PCS #5

Membuat JWT (JSON WEB TOKEN)

Goal

1. Mahasiswa mampu memahami konsep Token JWT (JSON Web Token)
2. Mahasiswa mampu memahami penerapan Token JWT pada REST API

Teori

JWT merupakan sebuah token berbentuk string random yang berfungsi untuk mengerjakan metode otentikasi dan pertukaran data. Biasanya untuk menjalankan login kita umumnya menerapkan session untuk mencatat siapa yang sedang login. Tapi didalam API sendiri kita akan menerapkan konsep JWT. Laman resminya ada di jwt.io

1. Apa itu JWT?

   Seperti yang telah dibahas sebelumnya, JWT merupakan singkatan dari JSON Web Token, yang berarti token ini memakai tipe data JSON (Javascript Object Notation), kemudian token ini memungkinkan kita untuk mengirimkan data yang bisa diverifikasi oleh dua pihak atau lebih.

2. Komponen
   a. Header

      Komponen pertama disebut dengan header. Header berisi data perihal algoritma dan tipe token yang diaplikasikan.

   b. Payload

      Komponen kedua adalah payload. Payload biasanya berisi informasi yang ingin disubmit via token.

   c. Signature

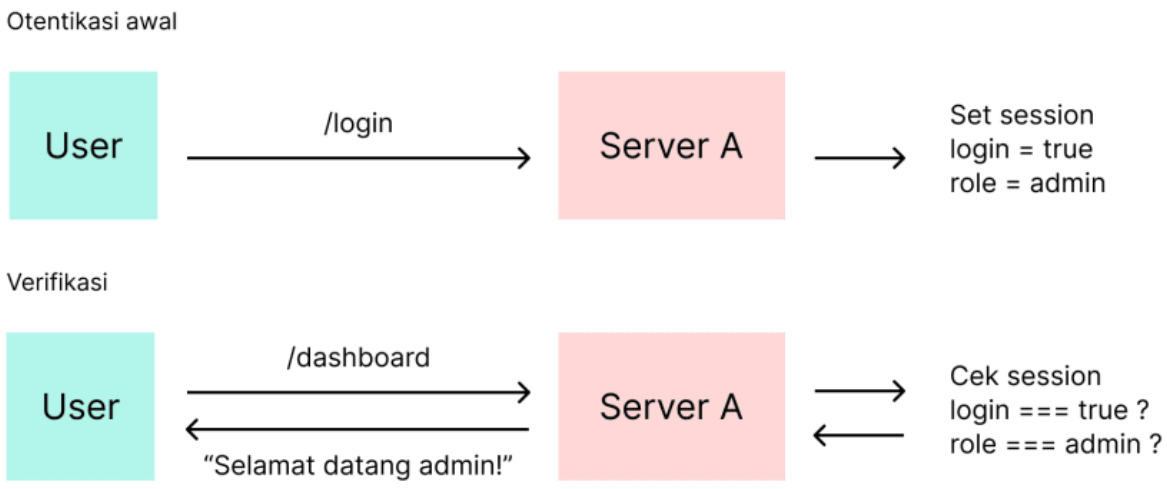
Komponen ketiga merupakan signature. Signature merupakan hash gabungan dari header, payload dan sebuah secret key (berupa string random panjang)

3. Penerapan Oleh Dua Pihak atau Lebih

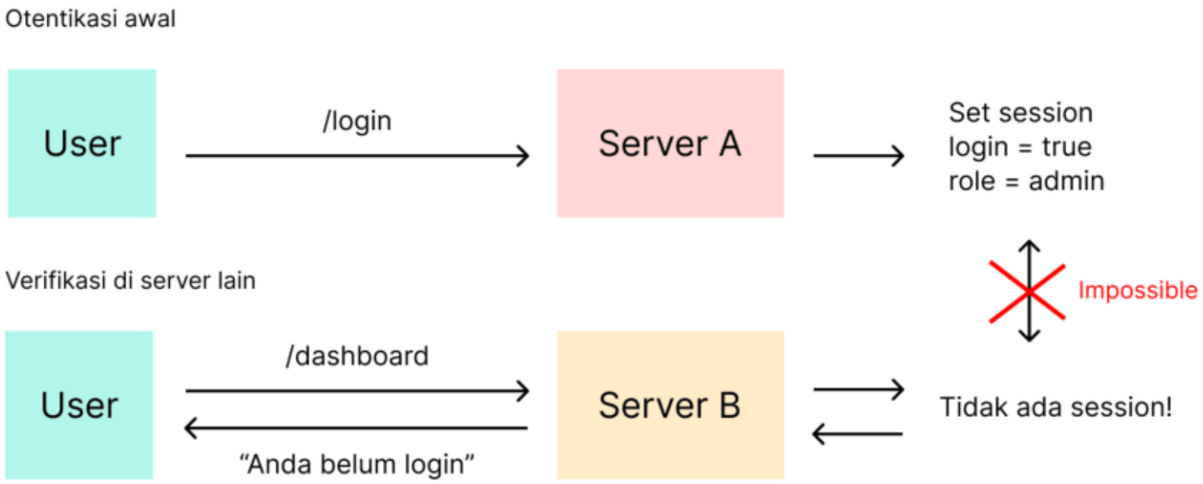
Ini adalah alasan kenapa JWT diciptakan. Pada proses otentikasi/otorisasi biasanya kita menerapkan session, yang mana saat user login di sebuah situs, maka server akan merekam data user tersebut.

Data session yang tersimpan itu akan digunakan untuk melakukan verifikasi otentikasi; memastikan user sudah login atau belum, dan otorisasi; memastikan hak akses user yang login.

```php
$sudah_login = $_SESSION['login'] === true;
$adalah_admin = $_SESSION('role') === 'admin';
if ($sudah_login) {
  if ($adalah_admin) {
    echo 'Selamat datang admin!';
  } else {
    echo 'Selamat datang user!';
  }
} else {
  echo 'Silahkan login';
}
```



Sejauh ini tak ada permasalahan. Tetapi, bila Anda mempunyai dua aplikasi atau lebih dan bermaksud agar dua aplikasi tersebut dapat berbagi session, tentu tak memungkinkan untuk menerapkan session, sebab session user hanya disimpan oleh situs yang diakes user pada login sebelumnya.



Dikarenakan valid atau tidaknya JWT dapat diverifikasi secara mandiri dengan signature, ini memungkinkan untuk aplikasi lain untuk dapat menggunakan token yang sama asalkan mempunyai secret key yang sama.

Setelah proses verifikasi JWT pada server B sukses, kita dapat lanjut mengerjakan proses pengecekan hak akses dengan memakai data dari payload; UUID, email, dan role, untuk dicocokkan dengan data yang ada di database, dan seterusnya.

4. Kesimpulan
    a. JWT dapat memastikan integritas dari data yang dikirim (Data yang ada di dalam token tak bisa diubah).
    b. JWT dapat digunakan untuk proses otentikasi/otorisasi oleh dua aplikasi yang berbeda.

Praktek

1. Pastikan anda sudah menyelesaikan praktek dari pertemuan 1-4.
2. Link Video Pertama - Melengkapi REST API ⟶ https://youtu.be/KH2ccOJKt-U
3. Link Video Kedua- Membuat JWT ⟶ https://youtu.be/RBd0-jpNcxI

   Didalam video tersebut anda akan diminta untuk :

   1). Download Library, dan berikut ini link downloadnya https://drive.google.com/file/d/1RGCL4f13h_6kkpAPiwGctu2XZMjsLvFz/view?usp=sharing

Sumber / Referensi :

1. Achmad Fauzi (2021). SQL for Beginners: Learn SQL using MySQL and Database Design, buildwithangga.com
2. AmperaKoding (2021). Membangun RestFul API di Codeigniter 3. Youtube
3. https://medium.com/@bojanmajed/standard-json-api-response-format-c6c1aabcaa6d
4. https://ruangkoding.id/apa-itu-jwt/