

**Name: Muhammad Ryan Marcellino**

## Backend Developer - Take Home Project

### Requirement

Using Go, Build a REST API for Product entity with following feature:

- Create, Read, Update, Delete
- Filtering
- Pagination

You can define columns/properties of Product by yourself.

### How to submit

Please upload the code for this project to GitHub, add rahmat@dagangan.com, hidayat@dagangan.com, and adwan.sopyar@dagangan.com as a member, and post a link to your repository below. Maximum submission is 14 days from today

## 1) Screenshot: Implementation of *Create Product (CRUD)*

The screenshot displays a REST client interface for a 'product-dagangan' project, specifically for the 'Create Product' endpoint. The request is a POST to 'localhost:8080/api/product' with a JSON body. The response is a 200 OK status, indicating a successful creation of the product.

**Request Details:**

- Method: POST
- URL: localhost:8080/api/product
- Body Type: JSON
- Body Content:

```
1 {
2   "name": "sarung tangan motor",
3   "merchant": "toko aksesoris motor",
4   "desc": "sarung tanga",
5   "stock": 14,
6   "price": 20000
7 }
```

**Response Details:**

- Status: 200 OK
- Time: 23 ms
- Size: 255 B
- Body Type: JSON (Pretty)
- Body Content:

```
1 {
2   "product": {
3     "id": 14,
4     "name": "sarung tangan motor",
5     "merchant": "toko aksesoris motor",
6     "desc": "sarung tanga",
7     "stock": 14,
8     "price": 20000
9   }
10 }
```

## 2) Screenshot: Implementation of *Read All Product (CRUD)*

The screenshot displays a REST client interface for a project named "product-dagangan". The active tab is "Read Product (All)". The request is a GET method to the URL "localhost:8080/api/product". The response status is "200 OK" with a time of "4 ms" and a size of "1.06 KB". The response body is shown in "Pretty" JSON format, containing an array of two product objects.

**Query Params**

KEY	VALUE	DESCRIPTION
Key	Value	Description

**Body**

```
1  {
2    "product": [
3      {
4        "id": 1,
5        "name": "sepatu puma",
6        "merchant": "toko sepatu",
7        "desc": "sepatu futsal",
8        "stock": 10,
9        "price": 599000
10     },
11     {
12       "id": 2,
13       "name": "sepatu nike",
14       "merchant": "toko sepatu",
15       "desc": "sepatu sepak bola",
16       "stock": 20,
17       "price": 799000
18     },
19   ]
20 }
```

At the bottom of the interface, there are icons for "Cookies", "Capture requests", "Runner", "Trash", and a window management icon.

### 3) Screenshot: Implementation of *Read Single Product (CRUD)*

The screenshot displays a REST client interface for a project named "product-dagangan". The active tab is "Read Product (Single)". The request is a GET method to the URL "localhost:8080/api/product/8". The response status is 200 OK, with a response time of 5 ms and a size of 242 B.

**Query Params**

KEY	VALUE	DESCRIPTION
Key	Value	Description

**Body**

The response body is displayed in JSON format:

```
1  {
2    "product": {
3      "id": 8,
4      "name": "buku menggambar hewan",
5      "merchant": "toko buku",
6      "desc": "buku anak",
7      "stock": 28,
8      "price": 15000
9    }
10 }
```

The bottom status bar includes icons for Cookies, Capture requests, Runner, Trash, and a full-screen icon.

#### 4) Screenshot: Implementation of *Update Product (CRUD)*

The screenshot displays a REST client interface for a PUT request to `localhost:8080/api/product/8`. The request body is a JSON object with the following fields:

```
{
  "name": "buku menggambar hewan",
  "merchant": "toko buku",
  "desc": "buku anak",
  "stock": 26,
  "price": 15000
}
```

The response is a 200 OK status with a JSON body containing the following fields:

```
{
  "Desc": "buku anak",
  "Merchant": "toko buku",
  "Name": "buku menggambar hewan",
  "Price": 15000,
  "Stock": 26,
  "message": "it's already updated!"
}
```

The interface includes tabs for Params, Authorization, Headers (3), Body, Pre-request Script, Tests, and Settings. The Body tab is selected, and the response is displayed in the bottom section. The status bar at the bottom shows Cookies, Capture requests, Runner, Trash, and other utility icons.

## 5) Screenshot: Implementation of *Delete Product (CRUD)*

The screenshot displays a REST client interface for a project named "product-dagangan". The active tab is "Delete Product". The request method is "DELETE" and the URL is "localhost:8080/api/product/13". The "Send" button is visible.

Below the request details, the "Params" tab is selected, showing "Query Params". A table lists the parameters:

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

The "Body" tab is also visible, showing the response details. The status is "200 OK", the time is "12 ms", and the size is "158 B". The response body is displayed in "Pretty" JSON format:

```
1 {  
2   "message": "it's already deleted!"  
3 }
```

The bottom status bar includes icons for "Cookies", "Capture requests", "Runner", "Trash", and a settings icon.

## 6) Screenshot: Implementation of *Read with Filter Merchant*

The screenshot displays a REST client interface for a GET request to `localhost:8080/api/product/filter?merchant=toko sepatu`. The request is successful, returning a 200 OK status with a response time of 31 ms and a size of 360 B.

**Query Params**

KEY	VALUE	DESCRIPTION
merchant	toko sepatu	

**Body**

Pretty Raw Preview Visualize JSON

```
1  {
2    "data": [
3      {
4        "id": 1,
5        "name": "sepatu puma",
6        "merchant": "toko sepatu",
7        "desc": "sepatu futsal",
8        "stock": 10,
9        "price": 599000
10     },
11     {
12       "id": 2,
13       "name": "sepatu nike",
14       "merchant": "toko sepatu",
15       "desc": "sepatu sepak bola",
16       "stock": 20,
17       "price": 799000
18     }
19   ],
20   "totalData": 2
21 }
```

At the bottom of the interface, there are icons for Cookies, Capture requests, Runner, Trash, and a help icon.

## 7) Screenshot: Implementation of *Read with Pagination*

The screenshot shows a REST client interface for a request to `localhost:8080/api/product/page/2`. The response is a JSON array of three product objects, displayed in the 'Body' tab. The status is 200 OK, with a response time of 10 ms and a size of 426 B.

**Query Params**

KEY	VALUE	DESCRIPTION
-----	-------	-------------

**Body** (Pretty)

```
1 {
2   {
3     "id": 7,
4     "name": "nanas",
5     "merchant": "toko buah",
6     "desc": "buah tropis",
7     "stock": 30,
8     "price": 10000
9   },
10  {
11    "id": 8,
12    "name": "buku menggambar hewan",
13    "merchant": "toko buku",
14    "desc": "buku anak",
15    "stock": 28,
16    "price": 15000
17  },
18  {
19    "id": 9,
20    "name": "bumi manusia",
21    "merchant": "toko buku",
22    "desc": "novel fiksi",
23    "stock": 17.
```