component.html* dengan menambahkan parameter **newtitle** dan **newlink** seperti berikut ini:

```
<button (click)="addArticle(newtitle, newlink)"
  class="ui positive right floated button">
  Submit link
</button>
```

- e. Sampai disini anda sudah berhasil membuat aplikasi yang akan melewatkan parameter **title** dan **link**. Jalankan aplikasi tersebut dan lihat hasilnya di browser menggunakan mode console seperti berikut:

Pada chrome gunakan fitur lainnya -> alat pengembang -> console (atau shift+ctrl+i), kemudian isikan kedua teks input dan tekan tombol lalu lihat hasilnya seperti berikut



4. Membuat aplikasi voting extended component.

Meneruskan proyek sebelumnya sekarang kita akan membuat fungsi voting-nya dengan komponen baru di aplikasi.

- Kembali ke console di dalam folder voting-npm buatlah komponen baru dengan nama **article** sbb ini:

```
ng generate component article
```

Secara otomatis angular akan meng-generate komponen-komponen yang diperlukan dengan nama **article** tersebut.

- Masuk ke folder article (lihat *path* dibawah) dan buka file *article.component.html*.

voting-npm ▶ src ▶ app ▶ article

Ganti isi pada file tersebut dengan tag berikut ini

```
1 <div class="four wide column center aligned votes">
2   <div class="ui statistic">
3     <div class="value">
4       {{ votes }}
5     </div>
6     <div class="label">
7       Points
8     </div>
9   </div>
10 </div>
11
12 <div class="twelve wide column">
13   <a class="ui large header" href="{{ link }}">
14     {{ title }}
15   </a>
16   <ul class="ui big horizontal list voters">
17     <li class="item">
18       <a href (click)="voteUp()">
19         <i class="arrow up icon"></i>
20         upvote
21       </a>
22     </li>
23     <li class="item">
24       <a href (click)="voteDown()">
25         <i class="arrow down icon"></i>
26         downvote
27       </a>
28     </li>
29   </ul>
30 </div>
```

- c. Berikutnya kita akan mem-bind template di atas dengan action-nya. Buka file *article.component.ts* dan ubah isinya seperti berikut ini:

```
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-article',
5    templateUrl: './article.component.html',
6    styleUrls: ['./article.component.css'],
7    host: {class: 'row'}
8  })
9  export class ArticleComponent implements OnInit {
10    votes: number;
11    title: string;
12    link: string;
13
14    constructor() {
15      this.title = 'IT Del';
16      this.link = 'https://del.ac.id';
17      this.votes = 10;
18    }
19    voteUp() {
20      this.votes += 1; return false;
21    }
22    voteDown() {
23      this.votes -= 1; return false;
24    }
25    ngOnInit() {
26    }
27
28  }
```

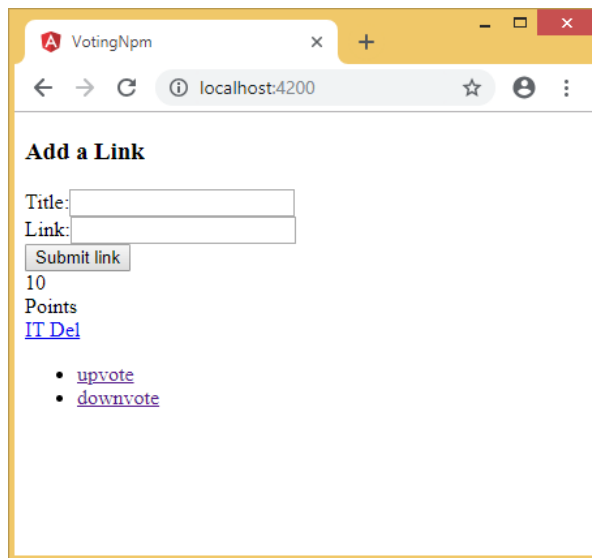
- d. Setelah membuat komponen article, sekarang kita akan panggil komponen tersebut di file utama. Untuk itu kembali ke file *app.component.html* dan tambahkan tag berikut di bawah tag form. Kemudian lihat kembali tampilan browser anda.

```
15
16  <div class="ui grid posts">
17    <app-article>
18  </app-article>
19  </div>
```

- e. Kemudian untuk memberitahukan *AppComponent* bahwa kita memiliki komponen yang bernama *ArticleComponent*, maka kita harus menuliskannya di *NgModule*. Buka file *app.module.ts* dan tambahkan kode berikut:

```
7  @NgModule({
8    declarations: [
9      AppComponent,
10     ArticleComponent // <-- ini yang ditambahkan jika belum ada
11   ],
```

Sampai disini lihat kembali tampilan browser anda dan coba untuk meng-klik tulisan **upvote** dan **downvote** untuk melihat perubahan yang terjadi.



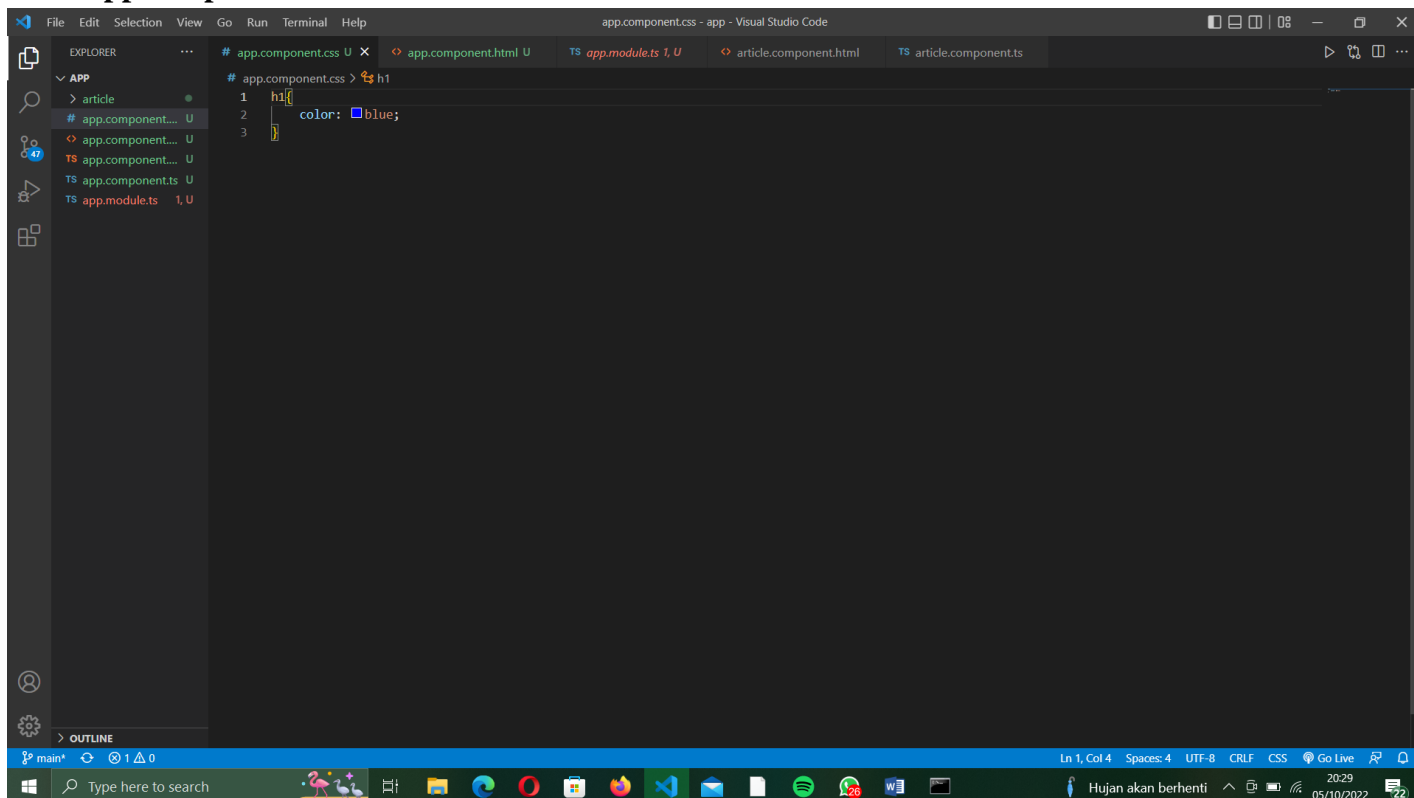
Tugas Praktikum:

1. Membuat aplikasi voting extended rendering.

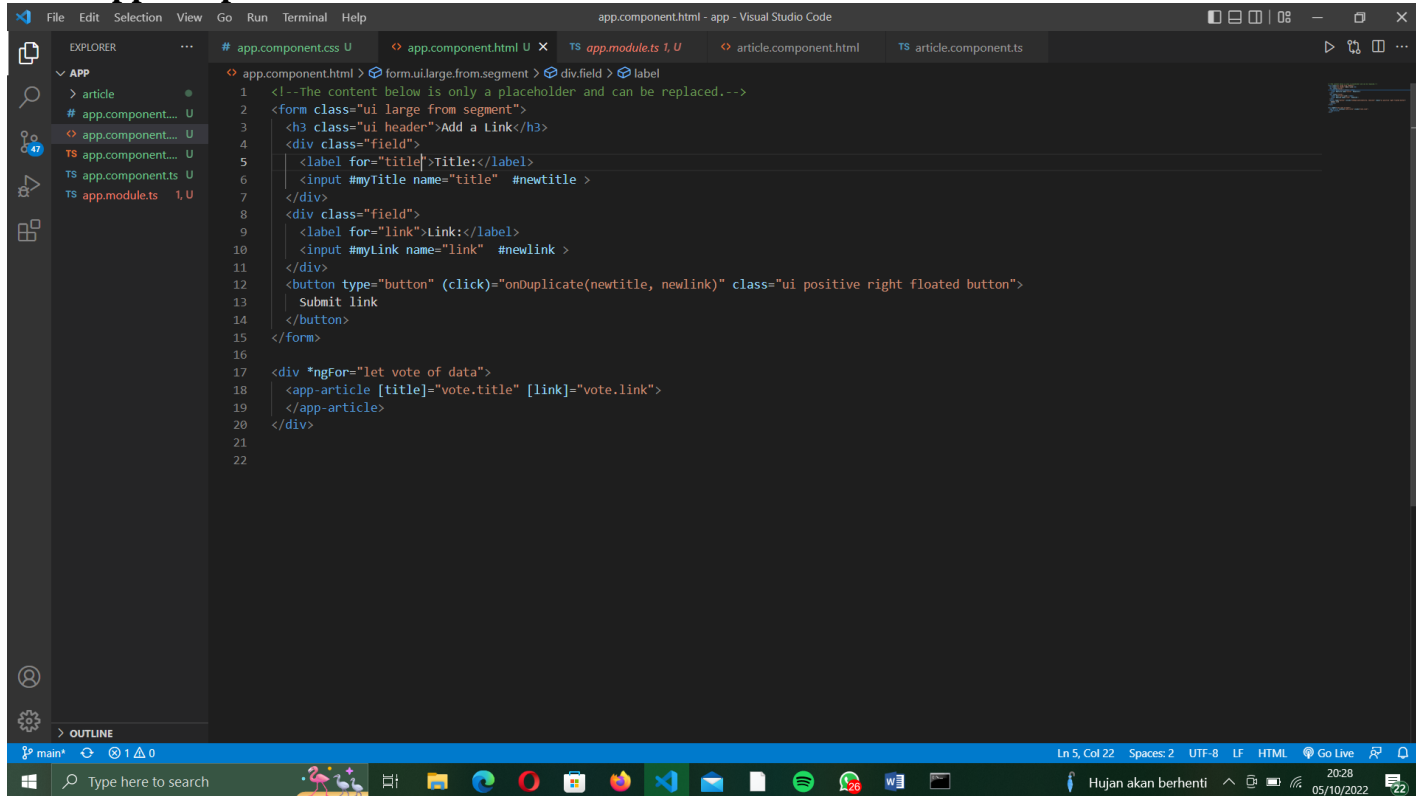
Silahkan anda dapat meneruskan aplikasi ini dengan membuat fungsi untuk meng-isolasi struktur data yang akan digunakan untuk me-render fungsi voting-nya. Dengan cara ini fungsi voting dapat mempunyai lebih dari 1 artikel (pilihan voting) dimana data artikel tersebut di dapatkan dari masukan pada **input title** dan **input link**. Dengan demikian anda dapat menyatukan kerja fungsi **Submit link** dan fungsi **voting**.

=>

Kode app.component.css

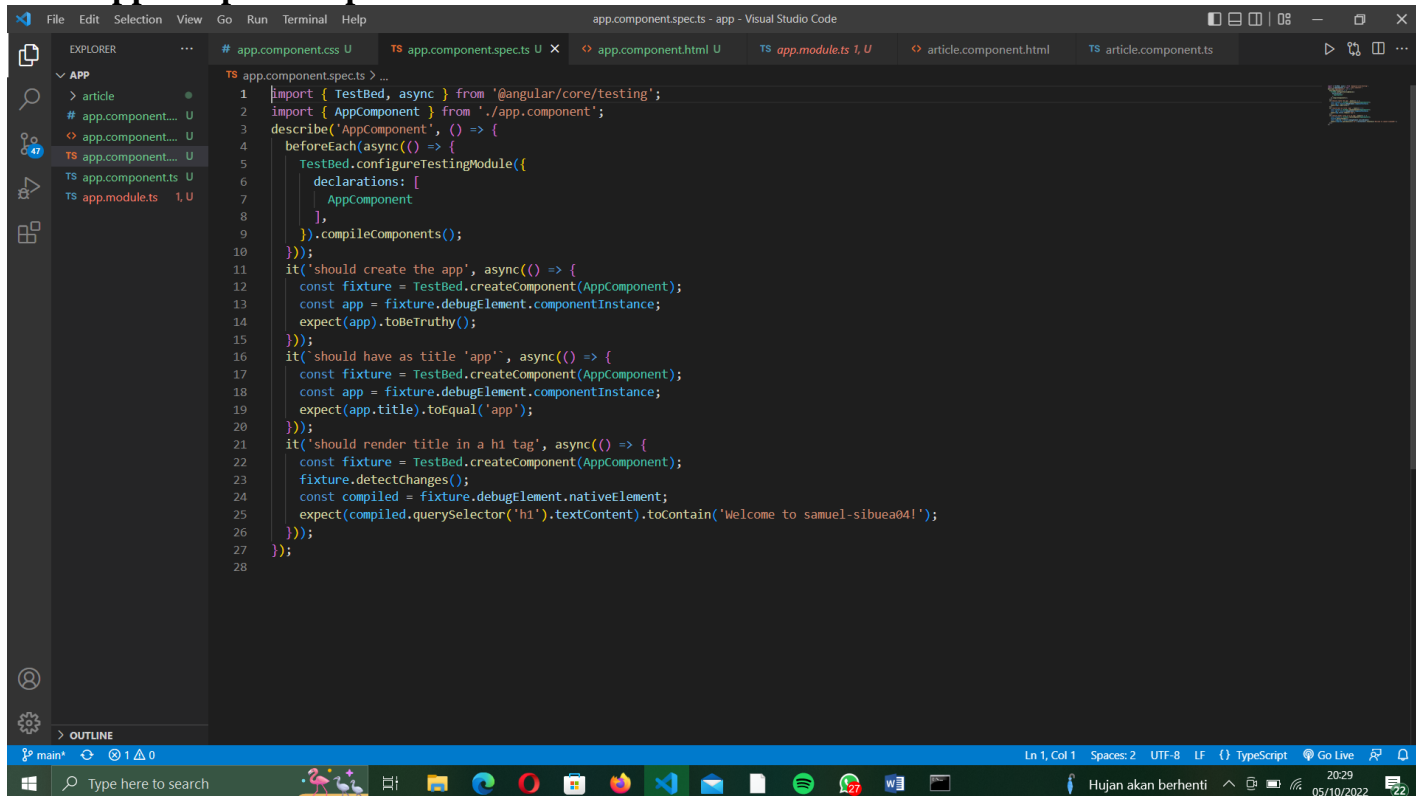


Kode app.component.html



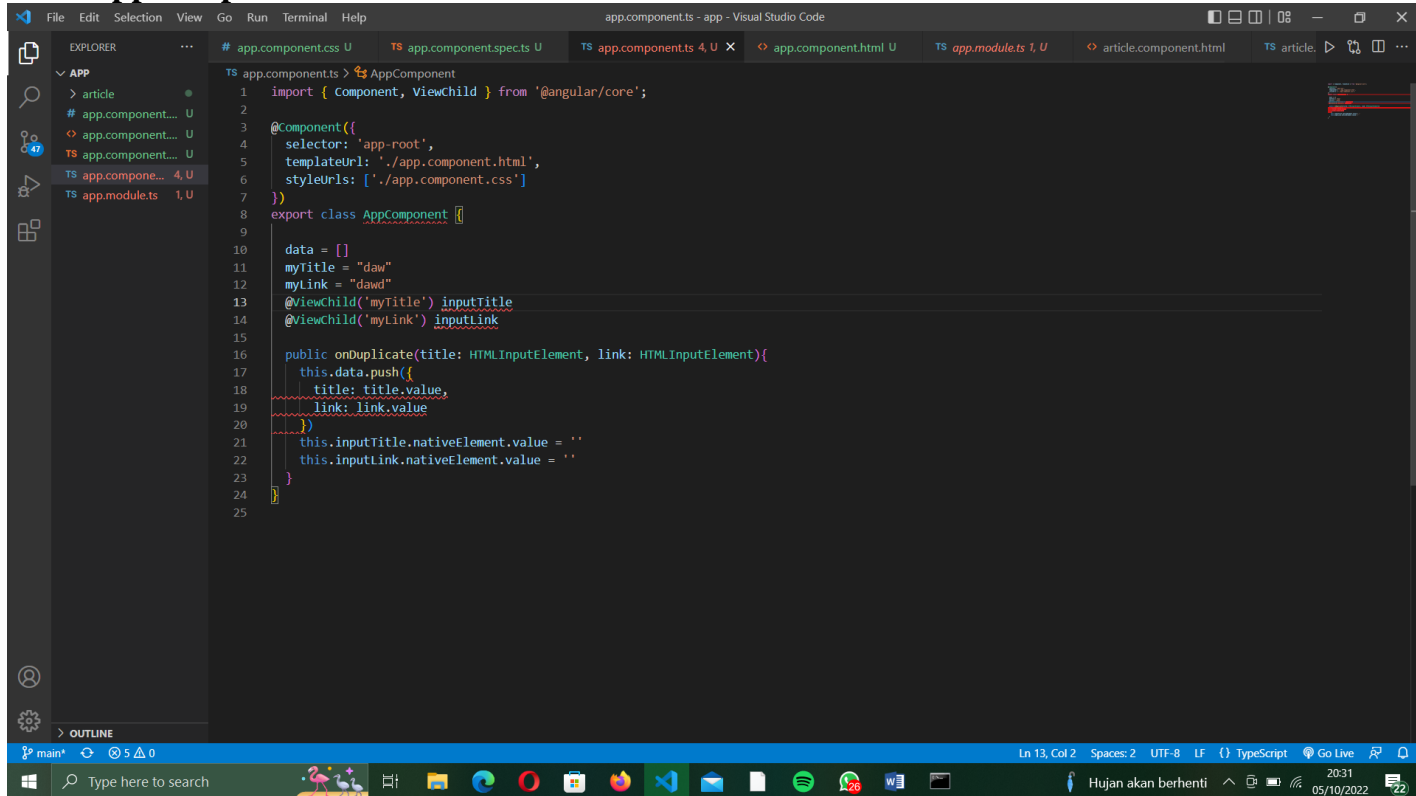
```
1 <!--The content below is only a placeholder and can be replaced.-->
2 <form class="ui large form segment">
3   <h3 class="ui header">Add a Link</h3>
4   <div class="field">
5     <label for="title">Title:</label>
6     <input #myTitle name="title" #newtitle >
7   </div>
8   <div class="field">
9     <label for="link">Link:</label>
10    <input #myLink name="link" #newlink >
11  </div>
12  <button type="button" (click)="onDuplicate(newtitle, newlink)" class="ui positive right floated button">
13    Submit link
14  </button>
15 </form>
16
17 <div *ngFor="let vote of data">
18   <app-article [title]="vote.title" [link]="vote.link">
19 </app-article>
20 </div>
21
22
```

Kode app.component.spec.ts



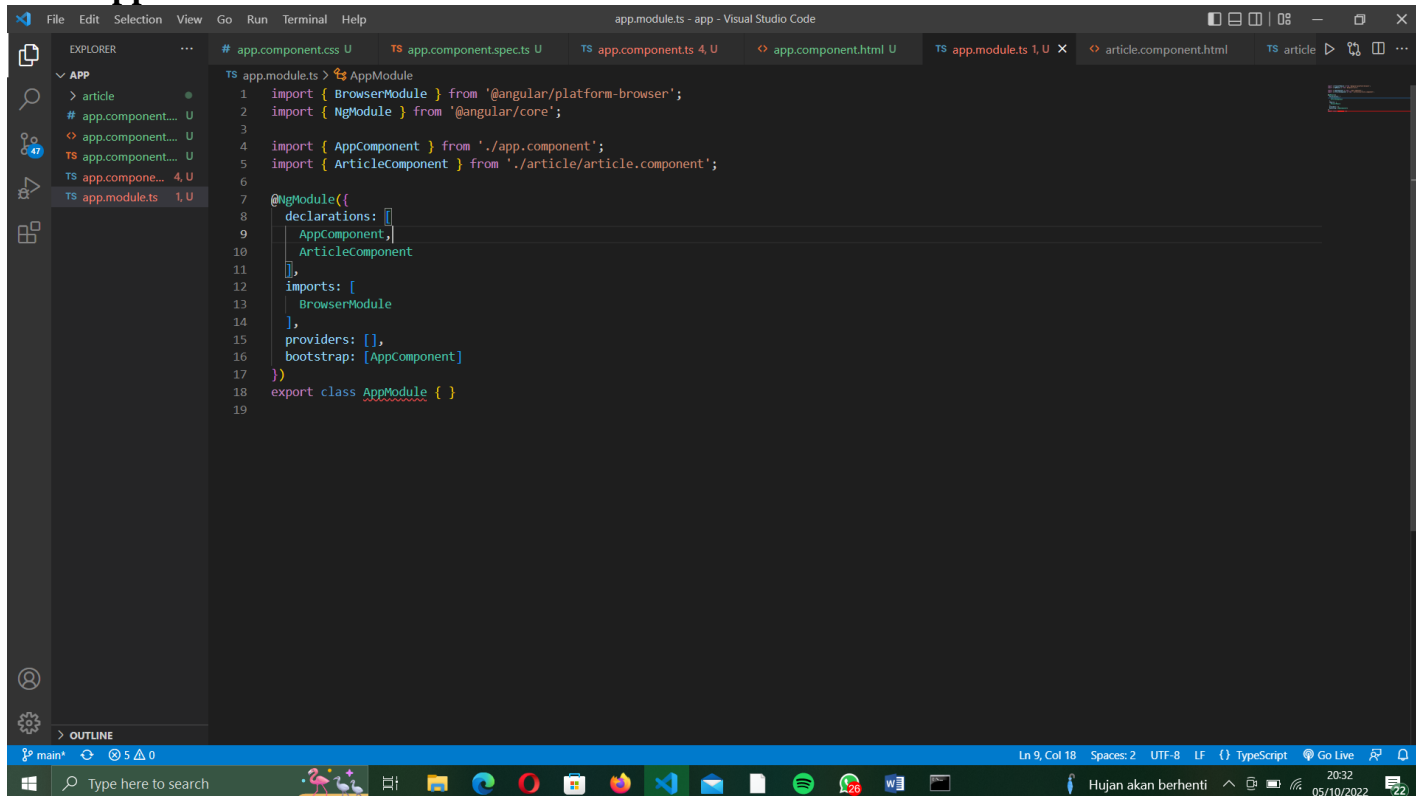
```
1 import { TestBed, async } from '@angular/core/testing';
2 import { AppComponent } from './app.component';
3 describe('AppComponent', () => {
4   beforeEach(async(() => {
5     TestBed.configureTestingModule({
6       declarations: [
7         AppComponent
8       ],
9     }).compileComponents();
10  });
11  it('should create the app', async(() => {
12    const fixture = TestBed.createComponent(AppComponent);
13    const app = fixture.debugElement.componentInstance;
14    expect(app).toBeTruthy();
15  });
16  it('should have as title \'app\'', async(() => {
17    const fixture = TestBed.createComponent(AppComponent);
18    const app = fixture.debugElement.componentInstance;
19    expect(app.title).toEqual('app');
20  });
21  it('should render title in a h1 tag', async(() => {
22    const fixture = TestBed.createComponent(AppComponent);
23    fixture.detectChanges();
24    const compiled = fixture.debugElement.nativeElement;
25    expect(compiled.querySelector('h1').textContent).toContain('Welcome to samuel-sibuea04!');
26  });
27 });
28
```

Kode app.component.ts



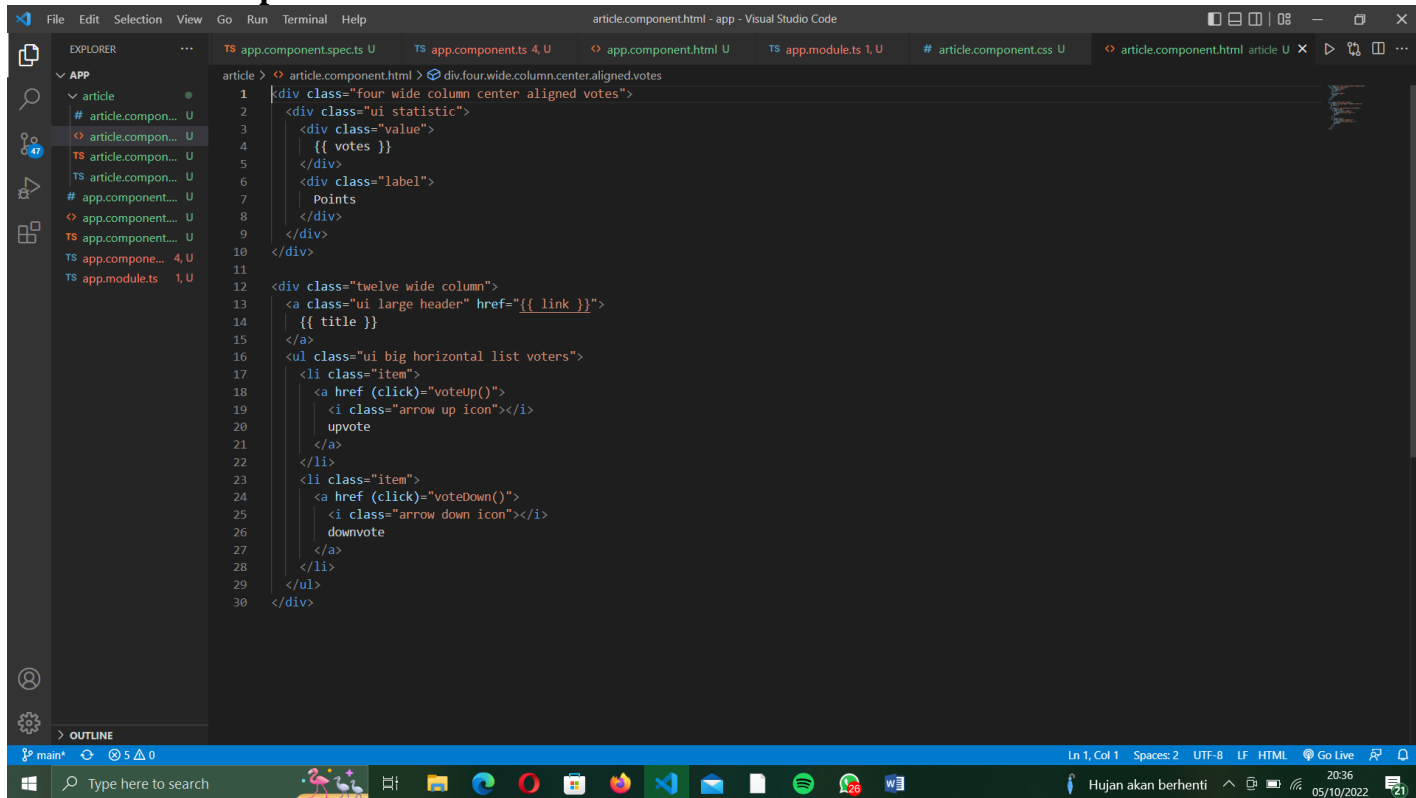
```
1 import { Component, ViewChild } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9
10  data = []
11  myTitle = "daw"
12  myLink = "dawd"
13  @ViewChild('myTitle') inputTitle
14  @ViewChild('myLink') inputLink
15
16  public onDuplicate(title: HTMLInputElement, link: HTMLInputElement){
17    this.data.push({
18      title: title.value,
19      link: link.value
20    })
21    this.inputTitle.nativeElement.value = ''
22    this.inputLink.nativeElement.value = ''
23  }
24 }
25
```

Kode app.module.ts



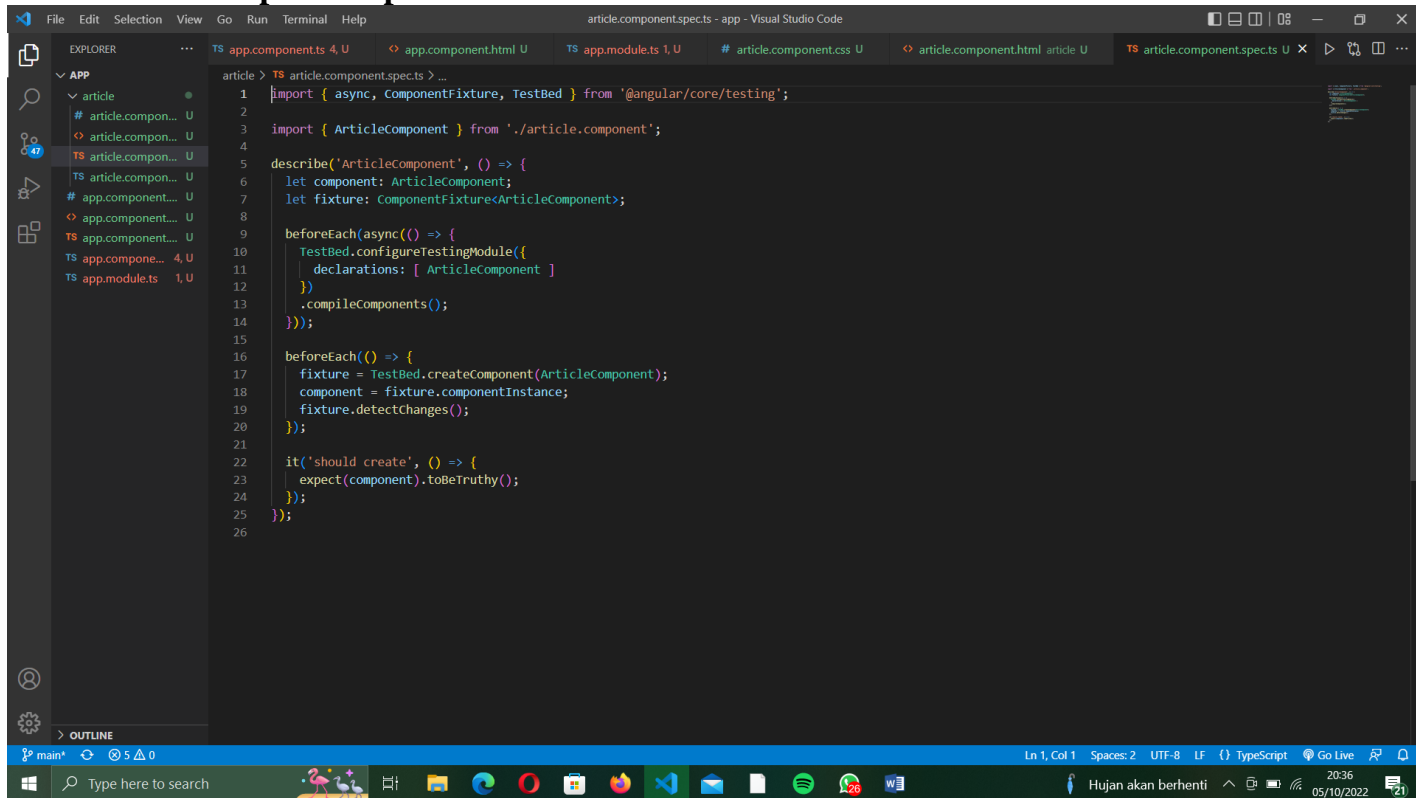
```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { ArticleComponent } from './article/article.component';
6
7 @NgModule({
8   declarations: [
9     AppComponent,
10    ArticleComponent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

Code article.component.html



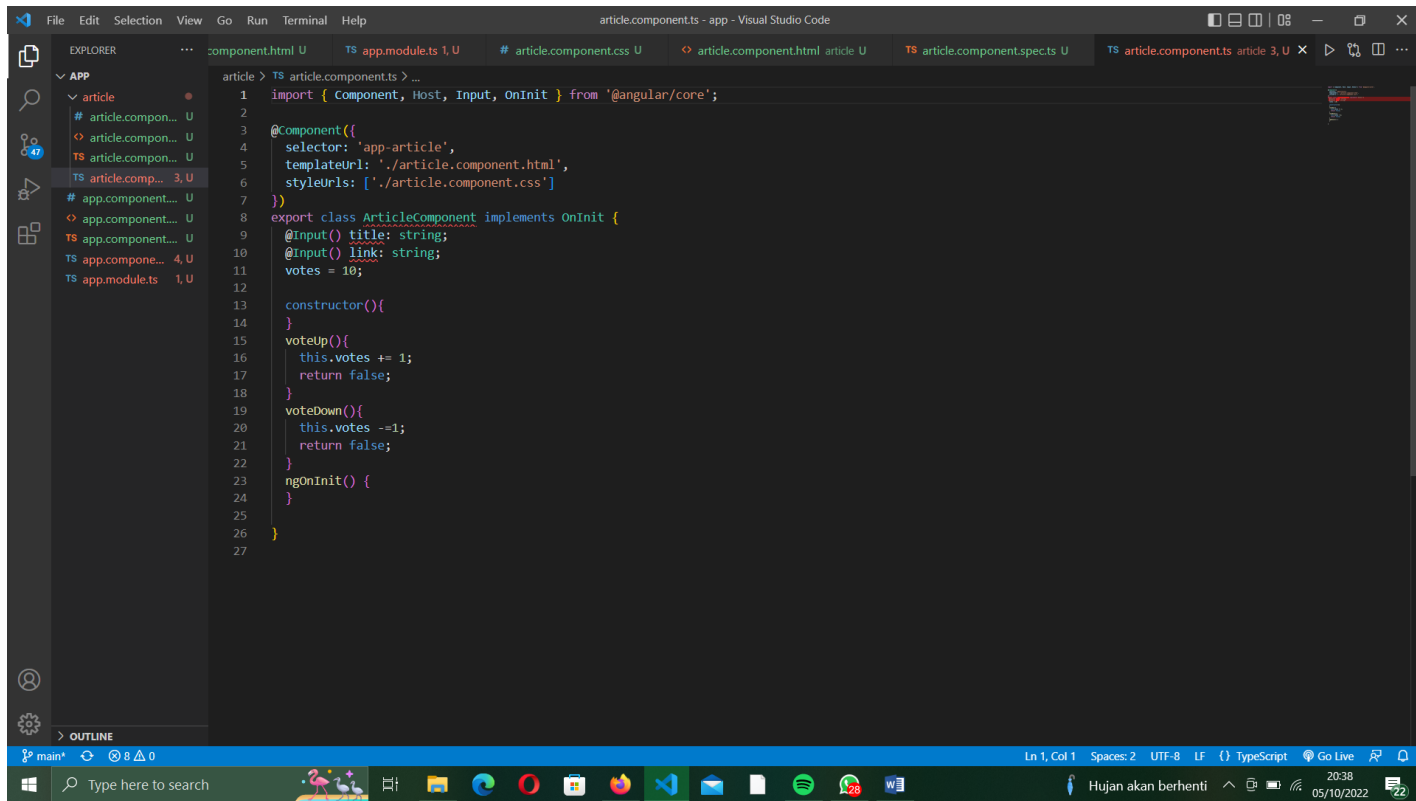
```
1 <div class="four wide column center aligned votes">
2   <div class="ui statistic">
3     <div class="value">
4       {{ votes }}
5     </div>
6     <div class="label">
7       Points
8     </div>
9   </div>
10 </div>
11
12 <div class="twelve wide column">
13   <a class="ui large header" href="{{ link }}">
14     {{ title }}
15   </a>
16   <ul class="ui big horizontal list voters">
17     <li class="item">
18       <a href (click)="voteUp()">
19         <i class="arrow up icon"></i>
20         upvote
21       </a>
22     </li>
23     <li class="item">
24       <a href (click)="voteDown()">
25         <i class="arrow down icon"></i>
26         downvote
27       </a>
28     </li>
29   </ul>
30 </div>
```

Code article.component.spec.ts



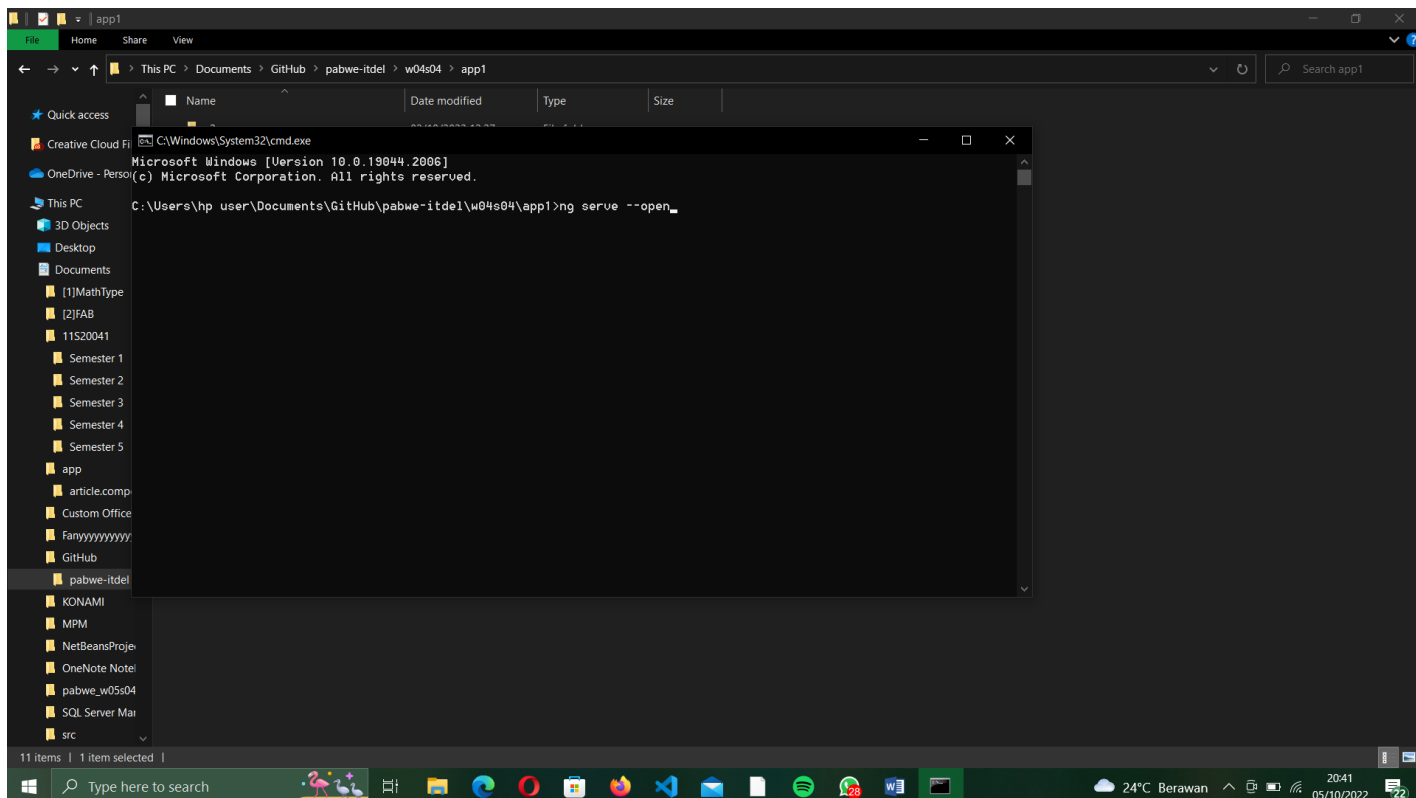
```
1 import { async, ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { ArticleComponent } from './article.component';
4
5 describe('ArticleComponent', () => {
6   let component: ArticleComponent;
7   let fixture: ComponentFixture<ArticleComponent>;
8
9   beforeEach(async(() => {
10     TestBed.configureTestingModule({
11       declarations: [ ArticleComponent ]
12     })
13     .compileComponents();
14   }));
15
16   beforeEach(() => {
17     fixture = TestBed.createComponent(ArticleComponent);
18     component = fixture.componentInstance;
19     fixture.detectChanges();
20   });
21
22   it('should create', () => {
23     expect(component).toBeTruthy();
24   });
25 });
```

Kode article.component.ts



```
1 import { Component, Host, Input, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-article',
5   templateUrl: './article.component.html',
6   styleUrls: ['./article.component.css']
7 })
8 export class ArticleComponent implements OnInit {
9   @Input() title: string;
10  @Input() link: string;
11  votes = 10;
12
13  constructor(){}
14
15  voteUp(){
16    this.votes += 1;
17    return false;
18  }
19  voteDown(){
20    this.votes -= 1;
21    return false;
22  }
23  ngOnInit() {
24  }
25
26 }
27
```

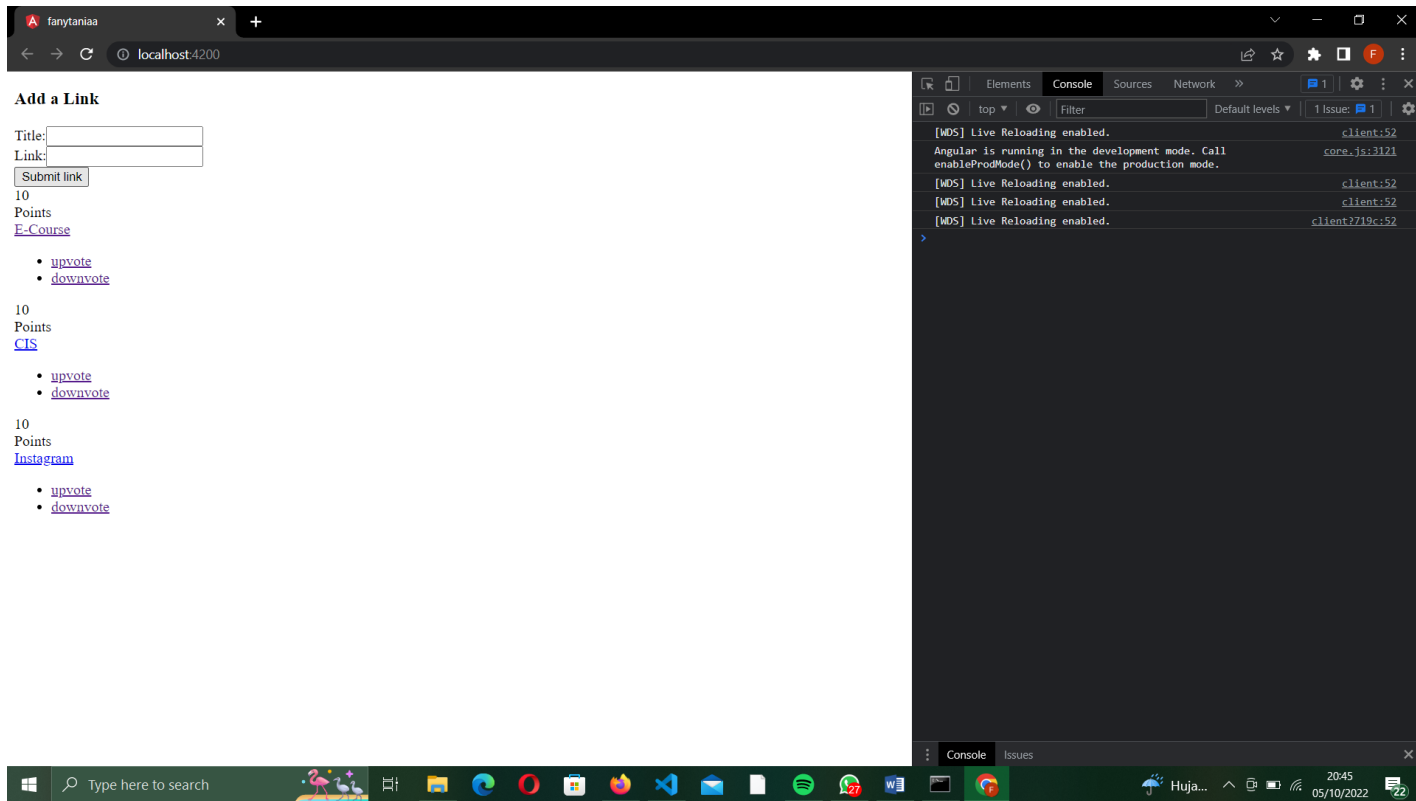
Setelah itu di bagian app1 cmd kan lalu mengetik ng serve --open seperti ini :



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp user\Documents\GitHub\pabwe-itdel\w04s04\app1>ng serve --open
```

Lalu menghasilkan output seperti ini :



Pengumpulan Praktikum:

1. Kumpulkan sofcopy ke ecourse dalam bentuk folder (file-file html dan laporan) yang telah di kompres (**PrakPABWE-0604_NIM**) dengan password berupa dua digit terakhir NIM anda.

Salam & Terima Kasih