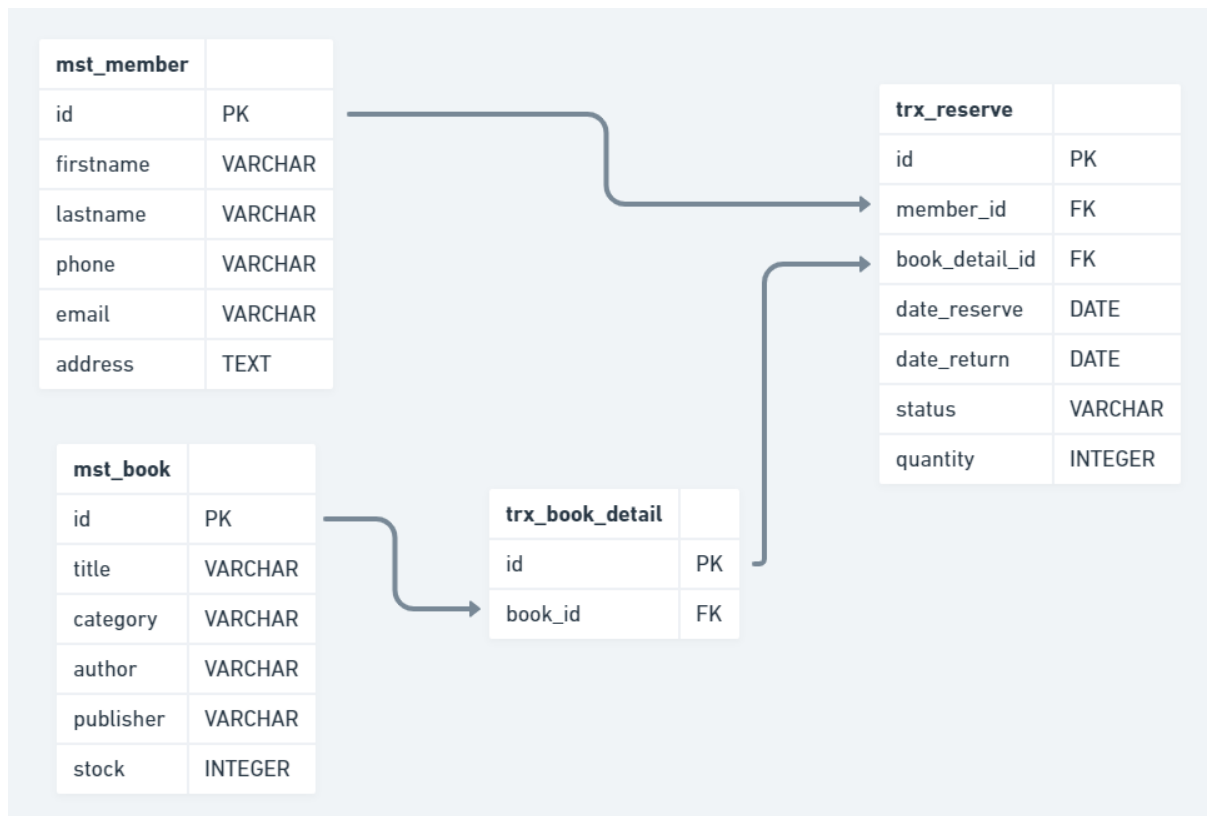


**Ahmad Fansyuri**

**17 April 2022**

**Tugas : Buatlah CRUD Library (Member dan Book). Nilai plus jika tiap pembelian buku dapat berkurang.**



Tabel 1 ERD library

Saya mendesain library ini menggunakan empat entitas. Dimana Entitas member dan book sebagai master, lalu book\_detail dan reserve sebagai tabel transaksi. Bersamaan dengan ERDnya saya sertakan variable tiap entitas.

## 1. Entitas mst\_member

- INSERT

Member / Add Member Save

POST ▼ `{{host}}/members`

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

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   ... "firstName": "Indra",
3   ... "lastName": "Subekti",
4   ... "phone": "082221082107",
5   ... "email": "indrabeckti@rocketmail.com",
6   ... "address": "Jln. Jakarta Raya, Jakarta, Indonesia"
7 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 246 ms Size: 269 B

Pretty Raw Preview Visualize **JSON** ≡

```
1 {
2   "id": "8a8ae46f8035b91d018035b930130000",
3   "firstName": "Indra",
4   "lastName": "Subekti",
5   "phone": "082221082107"
6 }
```

**TABEL 1.1 Penambahan Member**

- UPDATE

Member / Update Member

PUT ▼ `{{host}}/members`

Params Authorization Headers (9) **Body** Pre-request Script

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐

```
1 {
2   ... "id": "8a8ae46f8035b91d018035b930130000",
3   ... "firstName": "Indra",
4   ... "lastName": "Subektina",
5   ... "phone": "082221082107",
6   ... "email": "indrabeckti@yahoo.com",
7   ... "address": "Jln. Jakarta Raya, Jakarta, Indonesia"
8 }
```

	Data Output	Explain	Messages	Notifications					
	member_id [PK] character varying (255)	address character varying (255)	email character varying (255)	first_name character varying (255)	last_name character varying (255)	phone character varying (255)			
1	8a8ae46f803267800180326ad4eb0004	Jln. A. Dahlan RT 08 No. 75, Jakarta Selatan	effanhatta@gmail.com	Ahmad	Fansyuri	088286104160			
2	8a8ae46f803267800180326b6a3f0005	Jln. A. Dahlan DPP Gerindra	fajarpresiden@gmail.com	Fajar	Subianto	082221082107			
3	8a8ae46f8035b91d018035b930130000	Jln. Jakarta Raya, Jakarta, Indonesia	indrabeckti@yahoo.com	Indra	Subektina	082221082107			
4	8a8ae46f8035b91d018035ba45010001	Jln. Jakarta Selatan	Andre@yahoo.com	Andre	Suherman	08827171			
5	8a8ae46f8035b91d018035bcb8290002	Jln. Jakarta Raya, Jakarta, Indonesia	putragundala1999@yahoo.com	Gundala	Syaputra	082221082107			

## TABEL UPDATE

Dapat dilihat pada baris ketiga perubahan email dan last name.

- DELETE

DELETE
{{host}}/members?id=8a8ae46f8035b91d018035b930130000

	Data Output	Explain	Messages	Notifications					
	member_id [PK] character varying (255)	address character varying (255)	email character varying (255)	first_name character varying (255)	last_name character varying (255)	phone character varying (255)			
1	8a8ae46f803267800180326ad4eb0004	Jln. A. Dahlan RT 08 No. 75, Jakarta Selatan	effanhatta@gmail.com	Ahmad	Fansyuri	088286104160			
2	8a8ae46f803267800180326b6a3f0005	Jln. A. Dahlan DPP Gerindra	fajarpresiden@gmail.com	Fajar	Subianto	082221082107			
3	8a8ae46f8035b91d018035ba45010001	Jln. Jakarta Selatan	Andre@yahoo.com	Andre	Suherman	08827171			
4	8a8ae46f8035b91d018035bcb8290002	Jln. Jakarta Raya, Jakarta, Indonesia	putragundala1999@yahoo.com	Gundala	Syaputra	082221082107			

Pada tabel member dapat dilihat baris ketiga atas nama indra subektina sudah dihapus.

- PAGINATION (FILTERING & SEARCHING)

GET

{{host}}/members

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

2

3

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

Secara default saya menetapkan size dan page nya masing-masing di size=3 dan page=0.

## Member / Pagination


GET ▼ `{{host}}/members?size=7`

Params ● Authorization Headers (9) Body ● Pre-request Scri

● none ● form-data ● x-www-form-urlencoded ● raw ● binary

```
1  [
2  ... "searchFirstName" : "F"
3  ]
```


Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1  [
2  "content": [
3    {
4      "id": "8a8ae46f803267800180326b6a3f0005",
5      "firstName": "Fajar",
6      "lastName": "Subianto",
7      "phone": "082221082107"
8    },
9  ],
10 "pageable": {
11   "sort": {
12     "empty": false,
13     " ...
14   }
15 }
```

```
1  [
2  ... "searchFirstName" : "A",
3  ... "searchLastName" : "Su"
4  ]
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON ▼ 

```
1  [
2    "content": [
3      {
4        "id": "8a8ae46f8035b91d018035ba45010001",
5        "firstName": "Andre",
6        "lastName": "Suherman",
7        "phone": "08827171"
8      },
9      {
10       "id": "8a8ae46f803267800180326b6a3f0005",
11       "firstName": "Fajar",
12       "lastName": "Subianto",
13       "phone": "082221082107"
14     }
15   ],
16   "pageable": {
```

## 2. Entitas mst\_book

- **INSERT**

POST

▼

{{host}}/books

Params

Auth

Headers (9)

Body ●

Pre-req.

Tests

Settings

raw ▼

JSON ▼

```
1 {
2   ... "title" : "The Adventures Sherlock Holmes",
3   ... "category" : "Drama-Mystery",
4   ... "author" : "Sir. Conan Doyle",
5   ... "publisher" : "George Newnes",
6   ... "quantity" : 3
7 }
```

Body ▼

🌐

Pretty

Raw

Preview

Visualize

JSON ▼

↺↻

```
1 {
2   "id": "8a8ae46f8035b91d018035ca305a0003",
3   "title": "The Adventures Sherlock Holmes",
4   "category": "Drama-Mystery",
5   "author": "Sir. Conan Doyle",
6   "publisher": "George Newnes"
7 }
```

**TABEL 2.1 SAVE**

- **UPDATE**

```
1 {
2   ... "id" : "8a8ae46f8035b91d018035ca305a0003",
3   ... "title" : "The Adventures Sherlock Holmes",
4   ... "category" : "Drama-Mystery",
5   ... "author" : "Sir. Conan Doyle",
6   ... "publisher" : "George Newnes",
7   ... "quantity" : 5
8 }
9
```

	Data Output	Explain	Messages	Notifications					
	book_id [PK] character varying (255)	author character varying (255)	category character varying (255)	publisher character varying (255)	quantity integer	title character varying (255)			
1	8a8ae46f803267800180326931680000	Jk. Rowling	Fantasy	Bloomsbury		1 Harry Potter and the Deathly Hollows			
2	8a8ae46f8032678001803269b65c0001	Jk. Rowling	Fantasy	Bloomsbury		2 Harry Potter the Order of Phoenix			
3	8a8ae46f8032678001803269e92d0002	Jk. Rowling	Fantasy	Bloomsbury		1 Harry Potter the Chamber of Secrets			
4	8a8ae46f803267800180326a1c770003	Jk. Rowling	Fantasy	Bloomsbury		2 Harry Potter the Prisoner of The Azkaban			
5	8a8ae46f8033249a0180332bae260000	TheAnonymous	Filosofi	Gremlin		2 Dunia Sophie			
6	8a8ae46f8035b91d018035ca305a0003	Sir. Conan Doyle	Drama-Mystery	George Newnes		5 The Adventures Sherlock Holmes			

**TABEL 2.2 UPDATE BOOK**

Dapat dilihat pada baris 6 perubahan Quantity.

- DELETE**

DELETE

{{host}}/books?id=8a8ae46f8035b91d018035ca305a0003

	book_id [PK] character varying (255)	author character varying (255)	category character varying (255)	publisher character varying (255)	quantity integer	title character varying (255)			
1	8a8ae46f803267800180326931680000	Jk. Rowling	Fantasy	Bloomsbury		1 Harry Potter and the Deathly Hollows			
2	8a8ae46f8032678001803269b65c0001	Jk. Rowling	Fantasy	Bloomsbury		2 Harry Potter the Order of Phoenix			
3	8a8ae46f8032678001803269e92d0002	Jk. Rowling	Fantasy	Bloomsbury		1 Harry Potter the Chamber of Secrets			
4	8a8ae46f803267800180326a1c770003	Jk. Rowling	Fantasy	Bloomsbury		2 Harry Potter the Prisoner of The Azkaban			
5	8a8ae46f8033249a0180332bae260000	TheAnonymous	Filosofi	Gremlin		2 Dunia Sophie			

**TABEL 2.3 DELETE BOOK**

Pada tabel book Sherlock Holmes

- PAGINATION (FILTERING & SEARCHING)**

GET ▼ `{{host}}/books`

Params Auth Headers (9) **Body** ● Pre-req. Tests Settings

raw ▼ **JSON** ▼

1 ⌵  
2 ...  
3 ⌵

Body ▼ 🌐 200 OK 22 r

Pretty Raw Preview Visualize **JSON** ▼ ≡

```
1 ⌵  
2 "content": [  
3   {  
4     "id": "8a8ae46f8033249a0180332bae260000",  
5     "title": "Dunia Sophie",  
6     "category": "Filosofi",  
7     "author": "TheAnonymous",  
8     "publisher": "Gremlin"  
9   },  
10  {  
11    "id": "8a8ae46f803267800180326931680000",  
12    "title": "Harry Potter and the Deathly Hollows",  
13    "category": "Fantasy",  
14    "author": "Jk. Rowling",  
15    "publisher": "Bloomsbury"  
16  },  
17  {  
18    "id": "8a8ae46f8032678001803269e92d00002",  
19    "title": "Harry Potter the Chamber of Secrets",  
20    "category": "Fantasy",  
21    "author": "Jk. Rowling",  
22    "publisher": "Bloomsbury"  
23  }  
]
```

Secara default saya menetapkan size dan page nya masing-masing di size=3 dan page=0.



```
1  {
2  | ... "searchBookAuthor" : "anonymous"
3  }
```

y Cookies Headers (5) Test Results

pretty Raw Preview Visualize JSON ▾

```
1  {
2  |   "content": [
3  |     {
4  |       "id": "8a8ae46f8033249a0180332bae",
5  |       "title": "Dunia Sophie",
6  |       "category": "Filosofi",
7  |       "author": "TheAnonymous",
8  |       "publisher": "Gremlin"
9  |     }
10 |   ],
11 |   "pageable": {
12 |     "sort": {
```

```
"searchBookTitle" : "deathly"
```

is Headers (5) Test Results

Raw Preview Visualize JSON ▾

```
"content": [
  {
    "id": "8a8ae46f803267800180326931680000",
    "title": "Harry Potter and the Deathly Hallows",
    "category": "Fantasy",
    "author": "Jk. Rowling",
    "publisher": "Bloomsbury"
  }
],
```

```
... "searchBookCategory": "filosofi"
```

cookies Headers (5) Test Results

Raw

Preview

Visualize

JSON



```
"content": [
  {
    "id": "8a8ae46f8033249a0180332bae260000",
    "title": "Dunia Sophie",
    "category": "Filosofi",
    "author": "TheAnonymous",
    "publisher": "Gremlin"
  }
],
```

GET

▼

{{host}}/books?size=10

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Se

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

▼

1

2

3

...

searchBookAuthor: "Jk. Rowling"

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

▼

↺

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

...

1

"content": [

{

"id": "8a8ae46f803267800180326931680000",

"title": "Harry Potter and the Deathly Hollows",

"category": "Fantasy",

"author": "Jk. Rowling",

"publisher": "Bloomsbury"

},

{

"id": "8a8ae46f8032678001803269e92d0002",

"title": "Harry Potter the Chamber of Secrets",

"category": "Fantasy",

"author": "Jk. Rowling",

"publisher": "Bloomsbury"

},

{

"id": "8a8ae46f8032678001803269b65c0001",

"title": "Harry Potter the Order of Phoenix",

"category": "Fantasy",

"author": "Jk. Rowling",

"publisher": "Bloomsbury"

},

{

"id": "8a8ae46f803267800180326a1c770003",

"title": "Harry Potter the Prisoner of The Azkaban",

"category": "Fantasy",

"author": "Jk. Rowling",

"publisher": "Bloomsbury"

}


]


### 3. ENTITAS TRX\_BOOK\_DETAILS

Pada dasarnya transaksi book details sama saja dengan kedua di atas akan tetapi ada sedikit perbedaan dari segi code. Dikarenakan trx book details mengambil id mst\_book sebagai foreign\_keynya. Berikut saya sertakan codenya untuk implementasi dari bookdetails.

```
@Override
public void saveDetailService(String id) {
    Book book = bookRepository.findById(id).get();
    BookDetails bookDetails = new BookDetails();
    bookDetails.setBook(book);
    bookDetailsRepository.save(bookDetails);
}
```

Khusus pada transaksi yang memiliki hubungan many to one sampai saat ini saya belum tau cara mengisi data menggunakan body, sehingga saya menggunakan @RequestParam untuk mengisi nilai.

**POST**  `{{host}}/bookdetails?id=8a8ae46f803267800180326a1c770003`

Params  Authorization Headers (8) Body Pre-request Script Tests Settings

	Data Output	Explain	Messages	Notifications
	<b>book_details_id</b> [PK] integer		<b>book_id</b> character varying (255)	
1		1	8a8ae46f803267800180326931680000	
2		2	8a8ae46f8032678001803269b65c0001	
3		3	8a8ae46f8032678001803269b65c0001	
4		4	8a8ae46f8032678001803269b65c0001	
5		5	8a8ae46f8032678001803269e92d0002	
6		6	8a8ae46f8032678001803269e92d0002	
7		7	8a8ae46f803267800180326a1c770003	
8		8	8a8ae46f803267800180326a1c770003	

Alasan saya menggunakan entitas trx\_book\_details untuk memberikan unique id pada tiap kuantitas buku. Untuk tiap satu judul buku memiliki UUID tetapi untuk tiap kuantitasnya diberi lagi id (UUID – ID).

GET

▼

{{host}}/bookdetails

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Set

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

≡

```

1  {
2    {
3      "id": 1,
4      "book": {
5        "id": "8a8ae46f803267800180326931680000",
6        "title": "Harry Potter and the Deathly Hollows",
7        "category": "Fantasy",
8        "author": "Jk. Rowling",
9        "publisher": "Bloomsbury"
10     }
11   },
12   {
13     "id": 2,
14     "book": {
15       "id": "8a8ae46f8032678001803269b65c0001",
16       "title": "Harry Potter the Order of Phoenix",
17       "category": "Fantasy",
18       "author": "Jk. Rowling",
19       "publisher": "Bloomsbury"
20     }
21   },
22   {
23     "id": 3,
24     "book": {
25       "id": "8a8ae46f8032678001803269b65c0001",
26       "title": "Harry Potter the Order of Phoenix",
27       "category": "Fantasy",
28       "author": "Jk. Rowling",
29       "publisher": "Bloomsbury"
30     }
31   },

```

Dapat dilihat pada saat GET dari tabel entitas `trx_book_details`, tiap kuantitas memiliki buku.

#### 4. ENTITAS TRX\_RESERVE

Masih sama dengan transaksi book details ada sedikit perbedaan dari segi code. Dikarenakan trx\_reserve mengambil id mst\_member serta id trx\_book\_details sebagai foreign\_keynya. Berikut saya sertakan codenya untuk implementasi dari bookdetails.

```
@Override
public void saveReserve(Date dateReserve, Date dateReturn, String memberId, Integer bookDetailsId) {
    Member member = memberRepository.findById(memberId).get();
    BookDetails bookDetails = bookDetailsRepository.findById(bookDetailsId).get();
    Reserve reserve = new Reserve();

    int getBookQty = bookDetails.getBook().getQuantity();

    if (getBookQty >= 0) {
        bookDetails.getBook().setQuantity(getBookQty-1);
        reserve.setBookDetails(bookDetails);
        reserve.setMember(member);
        reserve.setDateReserve(dateReserve);
        reserve.setDateReturn(dateReturn);

        reserveRepository.save(reserve);
    }
}
```

Khusus pada transaksi yang memiliki hubungan many to one sampai saat ini saya belum tau cara mengisi data menggunakan body, sehingga saya menggunakan @RequestParam untuk mengisi nilai. Pada code sudah terdapat pengurangan buku tiap peminjaman dan kondisi tidak dapat dipinjam jika buku kurang dari 0.

POST	({{host}})/reserve?dateReserve=2022-04-20&dateReturn=2022-04-23&memberId=8a8ae46f803267800180326ad4eb0004&bookDetailsId=6 ...	
Params	Authorization	Headers (8)
Body		
Pre-request Script		
Tests		
Settings		
Query Params		
	KEY	VALUE
<input checked="" type="checkbox"/>	dateReserve	2022-04-20
<input checked="" type="checkbox"/>	dateReturn	2022-04-23
<input checked="" type="checkbox"/>	memberId	8a8ae46f803267800180326ad4eb0004
<input checked="" type="checkbox"/>	bookDetailsId	6
	Key	Value

#### TABEL SAVE TRX\_RESERVE


Data Output	Explain	Messages	Notifications			
	<div>book_id</div> <div>[PK] character varying (255)</div>	<div>author</div> <div>character varying (255)</div>	<div>category</div> <div>character varying (255)</div>	<div>publisher</div> <div>character varying (255)</div>	<div>quantity</div> <div>integer</div>	<div>title</div> <div>character varying (255)</div>
1	8a8ae46f803267800180326931680000	Jk. Rowling	Fantasy	Bloomsbury	1	Harry Potter and the Deathly Hollows
2	8a8ae46f8032678001803269b65c0001	Jk. Rowling	Fantasy	Bloomsbury	2	Harry Potter the Order of Phoenix
3	8a8ae46f8032678001803269e92d0002	Jk. Rowling	Fantasy	Bloomsbury	1	Harry Potter the Chamber of Secrets
4	8a8ae46f803267800180326a1c770003	Jk. Rowling	Fantasy	Bloomsbury	2	Harry Potter the Prisoner of The Azkaban
5	8a8ae46f8033249a0180332bae260000	TheAnonymous	Filosofi	Gremlin	5	Dunia Sophie


Perhatikan kuantitas pada buku dunia sophie. Saat ini saya akan melakukan peminjaman pada buku tersebut. Dimana tiap buku dunia sophie memiliki id dari 9 hingga 13.

	book_details_id [PK] integer	book_id character varying (255)
1	1	8a8ae46f803267800180326931680000
2	2	8a8ae46f8032678001803269b65c0001
3	3	8a8ae46f8032678001803269b65c0001
4	4	8a8ae46f8032678001803269b65c0001
5	5	8a8ae46f8032678001803269e92d0002
6	6	8a8ae46f8032678001803269e92d0002
7	7	8a8ae46f803267800180326a1c770003
8	8	8a8ae46f803267800180326a1c770003
9	9	8a8ae46f8033249a0180332bae260000
10	10	8a8ae46f8033249a0180332bae260000
11	11	8a8ae46f8033249a0180332bae260000
12	12	8a8ae46f8033249a0180332bae260000
13	13	8a8ae46f8033249a0180332bae260000

Alasan saya menggunakan entitas trx\_book\_details untuk memberikan unique id pada tiap kuantitas buku. Untuk tiap satu judul buku memiliki UUID tetapi untuk tiap kuantitasnya diberi lagi id (UUID – ID).

Saya melakukan peminjaman dengan member id Gundala

POST  {{host}}/reserve?dateReserve=2022-04-20&dateReturn=2022-04-23&memberId=8a8ae46f8035b91d018035bcb8290002&bookDetailsId=9

Params  Authorization Headers (8) Body Pre-request Script Tests Settings

## Query Params

KEY	VALUE
<input checked="" type="checkbox"/> dateReserve	2022-04-20
<input checked="" type="checkbox"/> dateReturn	2022-04-23
<input checked="" type="checkbox"/> memberId	8a8ae46f8035b91d018035bcb8290002
<input checked="" type="checkbox"/> bookDetailsId	9
Key	Value

Data Output	Explain	Messages	Notifications			
<div><div>book_id</div><div>[PK] character varying (255)</div></div>	<div><div>author</div><div>character varying (255)</div></div>	<div><div>category</div><div>character varying (255)</div></div>	<div><div>publisher</div><div>character varying (255)</div></div>	<div><div>quantity</div><div>integer</div></div>	<div><div>title</div><div>character varying (255)</div></div>	
1	8a8ae46f803267800180326931680000	Jk. Rowling	Fantasy	Bloomsbury	1	Harry Potter and the Deathly Hollows
2	8a8ae46f8032678001803269b65c0001	Jk. Rowling	Fantasy	Bloomsbury	2	Harry Potter the Order of Phoenix
3	8a8ae46f8032678001803269e92d0002	Jk. Rowling	Fantasy	Bloomsbury	1	Harry Potter the Chamber of Secrets
4	8a8ae46f803267800180326a1c770003	Jk. Rowling	Fantasy	Bloomsbury	2	Harry Potter the Prisoner of The Azkaban
5	8a8ae46f8033249a0180332bae260000	TheAnonymous	Filosofi	Gremlin	4	Dunia Sophie

Dapat dilihat pada tabel mst\_book baris ke-5 dengan judul dunia sophie terdapat pengurangan kuantitas sebanyak satu buah. Jika kita melakukan GET pada tabel reserve didapatkan seperti ini.

```

93     {
94         "id": 14,
95         "dateReserve": "2022-04-20",
96         "dateReturn": "2022-04-23",
97         "member": {
98             "id": "8a8ae46f8035b91d018035bcb8290002",
99             "firstName": "Gundala",
100             "lastName": "Syaputra",
101             "phone": "082221082107"
102         },
103         "bookDetails": {
104             "id": 9,
105             "book": {
106                 "id": "8a8ae46f8033249a0180332bae260000",
107                 "title": "Dunia Sophie",
108                 "category": "Filosofi",
109                 "author": "TheAnonymous",
110                 "publisher": "Gremlin"
111             }
112         }
113     }
114 }

```



## REVISI (PENAMBAHAN FITUR):

Penambahan fitur pengembalian. Program dapat meminjam buku dan mengurangi jumlah buku di stock dengan kondisi dapat meminjam jika kuantitas peminjaman kurang dari 3.

The screenshot shows a REST client interface with a POST request to `{{host}}/reserve?dateReserve=2022-04-20&dateReturn=2022-04-23&memberId=8a8ae46f80361f2f0180362299ad0001&bookDetailsId=1`. The 'Query Params' tab is active, displaying a table of parameters:

KEY	VALUE
<input checked="" type="checkbox"/> dateReserve	2022-04-20
<input checked="" type="checkbox"/> dateReturn	2022-04-23
<input checked="" type="checkbox"/> memberId	8a8ae46f80361f2f0180362299ad0001
<input checked="" type="checkbox"/> bookDetailsId	1
Key	Value

Below the table, the 'body' tab is selected, showing a 'Pretty' view of the request body with a single line of text: '1'.

## TABEL SAVE TRX\_RESERVE

Setelah melakukan save untuk `trx_reserve` selanjutnya value pada transaksi reserve akan menjadi sebagai berikut.

Data Output Explain Messages Notifications							
	id [PK] integer	date_reserve date	date_return date	qty integer	status character varying (255)	book_details_id integer	member_id character varying (255)
1	6	2022-04-20	2022-04-23	1	active	1	8a8ae46f80361f2f0180362299ad0001

Sedangkan pada tabel stock buku akan terjadi pengurangan sebanyak satu buah untuk tiap peminjaman.

Data Output Explain Messages Notifications						
	book_id [PK] character varying (255)	author character varying (255)	category character varying (255)	publisher character varying (255)	quantity integer	title character varying (255)
1	8a8ae46f80361f2f01803622047d0000	Sir. Conan Doyle	Drama-Mystery	George Newnes	4	The Adventures Sherlock Holmes

PUT		{{host}}/reserve?id=6	
Params		Authorization	Headers (8)
Data Output		Explain	Messages
		Notifications	
	id [PK] integer	date_reserve date	date_return date
	qty integer	status character varying (255)	book_details_id integer
	member_id character varying (255)		
1	6	2022-04-20	2022-04-23
	0	Inactive	1
			8a8ae46f80361f2f0180362299ad0001

TABEL RESERVE SETELAH PENGEMBALIAN

Pada tabel dapat dilihat qty menjadi 0 dan status menjadi inactive.

Data Output		Explain	Messages	Notifications	
	book_id [PK] character varying (255)	author character varying (255)	category character varying (255)	publisher character varying (255)	quantity integer
	title character varying (255)				
1	8a8ae46f80361f2f01803622047d0000	Sir. Conan Doyle	Drama-Mystery	George Newnes	5
					The Adventures Sherlock Holmes

TABEL MST\_BOOK SETELAH PENGEMBALIAN

Pada tabel dapat dilihat field quantity bertambah satu setelah pengembalian