

MAJOO TEST CASE 1: Membuat API

*Catatan: Hasil output dari API endpoint ini juga dapat dilihat pada postman,
<https://www.postman.com/portaonesia/workspace/majoo-test-cases>*

- A. Fungsi *login* (**terdapat pada source code**)
- B. *Authorization* (**terdapat pada source code**)
- C. Laporan nama merchant, omzet per hari dalam pada bulan november mulai tanggal 1 sampai dengan tanggal 30 dengan pagination. Apabila tidak ada transaksi pada tanggal itu, omzet akan bernilai 0 (**terdapat pada source code**)
- D. Laporan nama merchant, nama outlet, omzet per hari dalam pada bulan november mulai tanggal 1 sampai dengan tanggal 30 dengan pagination. Apabila tidak ada transaksi pada tanggal itu, omzet akan bernilai 0 (**terdapat pada source code**)
- E. Poin C, user tidak bisa melakukan akses pada merchant_id yang bukan miliknya (**terdapat pada source code**)
- F. Poin D, user tidak bisa melakukan akses pada outlet_id yang bukan miliknya (**terdapat pada source code**)
- G. Dari *test case* pada point C dan D, apakah ERD yang dibentuk sudah optimal? Berikan penjelasannya!

Menurut saya, struktur ERD yang dibentuk sudah optimal. Dalam ERD tersebut sudah terdapat:

- 1. Entitas, yaitu nama database
 - 2. Attribute, yaitu nama kolom
 - 3. Relationship, yaitu hubungan antara kolom pada tabel yang satu dengan kolom pada tabel lainnya. Dimana hubungannya tersebut juga sudah jelas menggambarkan, apakah hubungannya *one-to-one*, *one-to-many* atau *many-to-many*
- H. Dokumen teknis Data Manipulation Language (DML) (**terdapat pada halaman selanjutnya**)

Data Manipulation Language

Mendapatkan Data User

Function 1: GetUserFromLogin (GET)

Query 1: Mendapatkan tabel user. Hasil query hanya akan menampilkan user yang memiliki user_name 'admin1'

```
SELECT user_name,password
FROM users
WHERE user_name = 'admin1'
LIMIT 1;
```

Function 2: GetUserFromAuthorization (GET)

Query 2: Mendapatkan tabel user. Hasil query hanya akan menampilkan user yang memiliki id = '1'

```
SELECT *
FROM users
WHERE id = '1'
LIMIT 1;
```

Mendapatkan Laporan Transaksi

Function 1: GetTransactions (GET)

Query 1: Mendapatkan laporan nama merchant dan omzet per hari mulai tanggal 1 November – 30 November 2021 untuk user_id = '1'. Menggunakan pagination page = '1' dan page_size = '15'. Omzet merupakan total bill_total per hari. Apabila tidak ada transaksi pada tanggal itu, omzet akan bernilai 0.

```
WITH recursive cte AS (
    SELECT '2021-11-01' as created_at
    UNION ALL
    SELECT created_at + INTERVAL 1 DAY FROM cte WHERE DATE(created_at) < '2021-11-30'
)
SELECT
    cte.created_at as date
    merchants.merchant_name,
    IF(
        DATE(cte.created_at) = DATE(transactions.created_at),
        SUM(transactions.bill_total),
        0
    ) as omzet
FROM
    cte
    LEFT JOIN transactions ON DATE(transactions.created_at) <= cte.created_at
    INNER JOIN merchants ON merchants.id = transactions.merchant_id
    INNER JOIN users ON users.id = merchants.user_id
WHERE
    users.id = '1'
GROUP BY DATE(cte.created_at)
ORDER BY cte.created_at ASC
LIMIT 0,15;
```

Function 2: GetTransactionsWithOutlet (GET)

Query 2: Mendapatkan laporan nama merchant, nama outlet dan omzet per hari mulai tanggal 1 November – 30 November 2021 untuk user_id = '1'. Menggunakan pagination page = '1' dan page_size = '15'. Omzet merupakan total bill_total per hari. Apabila tidak ada transaksi pada tanggal itu, omzet akan bernilai 0

```
WITH recursive cte AS (  
    SELECT '2021-11-01' as created_at  
    UNION ALL  
    SELECT created_at + INTERVAL 1 DAY FROM cte WHERE DATE(created_at) < '2021-11-30'  
)  
SELECT  
    cte.created_at as date  
    merchants.merchant_name,  
    outlets.outlet_name,  
    IF(  
        DATE(cte.created_at) = DATE(transactions.created_at),  
        SUM(transactions.bill_total),  
        0  
    ) as omzet  
FROM  
    cte  
    LEFT JOIN transactions ON DATE(transactions.created_at) <= cte.created_at  
    INNER JOIN merchants ON merchants.id = transactions.merchant_id  
    INNER JOIN users ON users.id = merchants.user_id  
    INNER JOIN outlets  
        ON outlets.id = transactions.outlet_id  
        AND merchants.id = outlets.merchant_id  
WHERE  
    users.id = '1'  
GROUP BY DATE(cte.created_at)  
ORDER BY cte.created_at ASC  
LIMIT 0,15;
```