# TUGAS TAMBAHAN

## **Nama: Putra Akbar**

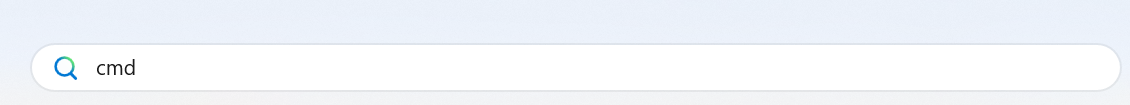
**Absen: 27**

# DASAR PENGGUNAAN

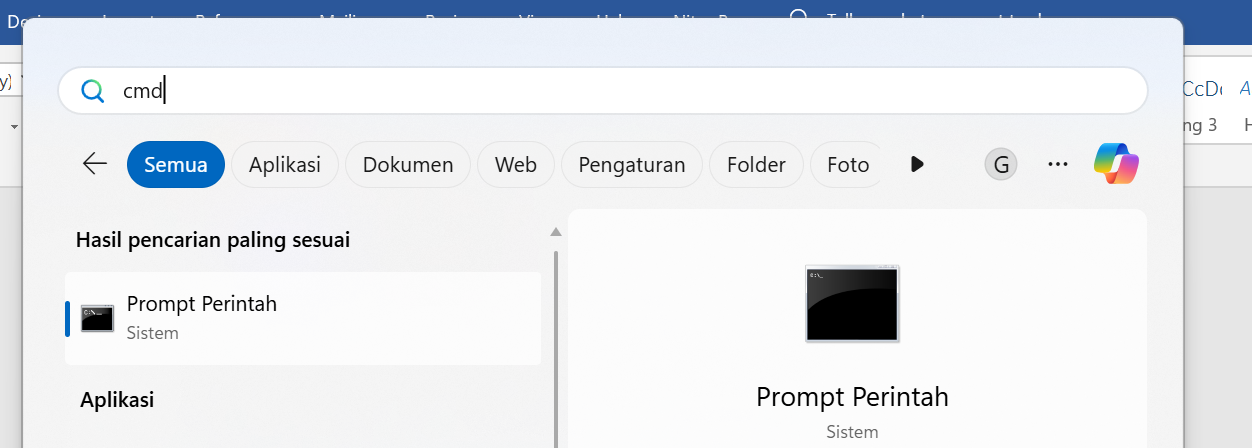
# Jalankan XAMPP Menggunakan XAMPP Control

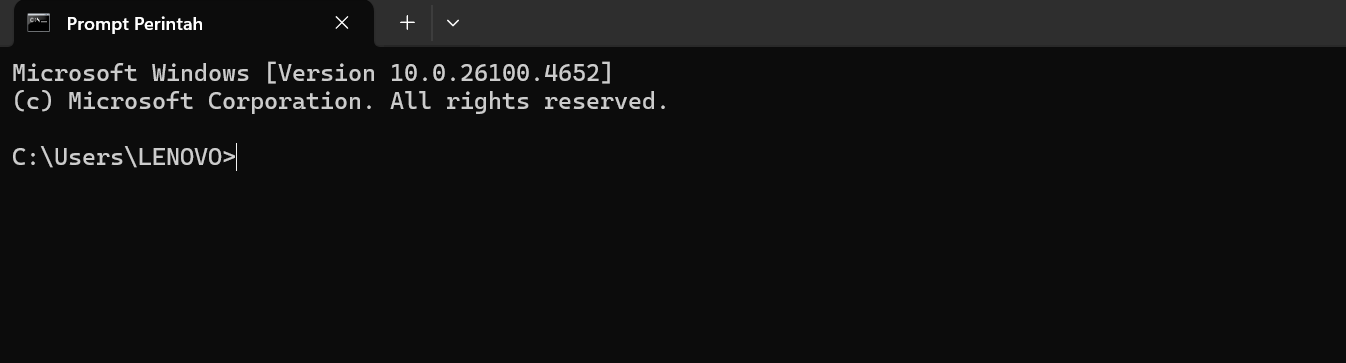


Setelah XAMPP berhasil dijalankan maka buka aplikasi Command Prompt di windows dengan cara ketik (cmd) di pencarian

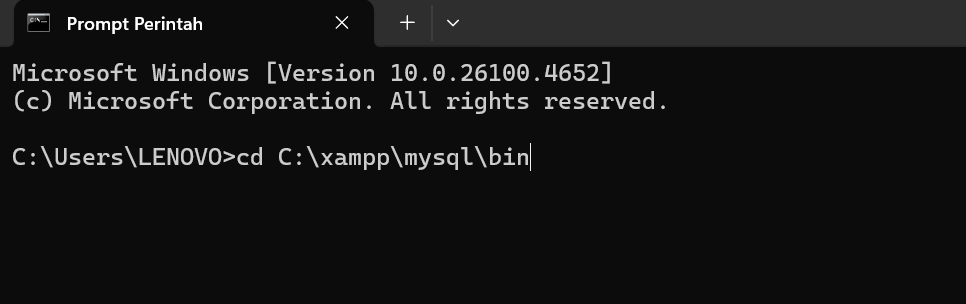


Pilih Prompt Perintah

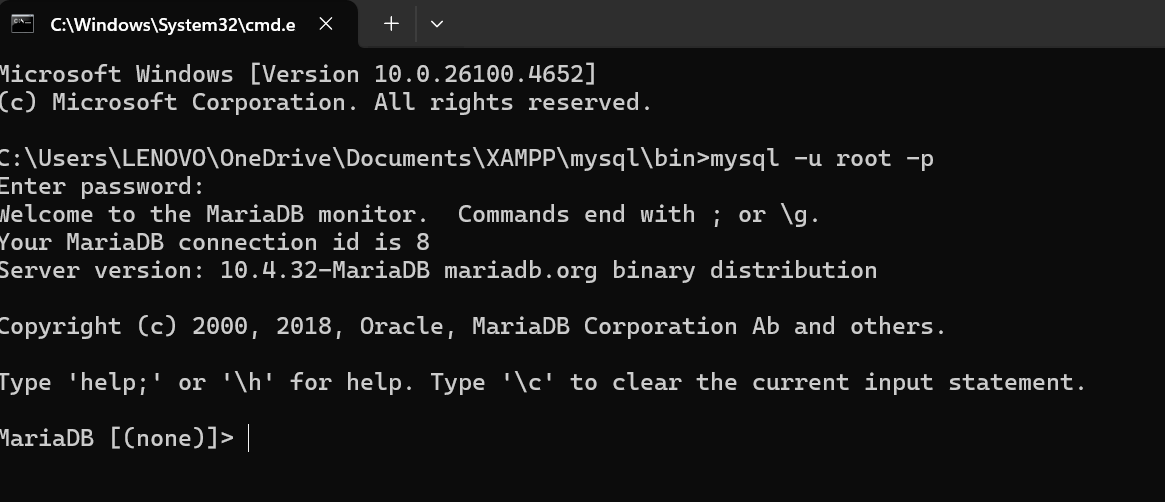




Setelah terbuka jalankan MySQL dengan mengetik

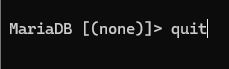


**Karena password default atau bawaan XAMPP nya kosong, langsung ketik enter saja. Jika MySQL berjalan maka akan tampil**



# KELUAR DARI MYSQL

Gunakan Perintah :

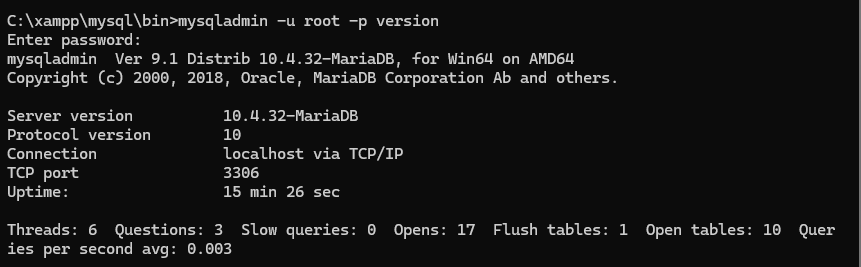


# MEMERIKSA MYSQL JALAN ATAU TIDAK

Untuk mengecek MySQL berjalan atau tidak, pastikan keluar dari MySQL dulu kemudian gunakan perintah dibawah. Tekan enter jika passwordnya kosong



# CEK VERSI MYSQL Kita bisa mengecek versi MySQL



# MENAMPILKAN HELP MYSQL

Anda harus masuk dulu ke MySQL dengan perintah



Tanda jika anda sudah ada dalam MySQL adalah



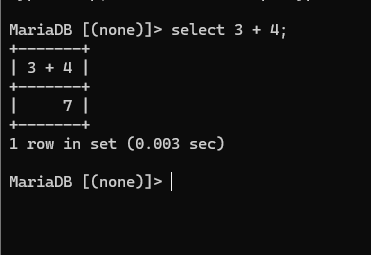
Untuk menampilkan HELP bisa menggunakan perintah

Atau gunakan 

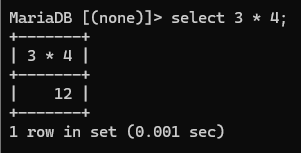
# MENGGUNAKAN KALKULATOR

Untuk menggunakan kalkulator di MySQL gunakan perintah sebagai berikut

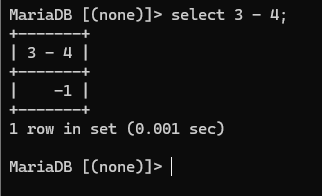
1. Penjumlahan



1. Perkalian



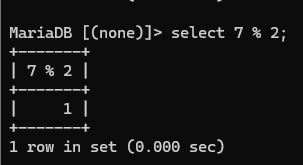
1. Pengurangan



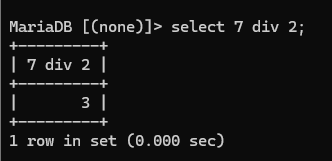
1. Pembagian



1. Modulus



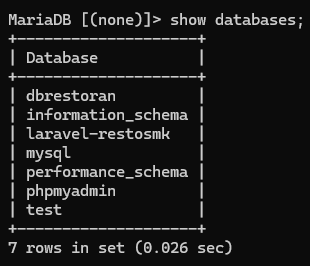
1. Hasil pembagian integer (utuh)



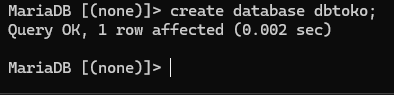
# Data Definition Language

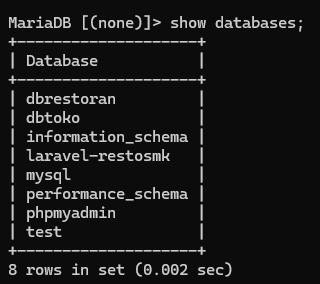
# Pembuatan DB, Tabel, & Index

# MENAMPILKAN DATABASE

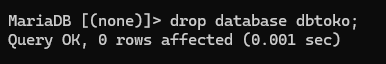


# MEMBUAT DATABASE

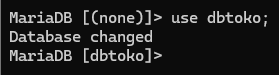




# MENGHAPUS DATABASE



# MENGAKTIFKAN DATABASE

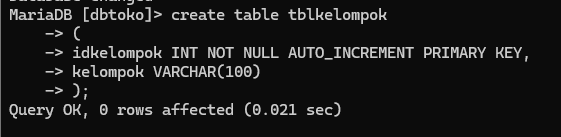


# MEMBUAT TABEL

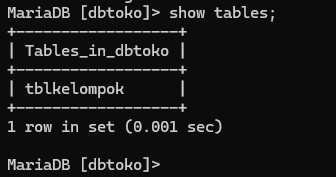
tabel yang pertama dibuat adalah tabel tblkelompok, dan cara membuat sebuah table adalah seperti berikut



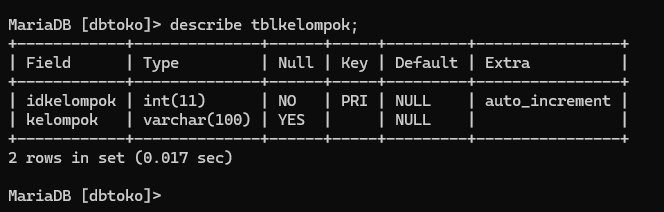
# MEMBUAT TABEL tblkelompok



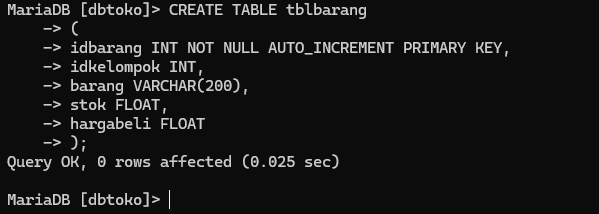
# MENAMPILKAN TABEL



# MENAMPILKAN STRUKTUR TABEL

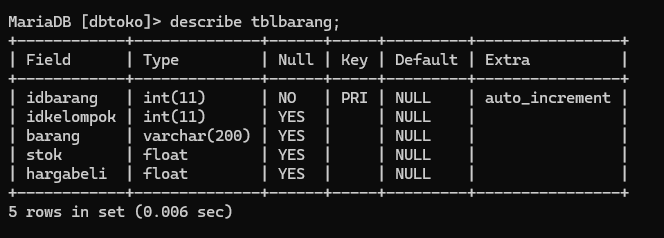


Untuk membuat tabel tblbarang gunakan perintah sebagai berikut

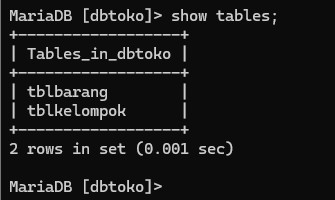


Karena pada MySQL tidak ada tipe data “ Currency “ maka kita bisa menggunakan tipe data “FLOAT”

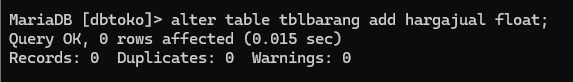
Periksa apakah struktur yang dibuat sudah benar

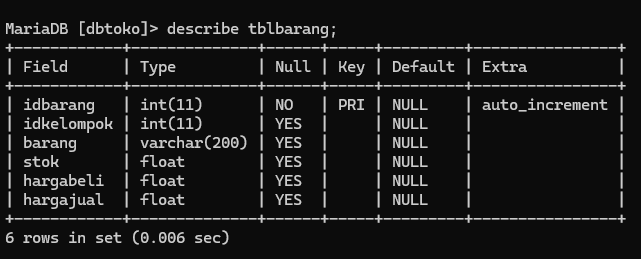


# Menampilkan Tabel

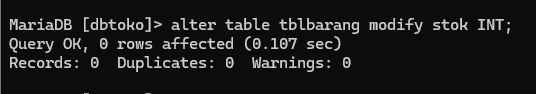


# MENAMBAH KOLOM

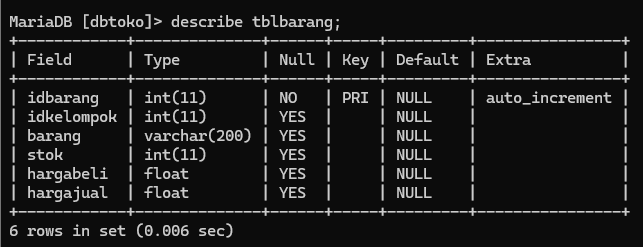




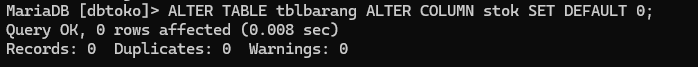
# MERUBAH TIPE DATA

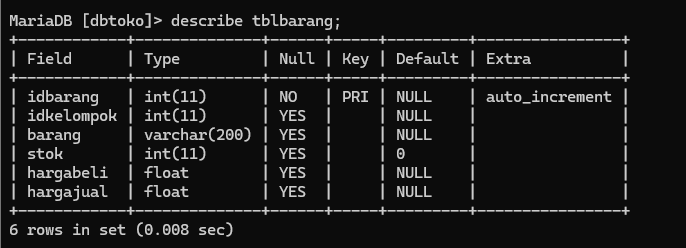


Periksa kembali tipe data yang digunakan menggunakan perintah berikut



# MEMBERI NILAI DEFAULT PADA KOLOM

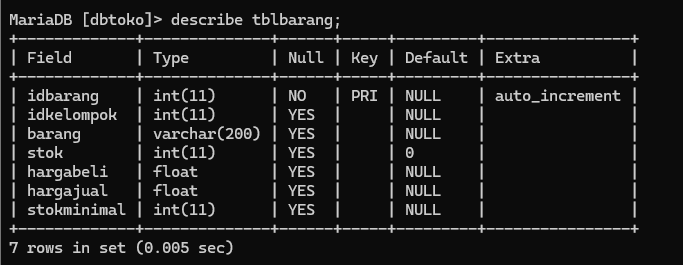




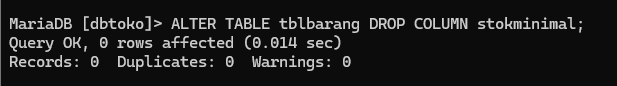
# MENGHAPUS KOLOM



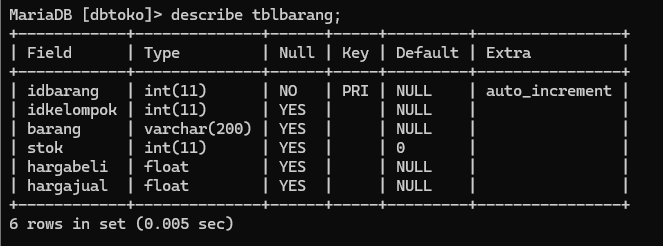
Periksa struktur tabel yang sudah dibuat



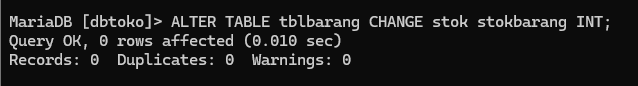
Pada struktur diatas kolom stokminimal akan dihapus. Untuk menghapus gunakan perintah berikut



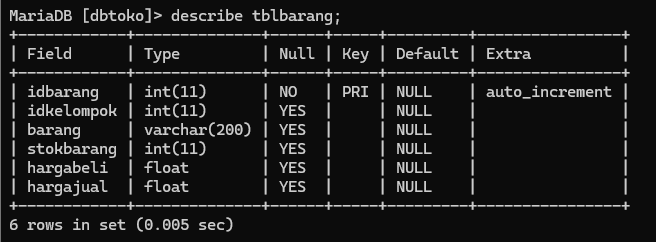
Periksa struktur tabel dengan perintah berikut



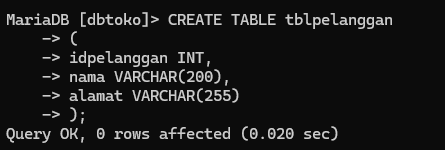
# MERUBAH NAMA KOLOM



Periksa struktur tabelnya

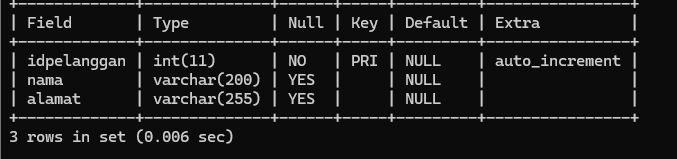


# MEMBUAT TABEL tblpelanggan



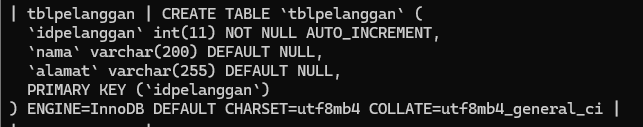
MENAMBAH PRIMARY KEY



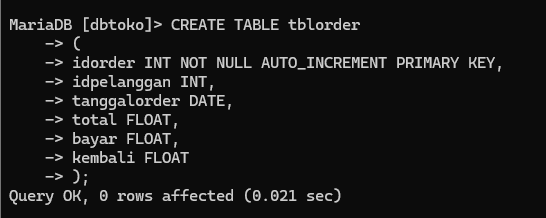


# MENAMPILKAN ENGINE YANG DIGUNAKAN

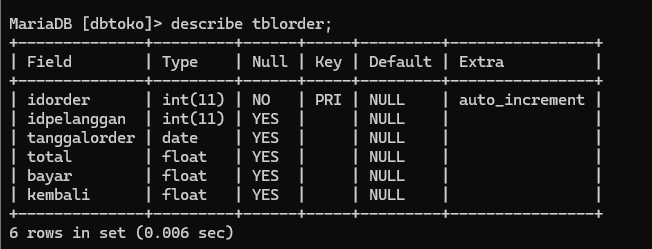




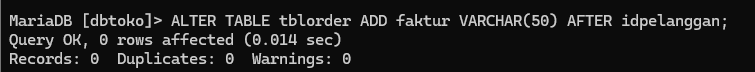
# MEMBUAT TABEL tblorder

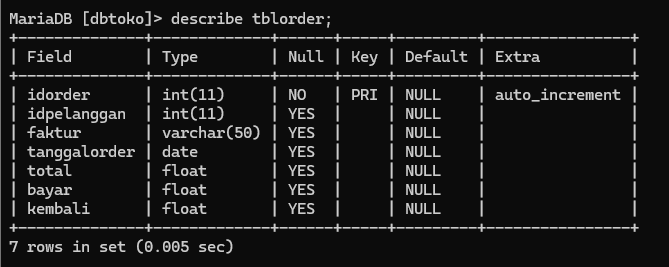


Struktur tabel

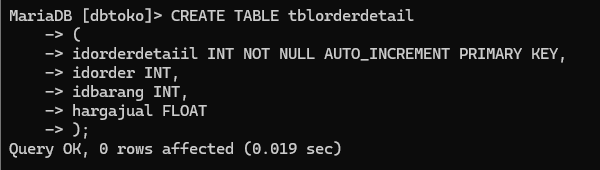


# MENAMBAH KOLOM SETELAH KOLOM

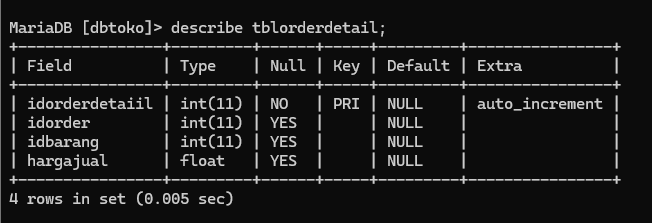




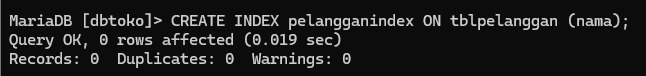
# MEMBUAT TABEL tblorderdetail



Struktur tabel yang sudah dibuat

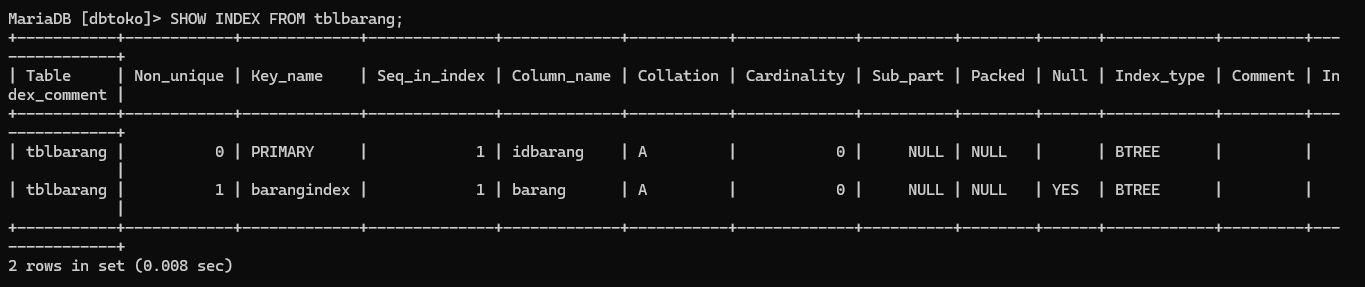


# MEMBUAT INDEX





# MENAMPILKAN INDEX



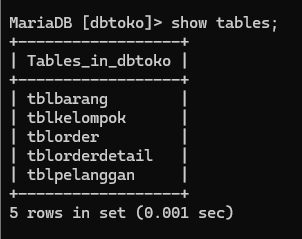
# MENGHAPUS INDEX

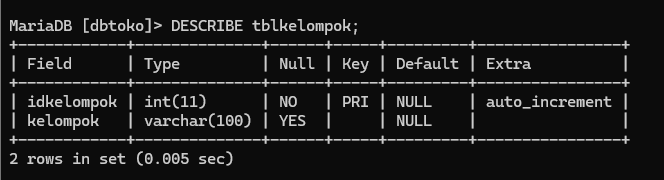


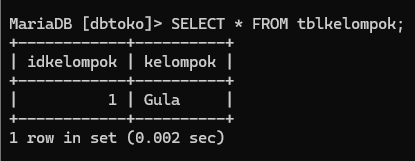
# Data Manipulation Language

Insert, Delete, Update, Select, View, Trigger, Procedure, & Function

# INSERT SEMUA KOLOM

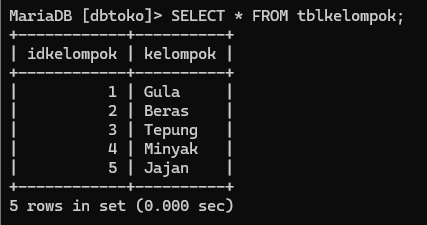






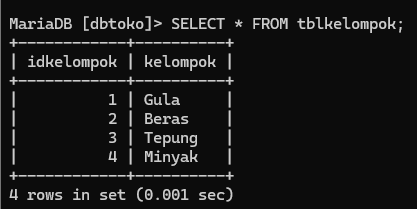
# INSERT SEBAGIAN KOLOM





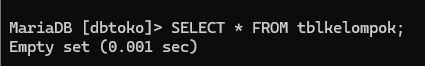
# DELETE SEBAGIAN *RECORD* (BARIS DATA)



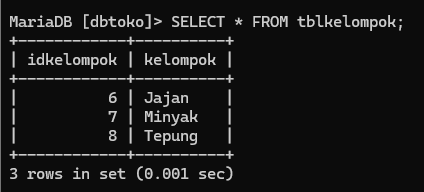


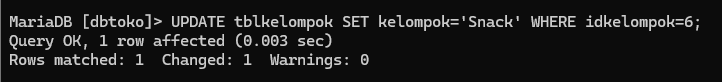
# DELETE SEMUA RECORD

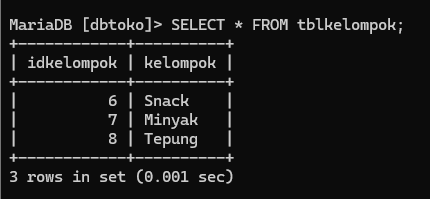




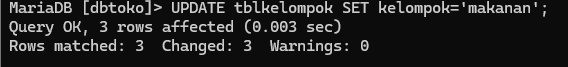
# UPDATE SEBAGIAN RECORD



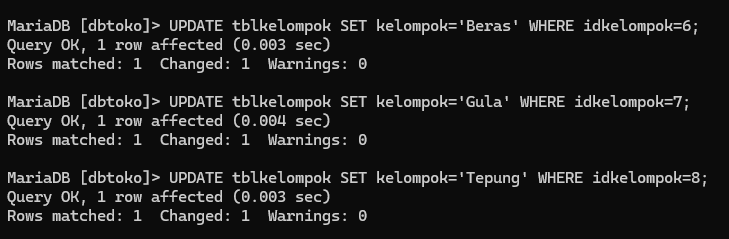




# UPDATE SEMUA RECORD





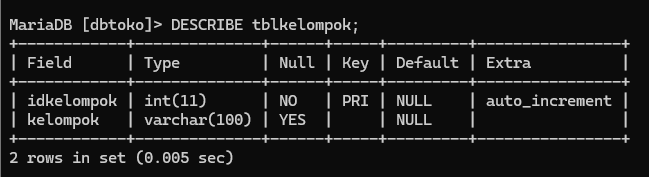
**



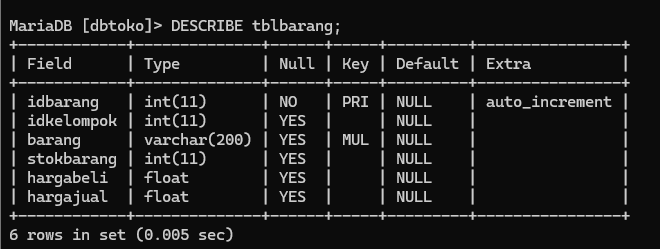
# TABEL MASTER DAN TABEL TRANSAKSI (DETAIL)

Lihat struktur tblkelompok dan tblbarang

tblkelompok

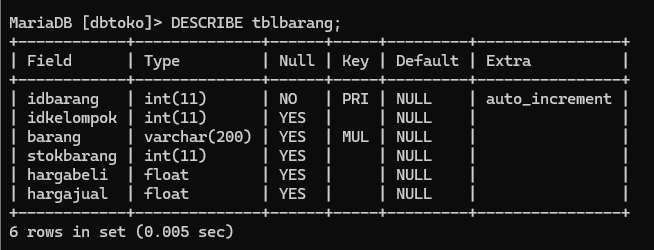


tblbarang

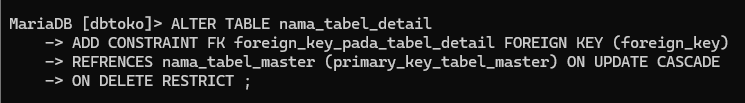


# MEMBUAT RELASI ANTAR TABEL

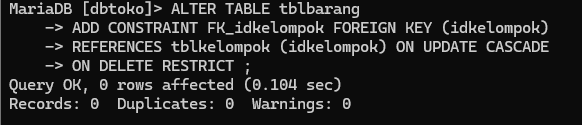
Lihat dulu struktur tabel yang akan dibat relasinya. Pada tblbarang terlihat struktur seperti pada gambar



Perintah untuk membuat relasi adalah sebagai berikut



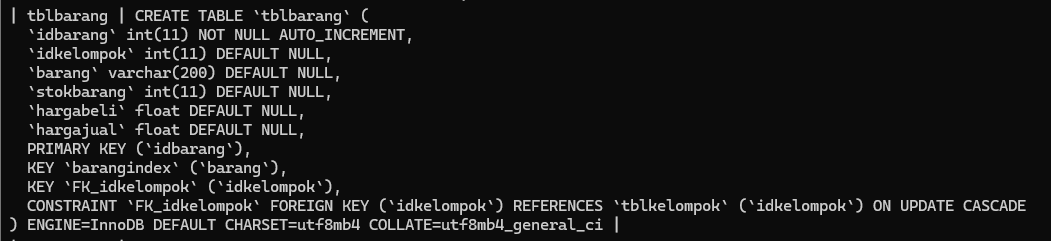
Kita akan coba pada tblbarang



Untuk melihat hasil pembuatan relasi gunakan perintah berikut



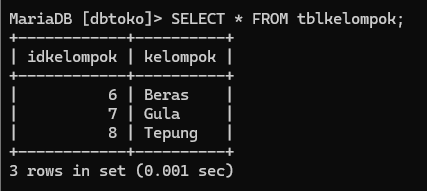
Kita akan lihat tblbarang



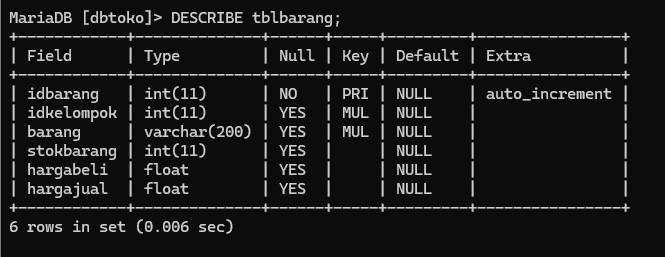
Dan diatas adalah hasilnya

# MENGUJI HASIL PEMBUATAN RELASI

Untuk menguji hasil pembuatan relasi lakukan INSERT data pada tabel master terlebih dahulu baru kemudian ke tabel transaksi.



Tampilkan struktur tabel untuk memudahkan proses insert data

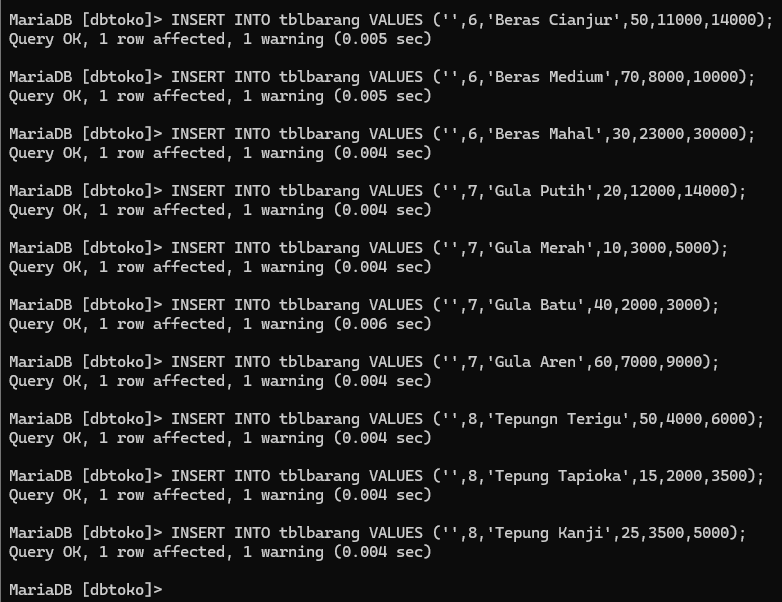


Isi tabel transaksi sesuai dengan idkelompok yang akan digunakan

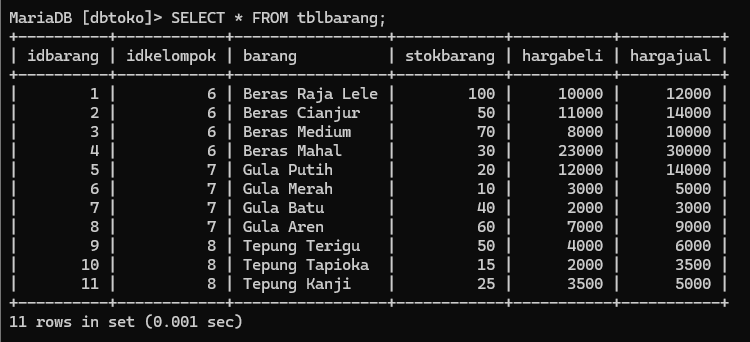
Karena idbarang AUTO\_INCREMENT maka kolom bisa diganti dengan ‘’ (petik kosong)



Kita lakukan insert beberapa jenis beras, gula, dan tepung

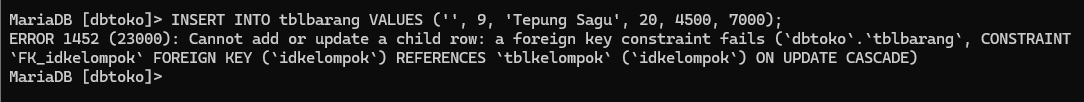


Dan kita akan lihat hasilnya



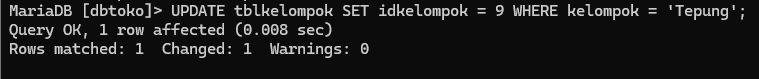
# PENGUJIAN INSERT

Insert pada tabel [tblbarang] HANYA BISA dilakukan menggunakan data yang tersedia pada tabel master. [Idkelompok] pada tabel master yang tersedia hanya [6,7,8].. Perhatikan hasil pengujian berikut; Pada pengujian INSERT idkelompok yang digunakan adalah 9 yang TIDAK TERSEDIA pada tabel master [tblkelompok].

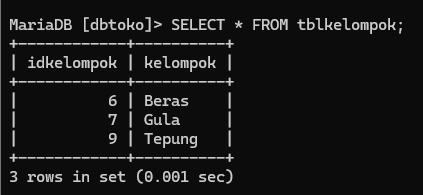


# PENGUJIAN UPDATE

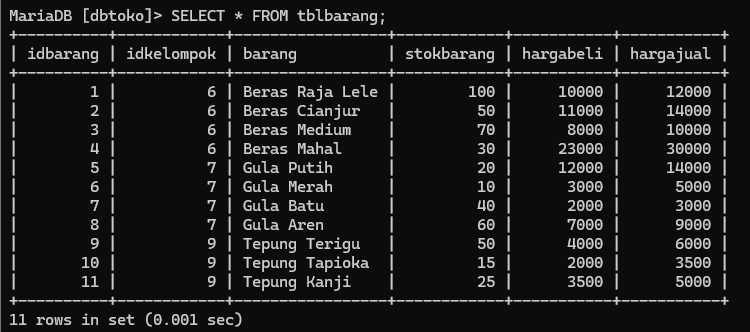
Coba lakukan update pada tabel master kemudian periksa di tabel transaksi. Ubah idkelompok tepung dari 8 menjadi 9 dengan perintah berikut



Periksa perubahan pada tblkelompok



Periksa pada tblbarang



# PENGUJIAN DELETE

Pengujian delete dilakukan dengan menghapus tabel master.



Karena [ idkelompok] 9 sudah digunakan pada tabel transaksi maka data tersebut tidak bisa dihapus seuai dengan relasi yang dibuat yaitu ON DELETE RESTRICT. Data yang bisa dihapus hanya yang belum digunakan pada tabel transaksi. coba INSERT kan data baru pada tabel master

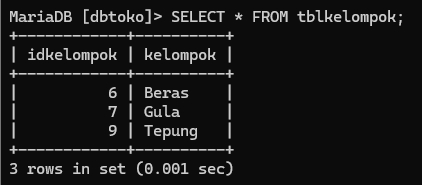


Periksa data yang sudah dimasukan



Karena kolom [idkelompok] 10 belum digunakan pada tabel transaksi maka kolom tersebut bisa dihapus. Lakukan penghapusan menggunakan perintah berikut

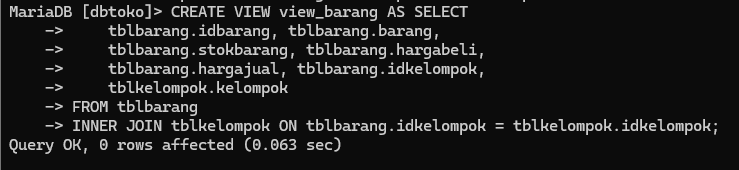


Periksa tabel yang sudah dihapus

# PEMBUATAN VIEW

View digunakan untuk menggabungkan tabel master dan tabel transaksi menjadi sebuah tabel baru agar lebih mudah dilihat dan digunakan.

Kita akan langsung mencobanya

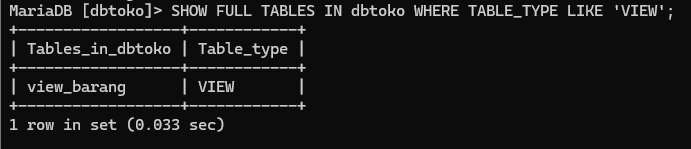


# MENAMPILKAN SEMUA VIEW YANG SUDAH DIBUAT

View yang sudah dibuat bisa dilihat dengan perintah



Kita akan lihat view dari dbtoko

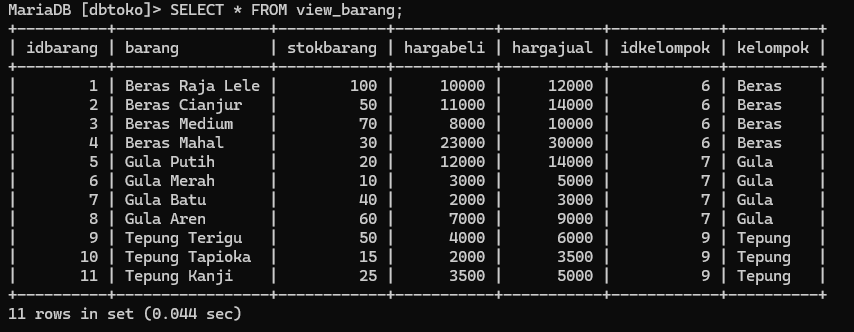


# MELIHAT ISI VIEW Untuk melihat isi VIEW

sama dengan melihat isi table



Kita akan lihat view\_barang



# MENGHAPUS VIEW Jika VIEW

yang sudah dibuat tidak digunakan lagi bisa di hapus dengan menggunakan perintah berikut



Kita akan coba untuk DROP view\_barang

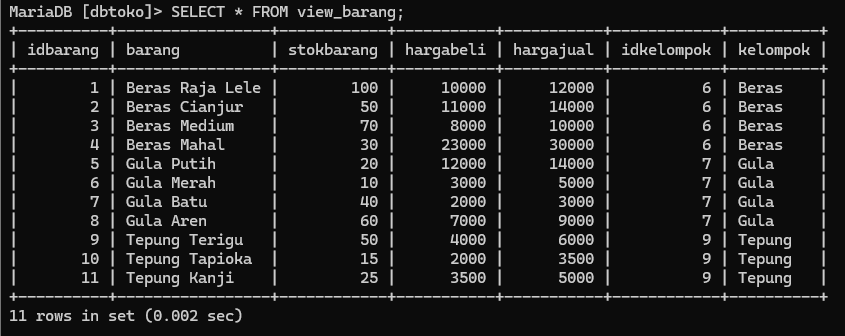


# SELECT SEMUA KOLOM (\*)

Select semua adalah menampilkan semua kolom dari tabel atau VIEW



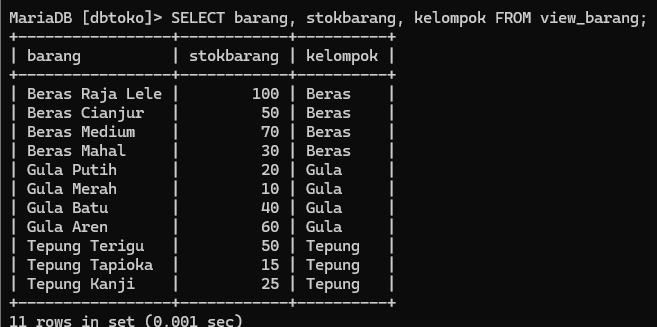
Kita akan buat view lagi dan kita lihat lagi view\_barang



# SELECT SEBAGIAN KOLOM

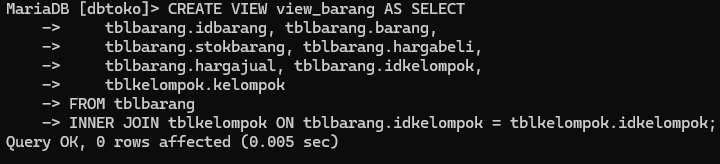
Select sebagian kolom adalah menampilkan hanya pada kolom yang dipilih





# SELECT ORDER Jika VIEW

pada latihan diatas sudah dihapus, silahkan buat ulang untuk latihan materi selanjutnya

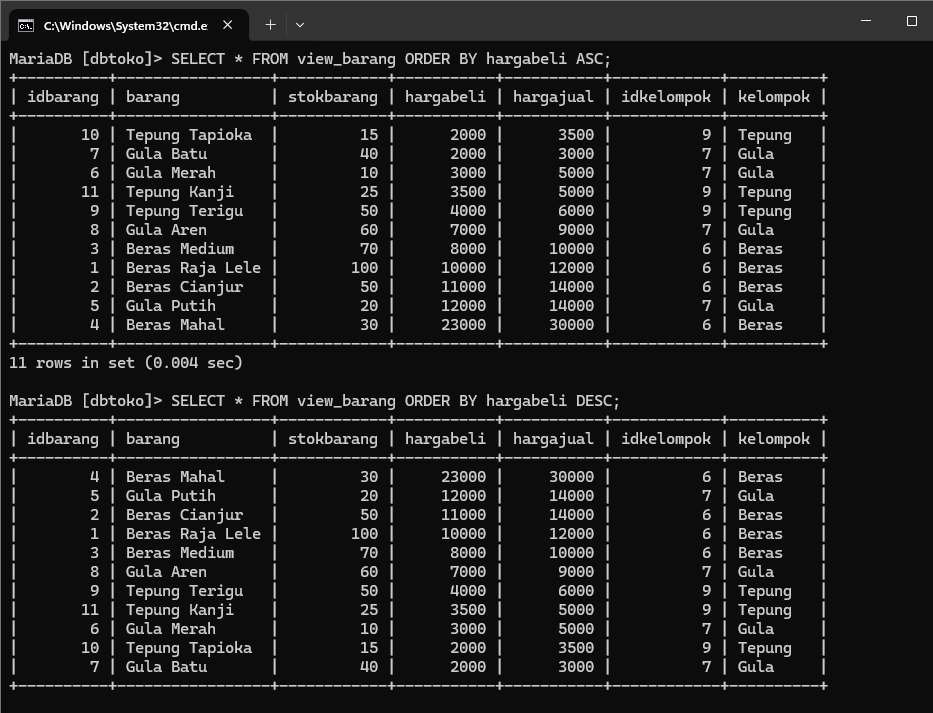


Kita sudah buat, dan kita akan melanjutkan materinya

SELECT ORDER adalah cara untuk menampilkan data dalam urutan naik atau turun Jika NAIK menggunakan ASC Jika TURUN menggunakan DESC



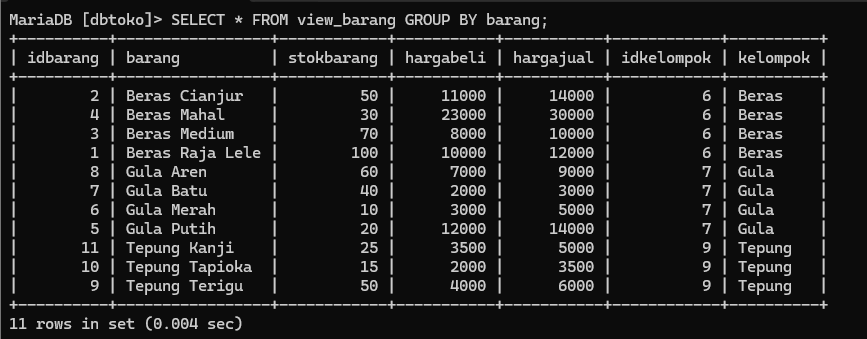
Hasil select ORDER BY nama\_kolom ASC dan DESC



# SELECT GROUP

Perintah SELECT GROUP digunakan untuk mengelompokan data sesuai dengan kelompok dari kolom yang dipilih.

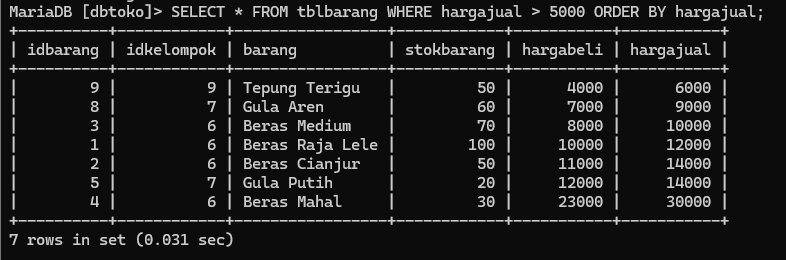


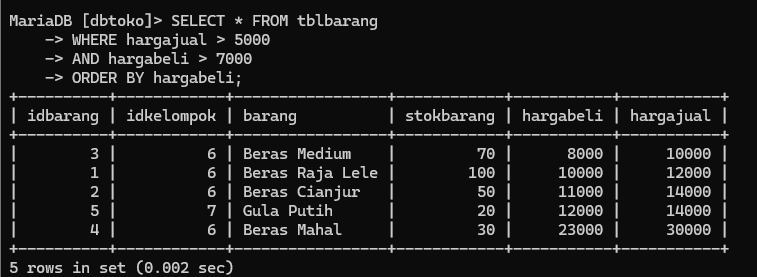


# PENGUJIAN WHERE

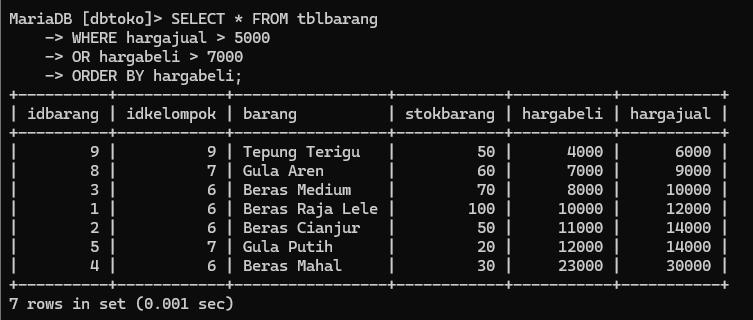


Kita akan mencoba pengujiannya menggunakakn tblbarang





Hasil di atas menampilkan data dari tabel **tblbarang** yang disaring menggunakan kondisi **hargajual > 5000** dan **hargabeli > 7000**, kemudian diurutkan berdasarkan **hargabeli**.  
Dari hasil tersebut, terdapat 5 barang yang memenuhi kriteria, yaitu berbagai jenis beras dan gula dengan harga beli antara **8000 hingga 23000**, serta harga jual antara **10000 hingga 30000**.



hasil di atas menampilkan data dari **tblbarang** dengan kondisi **hargajual > 5000** **atau** **hargabeli > 7000**, lalu diurutkan berdasarkan **hargabeli**.  
Dari hasilnya terdapat **7 barang**, mulai dari **Tepung Terigu** hingga **Beras Mahal**, dengan rentang **harga beli 4000–23000** dan **harga jual 6000–30000**



%a artinya awalan bebas yang penting akhiranya a

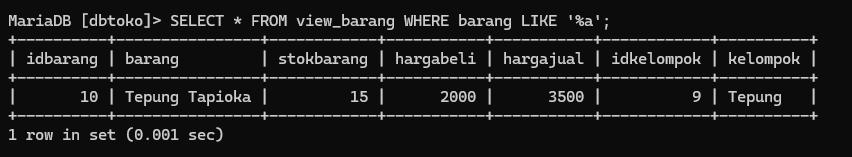


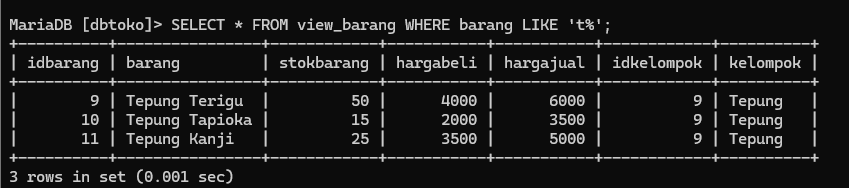
a% artinya inisial a akhiran bebas

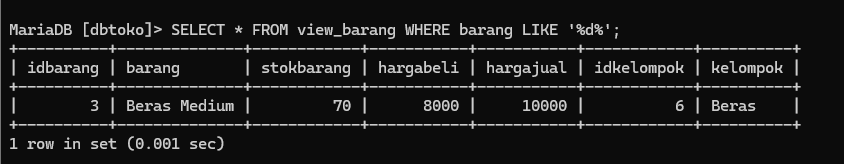


%a% artinya awalan sembarang dan akhiran sembarang yang tengahnya ada huruf a

contoh :





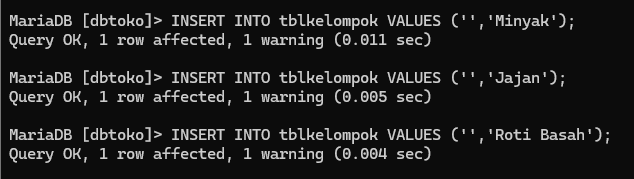


Dan jika tidak ada / kosong, maka



# SUBQUERY (SELECT IN SELECT)

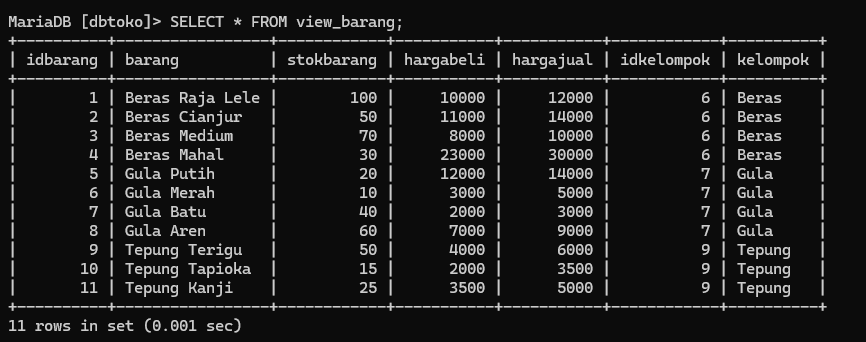
Subquery adalah SELECT yang ada di dalam SELECT. Sebelum berlatih menggunakan subquery, kita akan tambahkan data pada [tblkelompok].



sehingga datanya akan tampil seperti berikut



# Periksa [view\_barang]



Pada [view\_barang] ada kelompok yang sudah menggunakan data dari tabel [tblkelompok]. Ada pertanyaan yang muncul dari data diatas adalah:

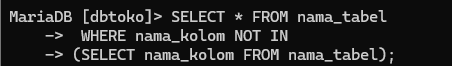
1. Ada berapa data dari tabel [tblkelompok] yang SUDAH digunakan pada VIEW?
2. Ada berapa data yang BELUM digunakan pada VIEW?

Untuk menjawab pertanyaan diatas bisa menggunakan SUBQUERY.

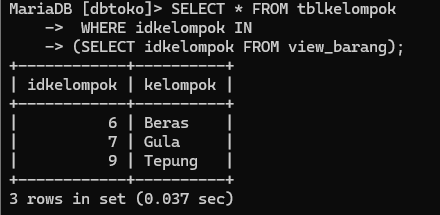
Gunakan perintah



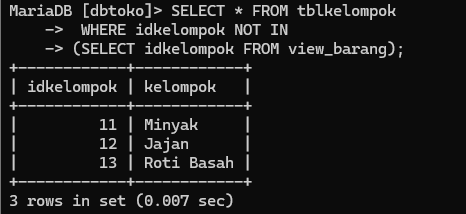
**IN** digunakan untuk **mengambil data** dari tabel utama yang **nilainya cocok** dengan hasil dari subquery.



**NOT IN** digunakan untuk **mengambil data** dari tabel utama yang **tidak ada** dalam hasil subquery.

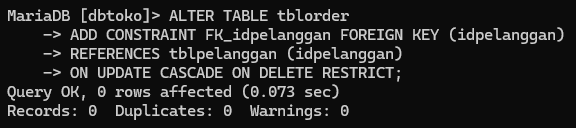


Menampilkan data **yang ada di view\_barang**, yaitu:  
**Beras (6)**, **Gula (7)**, dan **Tepung (9)**.



Menampilkan data **kelompok yang tidak ada di view\_barang**, yaitu:  
**Minyak (11)**, **Jajan (12)**, dan **Roti Basah (13)**.

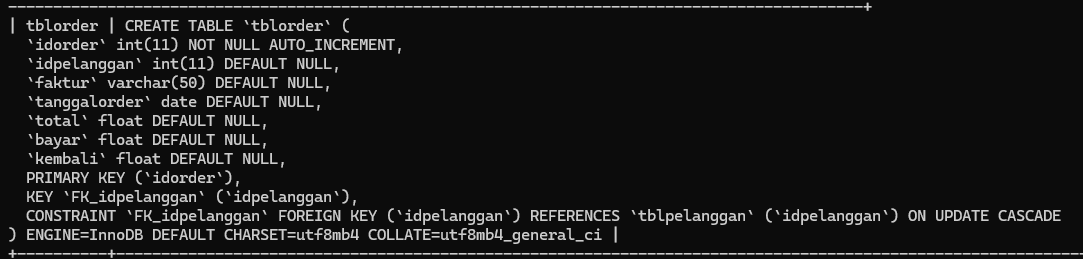
# MEMBUAT RELASI ANTAR TABEL



Periksa hasil pembuatan RELASI dengan perintah berikut

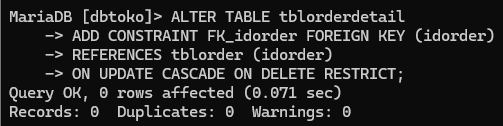


Hasilnya adalah

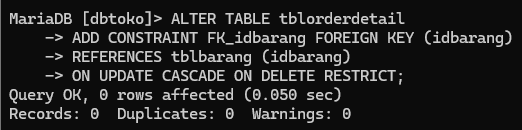


# RELASI 2 TABEL MASTER DAN 1 TABEL TRANSAKSI

Buat relasi antara tabel [tblorderdetail] dan tabel [tblorder]



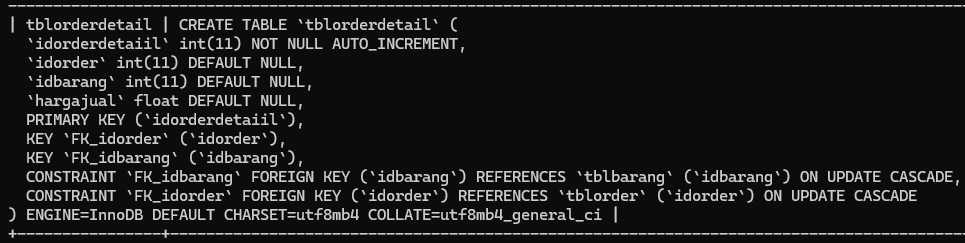
Buat relasi antara tabel [tblorderdetail] dan tabel [tblbarang]



Periksa hasil pembuatan relasi antara 2 tabel master [tblorder] [tblbarang] dan 1 tabel transaksi [tblorderdetail]



Dan ini hasilnya

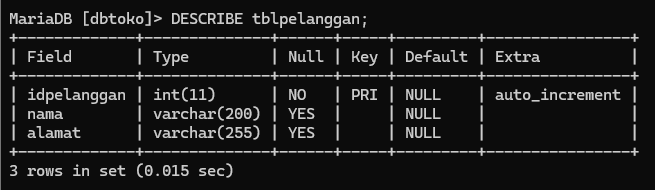


# DUMMY DATA

Dummy data adalah data yang tidak sebenarnya. Data ini diperlukan jika data sebenarnya tidak bisa diperoleh. Dummy data biasa digunakan selama pengujian database.

Pada tabel [tblpelanggan] aplikasi akan di isi dengan dummy data dengan nama pelanggan sebegai berikut

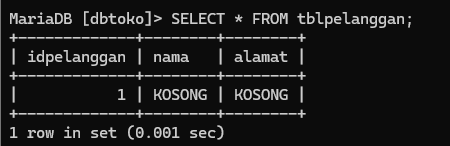
Tampilkan struktur tabel terlebih dahulu



Insert nama pelanggan dengan dummy data



Periksa hasil INSERT data



Pada tabel terdapat nama pelanggan dengan nama [KOSONG], jika terjadi pembelian dengan nama pelanggan yang tidak disebutkan maka aplikasi akan memberi nama pelanggan tersebut dengan nama [KOSONG]. Nama pelanggan dengan nama [KOSONG] ini disebut dengan dummy data.

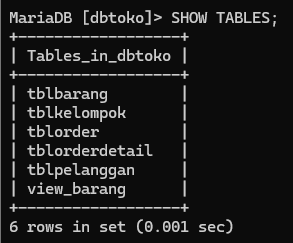
# KONSEP TRIGGER

Trigger adalah perintah INSERT, UPDATE, DELETE, FUNCTION, PROCEDURE yang ditanam pada MySQL yang akan DIJALANKAN pada kejadian berikut:

* AFTER INSERT (setelah INSERT) pada tabel yang dimaksud
* BEFORE INSERT (sebelum INSERT) pada tabel yang dimaksud
* AFTER DELETE (setelah DELETE) pada tabel yang dimaksud
* BEFORE DELETE (Sebelum DELETE) pada tabel yang dimaksud
* AFTER UPDATE (Setelah UPDATE) pada tabel yang dimaksud
* BEFORE UPDATE (Sebelum UPDATE) pada tabel yang dimaksud

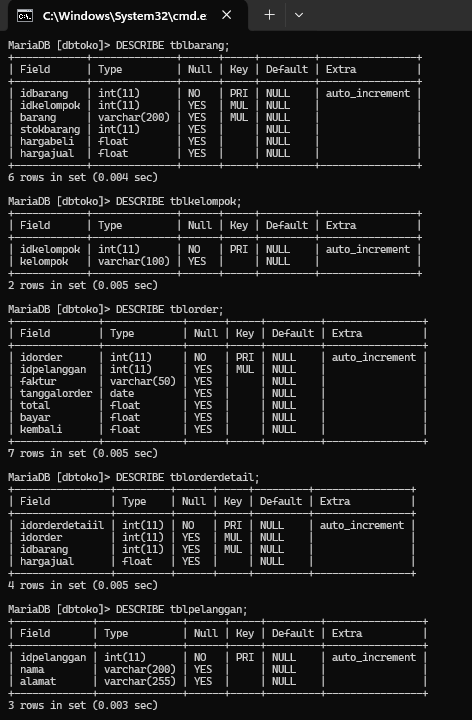
Sebelum membuat trigger pastikan anda memahami proses bisnis dari pembuatan database yang anda lakukan. Sebagai latihan buka semua database yang sudah dibuat pada materi sebelumnya, ikuti langkah berikut

1. TAMPILKAN SEMUA TABEL



# LIHAT STRUKTUR SEMUA TABEL

* [tblbarang]
* [tblkelompok]
* [tblorder]
* [tblorderdetail]
* [tblpelanggan]

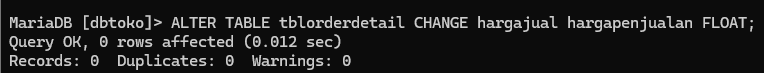


# PENYUSUNAN PROSES BISNIS

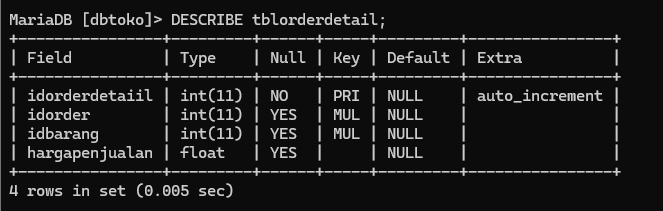
* kolom [stokbarang] pada tabel [tblbarang] akan BERKURANG jika terjadi INSERT data pada tabel [tblorderdetail]
* kolom [stokbarang] pada tabel [tblbarang] akan BERTAMBAH jika terjadi DELETE data pada tabel [tblorderdetail]
* kolom [total] pada tabel [tblorder] akan BERTAMBAH jika terjadi INSERT data pada tabel [tblorderdetail]
* kolom [total] pada tabel [tblorder] akan BERKURANG jika terjadi DELETE data pada tabel [tblorderdetail]

# PEMERIKSAAN NAMA KOLOM YANG SAMA

Sebelum membuat TRIGGER pastikan tidak ada nama kolom yang sama pada setiap tabel. Jika melihat tabel diatas terdapat nama kolom yang sama. Yaitu [hargajual] pada tabel [tblbarang] dan [hargajual] pada tabel [tblorderdetail]. Ubah nama kolom [hargajual] di tabel [tblorderdetail] menjadi [hargapenjualan]



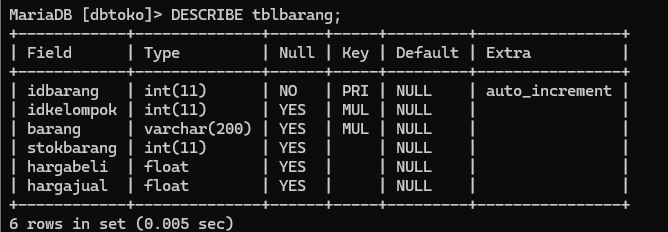
Hasil :



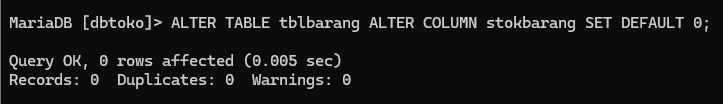
# PEMERIKSAAN KOLOM UNTUK PERUBAHAN HASIL TRIGGER

Lakukan pemeriksaan pada pada kolom yang akan terkena dampak dari proses trigger. Jika yang diproses oleh trigger adalah operasi matematika (penjumlahan, pengurangan, pembagian, perkalian). Pastikan SET DEFAULT VALUE pada kolom tabel tersebut di isi dengan ANGKA NOL. Jika belum di isi angka NOL lakukan ALTER untuk merubah kolom tersebut. Lihat contoh berikut

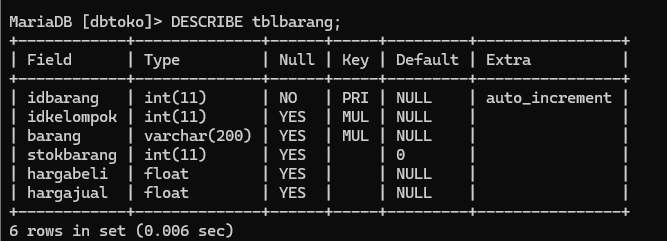
Kolom [stokbarang] belum di SET DEFAULT dengan ANGKA NOL



Command ALTER pada tblbarang

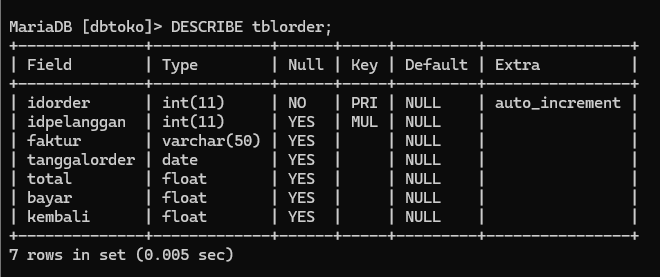


Hasil :

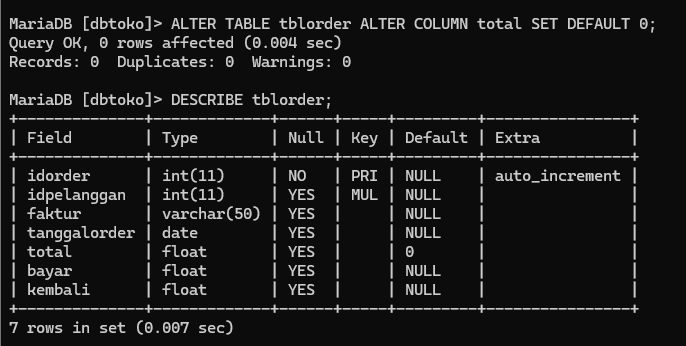


Default pada stokbarang menjadi 0

Pemeriksaan kolom [total] pada tabel [tblorder]. Belum di set DEFAULT ANGKA NOL



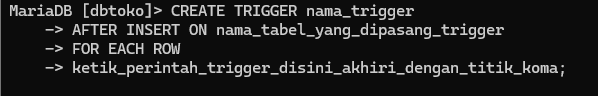
Lakukan perubahan dengan ALTER



# PEMBUATAN TRIGGER

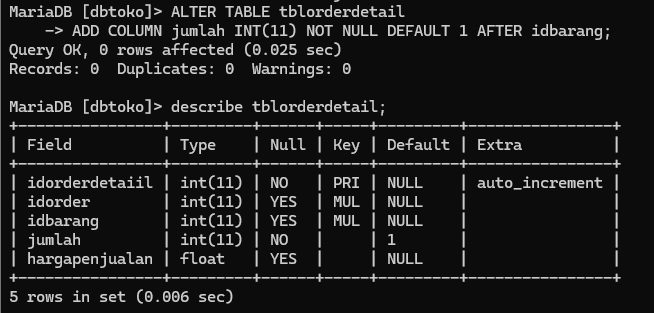
Setelah semua konsep, proses bisnis, dan pemeriksaan tabel dilakukan sekarang waktunya membuat TRIGGER. Ada 4 trigger yang akan dibuat.

|  |  |  |
| --- | --- | --- |
| **Nama Trigger** | **Kejadian** | **Fungsi** |
| kurang\_stok | BEFORE INSERT ON | Mengurangi [stokbarang] pada tabel [tblbarang] SEBELUM INSERT pada tabel [tblorderdetail] |
| tambah\_total | AFTER INSERT ON | Menambah [total] pada tabel [tblorder] SESUDAH INSERT pada tabel [tblorderdetail] |
| tambah\_stok | BEFORE DELETE ON | Manambah [stokbarang] pada tabel [tblbarang] SEBELUM DELETE pada tabel [tblorderdetail] |
| kurang\_total | AFTER DELETE ON | Mengurangi [total] pada tabel [tblorder] SESUDAH DELETE pada tabel [tblorderdetail] |

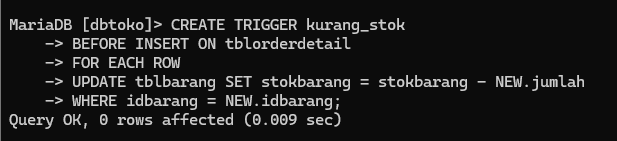


# kurang\_stok

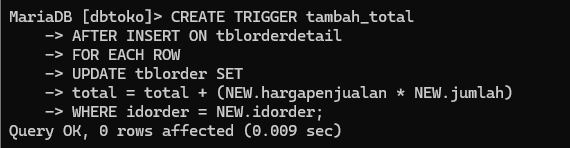
Buat column [jumlah] di dalam tblorderdetail



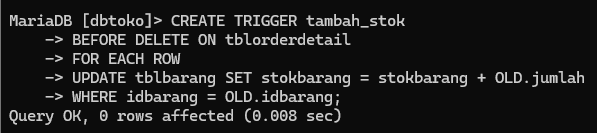
Sekarang kita buat trigger nya



# tambah\_total



# tambah\_stok



# kurang\_total

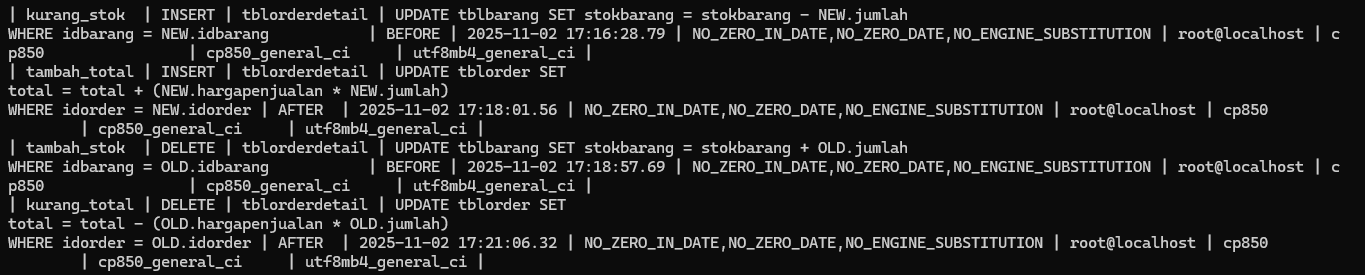


# MENAMPILKAN TRIGGER

Trigger yang sudah dibuat bisa ditempilkan dengan cara



Dan ini hasilnya

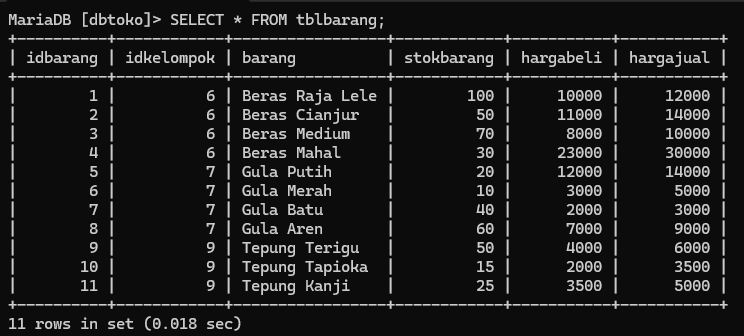


# PENGUJIAN TRIGGER

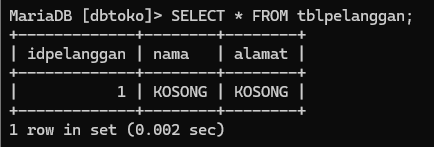
Setelah trigger dibuat maka langkah selanjutnya adalah menguji trigger

1. **PERSIAPAN**

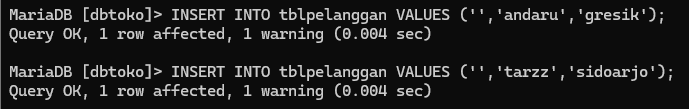
Tampilkan tabel [tblbarang]



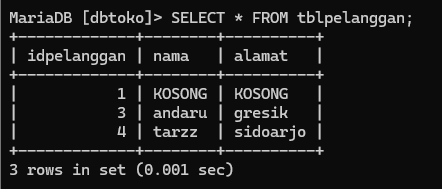
Tampilkan tabel [tblpelanggan]



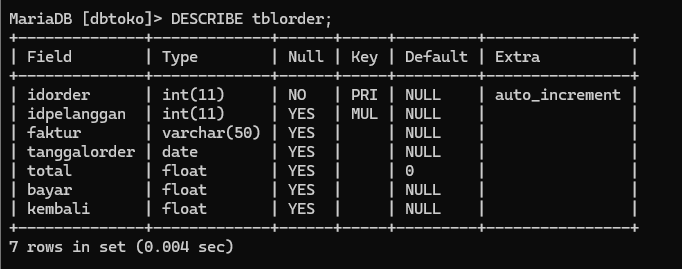
Karena data pelanggan hanya satu, tambahkan 2 pelanggan lagi untuk pengujian



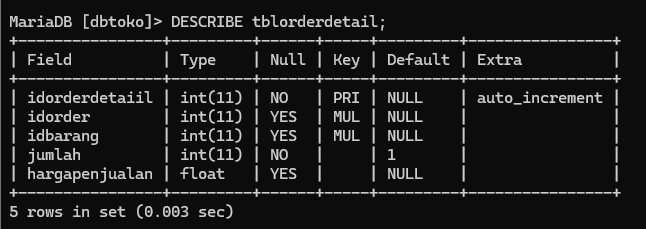
Tampilkan kembali data pelanggan



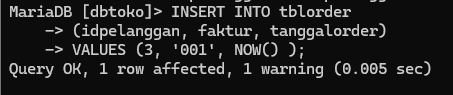
Tampilkan struktur tabel [tblorder]



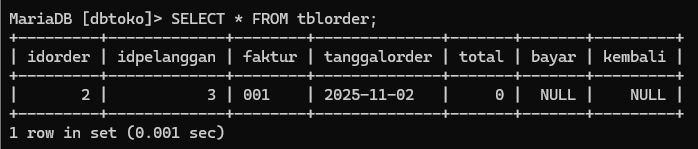
Tampilkan struktur tabel [tblorderdetail]



# PENGUJIAN INSERT PADA TABEL [tblorderdetail]

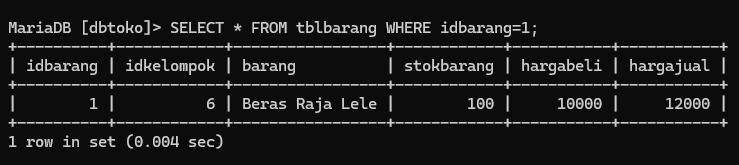


Periksa dengan perintah sebagai berikut

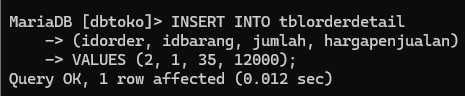


Buat INSERT data pada tabel [tblorderdetail] dengan menggunakan [idorder = 1] sesuai yang ada di tabel [tblorder] dengan [idbarang = 1] pada tabel [tblbarang].

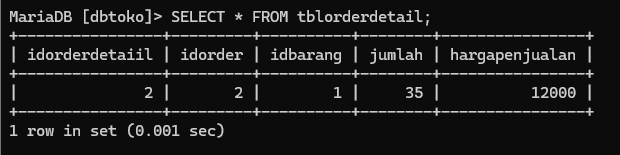
Tabel SEBELUM PROSES INSERT pada tabel [tblorderdetail]



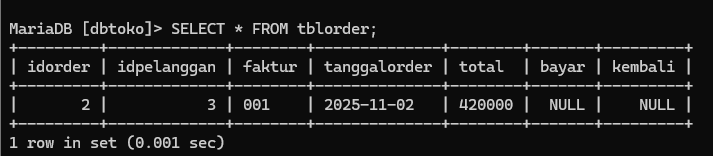
INSERT DATA pada tabel [tblorderdetail]



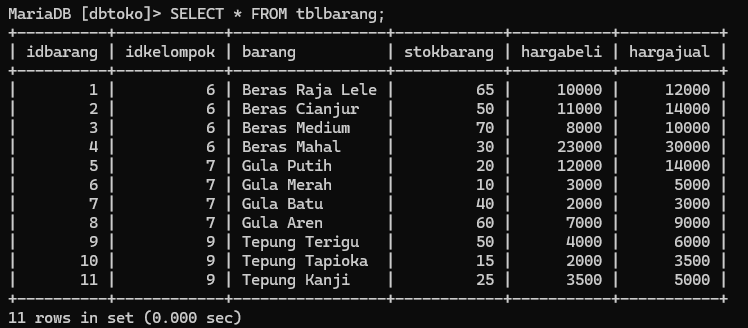
Hasil pada tabel [tblorderdetail]



Hasil pada tabel [tblorder]



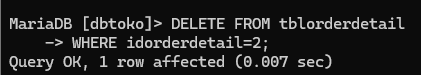
Hasil pada tabel [tblbarang]

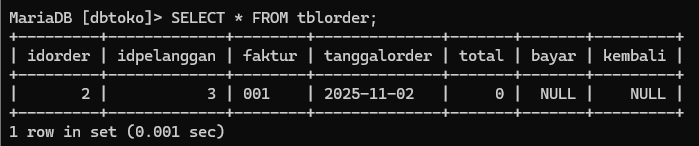


Stok beras raja lele atau idbarang=1 berkurang menjadi 65

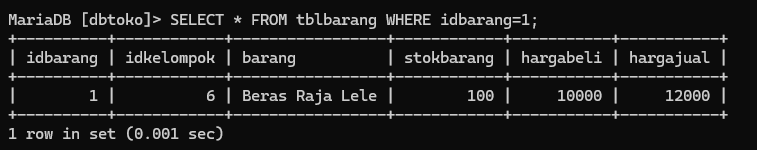
# PENGUJIAN DELETE PADA TABEL

[tblorderdetail] Lakukan DELETE pada tabel [tblorderdetail] dengan perintah sebagai berikut



Periksa hasil pada tabel [tblorder], setelah dihapus data pada tabel [tblorderdetail] kolom [total] akan kembali kenilai awal

Periksa pada tabel [tblbarang], setelah dihapus data pada tabel [tblorderdetail] kolom [stokbarang] akan kembali kenilai awal



# HAPUS TRIGGER

Jika trigger ada yang salah atau tidak digunakan lagi bisa dihapus dengan cara sebagai berikut

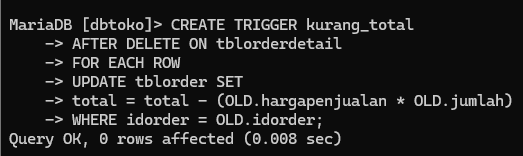


Cek setelah kita DROP kurang\_total

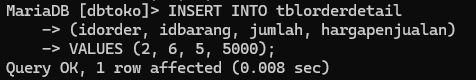


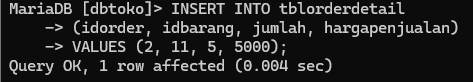
# JOIN (GABUNGAN TABEL)

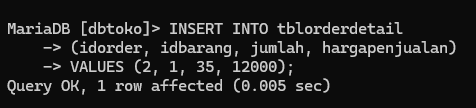
Sebelum belajar JOIN buat kembali Trigger yang telah dihapus pada materi diatas



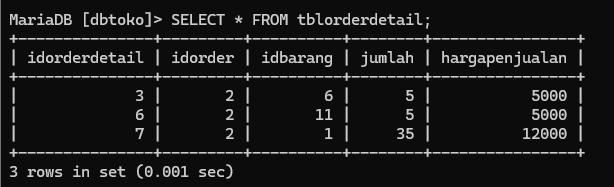
Setelah TRIGGER dibuat kembali lakukan insert data kembali dengan data sebagai berikut



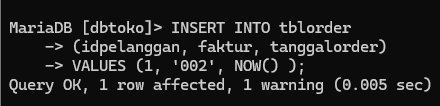




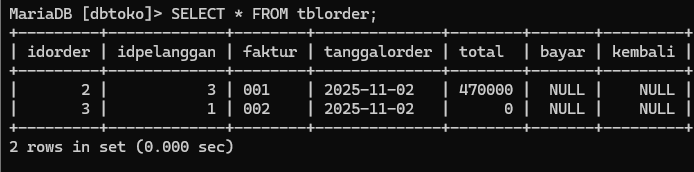
PERIKSA tabel [tblorderdetail]



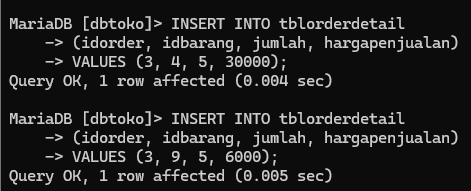
Buat INSERT data pada tabel [tblorder] lagi dengan data sebagai berikut



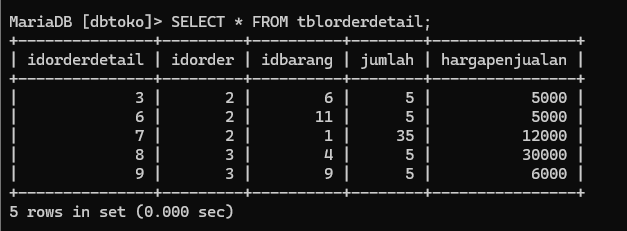
Periksa tabel [tblorder]



Lakukan insert data pada [tblorderdetail]

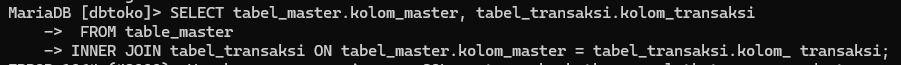


PERIKSA tabel [tblorderdetail]



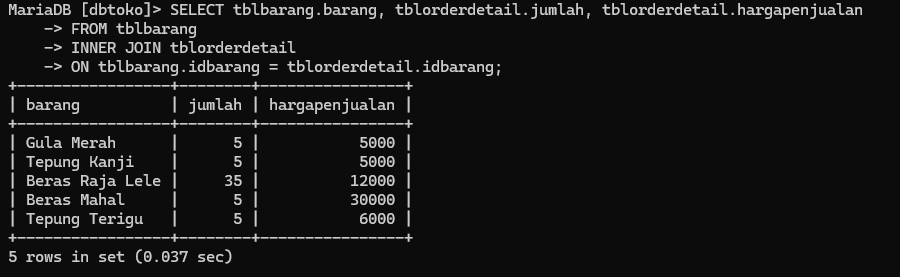
# INNER JOIN (MENGAMBIL BAGIAN YANG ADA DI TABEL MASTER DAN TABEL TRANSAKSI)

INNER JOIN untuk mengetahui barang apa yang sudah laku.



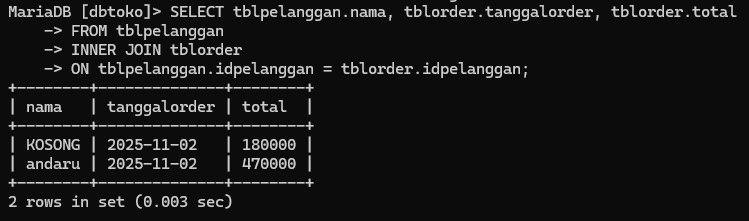
**JOIN HARUS DIMULAI DARI TABEL MASTER**

Kita akan langsung coba



Setelah kita enter command, akan langsung keluar table

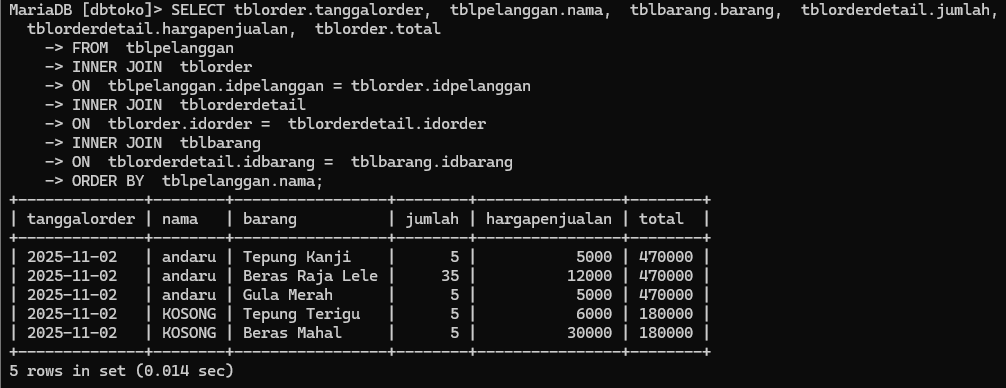
INNER JOIN untuk mengetahui pelanggan yang melakukan ORDER



Dan langsung tertera nama pelanggan yang order

# INNER JOIN BANYAK TABEL

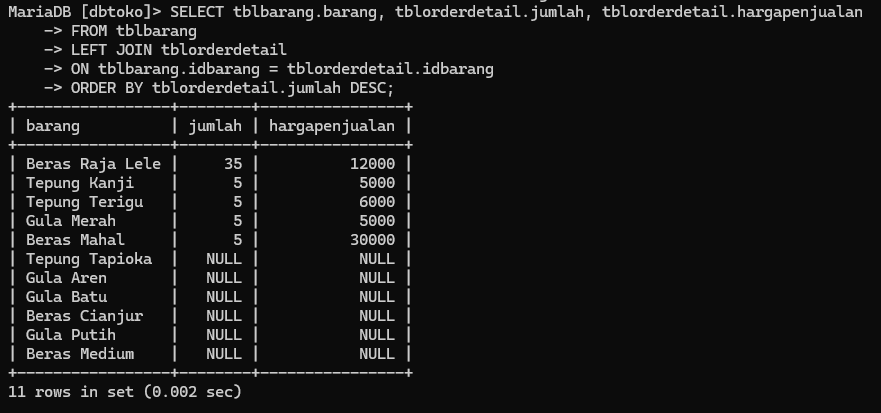
Dengan menggunakan INNER JOIN banyak tabel bisa diketahui barang yang dibeli pelanggan dan total pembelian oleh pelanggan.



Dan langsung keluar hasil INNER JOIN dengan banyak table

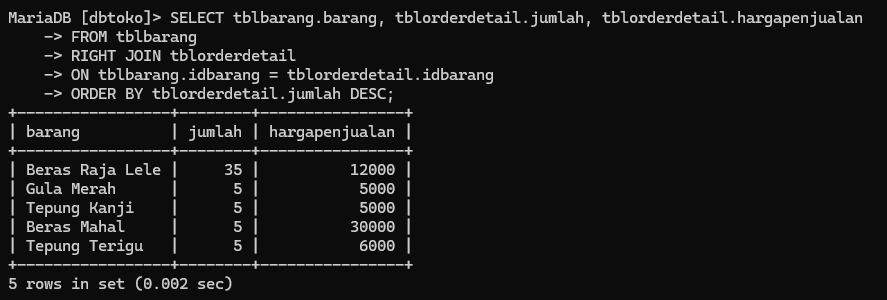
# LEFT JOIN (MENAMPILKAN TABEL MASTER)

Left join digunakan untuk menampilkan semua yang ada di tabel master dan mengambil sebagian yang ada di tabel transaksi



# RIGHT JOIN (MENAMPILKAN TABEL TRANSAKSI)

Right Join digunakan untuk menampilkan semua yang ada di tabel transaksi dan sebagian yang ada di tabel master.



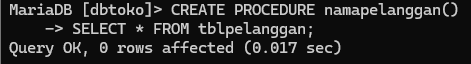
# STORE PROCEDURE

Store Procedure adalah blok program yang diletakan pada MySQL. Blok yang sudah dibuat bisa dipanggil jika diperlukan. PROCEDURE TIDAK MEMILIKI RETURN

* PROCEDURE TANPA PARAMETER



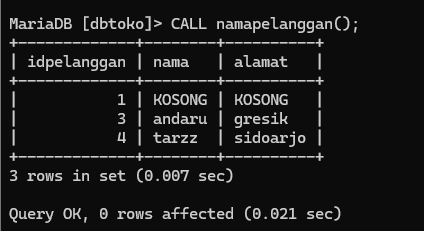
Kita buat prosedur dengan nama [namapelanggan()]



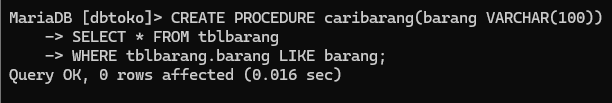
* PEMANGGILAN PROCEDURE



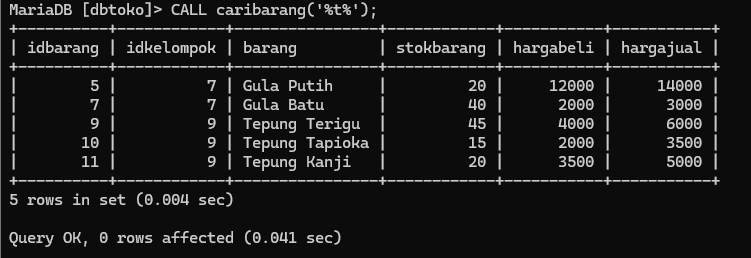
Kita akan coba memanggil [namapelanggan()]



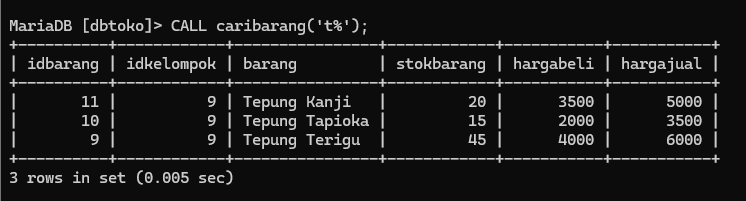
* PROCEDURE DENGAN PARAMETER



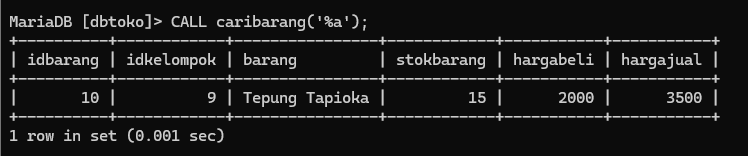
Panggil procedure yang sudah dibuat dengan parameter yang dimasuka [‘%t%’] yang menunjukan nama barang yang mengandung huruf [t]



Bisa juga memanggil Panggil procedure yang sudah dibuat dengan parameter yang dimasuka [‘t%’] yang menunjukan nama barang yang berinisial huruf [t]



Atau Bisa juga memanggil Panggil procedure yang sudah dibuat dengan parameter yang dimasuka [‘%a’] yang menunjukan nama barang yang berakhiran huruf [a]



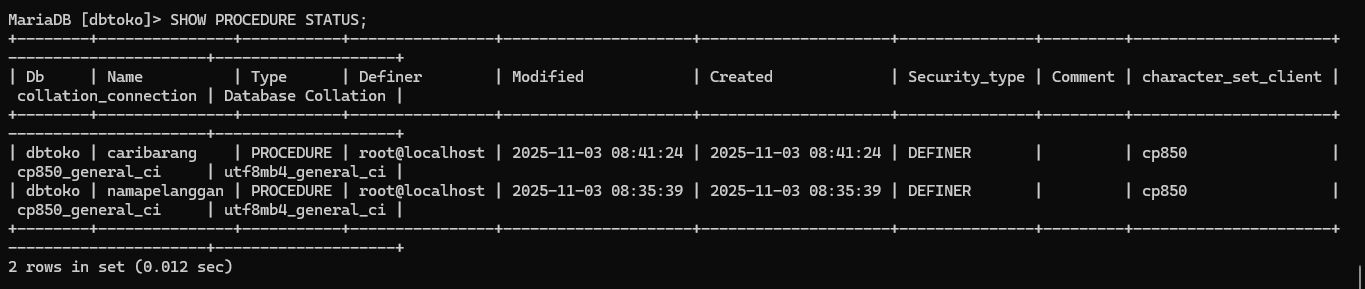
Dengan menggunakan PROCEDURE kita bisa membuat semua QUEREY atau SELECT di dalam procedure sehingga memudahkan dalam pembuatan aplikasi.

# MENAMPILKAN SEMUA PROCEDURE

MySQL bisa menampilkan semua isi procedure dengan perintah berikut



Procedure yang sudah dibuat bisa dilihat pada

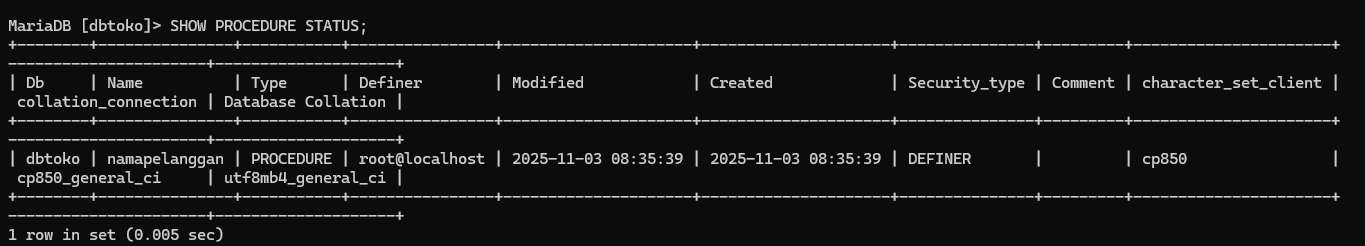


# MENGHAPUS STORE PROCEDURE

Untuk menghapus store procedure bisa menggunakan perintah



Periksa kembali procedure dengan perintah



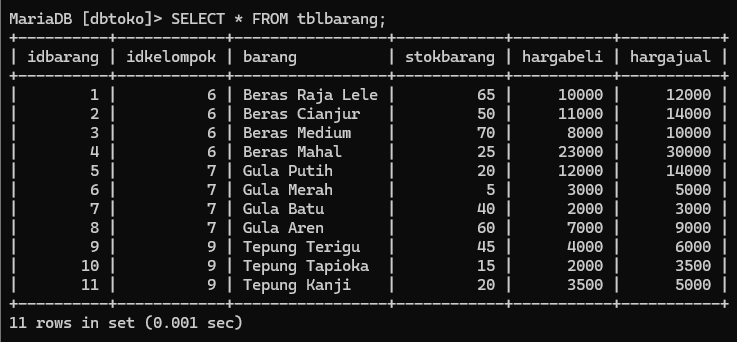
# FUNCTION

Function adalah blok program yang disimpan di MySQL yang bisa menerima INPUT atau PAREMETER dan MEMILIKI RETURN

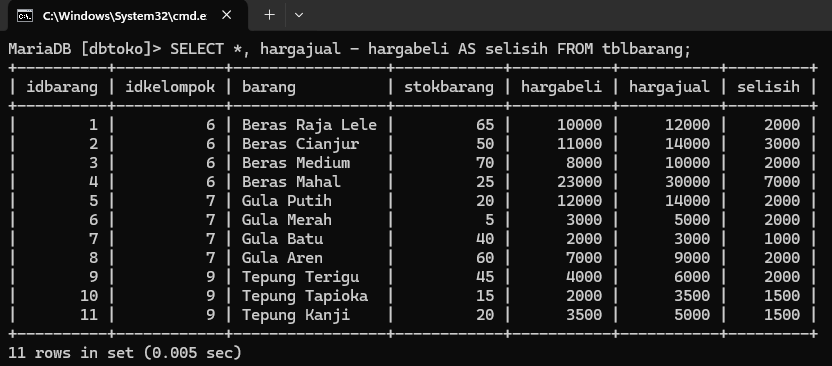
***MENYIAPKAN ISI FUNCTION***

Sebelum membuat Function harus disiapkan dulu code yang akan dibuat sebagai isi function sehingga hasil dari function sesuai dengan yang diharapkan. Pada materi belajar ini kita akan mebuat function untuk menampilkan selisih [hargajual] dan [hargabeli] pada tebel [tblbarang]

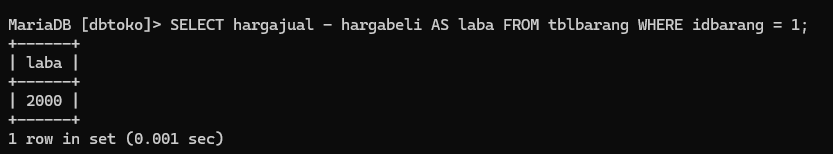
Tampilkan tabel [tblbarang]



Pada tabel diatas akan ditambahkan 1 kolom lagi sebagai selisih dari [hargajual] dikurangi dengan [hargabeli]



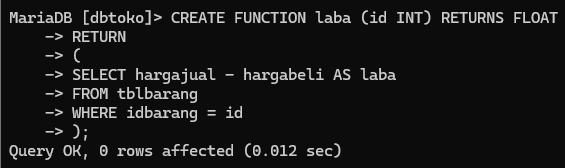
Selisih [hargajual] dan [hargabeli] ini disebut dengan [laba]. Sekarang akan diambil selisih perbarang untuk dihitung sebagai laba. Sebagai contoh diambil satu barang dengan [idbarang = 1] maka selisihnya akan ditampilkan seperti gambar dibawah.



Setelah isi dibuat, saatnya membuat function. Cara membuat function sebagai berikut



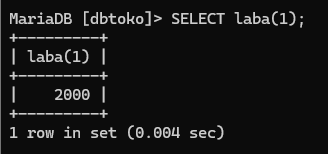
Isi parameter tipe data sesuai dengan data yang dimasukan. Pada contoh diatas parameter yang dimasukan adalah [idbarang] dengan tipe data INT. Output yang dihasilkan dari isi function diatas adalah selisih [hargajual] – [hargabeli] yang bertipe data FLOAT



# MEMANGGIL FUNCTION

Untuk memanggil function gunakan perintah berikut

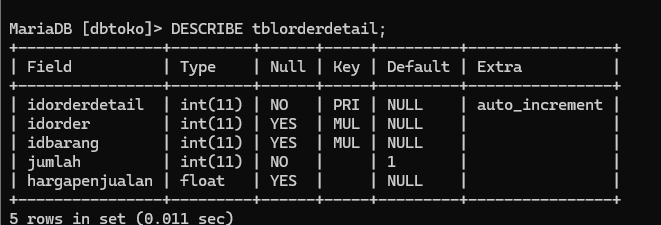




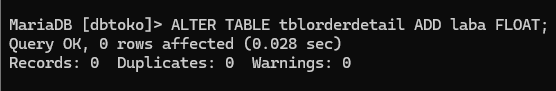
# PENGGUNAAN FUNCTION

Materi berikut akan menunjukan penggunaan function pada aplikasi yang sedang dibuat.

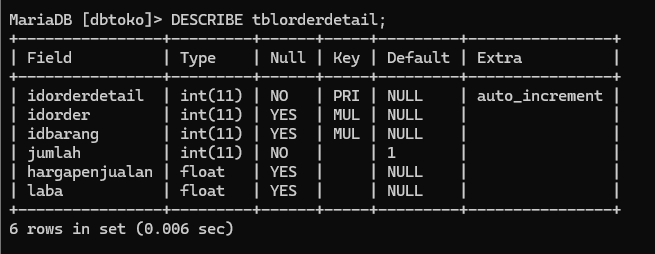
Lihat struktur tabel [tblorderdetail]



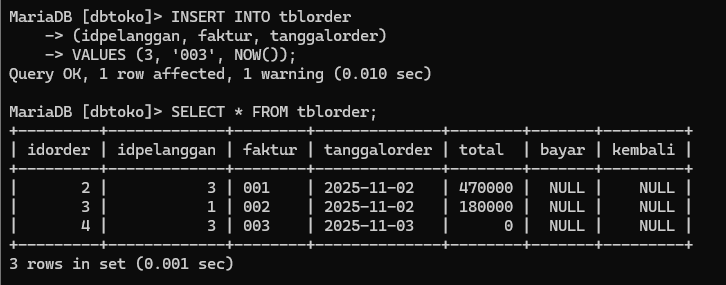
Tambahkan sebuah kolom dengan nama [laba]



Periksa kembali dengan describe

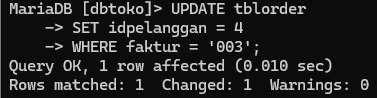


Lakukan proses order dengan melakukan insert pada tabel [tblorder], lalu cek [tblorder]

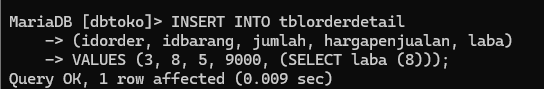


Buat insert pada tabel [tblorderdetail]

Dikarenakan ada 2 idpelanggan yang sama, maka idpelanggan dengan faktur = 003 akan saya ubah ke idpelanggan = 4

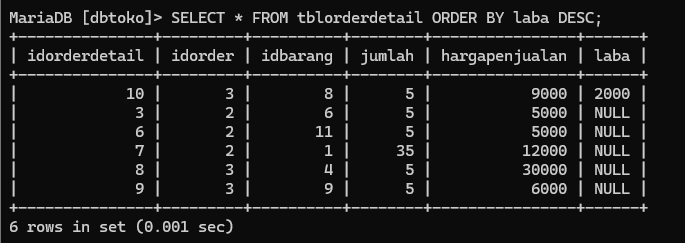


Selanjutnya kita akan lanjutkan insert

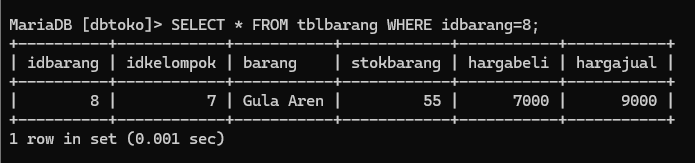


Pada saat INSERT kolom [laba] akan di isi dengan function yang menghitung [hargajual] dikurangi [hargabeli] dimana [idbarang] dimasukan sebagai parameter input pada function.

Periksa tabel [tblorderdetail]

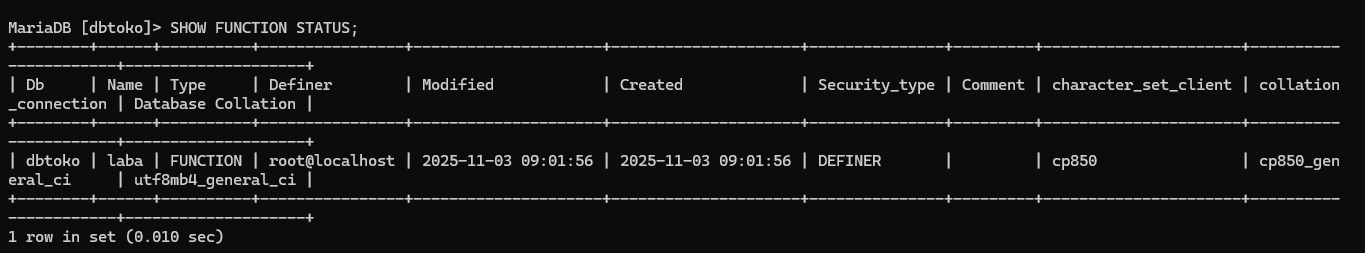
****

Periksa selisih [hargajual] – [hargabeli] pada tabel [tblbarang]. Jika sesuai berarti function sudah berjalan sesuai dengan yang diharapkan.



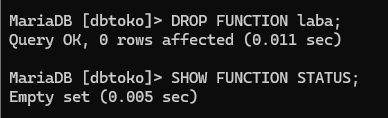
# MENAMPILKAN SEMUA FUNCTION

Untuk menampilkan function yang sudah dibuat bisa menggunakan

****

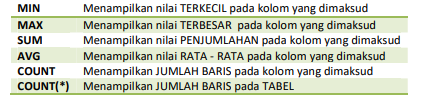
# MENGHAPUS FUNCTION

Untuk menghapus function bisa menggunakan perintah berikut, dan akan kita cek setelah menghapus function



# SELECT AGGREGATE

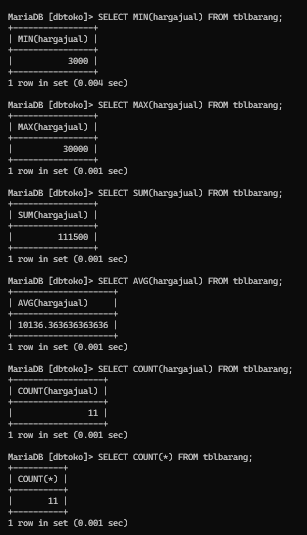
SELECT AGREGATE adalah SELECT yang menampilkan function bawaan dari MySQL. Ada beberapa select aggregate yang akan dipelajari yaitu



Src img : belajar mysql dari 0 | by : Pak Isa Hamdan, S.Kom



Antara nama aggregate dan kurung tidak boleh ada spasi



# SELECT BETWEEN (SELECT ANTARA DUA NILAI)

Select yang digunakan untuk menampilkan data antara dua nilai terendah dan tertinggi



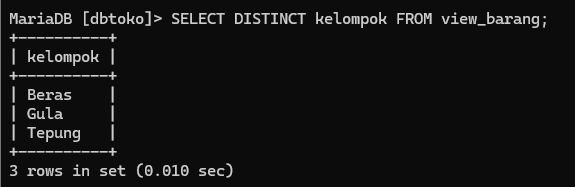


Disini hanya menampilkan data dengan hargajual di antara 5000 – 15000

# SELECT DISTINCT (MENAMPILKAN DATA YANG SAMA HANYA SATU KALI)

Untuk menampilkan data yang sama satu kali.



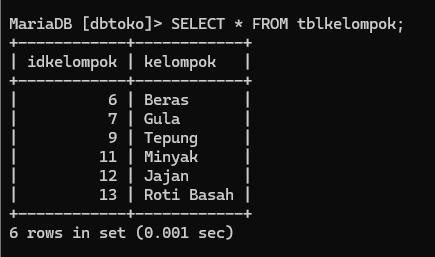


Disini hanya menampilkan kelompok kelompok yang ada di view\_barang berdasarkan kelompok / [idkelompok]

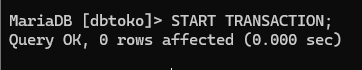
# START TRANSACTION, COMMIT, DAN ROLLBACK

Setiap perintah di MySQL yang masuk kelompok (INSERT, UPDATE, DELETE, SELECT) bisa dimasukan kedalam perintah START TRANSACTION, COMMIT, DAN ROLLBACK. Perintah yang dimulai dengan START TRANSACTION bisa dilakukan ROLLBACK (undo atau pembatalan perintah). COMMIT digunakan agar perintah yang dijalankan TIDAK BISA DI ROLLBACK (undo).

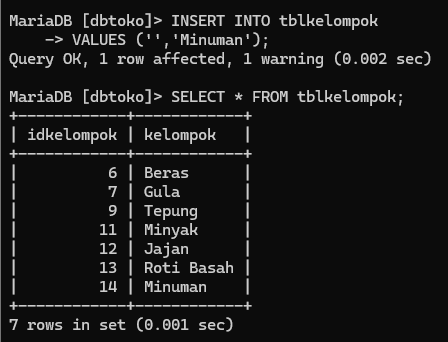
buka kembali tabel [tblkelompok]



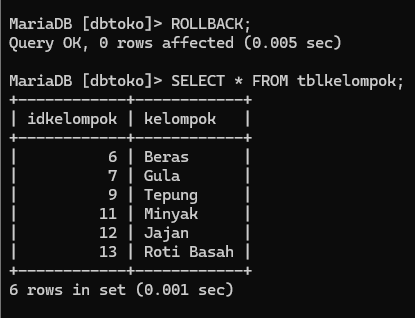
Lakukan perintah berikut, untuk memulai



Berikan perintah INSERT sebagai berikut, setelah INSERT langsung saja kita cek



Berikan perintah ROLLBACK lalu periksa kembali data yang sudah dimasukan



Maka insert yang sebelumnya akan hilang seperti saat kita ctrl + z (Undo) pada word, vscode, dll

Ulangi perintah INSERT sebelumnya



Jadi kesimpulannya adalah START TRANSACTION memulai transaksi, COMMIT menyimpan perubahan permanen, dan ROLLBACK membatalkan perubahan sebelum transaksi di-commit.

# DCL

# *Data Control Language*

# Management User dan Hak akses

# TENTANG DCL (Data Control Language)

Pengaturan USER diperlukan agar database yang sudah dibuat dengan susah payah tidak di acak – acak oleh orang yang tidak berhak.

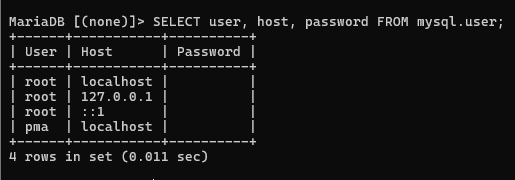
DCL adalah materi yang akan digunakan untuk mempelajari tentang menajemen atau pengelolaan USER dan Hak Akses yang akan menggunakan database MySQL.

Sebelum menggunakan MySQL kita di haruskan login terlebih dahulu dengan user awal [root] dan password [kosong].



# MENAMPILKAN SEMUA USER

Untuk menampilkan semua user yang audah ada di MySQL gunakan perintah berikut



HOST adalah lokasi komputer dimana database MySQL disimpan. Jika menggunakan komputer atau laptop sendiri maka HOST nya adalah localhost atau dengan menggunakan alamat IP 12.7.0.0.1

**Penambahan user dan hak akses hanya bisa diberikan oleh user [root]**

# MENAMBAH USER

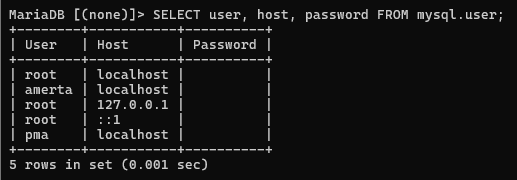
Untuk menambah user gunakan perintah berikut

****

jika pada komputer di lokasi yang lain misal pada server jaringan atau internet masukan IP dari SERVER tersebut pada HOST



Periksa dengan perintah berikut



# MEMBERIKAN PASSWORD PADA USER

User yang dibuat di awal BELUM ada passwordnya. Untuk memberikan password gunakan perintah berikut





Berikan perintah berikut agar MySQL membaca ulang (REFRESH) user dan password yang telah dibuat atau diubah.

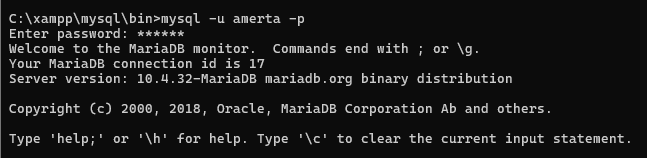


# MENGUJI USER DAN PASSWORD YANG TELAH DIBUAT

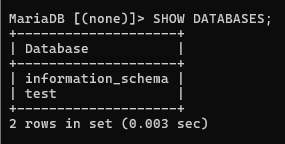
Lakukan quit untuk keluar dari MySQL (shortcut. gunakan ctr + c untuk quit)



Masuk lagi dengan perintah, user [amerta] password [amerta]



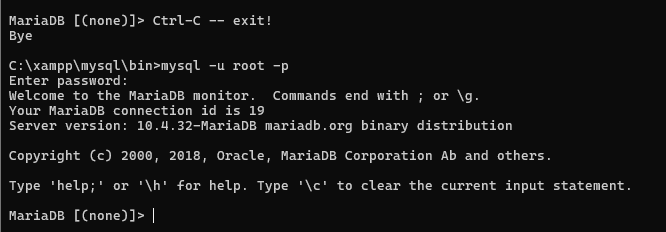
Periksa database



Pada saat menampilkan database tidak terdapat database seperti pada waktu masuk dengan user [root]

# MEMBERIKAN HAK AKSES (PRIVILEGES)

USER KE DATABASE Sebelummemberikan hak akses pada database pastikan anda QUIT terlebih dahulu, dan masuk kembali dengan user [root].



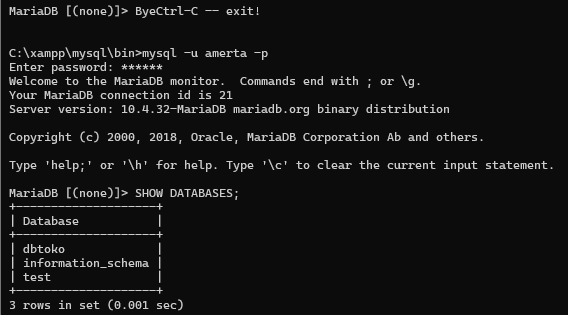
Untuk memberikan hak akses pada database gunakan perintah





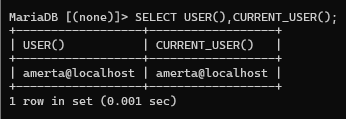
Setelah pemberian hak akses lakukan refresh dengan perintah



Untuk memeriksa apakah hak akses berhasil diberikan kita harus quit terlebih dahulu dari user [root] dan masuk kembali dengan user [amerta] password [amerta] dan tampilkan database

# MENAMPILKAN USER YANG SEDANG LOGIN (MASUK)

Untuk melihat user yang sedang login, gunakan perintah berikut



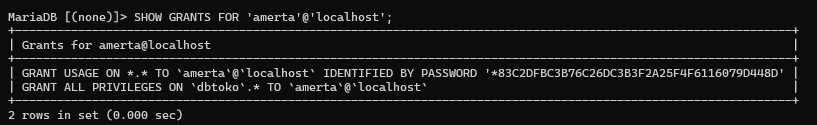
# MENAMPILKAN HAK AKSES YANG DIBERIKAN

Untuk memeriksa hak akses yang diberikan pada user, lakukan quit terlebih dahulu dan login kembali dengan user [root]



Gunakan perintah berikut untuk menampilkan hak akses

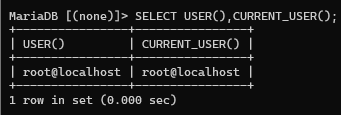




# MENGHAPUS HAK AKSES

Untuk menghapus hak akses pada user [smkrevit] anda harus login sebagai user [root] gunakan perintah sebagai berikut

Periksa dulu user login kalian



Untuk menghapus hak akses gunakan perintah

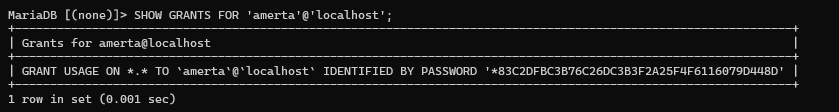




Refresh perubahan hak akses dengan perintah



Periksa hak akses yang sudah dihapus



# MEMBERIKAN HAK AKSES PADA DATABASE DENGAN TABEL

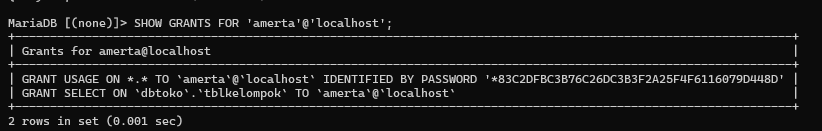
Gunakan perintah berikut untuk memberikan hak akses pada 1 tabel



Kita coba berikan hak akses tblkelompok pada user [amerta]



Periksa hak akses



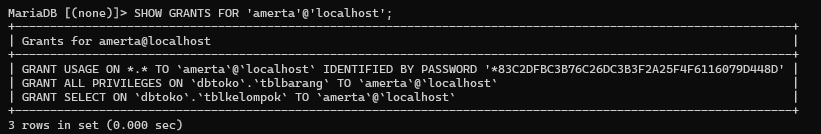
Untuk menambah tabel yang diberikan hak aksesnya gunakan perintah



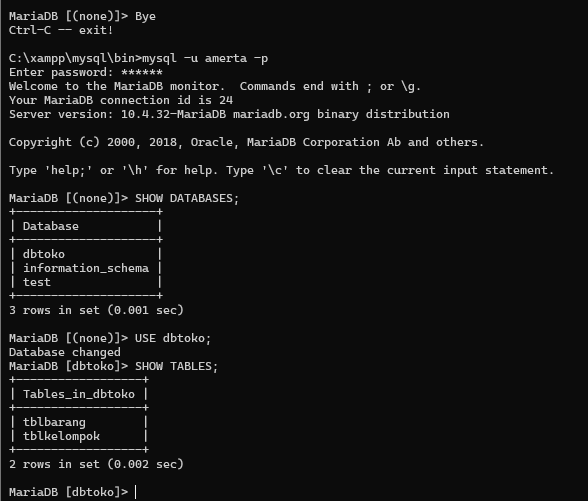
Kita akan coba tambahbkan akses tblbarangn pada user [amerta]



Periksa hasilnya



Untuk memeriksa hasil pemberian hak akses, lakukan quit dari user [root] dan login dengan user [smkrevit] lalu tampilkan database 🡪 Aktifkan database [dbtoko] 🡪 Periksa table



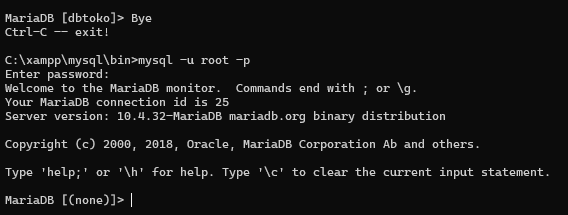
Disini kita [user amerta] hanya bisa melihat atau mengakses table yang diberi akses oleh root yaitu [tblkelompok] dan [tblbarang]

# MEMBERIKAN HAK AKSES PADA SELECT, INSERT, DELETE, UPDATE, PADA TABEL

MySQL memeberikan pengaturan agar user bisa diberikan hak akses tertentu sehingga dapat menjamin keamanan database.

Untuk memberikan hak akses SELECT, INSERT, DELETE, & UPDATE lakukan langkah langkah berikut

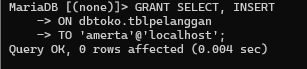
Lakukan QUIT 🡪 Login dengan user [root]



Berikan perintah berikut untuk memberikan hak akses SELECT DAN INSERT pada tabel [tblpelanggan].



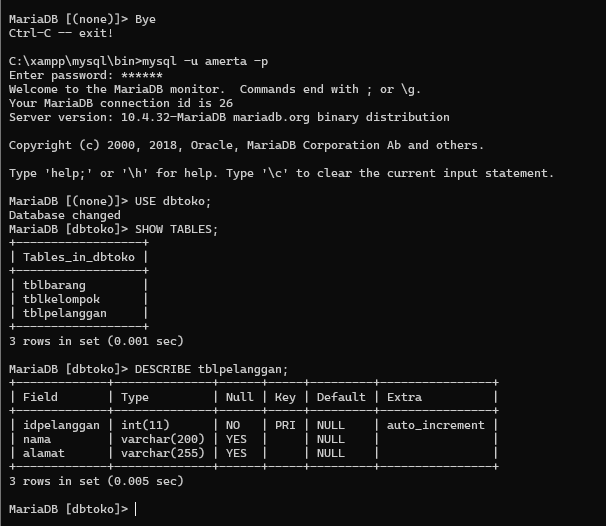
Jika setelah GRANT adalah ALL artinya user tersebut diberikan hak untuk SELECT INSERT DELETE UPDATE pada tabel tersebut.



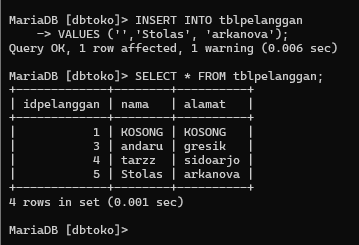
Disini kita memberikan akses SELECT dan INSERT pada table [tblpelanggan] untuk user amerta

Lakukan pengujian dengan cara sebagai berikut

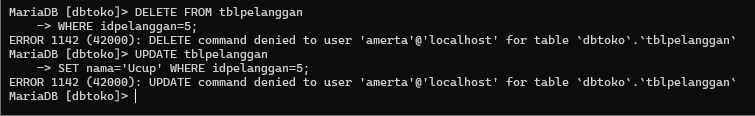
Lakukan quit terlebih dahulu 🡪 Login dengan user [amerta] 🡪 Aktifkan [dbtoko] 🡪 Periksa table 🡪 Lihat struktur tabel [tblpelanggan]



Kita akan coba melakukan INSERT dan SELECT



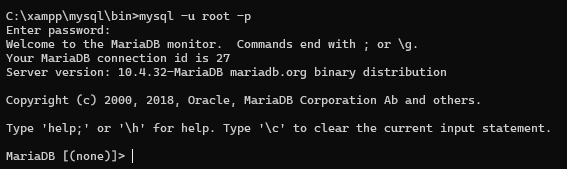
Kita akan coba untuk melakukan DELETE dan UPDATE untuk data yang baru saja kita tambahkan



Disini kita sebagai user [amerta] tidak bisa men-DELETE atau men-UPDATE karena kita tidak diberikan akses DELETE dan UPDATE oleh root

# UBAH PASSWORD USER

login sebagai user [root]



Ubah password dengan perintah berikut



Refresh :

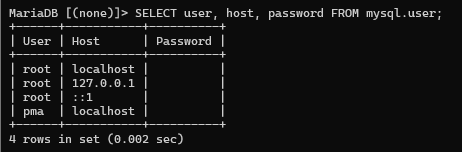


# HAPUS USER

Untuk menghapus user hanya bisa dilakukan oleh user [root]. Hapus user dengan perintah berikut



Periksa user yang telah dihapus



# MERUBAH PASSWORD USER [root]

Untuk merubah password user root gunakan perintah berikut

Pastikan sudah keluar dari MySQL sebelum merubah password user [root]

BERIKAN PERINTAH BERIKUT UNTUK SET PASSWORD, JIKA PASSWORD AWAL [kosong]



Lakukan login dengan user [root] dan password [123456]

Berikan perintah berikut jika ingin mengganti password lama dengan password baru





# ===================== TAMAT ======================