

DOCUMENTATION BLOGAPI TEST ON LOCAL

Langkah 1: Lakukan Install tools

Go Programming Language: Pastikan Anda telah menginstal Go di komputer Anda. Jika belum, unduh dan instal dari situs web resminya.

PostgreSQL Database: Pastikan Anda telah menginstal PostgreSQL di komputer Anda. Jika belum, unduh dan instal dari situs web resmi.

Kemudian, buka akun github. Dan lakukan git clone terhadap repository project saya. Lakukan git clone di terminal ubuntu ataupun terminal lainnya.

lalu, open project saya dan ikuti Langkah selanjutnya

Langkah 2: Menyiapkan Database dan konfigurasi db di Golang

Sebelum Anda menjalankan aplikasi, pastikan Anda telah membuat database di PostgreSQL. Gunakan nama yang sesuai, misalnya "apiblog", user dan password postgres. Lalu set di kode program golang agar terintegrasi dengan database yg telah kita buat baru.

Setelah itu, lakukan query di database baru, Create Table dan Column.

```
CREATE TABLE public.posts (  
    id serial4 NOT NULL,  
    title varchar(100) NOT NULL,  
    "content" text NOT NULL,  
    status varchar(10) NULL,  
    publish_date timestamp NULL,  
    CONSTRAINT posts_pkey PRIMARY KEY (id)  
);  
CREATE TABLE public.tags (  
    id serial4 NOT NULL,  
    "label" varchar(50) NOT NULL,  
    posts_id int4 NOT NULL,  
    CONSTRAINT tags_label_key UNIQUE (label),  
    CONSTRAINT tags_pkey PRIMARY KEY (id),  
    CONSTRAINT fk_posts FOREIGN KEY (posts_id) REFERENCES public.posts(id)  
);  
CREATE TABLE public.users (  
    id serial4 NOT NULL,  
    username varchar NOT NULL,  
    "password" varchar NOT NULL,  
    "role" varchar(10) NOT NULL,  
    CONSTRAINT users_pkey PRIMARY KEY (id),  
    CONSTRAINT users_username_key UNIQUE (username)  
);
```

maka table dan column akan tersedia didalam database baru.

Langkah 3: Instal Dependensi

Buka terminal atau command prompt.

Navigasikan ke direktori proyek Anda menggunakan perintah `cd`.

Jalankan perintah `go mod tidy` untuk menginstal semua dependensi yang diperlukan.

Langkah 4: Menjalankan Aplikasi

Buka terminal atau command prompt.

Navigasikan ke direktori proyek Anda menggunakan perintah `cd`.

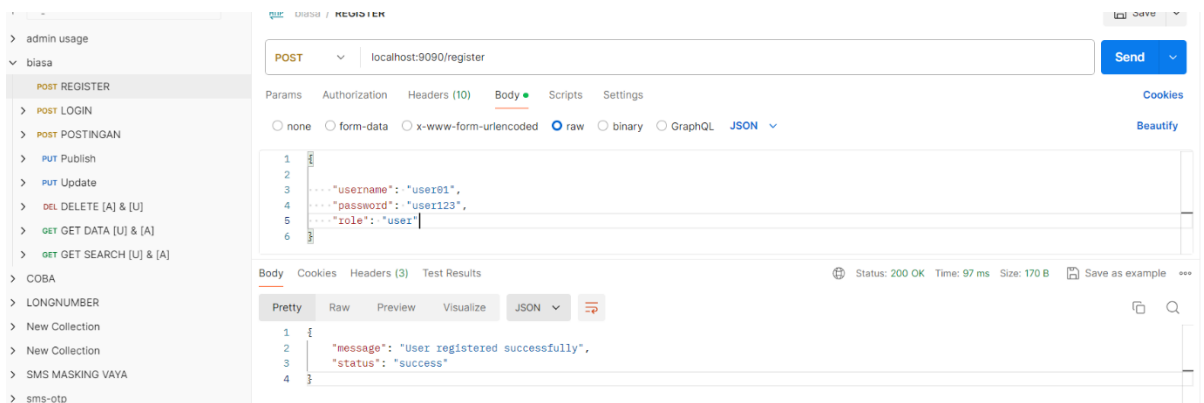
Jalankan perintah `go run main.go`. Ini akan memulai server HTTP lokal Anda.

Langkah 5: Pengujian API di Postman

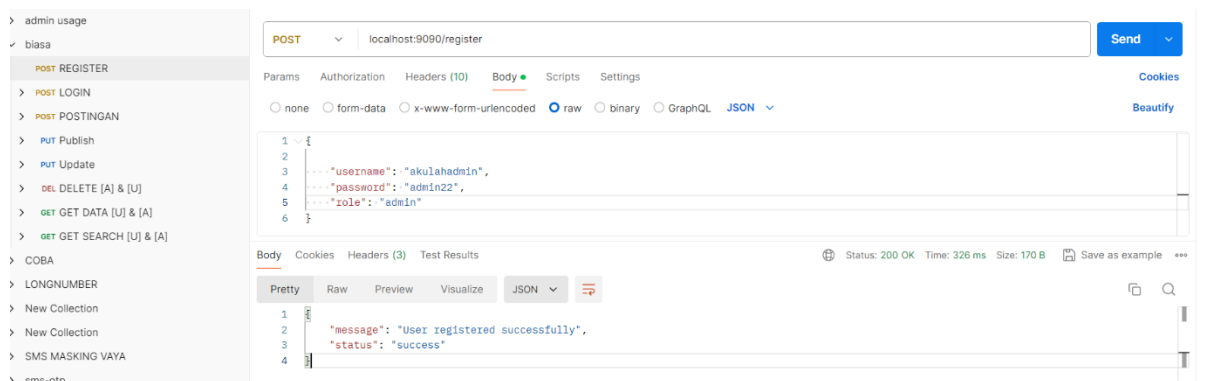
Setelah server berjalan, Anda dapat menggunakan alat seperti Postman atau lainnya untuk menguji endpoint-endpoint yang tersedia. Kali ini saya menggunakan Postman,

- Menguji `localhost:9090/register`

role user :



role admin :



- Menguji `localhost:9090/login`

admin usage

admin usage

POST

localhost:9090/login

Send

POST REGISTER

POST LOGIN

POST POSTINGAN

PUT Publish

PUT Update

DEL DELETE [A] & [U]

GET GET DATA [U] & [A]

GET GET SEARCH [U] & [A]

COBA

LONGNUMBER

New Collection

New Collection

SMS MASKING VAYA

sms-otp

sms-otp

POST

localhost:9090/login

Send

Params

Authorization

Headers (10)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```

1 {
2   "username": "skulahadmin",
3   "password": "admin22"
4 }

```

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Time: 245 ms

Size: 329 B

Save as exam

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "message": "Login successful",
3   "status": "success",
4   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFmdXNhaG9kbWluIiwicm9zeSI6ImkbWluIiwiaWF0IjOyMzE3MTkzOTAwZDQ7Ni1kPia7_95Fp_l2dkQgigR2a0CtMDRhiGibz7k"
5 }

```

admin usage

admin usage

POST

localhost:9090/login

Send

POST REGISTER

POST LOGIN

POST POSTINGAN

PUT Publish

PUT Update

DEL DELETE [A] & [U]

GET GET DATA [U] & [A]

GET GET SEARCH [U] & [A]

COBA

LONGNUMBER

New Collection

New Collection

SMS MASKING VAYA

sms-otp

sms-otp

POST

localhost:9090/login

Send

Params

Authorization

Headers (10)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```

1 {
2   "username": "user01",
3   "password": "user123"
4 }

```

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Time: 90 ms

Size: 321 B

Save as exam

Pretty

Raw

Preview

Visualize

JSON

```

1 {
2   "message": "Login successful",
3   "status": "success",
4   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InVZXiwiMSI6InR5bG91IiwiaWF0IjOyMzE3MTkzOTAwZDQ7Ni1kPia7_95Fp_l2dkQgigR2a0CtMDRhiGibz7k"
5 }

```

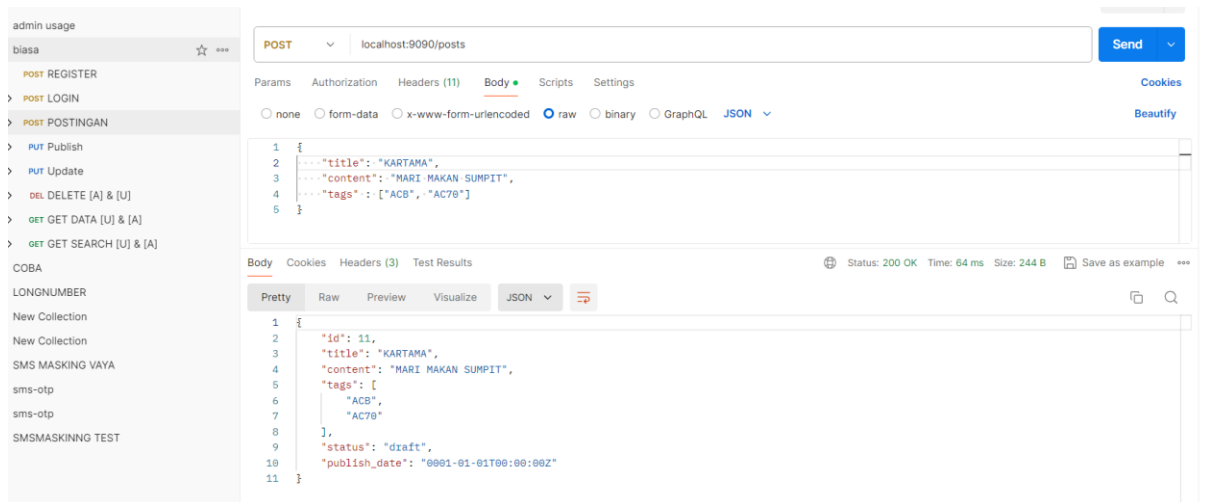
- > admin usage
 > biasa
 POST REGISTER
 > POST LOGIN
 > POST POSTINGAN
 PUT Publish
 PUT Update
 DEL DELETE [A] & [U]
 GET GET DATA [U] & [A]
 GET GET SEARCH [U] & [A]
 > COBA

POST localhost:9090/posts
 Params Authorization Headers (11) Body Scripts Settings
 Headers 9 hidden

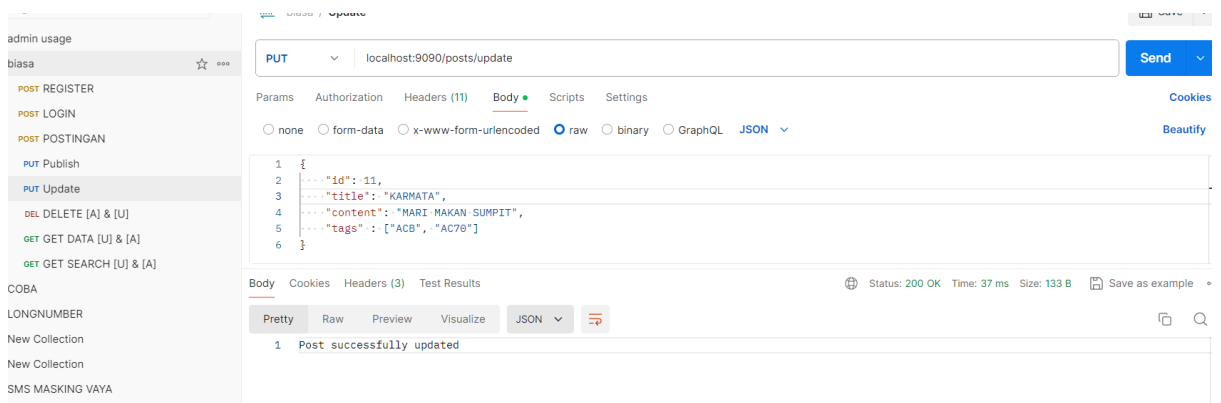
	Key	Value	Description	...	Build
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2...			
<input checked="" type="checkbox"/>	Content-Type	application/json			
	Key	Value	Description		

 Body Cookies Headers (3) Test Results
 Status: 200 OK Time: 28 ms Size: 244 B

Kemudian, lakukan pengisian body json , agar post tersimpan kedalam database. Hanya Akun User yang boleh melakukan posts tetapi status masih dalam draft.



- **Menguji localhost:9090/posts/update**
Posts bisa di update untuk perubahan isi , jikalau statusnya masih draft. Jika status sudah diubah oleh admin menjadi publish. Maka perubahan tidak bisa dilakukan.



- **Menguji localhost:9090/posts/publish?id=11**
Publish hanya bisa dilakukan oleh admin, fungsinya merubah status draft menjadi publish dan tidak bisa diganggu gugat isinya.



- **Menguji localhost:9090/posts/delete?id=11**
Fungsinya untuk menghapus posts yang menurut kita tidak layak lagi atau bosan , dan lainnya. Dimana admin memiliki akses untuk menghapus draft maupun publish posts yang ada. Sementara user memiliki akses hanya menghapus draft.

role user :

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** localhost:9090/posts/delete?id=11
- Query Params:**

Key	Value	Description
id	11	
- Status:** 403 Forbidden
- Time:** 22 ms
- Size:** 206 B
- Response Body:**

```
1 Forbidden: Only admin can delete published posts
2
```

role admin :

The screenshot shows a REST client interface with the following details:

- Method:** DELETE
- URL:** localhost:9090/posts/delete?id=11
- Headers:**

Key	Value	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2...	
Content-Type	application/json	
- Status:** 200 OK
- Time:** 24 ms
- Size:** 167 B
- Response Body:**

```
1 {
2   "message": "Post successfully deleted",
3   "status": "success"
4 }
```

- **Menguji localhost:9090/posts/get**
Fungsi api ini , admin dan user memiliki akses sama untuk mendapatkan keseluruhan posts baik yang publish maupun draft.

The screenshot shows a REST client interface with a sidebar on the left containing a list of endpoints. The main panel displays a GET request to `localhost:9090/posts/get`. The Headers tab is active, showing `Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2...` and `Content-Type: application/json`. The Body tab shows a JSON response with two draft posts.

```
1 [
2   {
3     "id": 2,
4     "title": "JUNAHF",
5     "content": "Seseorang yang berkelana dan mendapat jawaban arti dari SUNGAI ",
6     "tags": [
7       "{GOKIL,APQ}"
8     ],
9     "status": "draft",
10    "publish_date": "0001-01-01T00:00:00Z"
11  },
12  {
13    "id": 3,
14    "title": "JAIDIN12",
15    "content": "Sese yang berkelana dan mendapat jawaban arti dari SUNGAI ",
16    "tags": [
17      "{GOKS}"
18    ]
19  }
20 ]
```

- **Menguji localhost:9090/posts/search?tag=**
Fungsi Api ini , admin dan user memiliki akses yang sama yaitu : untuk melakukan mendapatkan posts dengan pencarian tag:label .

The screenshot shows a REST client interface with a sidebar on the left. The main panel displays a GET request to `localhost:9090/posts/search?tag=ABE`. The Headers tab is active, showing `Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2...` and `Content-Type: application/json`. The Body tab shows a JSON response with one draft post.

```
1 [
2   {
3     "id": 0,
4     "title": "SUNPAH",
5     "content": "MARI PULANG GAIS",
6     "tags": [
7       "{ABE}"
8     ],
9     "status": "draft",
10    "publish_date": "0001-01-01T00:00:00Z"
11  }
12 ]
```

Kemudian, untuk database yang saya miliki :

1. Table posts

Contoh table posts yang saya miliki

The screenshot shows a database management tool interface. On the left, a tree view displays the database structure: 'postgres' -> 'public' -> 'posts' (32K). The main area shows the 'posts' table with columns: id, title, content, status, and publish_date. The table contains four rows of data.

	id	title	content	status	publish_date
1	2	JUNAHF	Seseorang yang berkelana dan mendapat jawaban arti dari SUNGAI	draft	[NULL]
2	3	JAIDIN12	Sese yang berkelana dan mendapat jawaban arti dari SUNGAI	draft	[NULL]
3	5	SATUDUA	MARI PULANG	draft	[NULL]
4	8	SUMPAH	MARI PULANG GAIS	draft	[NULL]

2. Table tags

Contoh table tags yang saya miliki

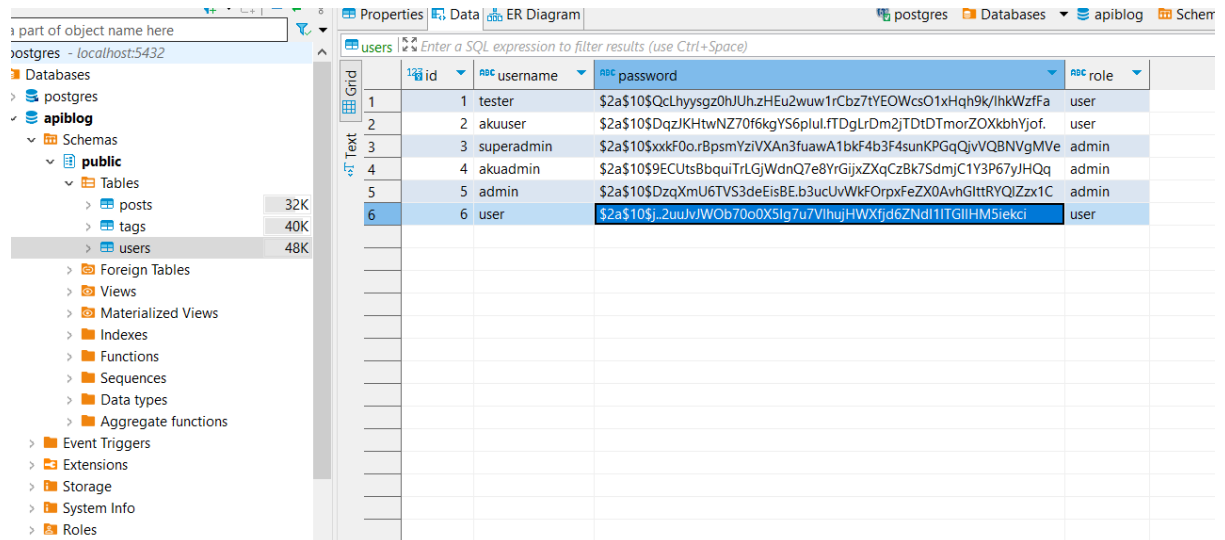
The screenshot shows a database management tool interface. On the left, a tree view displays the database structure: 'postgres' -> 'public' -> 'tags' (40K). The main area shows the 'tags' table with columns: id, label, and posts_id. The table contains seven rows of data.

	id	label	posts_id
1	3	GOKIL	2
2	4	APO	2
3	5	GOKS	3
4	8	AKU	5
5	9	APC	5
6	13	ABC	8
7	14	ABE	8

3. Table users

Sebenarnya ini tidak terlalu dibutuhkan kalau direquirement, tetapi saya membuat ini agar lebih bervariasi saja akun yg ingin digunakan, sehingga saya lakukan register.

Contoh users milik saya, saya gunakan haspassword karena bertujuan untuk praktik keamanan yang umum dilakukan dalam pengembangan perangkat lunak.



The screenshot shows a database management interface with the following structure:

- Database: postgres - localhost:5432
- Schema: public
- Table: users (48K)

	id	username	password	role
1	1	tester	\$2a\$10\$QcLhyysgz0hUhzHEu2wuw1rCbz7tYEOwCsO1xHqh9k/1hkWzfFa	user
2	2	akuuser	\$2a\$10\$DqzJKHtwNZ70f6kgYS6plul.fTDgLRdm2jTDtDTmorZOXkhhYjof.	user
3	3	superadmin	\$2a\$10\$xxkF0o.rBpsmYziVXAn3fuawA1bkF4b3F4sunKPGqQjvVQBnVgMVe	admin
4	4	akuadmin	\$2a\$10\$9ECUtsBbquiTrLGjWdnQ7e8YrGijxZXqCzBk7SdmjC1Y3P67yJHQq	admin
5	5	admin	\$2a\$10\$DzqXmU6TVS3deEisBE.b3ucUvWkFORpxFeZX0AvhGittRYQIZzx1C	admin
6	6	user	\$2a\$10\$j...2uuJvJWOb70c0XS1g7u7VihujHwXfjd6ZNdl1ITGIHM5iekci	user