**LAPORAN PERTEMUAN 9 TEORI**

**SISTEM BASIS DATA**



**Ditulis oleh :**

|  |  |
| --- | --- |
| NAMA | : Ilham Romadhani |
| NIM | : 23.230.0030 |
| KELAS | : 2P52 |

**Dosen Pengampu :**

Mosses Aidjilli, M.Kom

PROGRAM STUDI SISTEM INFORMASI

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER

(STMIK) WIDYA PRATAMA PEKALONGAN

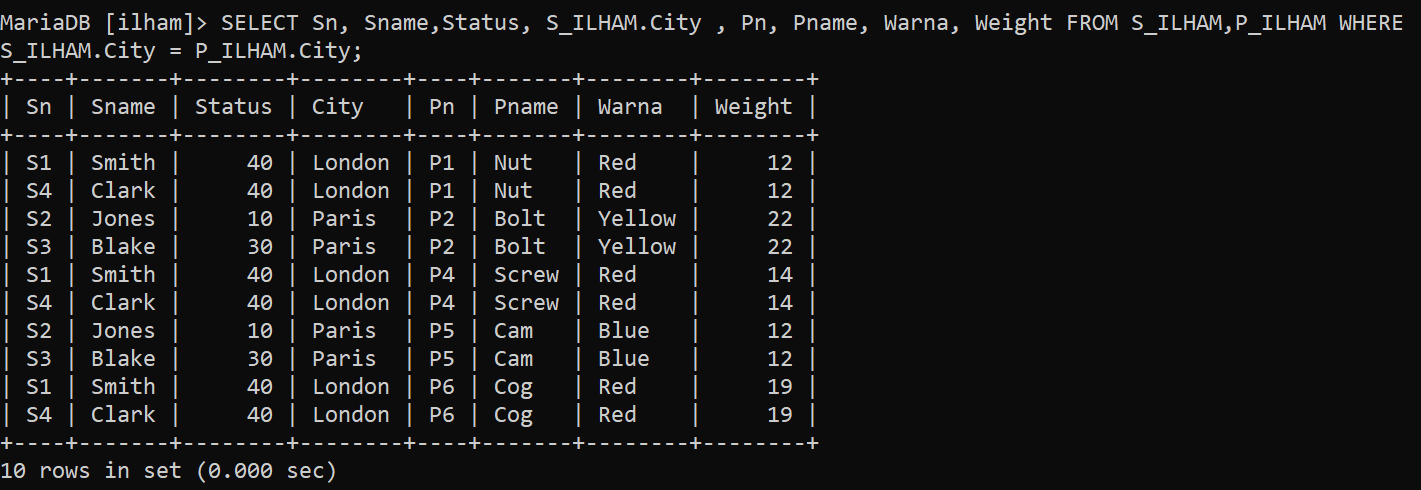
2024

## ➢ DATA DEFINITION LANGUAGE (DDL)

* **Menampilkan record (SELECT lebih dari satu tabel / JOIN)**

1. Menampilkan semua supplier dan part yang keduanya bertempat tinggal

pada kota yang sama

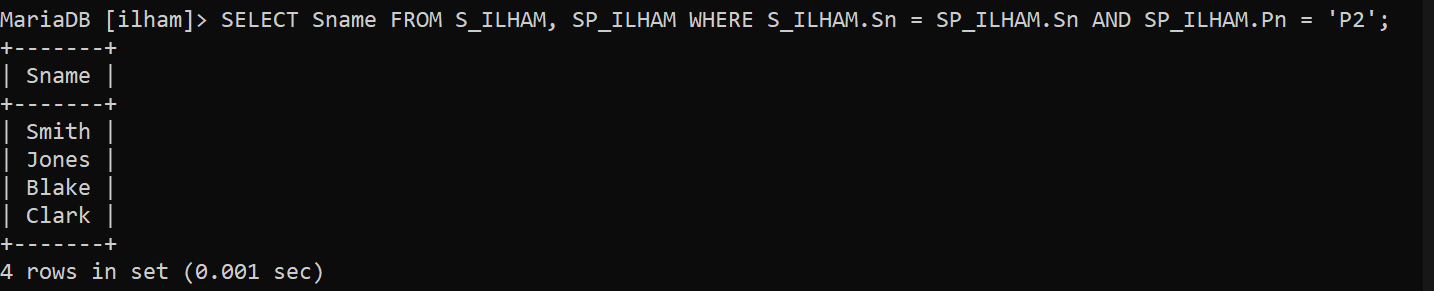


**Select sn,sname,status,S\_ilham.city,pn,pname,warna,weight** : memilih kolom-kolom dari tabel.

**from s\_ilham, p\_ilham** : dari 2 tabel.

**where s\_ilham.city = p\_ilham.city** : kondisi dimana nilai filed city dari kedua tabel tersebut harus sama. (namatabel.namafield).

1. Menampilkan nama supplier yang memasok barang dengan nomor part P2



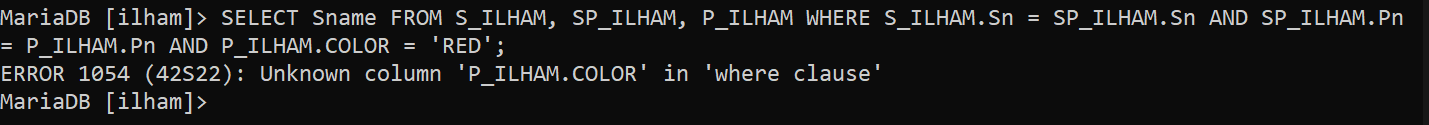
**Select Sname** : memilih field/kolom Sname.

**from s\_ilham, sp\_ilham** : dari tabel s\_ilham dan sp\_ilham.

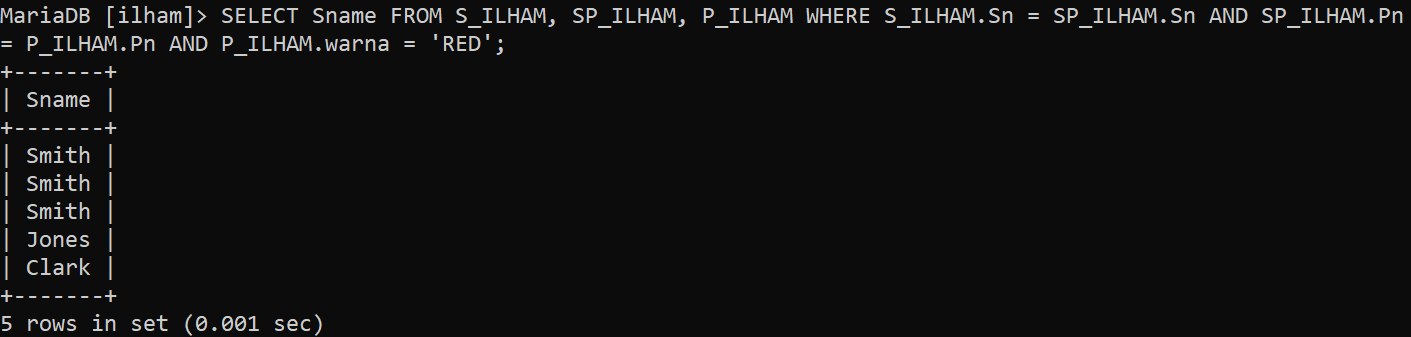
**where s\_ilham.sn = sp\_ilham.sn and sp\_ilham.pn = ‘P2’** : kondisi dimana nilai kolom sn dari tabel s\_ilham harus sama dengan nilai kolom sn pada tabel sp\_ilham. Serta ditambah dengan logika and dengan kondisi nilai kolom pn harus sama dengan ‘p2’

( string p2). Jadi karena ada logika AND, kedua kondisi tersebut harus benar.

1. Menampilkan nama supplier yang memasok part berwarna merah



**Error karena field sudah di ubah ke warna bukan color**

****

**Select Sname :** Memilih field Sname

**FROM s\_ilham, sp\_ilham, p\_ilham** : Dari 3 tabel tersebut.

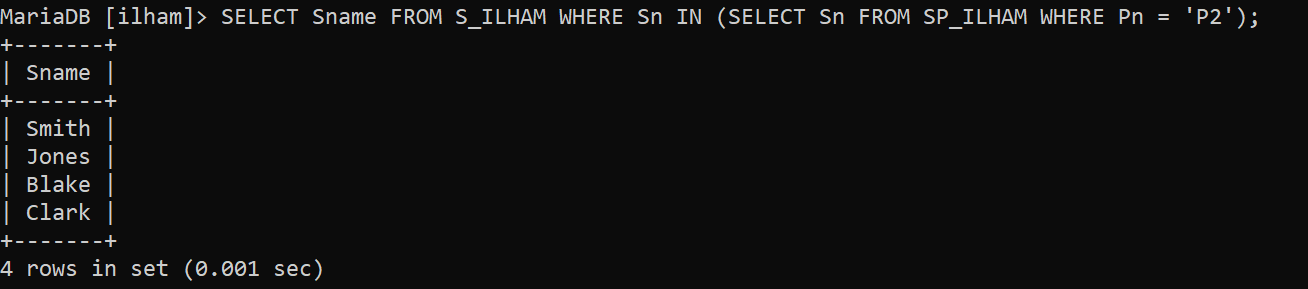
**Where s\_ilham.sn = sp\_ilham.sn**: kondisi dimana nilai pada kolom sn pada tabel s\_ilham sama dengan nilai pada kolom sn pada tabel sp\_ilham.

**AND sp\_ilham.pn = p\_ilham.pn** : ada logika AND dimana kondisi diantara keduanya harus benar = true. Kondisi selanjutnya dimana nilai kolom pn pada tabel sp\_ilham harus sama dengan nilai kolom pn pada tabel p\_ilham.

**AND p\_ilham.warna = ‘RED’** : ada logika AND lagi. Kondisi selanjutnya dimana kolom warna pada tabel p\_ilham harus sama dengan nilai = ‘RED’.

* **Menampilkan record (SELECT lebih dari satu tabel / SELECT Bertingkat)**

1. Menampilkan nama supplier yang memasok barang dengan nomor part P2



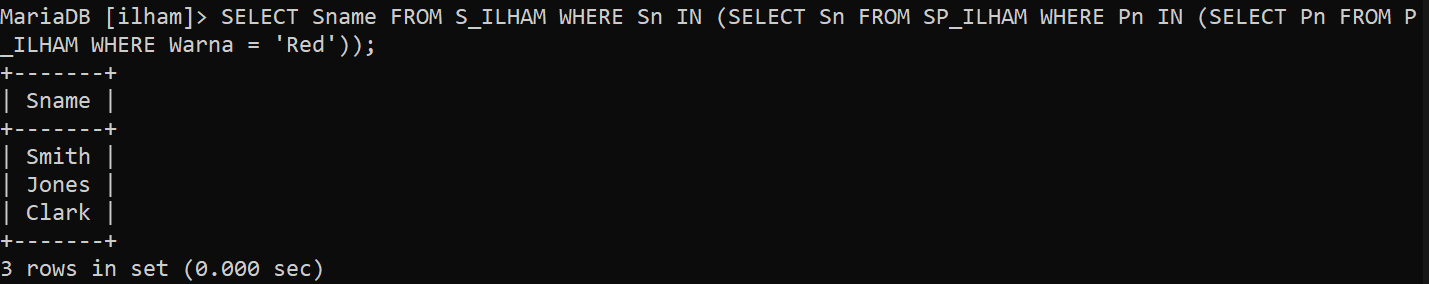
**Select Sname :** memilih kolom sname.

**FROM S\_Ilham :** dari tabel s\_Ilham.

Operator **IN** adalah cara untuk memeriksa apakah nilai kolom ada dalam daftar nilai tertentu.

**WHERE Sn IN (Select Sn FROM sp\_ilham WHERE Pn = ‘P2’)** : kondisi dimana nilai kolom Sn dari tabel S\_Ilham dicek apakah ada dalam daftar nilai Sn yang dihasilkan dari subquery SELECT Sn FROM sp\_ilham WHERE Pn = ‘P2’. Dengan kata lain, kita memilih baris-baris dari tabel S\_Ilham di mana nilai Sn-nya ada di dalam daftar nilai yang kita dapatkan dari subquery.

1. Menampilkan nama supplier yang memasok part berwarna merah

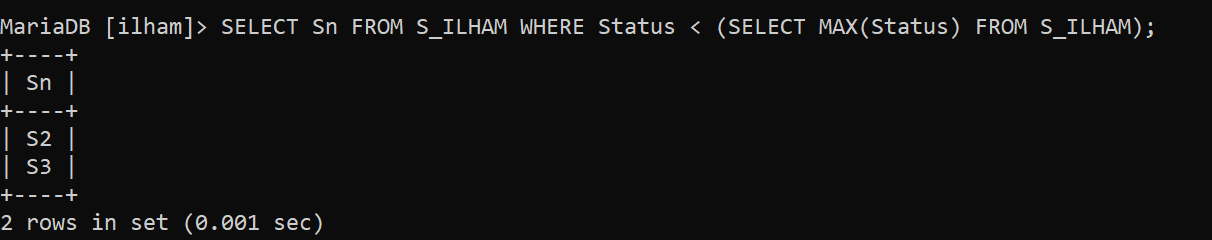


**SELECT Sname FROM S\_ILHAM :**  Memilih kolom Sname dari tabel s\_ilham.

**WHERE Sn IN (SELECT Sn FROM SP\_ILHAM WHERE Pn IN (SELECT Pn FROM P\_ILHAM WHERE Warna = ‘Red’)) :** kondisi untuk membatasi hasil yang kita inginkan. Di sini, kita menggunakan operator **IN** untuk memeriksa apakah nilai **Sn** (nomor supplier) ada dalam hasil subquery di dalam tanda kurung.

1. Menampilkan no.supplier dengan nilai status lebih kecil daripada nilai

maksimum status yang ada pada tabel S\_ilham



**SELECT Sn FROM S\_ILHAM :** Memilih kolom Sn dari tabel s\_ilham.

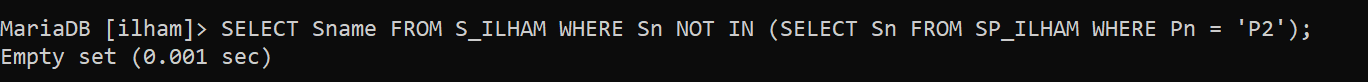
**WHERE status < (SELECT MAX(Status) FROM S\_ILHAM) :** kondisi dimana kolom nilai status pada tabel s\_ilham lebih kecil dari nilai maksimum status.

**(SELECT MAX(Status) FROM S\_ILHAM) :** subquery ini menghitung nilai maksimum dari kolom status di tabel s\_ilham.

**MAX(Status) :** Mengambil nilai terbesar dari kolom status.

1. Menampilkan nama supplier yang tidak memasok barang dengan nomor part

P2

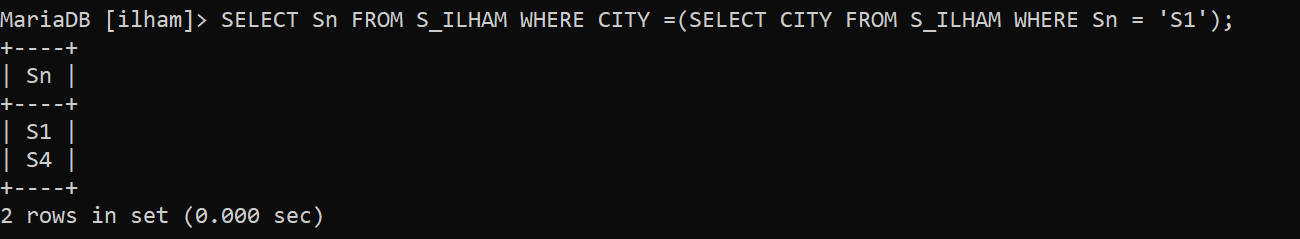


**SELECT Sname FROM s\_ilham :** Memilih kolom Sname dari tabel s\_ilham.

**WHERE Sn NOT IN :** kondisi dimana mencari Sn yang tidak ada dalam daftar nilai yang di tentukan atau di hasilkan dari subquery di bawah ini.

**(SELECT Sn FROM SP\_ILHAM WHERE Pn = ‘P2’) :** Mencari nilai Sn dengan kondisi dimana kolom Pn pada tabel SP\_ILHAM nilainya sama dengan ‘P2’.

1. Menampilkan semua nomor supplier yang sama lokasinya dengan S1



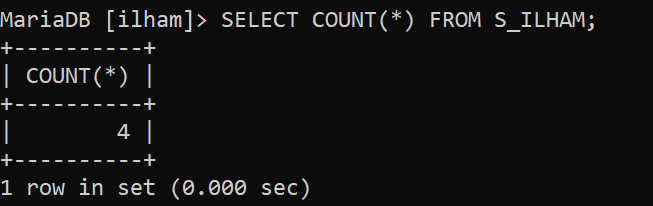
**SELECT Sn FROM S\_ILHAM :** Memilih kolom Sn dari tabel s\_ilham.

**WHERE CITY = (SELECT CITY FROM S\_ILHAM WHERE Sn = ‘S1’) :** Kondisi dimana field city pada tabel s\_ilham sama dengan subquery().

**(SELECT CITY FROM S\_ILHAM WHERE Sn = ‘S1’) :** Mencari nilai city dari tabel s\_ilham dimana jika nilai Sn pada tabel s\_ilham sama dengan ‘S1’.

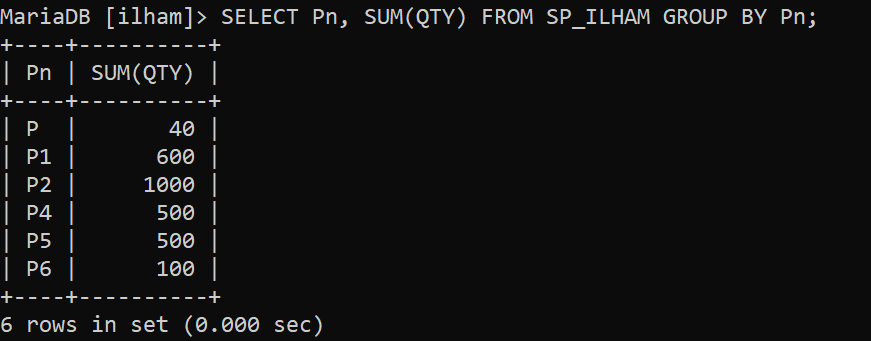
* **Fungsi Perhitungan**

1. Menghitung jumlah supplier



**SELECT COUNT(\*) FROM S\_ILHAM :** Menghitung jumlah record pada tabel.

1. Menampilkan nomor part dan total kuantitas pengiriman dari setiap part



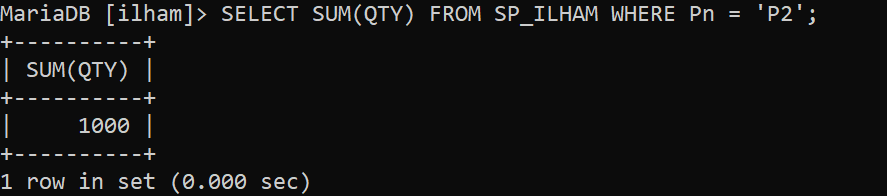
**SELECT Pn, SUM(QTY) :** Memilih kolom Pn dan fungsi SUM untuk menjumlahkan nilai di kolom QTY.

**FROM SP\_ILHAM :**  Dari tabel sp\_ilham.

**GROUP BY Pn:** Mengelompokkan baris yang memiliki nilai yang sama di kolom Pn.

Setelah pengelompokan, fungsi agregat seperti SUM diterapkan ke setiap grup. Dalam hal ini, kita menjumlahkan nilai **QTY** untuk setiap grup yang diidentifikasi oleh **Pn**.

1. Menghitung jumlah kuantitas dari P2 yang telah disupply

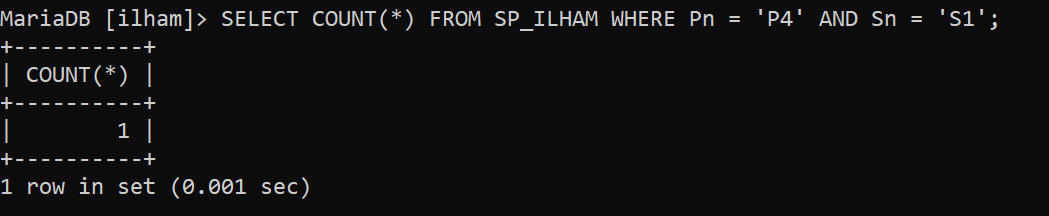


**SELECT SUM(QTY) FROM SP\_ILHAM :** Memilih field qty dengan fungsi SUM untuk menjumlahkan nilai pada kolom qty.

**WHERE Pn = ‘P2’ :** kondisi dimana field Pn = ‘P2’ (string P2). Akan menjumlahkan semua nilai qty dari kolom yang sama dengan P2.

1. Menampilkan jumlah pengiriman barang dengan nomor P4 dan dipasok oleh

nomor suppplier S1



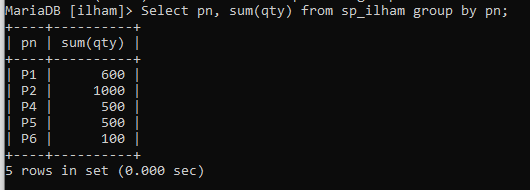
**SELECT COUNT(\*) FROM SP\_ILHAM :** Menjumlahkan total record atau baris dari tabel sp\_ilham, dengan kondisi di bawah ini.

**WHERE Pn = ‘P4’ AND Sn = ‘S1’ :**  Kondisi dimana record akan terhitung/true jika field Pn = ‘P4’ DAN Sn = ‘S1’, karena ada logika AND jadi kedua kondisi tersebut harus benar maka data akan di hitung oleh fungsi count.

1. Menampilkan nomor part dan total kuantitas dari masing-masing part



P3 menghasilkan kesalahan karena bukan nama kolom yang valid dalam tabel. Maka dari itu untuk menampilkan total kuantitas dari masing-masing part ganti dengan Pn.



**SELECT Pn, SUM(QTY) FROM SP\_ILHAM :** Memilih kolom Pn dan fungsi SUM untuk menjumlahkan nilai di kolom QTY.

**GROUP BY Pn :** Mengelompokkan baris yang memiliki nilai yang sama di kolom **Pn**. Setelah pengelompokan, fungsi agregat seperti SUM diterapkan ke setiap grup. Dalam hal ini, kita menjumlahkan nilai **QTY** untuk setiap grup yang diidentifikasi oleh **Pn**.

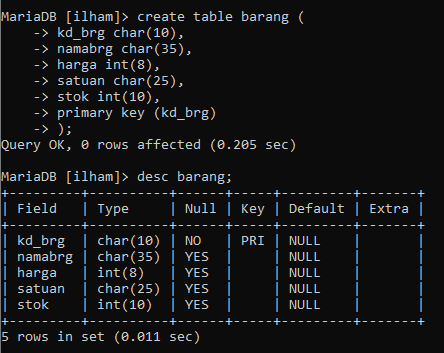
**TEORI YANG ADA DI PAPAN TULIS**

**MASTER = Primary Key**

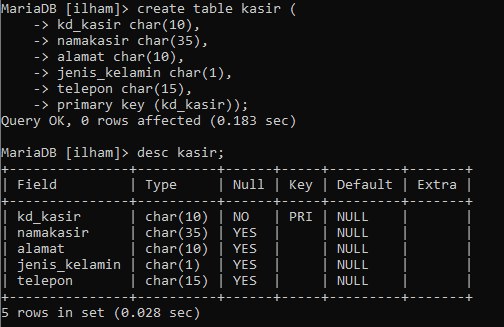
**TRANSAKSI = FOREIGN KEY**

**MASTER TRANSAKSI = PRIMARY KEY**

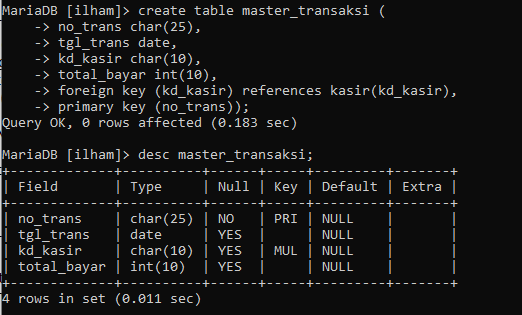
1. **BUATLAH TABEL BARANG**

****

1. **BUATLAH TABEL KASIR**

****

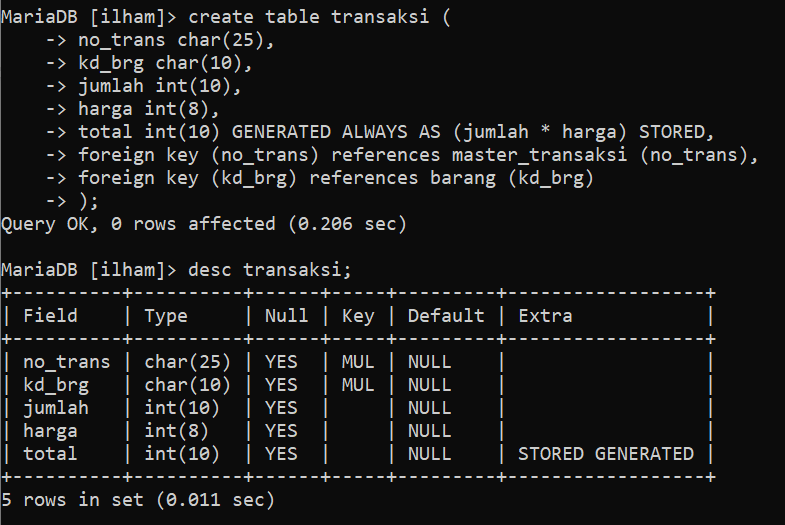
1. **BUATLAH TABEL MASTER TRANSAKSI**

****

*TABEL MASTER\_TRANSAKSI BERELASI DENGAN TABEL KASIR*

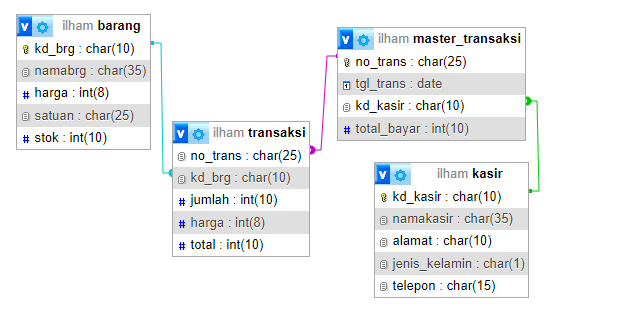
1. **BUATLAH TABEL TRANSAKSI**

**GENERATED ALWAYS AS (jumlah \* harga) STORED** adalah kolom yang nilainya otomatis dihitung dari jumlah x harga dan disimpan di tabel untuk kolom total di bawah ini.

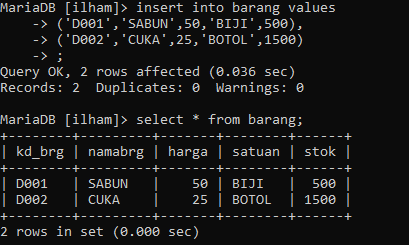
****

*TABEL TRANSAKSI BERELASI DENGAN TABEL BARANG DAN MASTER\_TRANSAKSI*

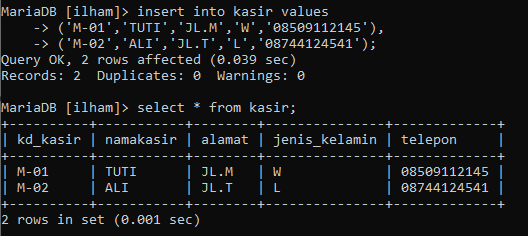
1. **TAMPILAN RELASINYA**

****

1. **INSERT DATA PADA TABEL BARANG**

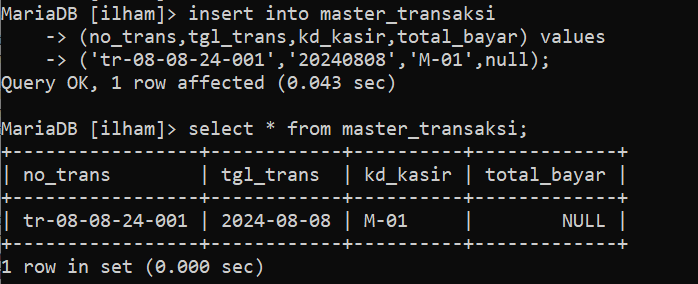
****

1. **INSERT DATA PADA TABEL KASIR**

****

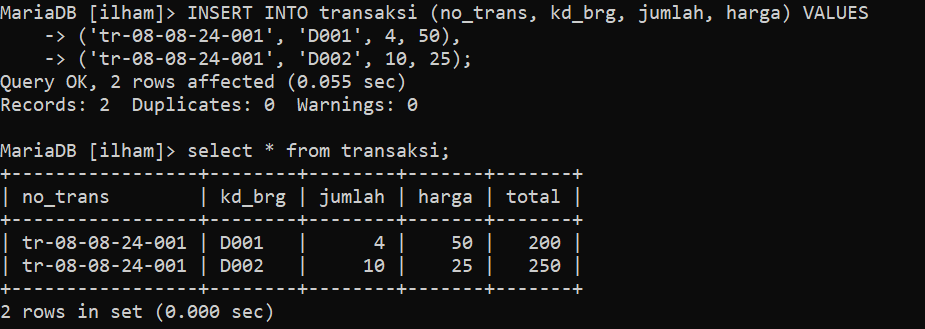
1. **INSERT DATA PADA TABEL MASTER\_TRANSAKSI**

Insert data akan tetapi total\_bayar tidak ada nilai, karena akan di tambah nilai menggunakan rumus, di mana nilai di dapat dari tabel transaksi kolom total transaksi.

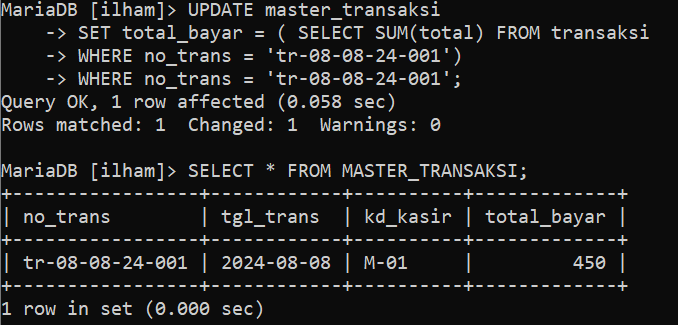
****

1. **INSERT DATA PADA TABEL TRANSAKSI**

Tidak perlu input nilai total, karena akan di isi otomatis dengan mengkalikan jumlah \* harga.

****

1. **MENGHITUNG/UPDATE TOTAL BAYAR TABEL MASTER\_TRANSAKSI DARI TABEL TRANSAKSI(TOTAL) DENGAN RUMUS SUM.**

****

Berikut penjelasan per bagian untuk pernyataan SQL diatas:

**1. UPDATE master\_transaksi**: memperbarui tabel master\_transaksi.

**2. SET total\_bayar = ( SELECT SUM(total) FROM transaksi WHERE no\_trans = 'tr-08-08-24-001')** : menetapkan nilai baru untuk kolom total\_bayar. menghitung jumlah (SUM) dari kolom total di tabel transaksi untuk baris-baris yang memiliki no\_trans sama dengan 'tr-08-08-24-001'. Nilai hasil dari subquery ini digunakan untuk memperbarui total\_bayar.

**4. WHERE no\_trans = 'tr-08-08-24-001'** : Memastikan bahwa hanya baris di tabel master\_transaksi dengan no\_trans yang sama dengan 'tr-08-08-24-001' yang diperbarui.

1. **MENAMPILKAN record kolom**

no\_trans -> master\_transaksi

namabrg -> barang

tgl\_trans -> transaksi

namakasir -> kasir

total\_bayar -> master\_transaksi

