

Baerikut Dokumentasi pengerjaan aplikasi :

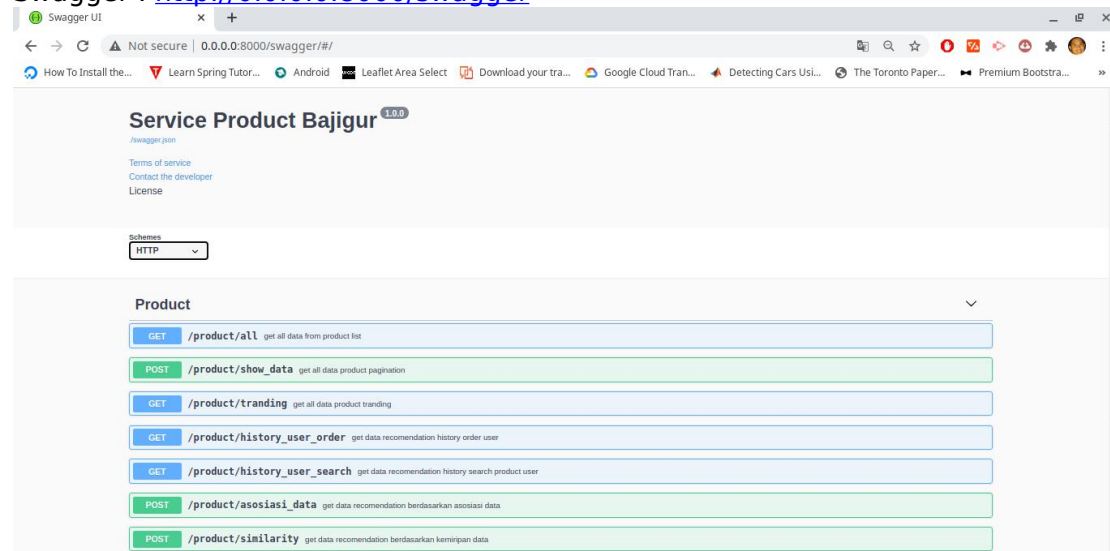
1. Instalasi
  - a) Install anaconda atau python 3.7
  - b) Intall Library pada file requirement.txt
  - c) Intall database MYSQL
2. Cara penggunaan
  - a) Clone project
  - b) Masuk ke dalam folder project
  - c) Jalakan perintah python main.py

```
(base) adil@adil-VivoBook-14-ASUS-Laptop-X407UF ~/Documents/Kerjaan/About_me/JB/python/API $ python main.py
[2020-11-16 00:42:36 +0700] [1340] [INFO] Goin' Fast @ http://0.0.0.0:8000
[2020-11-16 00:42:36 +0700] [1340] [INFO] Starting worker [1340]
```

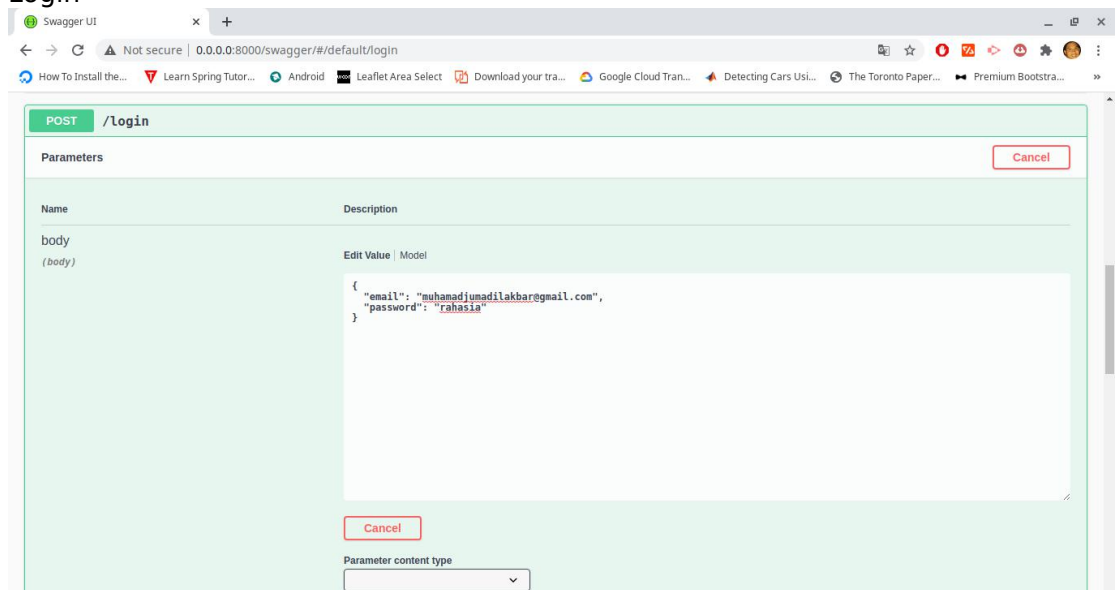
### 3. Hasil Run

Beberapa perintah harus menggunakan authorization dengan Baerer Token dan ada jg tidak perlu menggunakan token untuk yg bersifat umum  
Aplikasi ini sudah di lengkapi dengan swagger jadi bisa langsung di cba di swagger dan untuk yg memerlukan token maka gunakan Postman

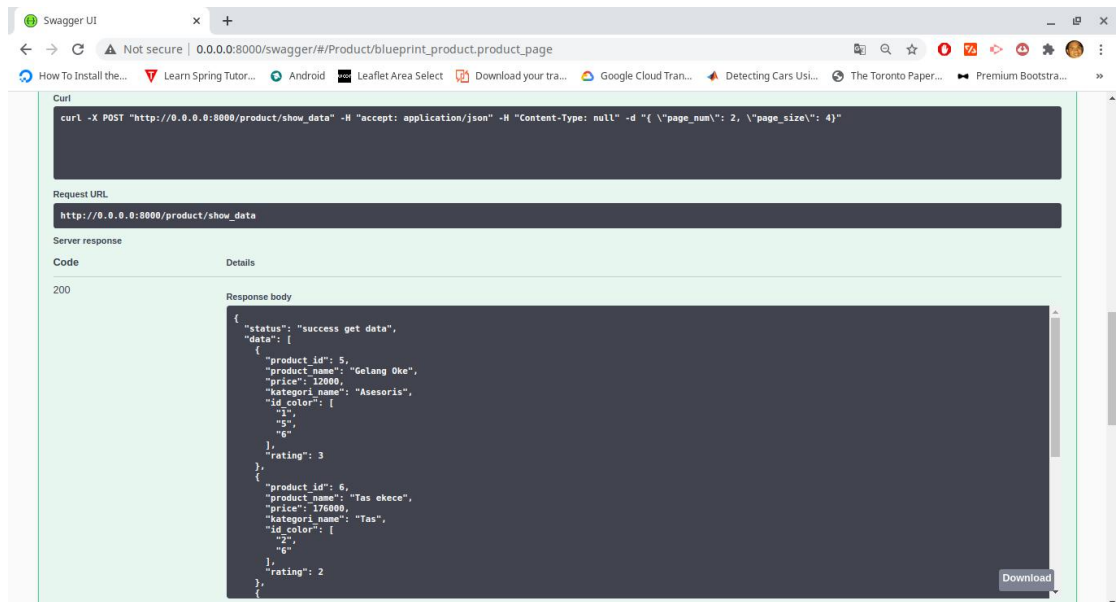
Swagger : <http://0.0.0.0:8000/swagger>



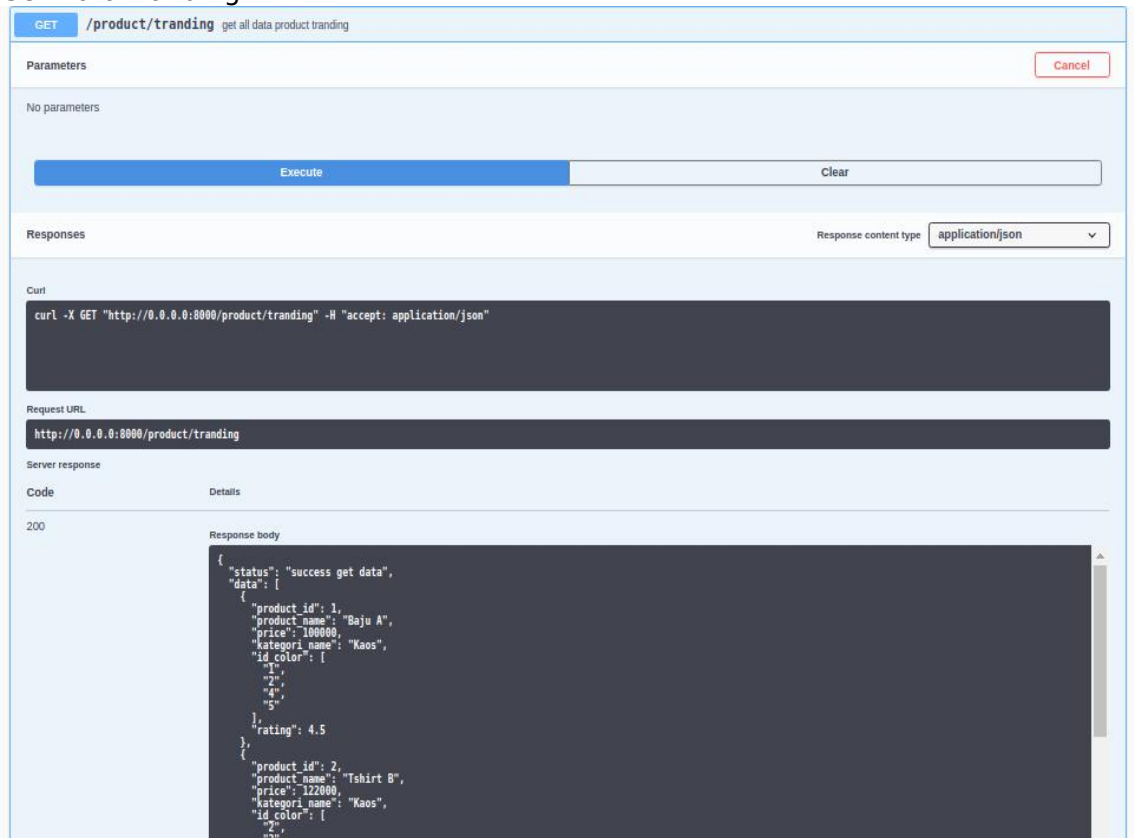
#### a) Login







### c) Get Data trading



### d) Get Data Recommendation by History user

[illegible]

## e) Get Data Asosiasi Data

Swagger UI

Not secure | 0.0.0.0:8000/swagger/#/Product/blueprint\_product.asosiasi\_data

POST /product/asosiasi\_data get data rekomendasi berdasarkan asosiasi data

Parameters

| Name               | Description  |
|--------------------|--|
| USER object (body) | <div>Edit Value</div> <div><pre>{ "product_id": 1 }</pre></div> <div>Cancel</div> <div>Parameter content type</div> <div></div> <div>Execute</div> |

Swagger UI

Not secure | 0.0.0.0:8000/swagger/#/Product/blueprint\_product.asosiasi\_data

curl -X POST "http://0.0.0.0:8000/product/asosiasi\_data" -H "accept: application/json" -H "Content-Type: null" -d "{ \"product\_id\": 1}"

Request URL

http://0.0.0.0:8000/product/asosiasi\_data

Server response

| Code | Details   |
|------|---|
| 200  | <div>Response body</div> <pre>{   "status": "success get data",   "data": [     {       "product_id": 2,       "product name": "Tshirt B",       "price": 122000,       "kategori name": "Kaos",       "id color": [         "2",         "3",         "5"       ],       "rating": 4     },     {       "product_id": 3,       "product name": "Celana Jeans A",       "price": 200000,       "kategori name": "Celana",       "id color": [         "3",         "6"       ],       "rating": 5     }   ] }</pre> |

## f) Get Data Product Similarity

The image shows two screenshots of the Swagger UI interface. The top screenshot displays the endpoint configuration for a POST request to `/product/similarity`. The description is "get data rekomendasi berdasarkan kemiripan data". The parameter is an object in the body with the value `{ "product_id": 1 }`. The bottom screenshot shows the executed response, which is a 200 status code with a JSON body containing product details and their ratings.

**Endpoint Configuration:**

- Method: POST
- URL: `/product/similarity`
- Description: get data rekomendasi berdasarkan kemiripan data
- Parameters: 

| Name               | Description                                    |
|--------------------|--|
| USER object (body) | Edit Value<br><code>{ "product_id": 1 }</code> |

**Response:**

```
curl -X POST "http://0.0.0.0:8000/product/similarity" -H "accept: application/json" -H "Content-Type: null" -d "{ \"product_id\": 1}"
```

Request URL: `http://0.0.0.0:8000/product/similarity`

Server response: 200

Response body:

```
{
  "status": "success get data",
  "data": [
    {
      "product_id": 1,
      "product_name": "Baju A",
      "price": 100000,
      "kategori_name": "Kaos",
      "id_color": [
        "1",
        "2",
        "4",
        "5"
      ],
      "rating": 4.5
    },
    {
      "product_id": 2,
      "product_name": "Tshirt B",
      "price": 122000,
      "kategori_name": "Kaos",
      "id_color": [
        "2",
        "3",
        "5"
      ],
      "rating": 4
    }
  ]
}
```