|  |  |
| --- | --- |
|  | Jurusan Teknologi Informasi Politeknik Negeri Malang  **Jobsheet-11: Authentication and Authorization**  **Mata Kuliah Pemrograman Web Lanjut**  Pengampu: Tim Ajar Pemrograman Web Lanjut  *Mei 2019* |

**Topik**

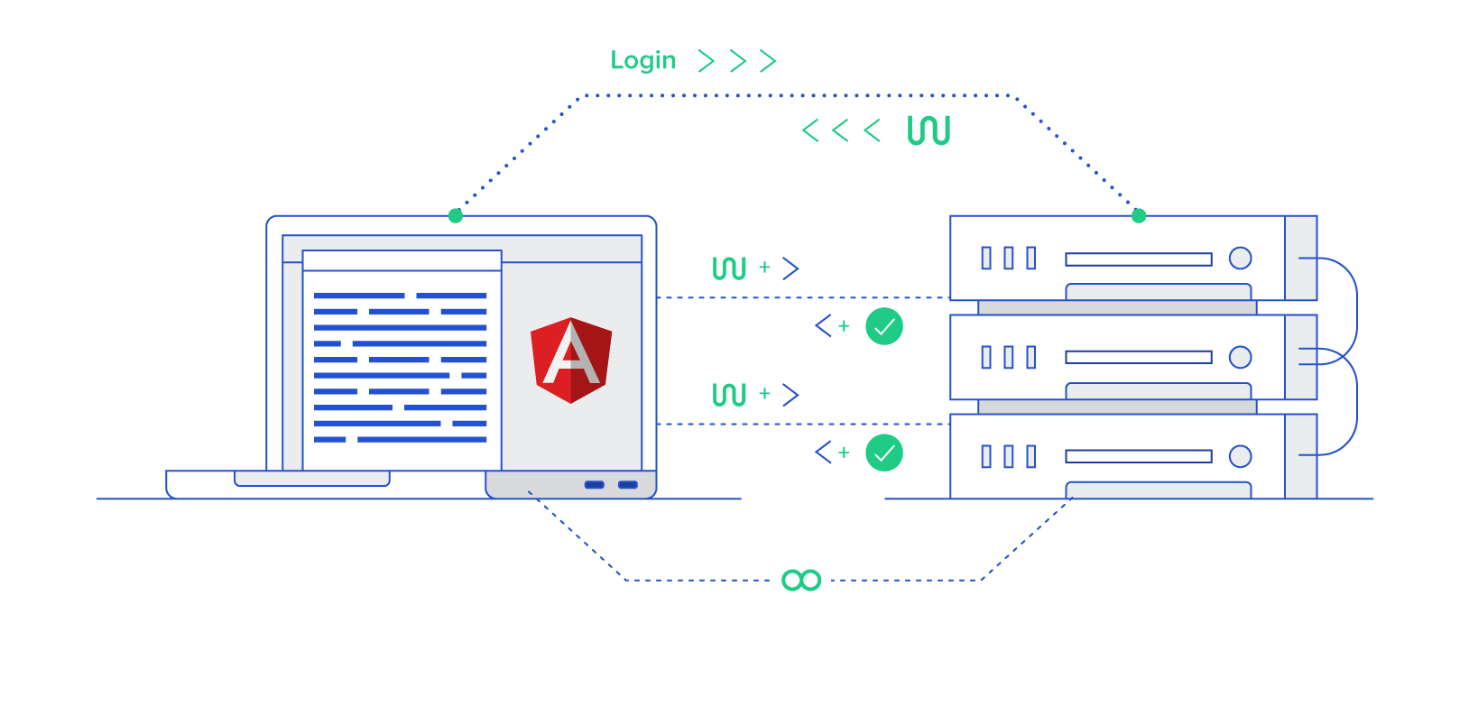
* Authentication and Authorization

**Tujuan**

Mahasiswa diharapkan dapat:

1. Memahami konsep Autentikasi dan Autorisasi
2. Mengimplementasikan Autentikasi dan Autorisasi pada aplikasi angular

**Pendahuluan**

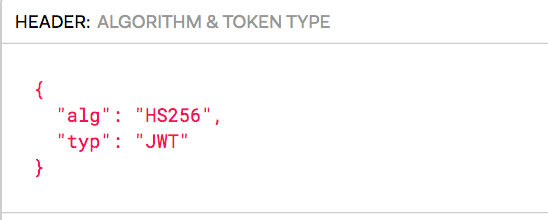


Json Web Tokens (JWT) sebuah token berbentuk string panjang yang sangat random yang gunanya sendiri untuk melakukan sistem Autentikasi dan Pertukaran Informasi. Umumnya untuk melakukan login tidak seperti pada aplikasi website biasa dimana kita menggunakan session untuk mengingat siapa yang sedang Login. Tapi didalam API sendiri kita menggunakan konsep JWT atau dibacanya sebagai "jot". Website resminta ada di jwt.io.

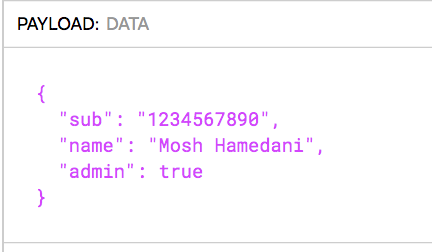
Dimana JWT atau Token ini seperti password jadi ketika users berhasil melakukan Login maka server akan memberikan sebuah Token. Nanti Token tersebut akan disimpan oleh users pada Local Storage atau Cookies Browser dan bila users ingin mengakses halaman halaman tertentu maka harus menyertakan token tersebut.  Untuk itu users akan mengirim balik token yang dikasih diawal tadi sebagai bukti bila user ini, sudah melakukan login.

Tokennya dimana terdiri dari tiga bagian yaitu yang pertama header lalu kedua bagian payloadnya atau datanya dan yang ketiga adalah bagian verify signature.

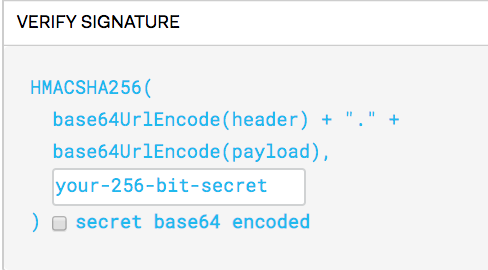
- **Headernya** sendiri hanya terdiri dari Algoritma HS256 yang kita gunakan dan Typenya jwt sebagai defaultnya.



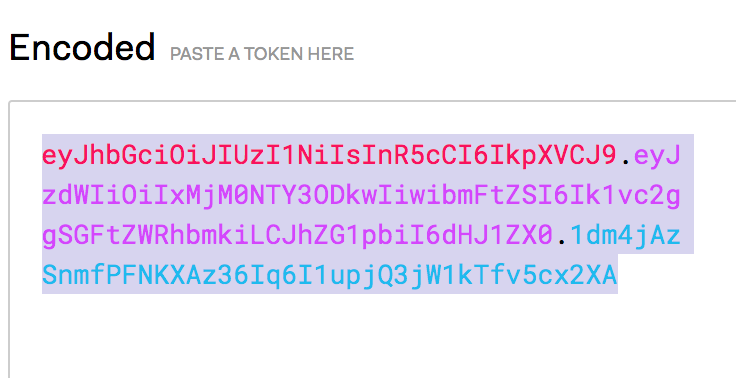
- **Payloadnya** sebagai infomasi atau data yang ingin kita kirimkan untuk users misalnya \_id usernya atau tanggal expirednya dan lain lain.



- **Verify Signature** adalah hasil dari Hash atau gabungan dari isi encode Header dan Payloadnya lalu ditambahkan kode secretnya.



Dan hasil ketiga bagian tersebut akan digabung dan otomatis di encode menjadi Token string random panjang seperti berikut



Dan apabila isi Header atau Payload dirubah maka isi Signature menjadi tidak valid.

**Praktikum – Bagian 1: Implementation Login**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buka folder auth-demo-starter kemudian open terminal dan lakukan ng serve. Catat hasil nya (soal 1) |
| 2 | Apabila terjadi error seperti dibawah ini :  Maka lakukan perintah untuk menginstall angular2-jwt, dengan perintah :  **npm../../../../../Desktop/Screen%20Shot%202019-04-22%20at%2017.01.37.png install angular2-jwt –save**  kemudian lakukan  **ng serve –open**  catat hasilnya (soal 2) |
| 3 | ../../../../../Desktop/Screen%20Shot%202019-04-22%20at%2017.10.16.png  Ketika kita klik menu **Login** untuk masuk ke aplikasi, hasilnya akan kembali  ke menu semula dengan kondisi [NAME] tidak ada perubahan. Sehingga akan ditambahkan autentikasi pada menu tersebut. |
| 4 | Buka file auth.service.ts lalu tambahkan perintah seperti berikut ini :  ../../../../../Desktop/Screen%20Shot%202019-04-23%20at%2015.42.26.png  ../../../../../Desktop/Screen%20Shot%202019-04-23%20at%2015.41.56.png |
| 5 | Jika sudah ditambahkan, jalankan perintah ng serve, dan klik bagian login lalu masukkan email address : [mosh@domai.com](mailto:mosh@domai.com) dan password 1234 kemudian klik sign in. perhatikan apa yang terjadi? Catat hasilnya (soal 3) |
| 6 | Kemudian lakukan login dengan email yang formatnya tidak valid contoh email kita isi 1234 dengan password 1234 maka ketika kita login dan kita inspect response apa yang muncul ?  Catat hasilnya (soal 4) |
| 7 | Buka file auth.service.ts kemudian lakukan perubahan seperti berikut :  ../../../../../Desktop/Screen%20Shot%202019-04-27%20at%2011.20.19.png |
| 7 | Kemudian jalankan dan lakukan login dengan username dan password yang sesuai pada no 5, kemudian lakukan inspect pilih menu application liat di local storage, apa yang terlihat ? catat hasilnya (soal 5) |

**Praktikum – Bagian 2: Implemetasi Logout**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Open file [home.component.html](http://home.component.html), tambahkan  ../../../../../Desktop/Screen%20Shot%202019-04-27%20at%2011.48.06.png |
| 2 | Open file auth.service.ts lalu tambahkan pada bagian logout :  ../../../../../Desktop/Screen%20Shot%202019-04-27%20at%2011.49.39.png |
| 3 | Jalankan aplikasi, lalukan login dengan username dan password seperti sebelumnya dan lakukan inspect, cek pada local storage harusnya token nya sudah ada di local storage. Kemudian lakukan logout. Apa yang terjadi pada local storage ?  Catat hasil (soal 6) |

**Praktikum - Bagian 3 : Getting the Current User**

|  |  |
| --- | --- |
| **Langkah** | **Keterangan** |
| 1 | Buka <https://jwt.io>, kita akan membuat token yang nantinya akan kita tempelkan di fake-backed.ts  ../../../../../Desktop/Screen%20Shot%202019-04-29%20at%2005.36.37.png../../../../../Desktop/Screen%20Shot%202019-04-29%20at%2005.36.25.png  lalu copy code hasil diatas. |
| 2 | Open auth.service.ts, lakukan modifikasi pada part isLoggeedIn() sehingga menjadi seperti berikut :  ../../../../../Desktop/Screen%20Shot%202019-04-29%20at%2005.38.00.png |
| 3 | Kemudian tambahkan code berikut tepat dibawah langkah no 2 :  get **currentUser**(){  let token = localStorage.**getItem**('token');  if (!token) return null;  return new **JwtHelper**().**decodeToken**(token);  }  } |
| 4 | Open fake-backend.ts, ganti token yang lama menjadi token pada no 1 |
| 5 | Open file home.component.html , modifikasi seperti berikut :  ../../../../../Desktop/Screen%20Shot%202019-04-29%20at%2005.41.09.png |
| 6 | Jalankan aplikasi. Cek apa yang terjadi dan beri penjelasan (soal 7) |

**Daftar pustaka :**

<https://nodejs.org/en/>

<https://angular.io/>

<https://www.udemy.com/the-complete-angular-master-class/learn/v4/content>