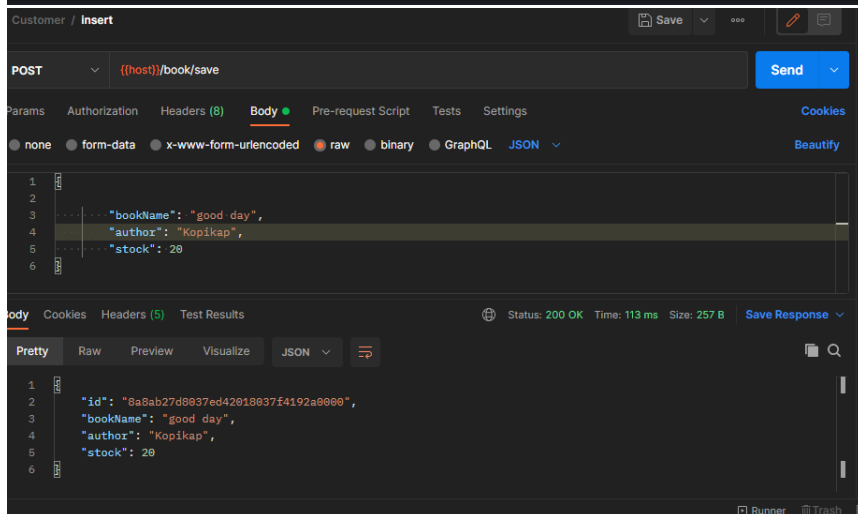
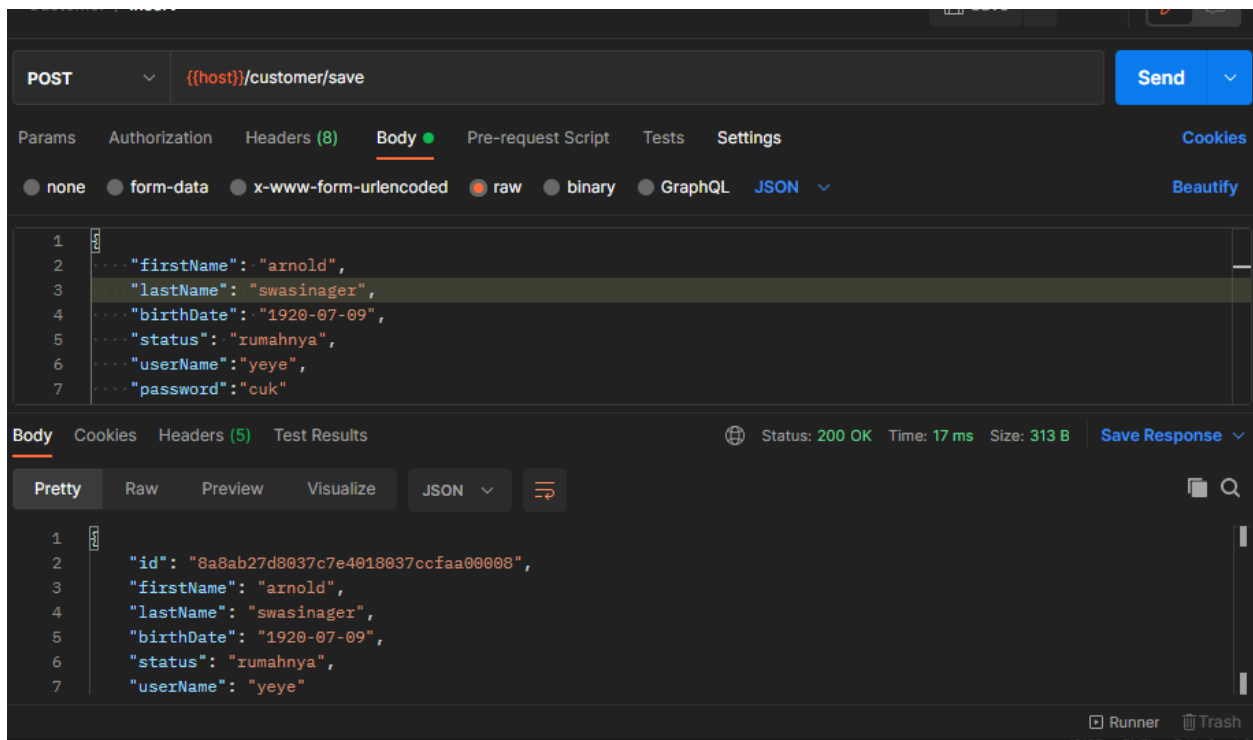


ERD

public
mst_customer
customer_id character varying(255)
birth_date timestamp without time zone
first_name character varying(255)
last_name character varying(255)
password character varying(255)
status character varying(255)
user_name character varying(255)

public
mst_book
book_id character varying(255)
author character varying(255)
book_name character varying(255)
stock integer

Create



Hasil dari insert data pada book dan customer

Read

Customer / **get all** [Save] [Send]

GET `{{host}}/book`

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (5) Test Results [Status: 200 OK Time: 13 ms Size: 1.25 KB Save Response]

Pretty Raw Preview Visualize JSON [Copy]

```
1 [
2   {
3     "id": "8a8ab27d8037e967018037e986cf0000",
4     "bookName": "tes",
5     "author": "tes",
6     "stock": 20
7   },
8   {
9     "id": "8a8ab27d8037e967018037e99ee50001",
10    "bookName": "set",
11    "author": "tes",
12    "stock": 20
13  },
14 ]
```

Runner Trash

Customer / **get all** [Save] [Send]

GET `{{host}}/customer`

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (5) Test Results [Status: 200 OK Time: 39 ms Size: 4.05 KB Save Response]

Pretty Raw Preview Visualize JSON [Copy]

```
1 [
2   {
3     "id": "8a8ab27d8036da61018036db51080000",
4     "firstName": "Papai",
5     "lastName": "Yuli",
6     "birthDate": "1999-07-09",
7     "status": "rumahnya",
8     "userName": "yeye"
9   },
10  {
11    "id": "8a8ab27d8036dc39018036dc81e60000",
12    "firstName": "Papai",
13    "lastName": "Yuli",
14    "birthDate": "1999-07-09",
15    "status": "rumahnya",
16    "userName": "yeye"
17  },
18 ]
```

Read data dengan get all pada book dan customer

Update

The screenshot shows a REST client interface with a PUT request to `{{host}}/book/update`. The request body is a JSON object with the following fields:

```
{  "id": "8a8ab27d8037ed42018037f4192a0000",  "bookName": "good night",  "author": "Kopikap",  "stock": 20}
```

The response status is 200 OK, with a time of 26 ms and a size of 259 B. The response body is a JSON object with the same fields as the request body.

The screenshot shows a REST client interface with a PUT request to `{{host}}/customer/update`. The request body is a JSON object with the following fields:

```
{  "lastName": "Yela",  "birthDate": "1920-07-09",  "status": "rumahnya",  "userName": "yeye",  "password": "cuk"}
```

The response status is 200 OK, with a time of 27 ms and a size of 276 B. The response body is a JSON object with the following fields:

```
{  "id": "1",  "firstName": "gerald",  "lastName": "Yela",  "birthDate": "1920-07-09",  "status": "rumahnya",  "userName": "yeye"}
```

Update pada book dan customer

Delete

Customer / Delete

Save ...

DELETE `{{host}}/book/delete?id=8a8ab27d8037df36018037e1ec100003` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	8a8ab27d8037df36018037e1ec100003			
	Key	Value	Description		

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 37 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

1

Runner Trash

Customer / Delete

Save ...

DELETE `{{host}}/customer/delete?id=8a8ab27d8036da61018036db51080000` Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	8a8ab27d8036da61018036db51080000			
	Key	Value	Description		

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 52 ms Size: 123 B Save Response

Pretty Raw Preview Visualize Text

1

Delete pada book dan customer

Pagination

The image displays two screenshots of a REST client interface, likely Postman, showing the results of a GET request to a pagination endpoint. The request URL is `{{host}}/book/pagination?page=0&size=3&direction=desc`.

Top Screenshot: The response body is shown in JSON format. It contains a single object with the following fields:

```
{  "searchBookName": "t",  "searchAuthor": "t"}
```

The status is 200 OK, Time: 831 ms, Size: 736 B.

Bottom Screenshot: The response body is shown in JSON format. It contains an array of two objects, each with the following fields:

```
{  "content": [    {      "id": "8a8ab27d8837e967018037e9af8e000b",      "bookName": "set",      "author": "tes",      "stock": 20    },    {      "id": "8a8ab27d8837e967018037e9ed1a000c",      "bookName": "hari poe",      "author": "jk jk",      "stock": 20    }  ]}
```

The status is 200 OK, Time: 97 ms, Size: 743 B.

Customer / Delete

Save

Send

DELETE

{{(host)}}/book/delete?id=8a8ab27d8037df36018037e1ec100003

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	id	8a8ab27d8037df36018037e1ec100003			
	Key	Value	Description		

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 37 ms

Size: 123 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

Runner

Trash

Customer / Search wildcard

Save

Send

GET

{{(host)}}/customer/pagination?page=0

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Body

Cookies

Headers (5)

Test Results

Status: 200 OK

Time: 25 ms

Size: 1.87 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1
2  "content": [
3    {
4      "id": "8a8ab27d8036da61018036db51080000",
5      "firstName": "Papai",
6      "lastName": "Yuli",
7      "birthDate": "1999-07-09",
8      "status": "xumahnya",
9      "userName": "yeye"
10   },
11   {
12     "id": "8a8ab27d8036dc39018036dc81e60000",
13     "firstName": "Papai",
14     "lastName": "Yuli",
15     "birthDate": "1999-07-09",
16     "status": "xumahnya",
17     "userName": "yeye"
```

Runner

Trash

Pagination dengan menggunakan dto ,sorting dan paging