



# Authentication & Authorization

**Silver**- Chapter 3 - Topic 6

---

**Selamat datang di Chapter 3 Topic 6 pada course  
React Native dari Binar Academy!**





**Good morning!** 🙌

Hari ini kita masuk di topik terakhir dari Chapter 3 React Native. Kalau di topik 5 kemarin kita sudah belajar tentang React Native HTTP Request, di topik 6 ini kita bakal belajar tentang **Authentication dan Authorization**.

Siap-siap sikat ilmu baru ya!



**Detailnya, kita bakal bahas hal-hal berikut ini:**

- Konsep dan alur Authentication dan Authorization
- Cara mengakses data dengan menggunakan Authorization





**Authentication** dan **authorization** adalah dua hal yang penting untuk menjaga keamanan data sebuah aplikasi.

Mari kita pelajari konsep dan alur penggunaannya, guys!





### Hati-hati, guys!

Dalam penggunaan sistem yang berbasis internet atau aplikasi, nggak bisa dipungkiri bahwa banyak pencurian data pengguna yang dilakukan.

Banyak orang-orang yang nggak bertanggung jawab berusaha menembus data pada suatu sistem, lalu menyalahgunakan data tersebut untuk keuntungan pribadinya.





### Kayak apa contohnya?

Yang paling sering jadi sasaran empuk pencurian data adalah data rekening tabungan. Banyak modus dilakukan mulai dari email palsu, click bait, dan sebagainya, yang tujuannya adalah menguras uang dalam tabungan si calon korban.

Karena itulah, setiap sistem yang menyimpan data-data pribadi **harus punya sistem autentikasi sebagai lapisan keamanan.**





### Apa itu autentikasi?

Autentikasi (authentication) adalah **proses verifikasi identitas** seseorang atau pengguna perangkat tertentu saat akan masuk ke dalam suatu sistem.

Verify your identity





### Apa manfaat verifikasi data?

Proses verifikasi identitas biasanya menggunakan Username dan Password pengguna.

Tujuan verifikasi adalah sebagai informasi untuk mengetahui apakah orang yang mengakses itu adalah **pemilik akun asli yang sah dan berhak masuk ke dalam sistem.**





### Hmm.. terus apa bedanya autentikasi dengan otorisasi?

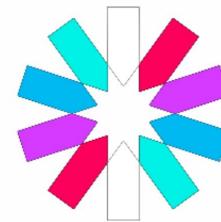
Kalau autentikasi mengarah pada proses verifikasi identitas, maka authorization (otorisasi) adalah proses untuk **memverifikasi seseorang** dalam perihal pengaksesan data.





### Nah pertanyaannya adalah otorisasi bentuknya yang kayak apa ya?

Otorisasi yang dibutuhkan supaya kamu bisa mengakses sebuah Private API **berbentuk JSON Web Token**, guys, atau yang sering disingkat JWT.



J W U T

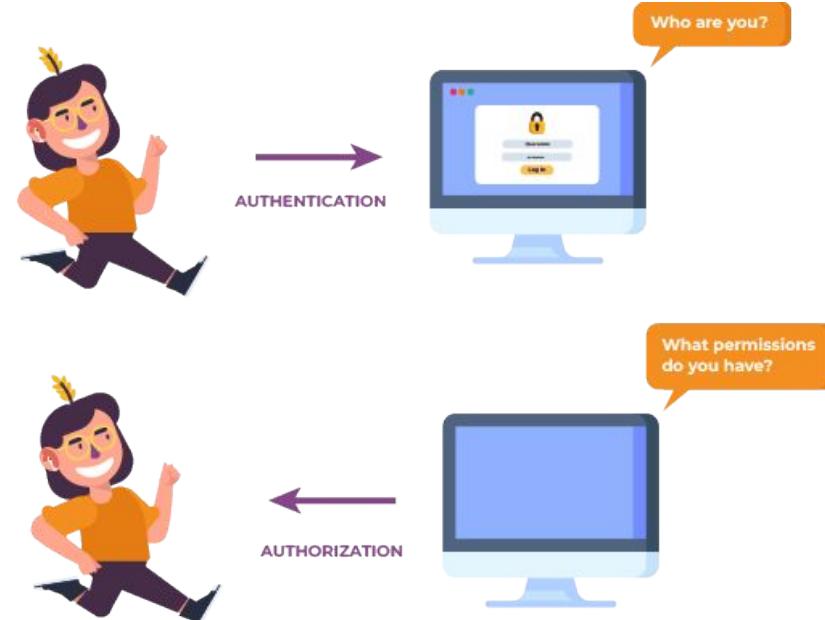


Dalam teknologi digital authentication dan authorization punya satu tujuan yang sama, yaitu untuk melakukan **verifikasi identitas diri** dan **verifikasi akses terhadap data-data yang bisa di akses oleh identitas tersebut**.



## Beda antara authentication dan authorization

| Authentication  | Authorization  |
|---|--|
| Authentication adalah <b>proses pemeriksaan identitas pengguna</b> untuk menyediakan akses ke sistem. | Authorization adalah <b>proses menentukan resources atau data apa yang bisa diakses</b> oleh pengguna. |
| Authentication dilakukan sebelum authorization.   | Authorization hanya bisa dilakukan oleh pengguna yang telah terautentifikasi.                          |





**Kita sudah bahas sebelumnya, ada 2 jenis API, yaitu Private dan Public.**

### Private API

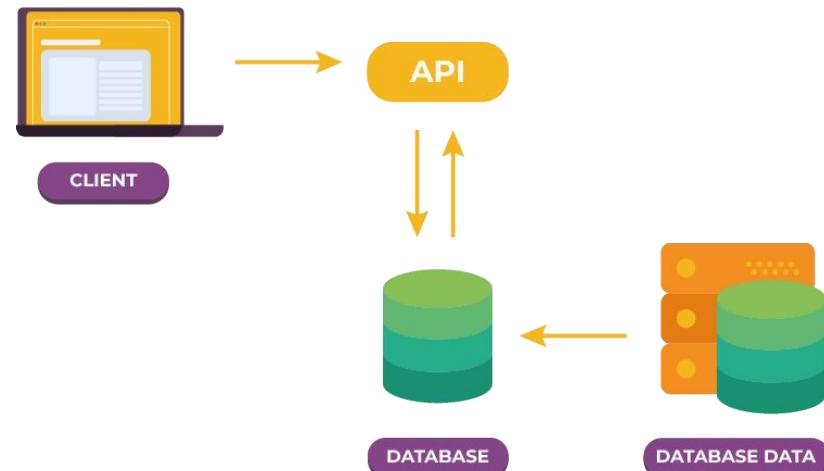
Private API adalah RESTful API yang nggak bisa di akses oleh semua orang.

Cuma orang-orang yang terdaftar dalam sistemlah yang bisa akses RESTful API itu.

### Public API

Public API adalah RESTful API yang bisa di akses oleh semua orang.

Contohnya RESTful API BMKG yang bisa diakses oleh siapapun, tanpa ia harus terdaftar pada server BMKG.



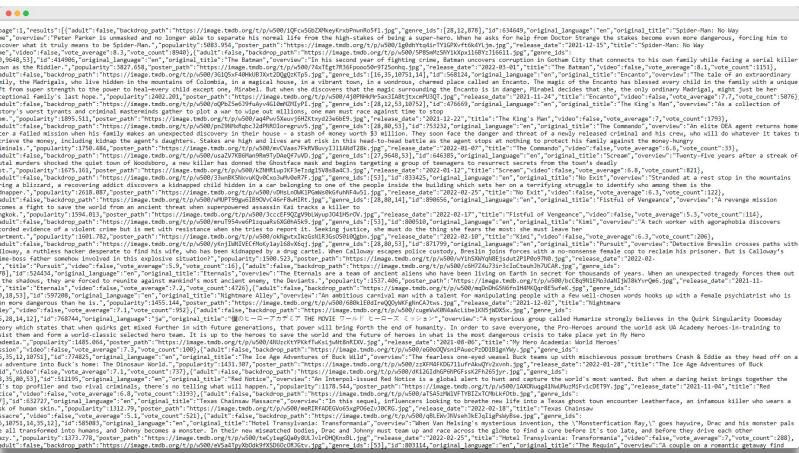


## Lalu apa kaitan API sama authentication dan authorization?

Nah, jadi kalau kamu **mengakses Public API**, kamu nggak perlu **melakukan verifikasi identitas diri** untuk bisa mengakses data-data pada server tersebut.

klik [di sini](#) untuk melihat contoh Public API.

Pas kalian buka URL API tersebut, maka server akan langsung ngasih kamu respon dalam bentuk data. Nah, nanti tampilannya kayak gambar di samping ya, guys!

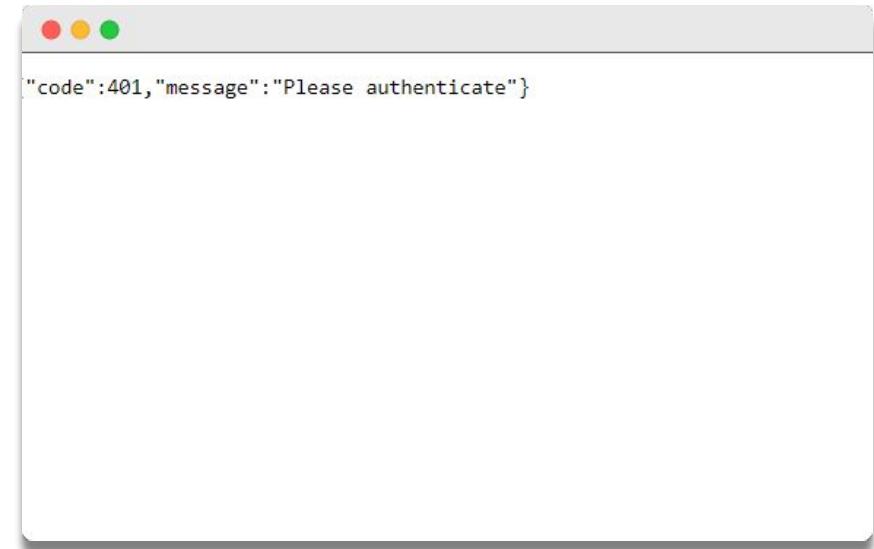




Lain ceritanya kalau kamu **mengakses Private API. Private API butuh verifikasi identitas** diri supaya kalian bisa mengakses data-datanya.

Klik [di sini](#) untuk melihat contoh Private API.

Kalau kalian membuka URL tersebut, maka server akan memberikan respon error dan ngasih tahu kalau kamu harus punya otorisasi supaya bisa mengakses data-data pada API tersebut. Tampilannya kayak gambar di samping.



```
"code":401,"message":"Please authenticate"}
```



Jangan sampai nggak paham tentang autentikasi dan otorisasi, gengs. Soalnya kalau sudah paham, kita mau lanjut nih ke materi **JWT (JSON WEB TOKEN)**.





### Kenalin nih, JWT..

JWT alias JSON Web Token ini ibarat kartu akses kalau dalam kehidupan nyata, guys.

Iho, bagaimana bisa kayak kartu akses ya? Simak slide selanjutnya yaa





## Begini analoginya..

Misalnya kamu kunjungan ke pabrik Pertamina. Sebelum masuk ke dalam pabrik, kamu harus verifikasi identitas dulu, caranya dengan menyerahkan data diri seperti KTP kepada satpam.





Setelah satpam berhasil mengidentifikasi identitasmu, satpam akan ngasih sebuah ID Card Visitor. Dengan kartu inilah kamu bisa berkeliling di dalam pabrik.

Nah, ID Card Visitor inilah yang disebut JWT atau alat untuk mengakses sesuatu sistem.





# Terus bagaimana bentuk JWT? Apakah bentuknya kartu juga?

Ya nggak dongg.

JWT berbentuk sebuah strings random yang nggak bisa dibaca kasat mata oleh manusia.

Strings ini berisi data-data yang telah di-encrypt dengan algoritma encrypt HMAC, SHA256, dan RSA.



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW  
IiOiI2MjI4MGUwNDE1YzZlZTNlMjU0ZTg4MWQiLCJpY  
XQiOjE2NDY3OTIyNTksImV4cCI6MTY0Njc5NDA1OSwi  
dHlwZSI6ImFjY2VzcyJ9.V9MH3sJ4gx6xr3JzFH2Q5j  
039mLJDv9VQsBXQs4hFis
```



## Kapan JWT kita bisa gunakan?

Token ini akan kamu pakai **setiap kali kamu membuat request ke Private API**, sebagai kartu akses kamu untuk mengakses data-data pada Private API tersebut.

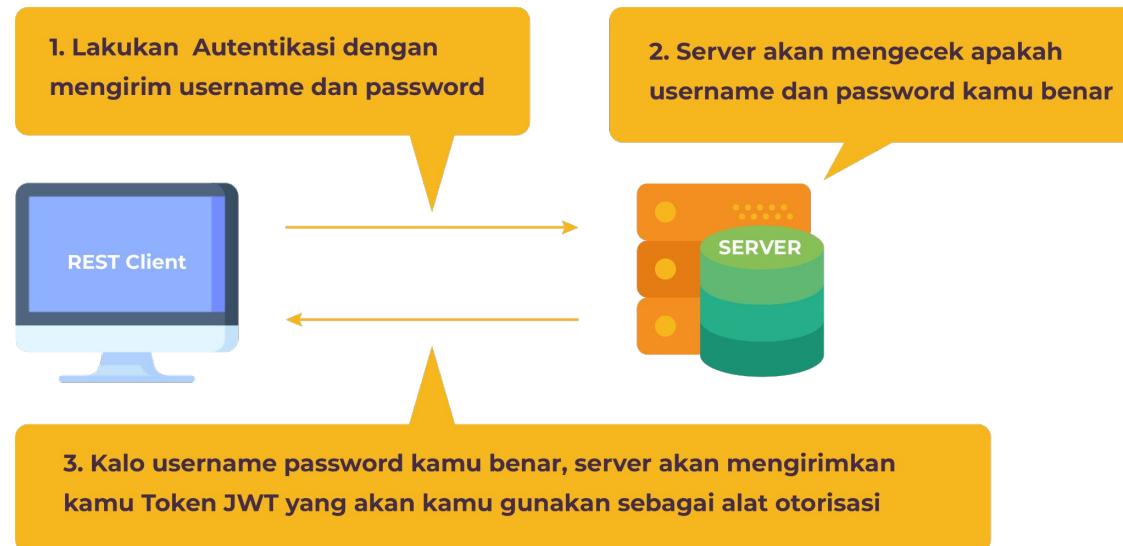
Nah, pertanyaan selanjutnya adalah bagaimana sih biar kamu bisa mendapatkan si JWT?



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdW  
IiOiI2MjI4MGUwNDE1YzZlZTNlMjU0ZTg4MWQiLCJpY  
XQiOjE2NDY3OTIyNTksImV4cCI6MTY0Njc5NDA1OSwi  
dHlwZSI6ImFjY2VzcyJ9.V9MH3sJ4gx6xr3JzFH2Q5j  
039mLJDv9VQsBXQs4hFis
```

## Bagaimana dapetin JWT? Pantengin caranya nih

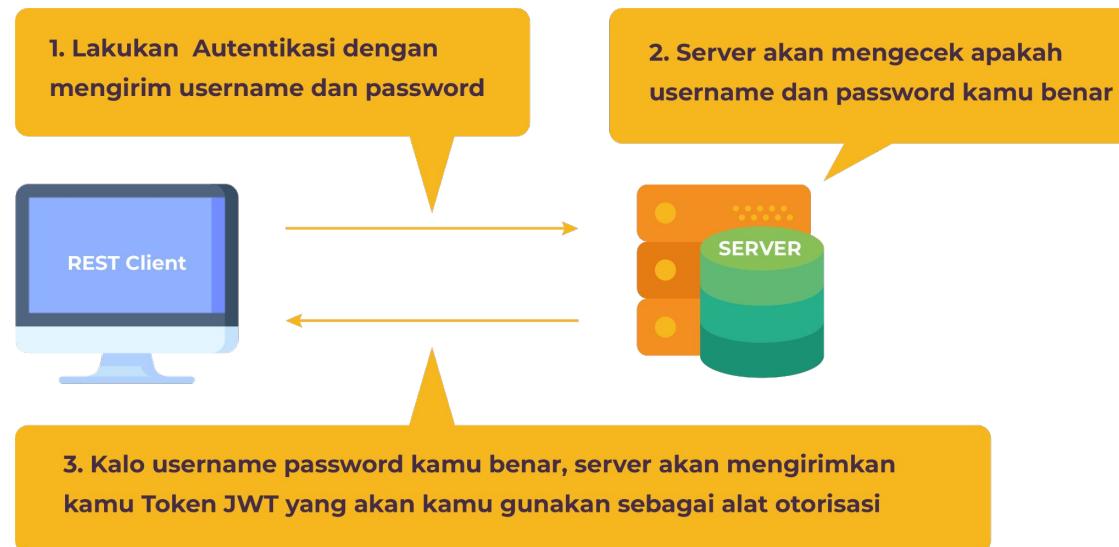
**Tahap pertama.** Seperti yang kita bahas di awal, sebelum mendapatkan authorization (otorisasi), kamu harus **mMelakukan authentication (autentikasi)** terlebih dahulu dengan mengirim informasi username dan password.





**Tahap kedua.** Jadi setelah kamu mengirimkan username dan password ke server, kemudian **server akan melakukan pengecekan** terhadap username dan password kamu.

**Tahap ketiga.** Kalau username dan password benar dan terdaftar, maka **server akan memberikan kamu JWT**. Sebaliknya, kalau username dan password kamu salah, server nggak akan memberikan kamu JWT





### Lalu gimana caranya agar setiap user bisa melakukan autentikasi?

Umumnya server akan menyediakan sebuah RESTFull API khusus seperti login, agar setiap user bisa melakukan autentikasi.

Sebagai contoh, kita bisa menggunakan API Autentikasi pada endpoint <http://code.aldipee.com/api/v1/auth/login>.

Jadi, kamu bisa mengirimkan username dan password, untuk mendapatkan JWT token dari server tersebut.





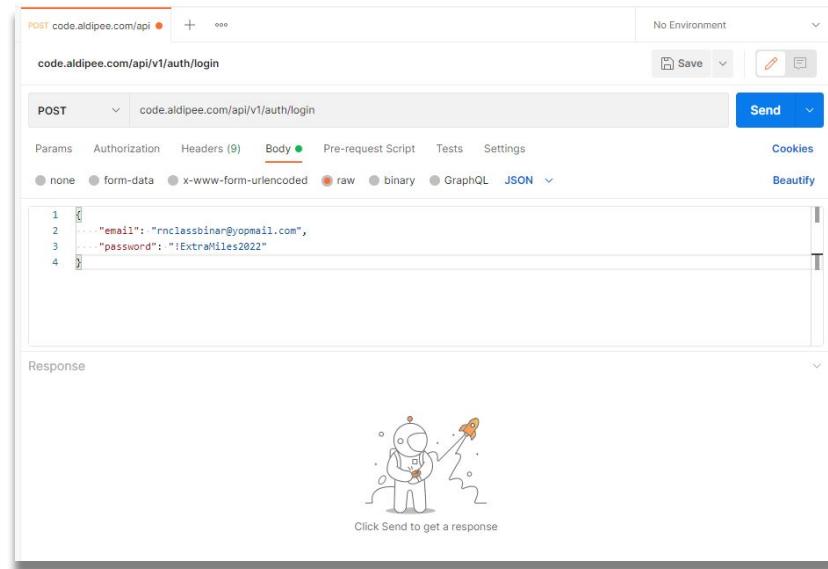
**Yuk, mari kita coba praktekin cara authentication pakai login! Cuss~**





## Begini caranya~

1. Sekarang kita akan coba melakukan autentikasi pada server [code.aldipee.com](http://code.aldipee.com).
2. Untuk melakukan autentikasi kita akan mengirimkan username dan password ke endpoint <http://code.aldipee.com/api/v1/auth/login>  
FYI, kamu bisa pakai username [rnclassbinar@yopmail.com](mailto:rnclassbinar@yopmail.com) dan password !ExtraMiles2022.
3. Setelah itu klik sent.



The screenshot shows the Postman application interface. A POST request is being made to the endpoint `code.aldipee.com/api/v1/auth/login`. The request method is set to POST, and the URL is displayed in the header bar. The 'Body' tab is selected, showing a JSON payload:

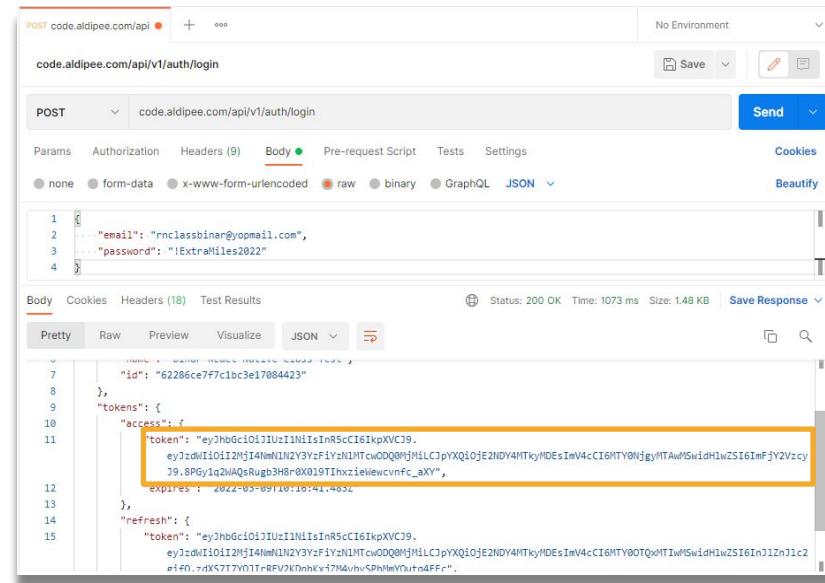
```
1 {  
2   "email": "rnclassbinar@yopmail.com",  
3   "password": "!ExtraMiles2022"  
4 }
```

Below the request, there is a 'Response' section which currently displays a cartoon illustration of an astronaut launching a rocket.



4. Setelah autentikasi berhasil, kamu bisa lihat server ngasih respon yang terdapat data token.

Naaaah, token inilah yang bisa kamu pakai sebagai alat otorisasi saat kamu ingin mengakses Private API.



The screenshot shows a Postman interface with a POST request to `code.aldipee.com/api/v1/auth/login`. The request body contains:

```
1 {
2   "email": "rnaclassbinar@yopmail.com",
3   "password": "!ExtraMiles2022"
4 }
```

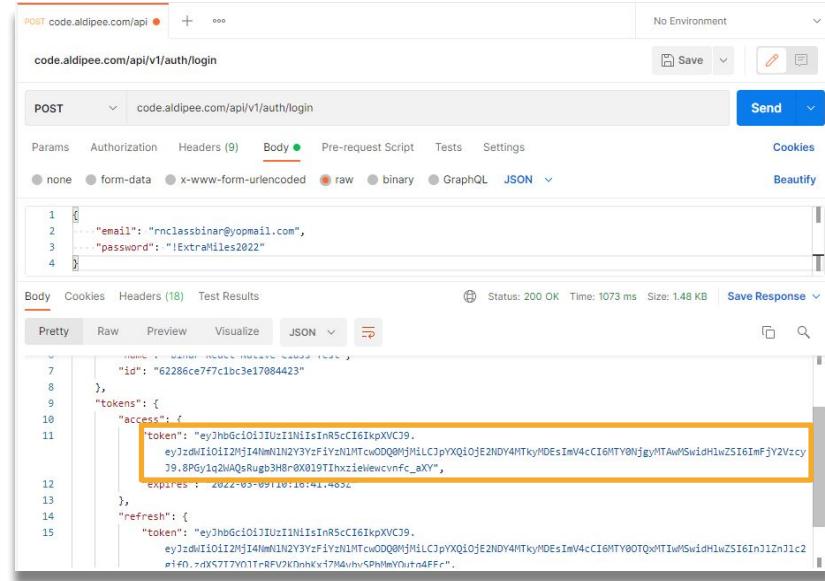
The response status is 200 OK, with a response body containing:

```
7   "id": "62286ce7f7c1bc3e17084423"
8 },
9 "tokens": {
10   "access": {
11     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiI2MjI4ImNlN2YzY2F1czIMTcwODQ0MjI1LCJpYXQiOjE2NDY4MTkyMDEsImV4cC16MTY0NjgyMTAwMSwidHlwZSI6ImFjY2Vzc
19.8PgY1g2mAdQsRugb3H8r0X019TlxzileViewcvnfc_aXY",
12     "expires": "2022-03-09T10:10:41.483Z
13   },
14   "refresh": {
15     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdWIiOiI2MjI4ImNlN2YzY2F1czIMTcwODQ0MjI1LCJpYXQiOjE2NDY4MTkyMDEsImV4cC16MTY0OTQ0MTIwMSwidHlwZSI6InJ12J1c
16     "expires": "2022-03-09T10:10:41.483Z
17   }
18 }
```



5. Selanjutnya, kamu **copy tokennya**, lalu gunakan token tersebut sebagai alat otorisasi untuk mengakses Private API pada <http://code.aldipee.com/api/v1/books>

Cuss langsung coba..



The screenshot shows a Postman interface with a POST request to `code.aldipee.com/api/v1/auth/login`. The request body is:

```
1 {
2   "email": "rnaclassbinar@yopmail.com",
3   "password": "!ExtraMiles2022"
4 }
```

The response status is 200 OK, with a response body containing two tokens:

```
7   "id": "62286ce7f7c1bc3e17084423"
8 },
9 "tokens": {
10   "access": {
11     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdW1Ioi123jI4ImNlNEYzYzFiyzhIMTcwODQ0MjIiLCJpYXQiOjE2NDY4MTkyMDEsImV4cC16MTY0NjgyMTAwMSwidHlwZSI6ImFjY2Vzc
19.8PgY1q2mAdQsRugb3Hr0X019TlxzileViewcvnfc_aXY",
12     "expires": "2022-03-09T10:10:41.482Z
13   },
14   "refresh": {
15     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJzdW1Ioi123jI4ImNlNEYzYzFiyzhIMTcwODQ0MjIiLCJpYXQiOjE2NDY4MTkyMDEsImV4cC16MTY0OTQ0MTIwMSwidHlwZSI6InJ12J1c2
16     "expires": "2022-03-09T10:10:41.482Z
17   }
18 }
```



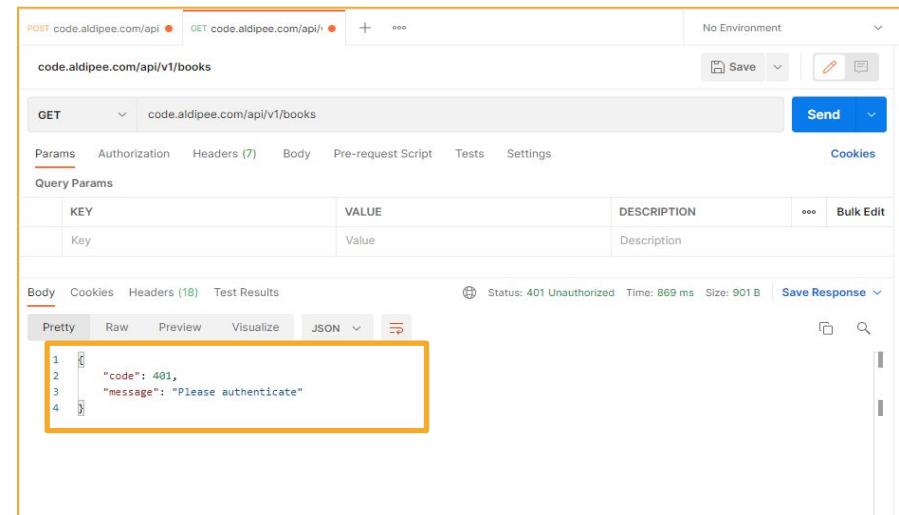
Belajar cara autentikasi dengan login, checked. Sekarang bagaimana caranya memakai **JWT** untuk proses authorization alias otorisasi?





Oke, jadi kalau kamu bikin request ke GET <http://code.aldipee.com/api/v1/books>, maka **server akan ngasih respon error**. Itu karena kamu **belum memasukan JWT-nya saat bikin request**.

Hmm.. terus bagaimana cara menggunakan JWT sebagai authorization saat membuat request?



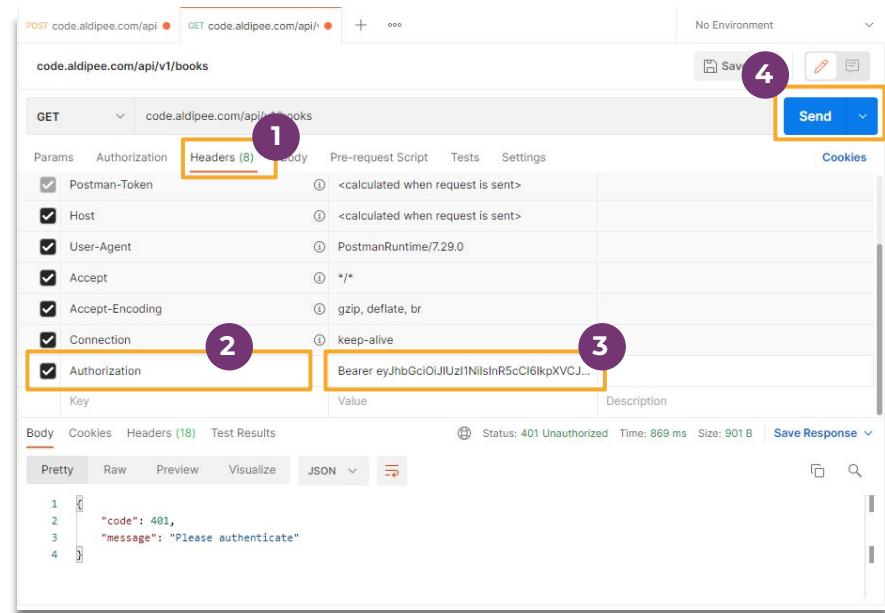
The screenshot shows a Postman collection with two requests. The first request is a POST to `code.aldipee.com/api/v1/books` with a status of 201 Created. The second request is a GET to `code.aldipee.com/api/v1/books`, which has failed with a status of 401 Unauthorized. The response body for the failed request is:

```
1 {  
2   "code": 401,  
3   "message": "Please authenticate"  
4 }
```



## Begini nih langkah menggunakan JWT untuk authorization:

1. Klik 'Headers'
2. Tambahkan 'Authorization' di bagian paling bawah
3. Masukkan token yang kamu dapatkan dari hasil proses login sebelumnya di sini. Lalu tambahkan kata "Bearer" dan spasi di depan token tersebut
4. Terakhir, klik tombol 'Send'



The screenshot shows the Postman application interface with the following steps highlighted:

1. A purple circle highlights the 'Headers (8)' tab in the Headers section of the request configuration.
2. A purple circle highlights the 'Authorization' field under the Headers section.
3. A purple circle highlights the value 'Bearer eyJhbGciOiJIUzI1Ni...'. This value is a JWT token consisting of three parts separated by dots: 'Bearer', the base64 encoded header, and the base64 encoded payload.
4. A purple circle highlights the blue 'Send' button in the top right corner of the interface.

The request details show a GET request to 'code.aldipee.com/api/v1/books'. The Headers section contains several fields: Postman-Token, Host, User-Agent, Accept, Accept-Encoding, Connection, and Authorization. The Authorization field has the value 'Bearer eyJhbGciOiJIUzI1Ni...'. The Body section is empty. The status bar at the bottom indicates a 401 Unauthorized response with a status of 869 ms and a size of 901 B.



Nah, setelah menggunakan token yang didapat saat login sebagai authorization, Private API yang sebelumnya error saat di akses, sekarang sudah bisa di akses dan menampilkan data yang dibutuhkan deh.

The screenshot shows the Postman application interface. A GET request is made to `code.aldipee.com/api/v1/books`. The Headers tab is selected, showing the following configuration:

| Key             | Value  | Description |
|-----------------|--|-------------|
| User-Agent      | PostmanRuntime/7.29.0                        |             |
| Accept          | /*   |             |
| Accept-Encoding | gzip, deflate, br                            |             |
| Connection      | keep-alive                                   |             |
| Authorization   | Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpVJC... |             |

The Body tab shows the response in Pretty format:

```
1 {
2   "results": [],
3   "page": 1,
4   "limit": 10,
5   "totalPages": 0,
6   "totalResults": 0
7 }
```



Itulah tadi contoh authorization yang digunakan oleh POSTMAN.

Nah selanjutnya, gimana caranya kalau kita mau pakai authorization pada Axios?





## Authorization pada Axios

Sama kayak POSTMAN, pada Axios kamu juga harus menambahkan keyword 'Authorization' pada bagian header.

Untuk menambahkan header pada Axios, kamu bisa memasukan config object pada parameter Axios.

Untuk method GET dan DELETE, config object dapat dimasukan sebagai parameter kedua.

Dan untuk method POST, PUT, PATCH, object config dapat dimasukan sebagai parameter ketiga.

```
const objectConfig = {
  headers : {
    Authorization : 'Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJdWIoiI2MjI4NmNlN2Y3YzFiYzNlMTcwODQ0Mj
MiLCJpYXQiOjE2NDY4MTkyMDEsImV4cC16MTY0NjgyMTAwMSwidHlwZSI6ImFjY2VzcycJ9.8PGy1q2W
AQsRugb3H8r0X0l9TIhxzieWewcvnfc_aXY '
  }
}

// Method GET
axios.get('http://code.aldipee.com/api/v1/books', objectConfig)

// Method DELETE
axios.delete('http://code.aldipee.com/api/v1/books', objectConfig)

// Method POST
axios.post('http://code.aldipee.com/api/v1/books', objectData, objectConfig)

// Method PUT
axios.put('http://code.aldipee.com/api/v1/books', objectData, objectConfig)

// Method PATCH
axios.patch('http://code.aldipee.com/api/v1/books', objectData, objectConfig)
```



Selanjutnya kita harus tahu juga dong cara **register** data untuk bisa melakukan proses authentication.

Simak baik-baik ya!





Pada slide sebelumnya kita menggunakan username rnaclassbinar@yopmail.com dan password !ExtraMiles2022 untuk melakukan proses autentikasi/login.

Proses autentikasi dengan data tersebut berhasil **karena data username dan password tersebut sebelumnya telah terdaftar di server.**

The screenshot shows a Postman interface for a POST request to `code.aldipee.com/api/v1/auth/login`. The request body is set to JSON and contains the following data:

```
1 {  
2   ... "email": "rnaclassbinar@yopmail.com",  
3   ... "password": "!ExtraMiles2022"  
4 }
```

The response area is currently empty, with the placeholder text "Click Send to get a response". Below the response area is a cartoon illustration of an astronaut holding a small Christmas tree.



## Tapi kamu perlu tahu ini nih..

Apa yang terjadi kalau data username belum atau nggak terdaftar pada server?

Yang terjadi adalah server nggak bakal ngasih JWT Token dan akan **memberikan response error** yang menginformasikan kalau username/password tersebut gagal melakukan proses autentikasi.

The screenshot shows a POST request to `code.aldipee.com/api/v1/auth/login`. The Body tab contains the following JSON payload:

```
1 ... "email": "meongg@yopmail.com",
2 ...
3 ...
4 ... "password": "gogogo"
```

The response body shows an error message:

```
1 ...
2 ...
3 ...
4 ...
5 {
6   "code": 401,
7   "message": "Incorrect email or password"
8 }
```

The status bar at the bottom indicates `Status: 401 Unauthorized`.



## Kalau kamu nggak register ke server, analoginya kayak gini~

Misalnya kamu mau cetak KTM (Kartu Tanda Mahasiswa) di kampus X, tapi kamu belum terdaftar di kampus tersebut. Otomatis pihak kampus akan menolak memberikan KTM ke kamu dong, karena kamu dianggap outsider alias bukan mahasiswa kampus yang berhak dapat KTM.

Sebaliknya, kalau kamu sudah terdaftar, pihak kampus dengan senang hati memberikan kamu KTM, sebagai kartu akses yang dapat kamu gunakan di lingkungan kampus.



## Server juga begitu..

Sama kayak sebuah aplikasi, server nggak akan ngasih kamu JWT Token kalau kamu nggak terdaftar pada server tersebut.

Umumnya, sebelum seorang user melakukan proses autentikasi/login, **user akan mendaftarkan diri lebih dulu** ke dalam server. Kalau pendaftaran berhasil, user tersebut dapat melakukan proses autentikasi/login.

Dan, ya, sama kayak di kampus, pendaftaran hanya perlu dilakukan sekali saja.





## Begini proses registrasinya, guys!

**Pertama**, untuk melakukan registrasi pada server, kamu bisa menggunakan API POST:

<http://code.aldipee.com/api/v1/auth/register>

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** code.aldipee.com/api/v1/auth/register
- Body (JSON):**

```
1 {
2   ... "email": "sabrinabinar@yopmail.com",
3   ... "password": "Binar2022",
4   ... "name": "sabrinabinar@yopmail.com"
5 }
```
- Body (Pretty):**

```
1 {
2   "success": true,
3   "message": "Please verified your email. We sent email verification to sabrinabinar@yopmail.com",
4   "data": [
5     {
6       "role": "user",
7       "isEmailVerified": false,
8       "email": "sabrinabinar@yopmail.com",
9       "name": "sabrinabinar@yopmail.com",
10      "id": "6228e1e0f7c1bc3e1708442b"
11    }
12  ]
```
- Status:** 201 Created



**Kedua**, agar register berhasil, kamu harus mengirimkan informasi, username, password, dan name yang akan kamu daftarkan.

Misal kamu akan mendaftarkan user dengan nama Sabrina, email sabrinabinar@yopmail.com dan password Binar2022.

Kemudian tekan tombol “Sent”.

POST `code.aldipee.com/api/v1/auth/register`

Params Authorization Headers (9) Body  Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   ...
3   "email": "sabrinabinar@yopmail.com",
4   "password": "Binar2022",
5   "name": "sabrinabinar@yopmail.com"
6 }
```

Body Cookies Headers (18) Test Results Status: 201 Created

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "message": "Please verified your email. We sent email verification to sabrinabinar@yopmail.com",
4   "data": {
5     "role": "user",
6     "isEmailVerified": false,
7     "email": "sabrinabinar@yopmail.com",
8     "name": "sabrinabinar@yopmail.com",
9     "id": "6228e1e0f7c1bc3e1708442b"
10   }
11 }
```



**Ketiga,** jika request berhasil, maka sekarang user `sabrinabinar@yopmail.com` telah terdaftar pada serve dan bisa melakukan autentikasi/login.

The screenshot shows a Postman interface with the following details:

- Method:** POST
- URL:** code.aldipee.com/api/v1/auth/register
- Body:** JSON (selected)
- Request Body Content:**

```
1 {
2     "email": "sabrinabinar@yopmail.com",
3     "password": "Binar2022",
4     "name": "sabrinabinar@yopmail.com"
5 }
```
- Response Status:** 201 Created
- Response Body (Pretty JSON):**

```
1 {
2     "success": true,
3     "message": "Please verified your email. We sent email verification to sabrinabinar@yopmail.com",
4     "data": [
5         {
6             "role": "user",
7             "isEmailVerified": false,
8             "email": "sabrinabinar@yopmail.com",
9             "name": "sabrinabinar@yopmail.com",
10            "id": "6228e1e0f7c1bc3e1708442b"
11        }
12 ]
```



## Sekarang kita coba buktikan ya~

Coba lakukan login dengan username `sabrinabinar@yopmail` dan password `Binar2020`.

Yang terjadi adalah server berhasil memproses autentikasi tersebut dan server memberikan JWT Token atas username `sabrinabinar@yopmail.com`.

Jadi intinya, **sebelum proses autentikasi/login, user harus terlebih dahulu terdaftar pada suatu system/aplikasi tersebut.**

The screenshot shows a Postman request to `code.aldipee.com/api/v1/auth/login`. The request method is POST, and the body is set to raw JSON:

```
1 {
2   "email": "sabrinabinar@yopmail.com",
3   "password": "Binar2020"
4 }
```

The response status is 200 OK, and the response body is:

```
1 {
2   "user": {
3     "role": "user",
4     "isEmailVerified": true,
5     "email": "sabrinabinar@yopmail.com",
6     "name": "sabrinabinar@yopmail.com",
7     "id": "6228e1e0f7c1bc3e1708442b"
8   },
9   "tokens": [
10     "access": {
11       "token": "eyJhbGciOiJIUzI1NiIsInR5cCIkIkpXVCJ9.eyJzdWIiOiI2MjAzTF1MGY3YzF1YzNlMTcwODQ0MiiLCJpYXQiOjE2NDY4NDY3MTgsImV4cCI6MTY0Njg0ODUxOCwidHlwZSI6ImFjY2Vzcj9.eyJyZGMxMxFvdvEVy0fk3Q08KEXXXrgPbw11leR3E54",
12       "expires": "2022-03-09T17:55:18.378Z"
13     },
14   ],
15 }
```



**Yeaayy!! Selamat yaa, guys!**

**Akhirnya kamu berhasil menuntaskan  
Semua Topik di Chapter 3 ini. Biar makin  
jago, yuk kerjain quiz berikut ini!**





# Saatnya kita Quiz!





## 1. Untuk mengakses Private API, yang perlu dilakukan adalah..

- A. Authentication
- B. Authorization
- C. Register



## 2. Pernyataan sebab akibat yang benar adalah..

- A. Apabila Autentikasi gagal maka Authorization dapat dilakukan dengan cara lain yaitu register
- B. Apabila Autentikasi gagal maka Tidak bisa melakuakn Authorization
- C. Apabila Autentikasi gagal maka Authorization tetap dapat dilakukan dengan cara login



### 3. Manakah pernyataan-pernyataan yang benar di bawah ini?

- A. Authorization dapat dilakukan setelah user melakukan Autentication
- B. Authentication dapat dilakukan setelah user melakukan Authorization
- C. Authentication dapat dilakukan setelah user melakukan login



## 4. Authorization dapat digunakan pada method?

- A. POST dan PUT
- B. GET dan DELETE
- C. Semua HTTP Method

# Terima Kasih!



Chapter ✓

completed