



# Studi Kasus

Membuat Rest API dengan Autentikasi & Keamanan



# Goal dari materi ini ?

- **Memahami struktur project**
- **Memahami cara kerja autentikasi & keamanan pada API**
- **Memahami pembuatan rest api dari awal**
- **Memahami logging out**
- **Memahami relationship pada mongodb**
- Memahami Sorting, Filtering, pagination
- Memahami Upload



# Apa yang akan di pelajari ?

- Middleware
- Bcrypt
- JWT (JSON Web Token)
- Mongoose
- Express



# Apa itu **JWT** ?

merupakan sebuah token berbentuk JSON, yang ukurannya sangat padat dari segi ukurannya, maksudnya yaitu, token JWT dapat dikirim melalui URL



```
<base64-encoded header>.<base64-encoded payload >.<base64-encoded signature>
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ
```



**Header** memiliki attribute alg (algoritma yang digunakan untuk signature) dan "typ"(tipe token).

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

JSON tersebut di encode dengan Base64Url, dan menjadi bagian pertama dari JWT.



**Payload** terdiri dari attribute iss (issuer), iat (expiration time), sub(subject), dan lain-lain.

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

JSON tersebut di encode dengan Base64Url, dan menjadi bagian kedua dari JWT.



**Signature** adalah gabungan dari header, payload, secret key yang di sign dengan sebuah algoritma yang sudah ditetapkan. Misalkan kita menggunakan algoritma HMAC SHA256,

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) secret base64 encoded
```





# Thanks!

Any questions?

You can find me on github :

**@randyviandaputra**