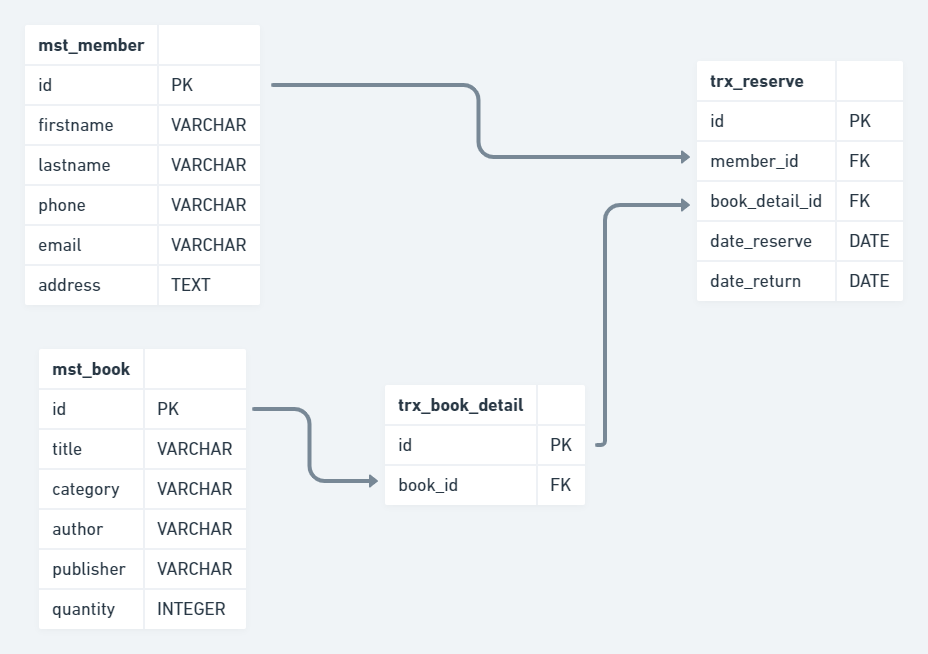
**Ahmad Fansyuri**

**17 April 2022**

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

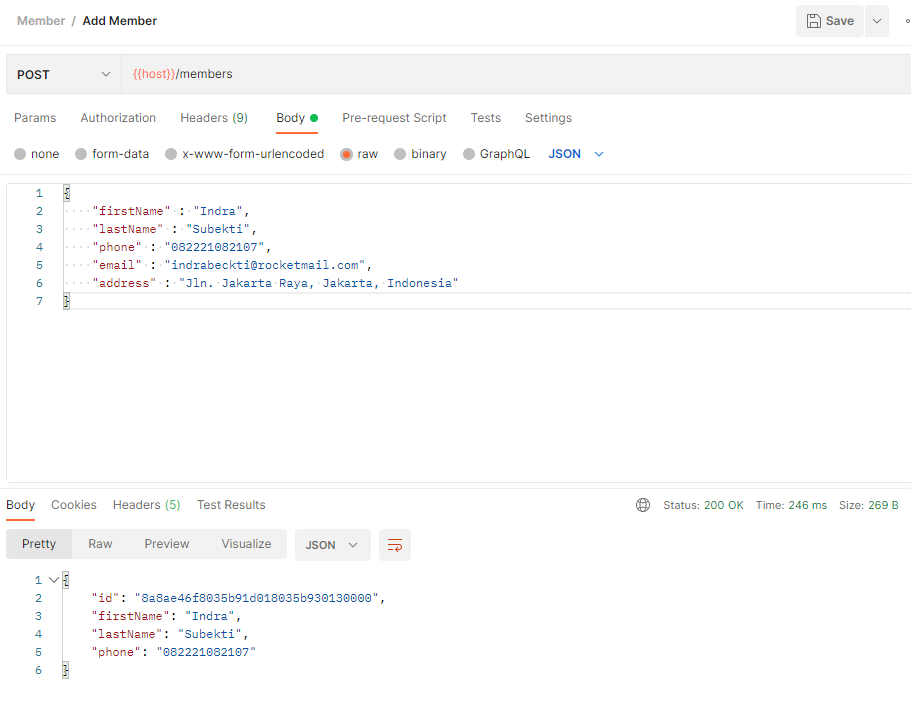


Tabel 1 ERD library

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.

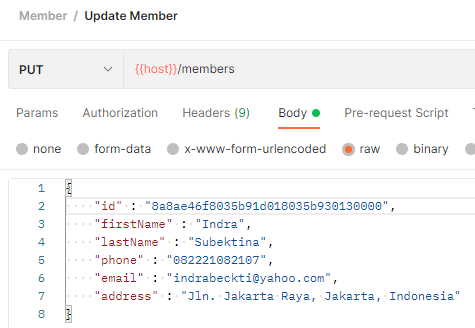
1. **Entitas mst\_member**

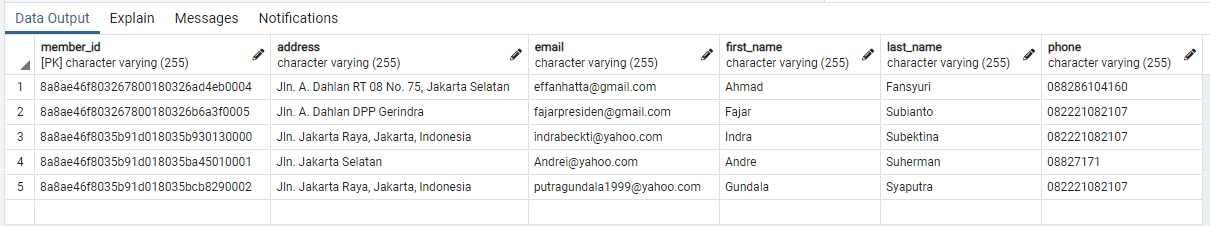
* **INSERT**



**TABEL 1.1 Penambahan Member**

* **UPDATE**

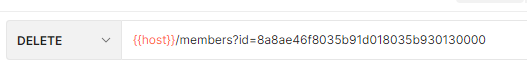
****

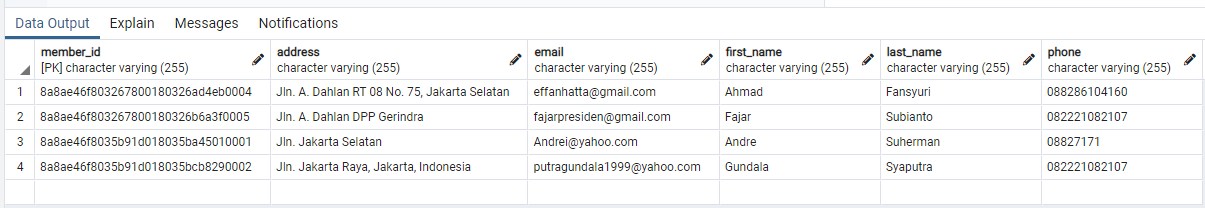


**TABEL UPDATE**

Dapat dilihat pada baris ketiga perubahan email dan last name.

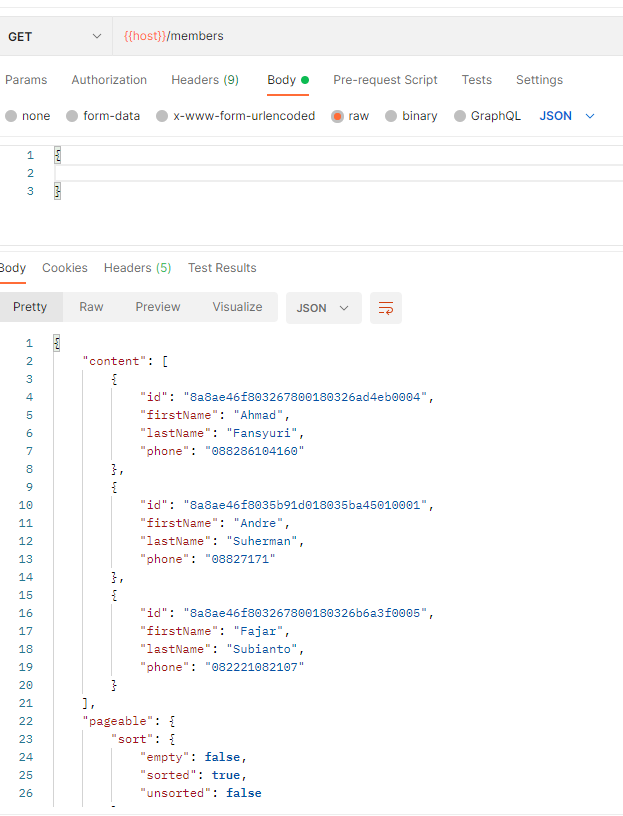
* **DELETE**



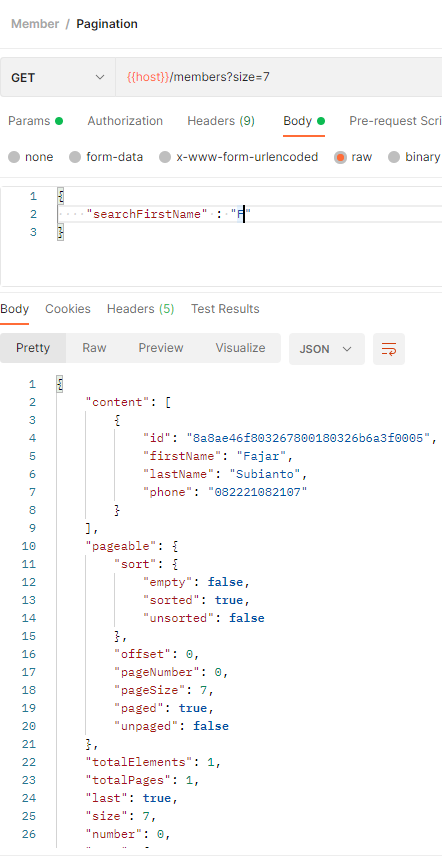


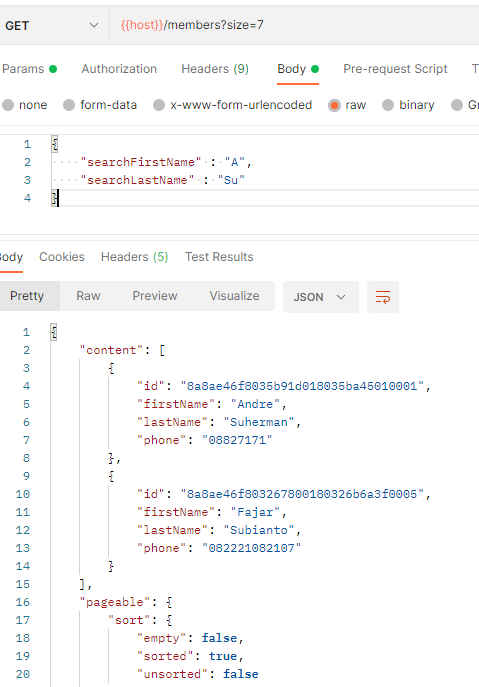
Pada tabel member dapat dilihat baris ketiga atas nama indra subektina sudah dihapus.

* **PAGINATION (FILTERING & SEARCHING)**



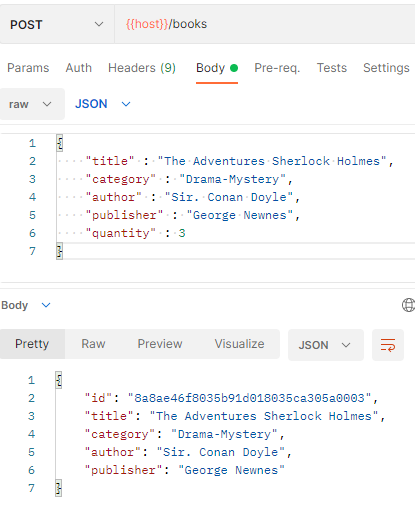
Secara default saya menetapkan size dan page nya masing-masing di size=3 dan page=0.





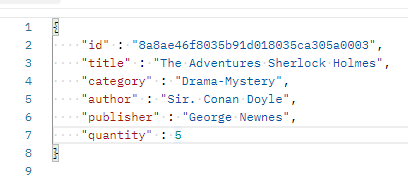
1. **Entitas mst\_book**

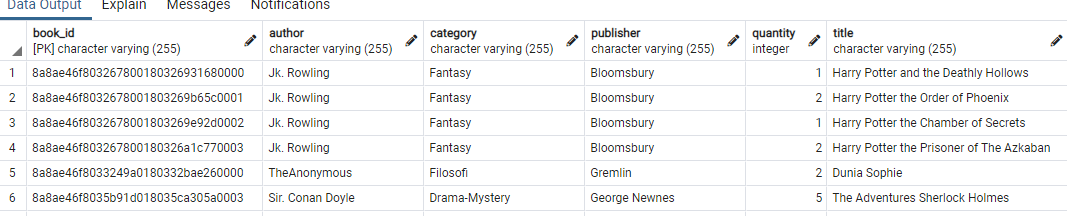
* **INSERT**



**TABEL 2.1 SAVE**

* **UPDATE**

****

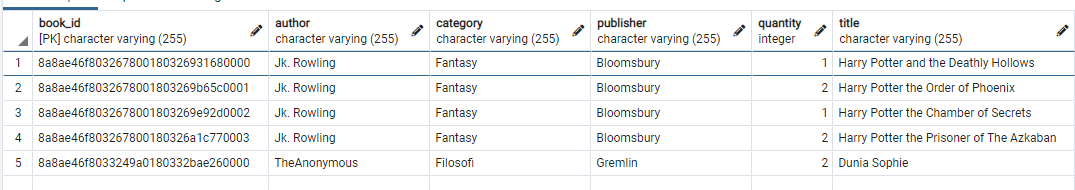
****

**TABEL 2.2 UPDATE BOOK**

Dapat dilihat pada baris 6 perubahan Quantity.

* **DELETE**

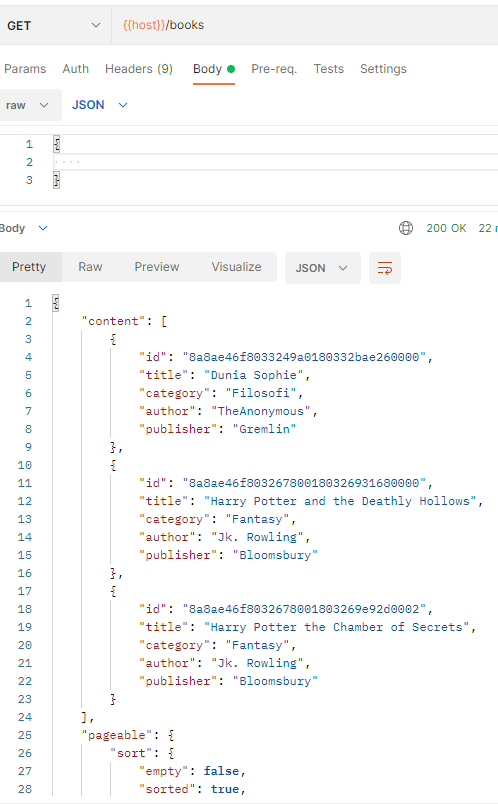




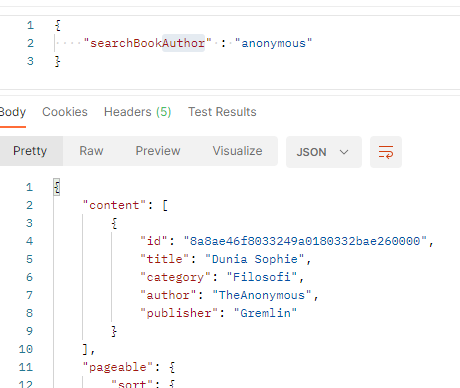
**TABEL 2.3 DELETE BOOK**

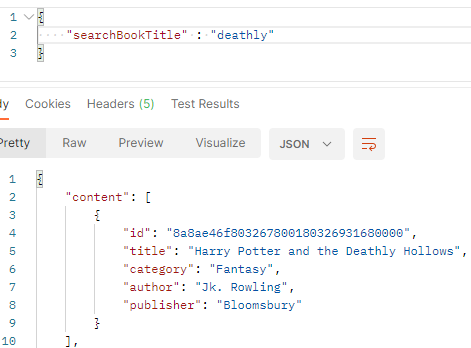
Pada tabel book Sherlock Holmes

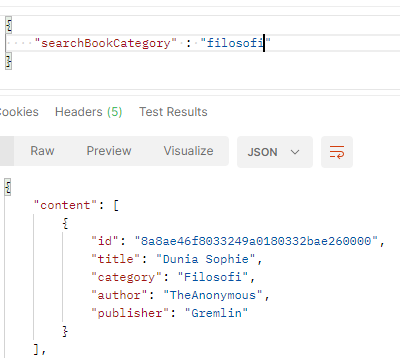
* **PAGINATION (FILTERING & SEARCHING)**

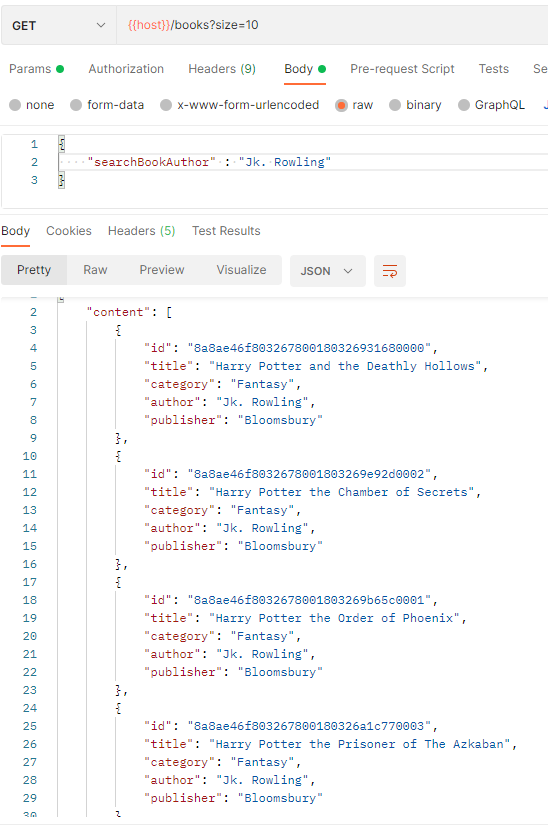


Secara default saya menetapkan size dan page nya masing-masing di size=3 dan page=0.



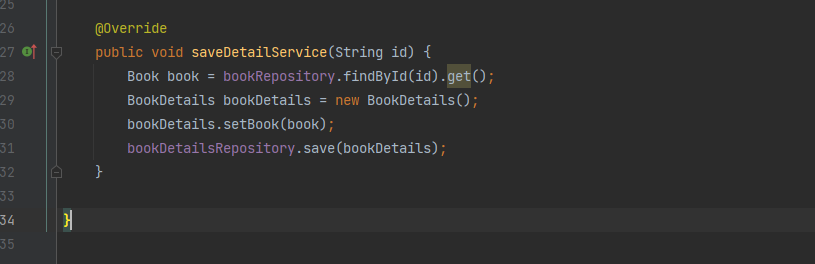




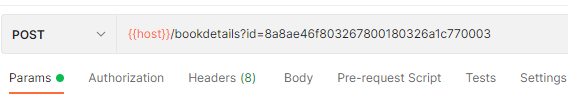


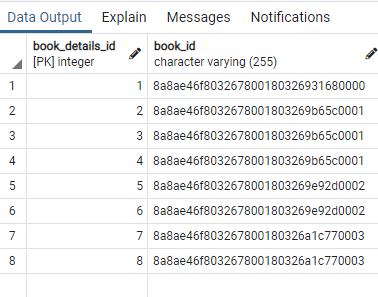
1. **ENTITAS TRX\_BOOK\_DETAILS**

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.

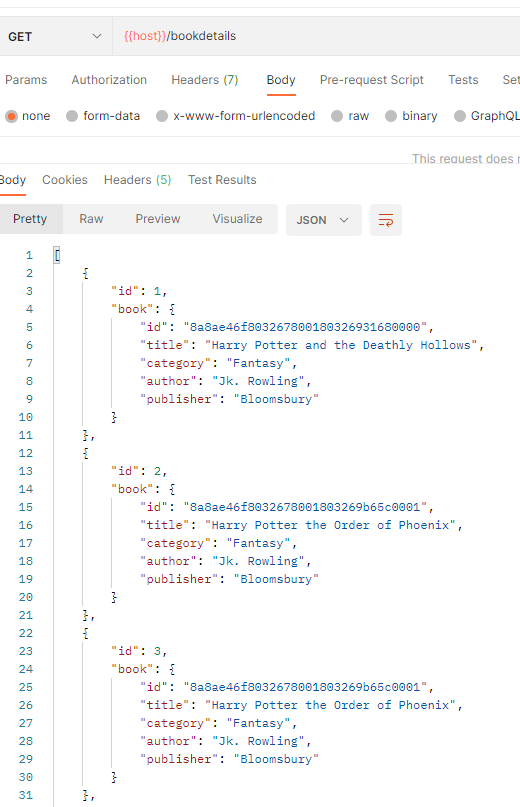


Khusus pada transaksi yang memiliki hubungan many to one sampai saat ini saya belum tau cara mengisi data menggunakan body, sehingga saya menggunakan @RequestParam untuk mengisi nilai.





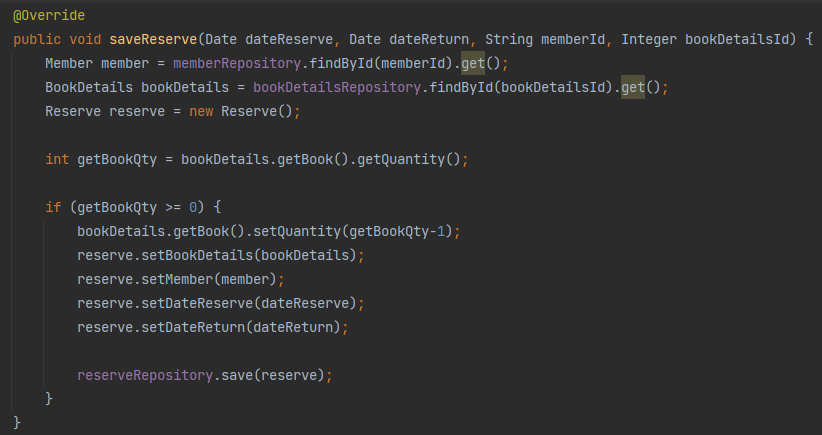
Alasan saya menggunakan entitas trx\_book\_details untuk memberikan unique id pada tiap kuantitas buku. Untuk tiap satu judul buku memiliki UUID tetapi untuk tiap kuantitasnya diberi lagi id (UUID – ID).



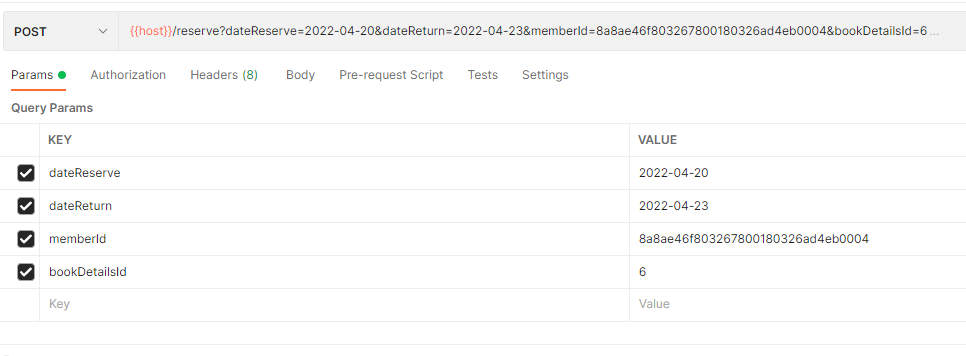
Dapat dilihat pada saat GET dari tabel entitas trx\_book\_details, tiap kuantitas memiliki buku.

**4. ENTITAS TRX\_RESERVE**

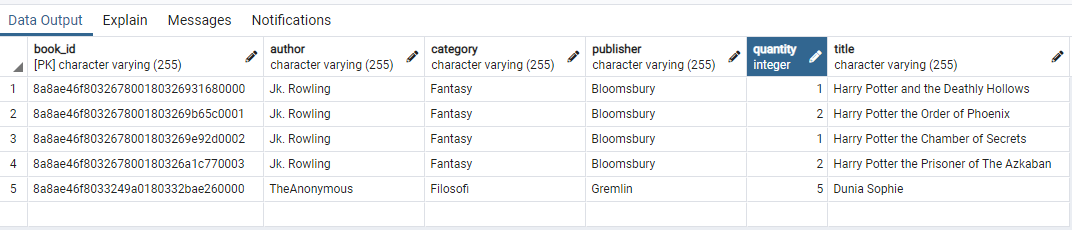
Masih sama dengan transaksi book details ada sedikit perbedaan dari segi code. Dikarenakan trx\_reserve mengambil id mst\_member serta id trx\_book\_details sebagai foreign\_keynya. Berikut saya sertakan codenya untuk implementasi dari bookdetails.



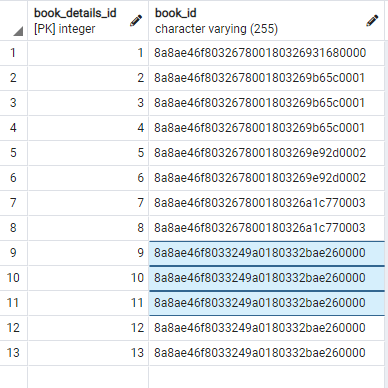
Khusus pada transaksi yang memiliki hubungan many to one sampai saat ini saya belum tau cara mengisi data menggunakan body, sehingga saya menggunakan @RequestParam untuk mengisi nilai. Pada code sudah terdapat pengurangan buku tiap peminjaman dan kondisi tidak dapat dipinjam jika buku kurang dari 0.



**TABEL SAVE TRX\_RESERVE**

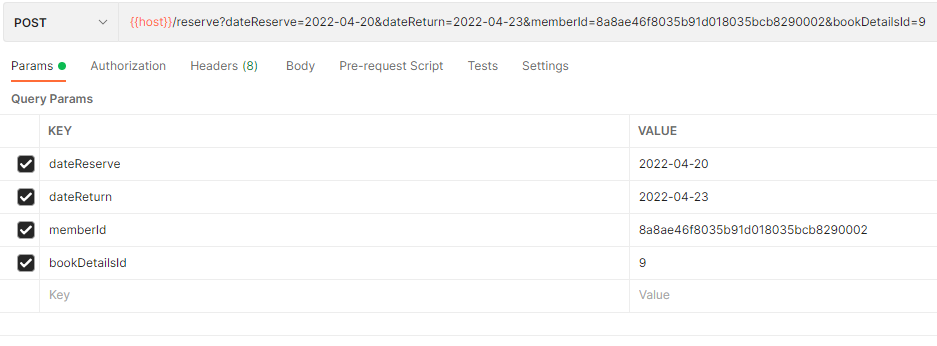


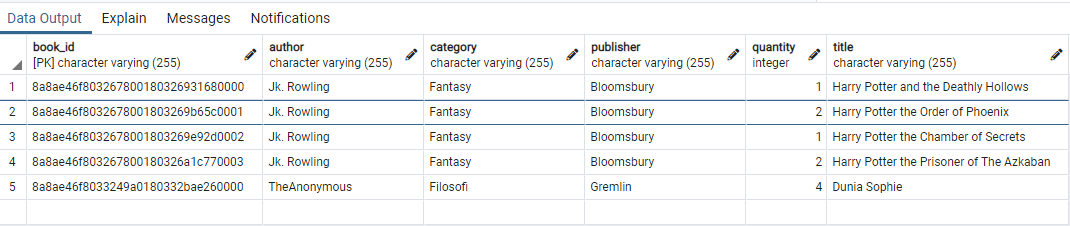
Perhatikan kuantitas pada buku dunia sophie. Saat ini saya akan melakukan peminjaman pada buku tersebut. Dimana tiap buku dunia sophie memiliki id dari 9 hingga 13.



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





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

