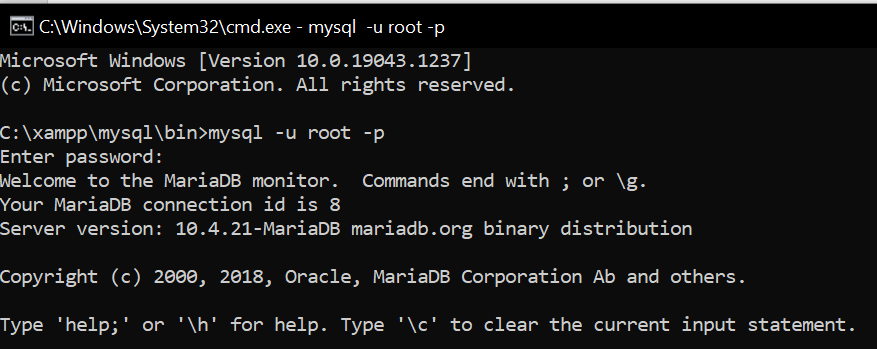
|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Basis Data Minggu Ke-11 SMST5 (Video Playlist Basis Data SMK Revit 18 (DDL, DML)) | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Melakukan proses download XAMPP di <https://www.apachefriends.org/download.html> dan instalasi agar terpasang di dalam device.

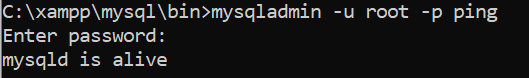
2. Untuk menjalankan mysql, buka folder xampp\mysql\bin kemudian ketikkan cmd di atas atau tempat keterangan tempat foldernya. Jika cmd sudah terbuka ketikkan mysql -u root -p kemudian enter, jika keluar Enter Password langsung klik enter lagi. Apabila berjalan dengan baik akan keluar seperti ini



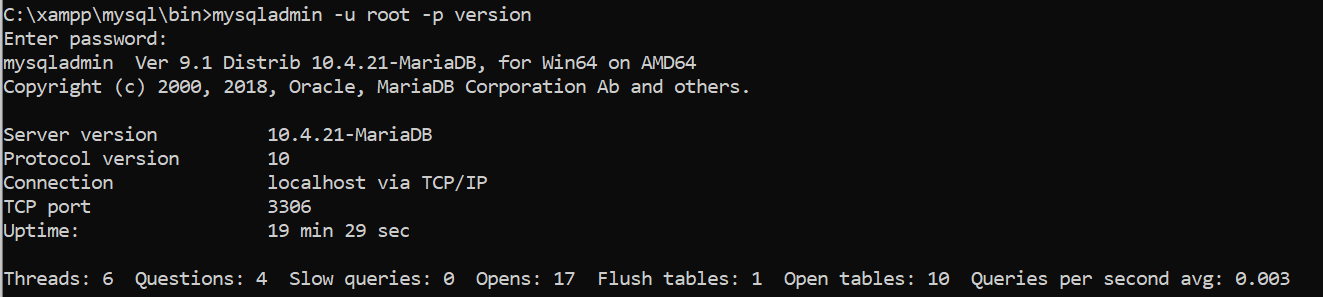
Kemudian untuk keluar dari mysql ketikkan quit di MariaDB [(none)]> seperti ini



3. Untuk memeriksa mysql jalan atau tidak ketik mysqladmin -u root -p ping di xampp\mysql\bin lalu enter seperti ini

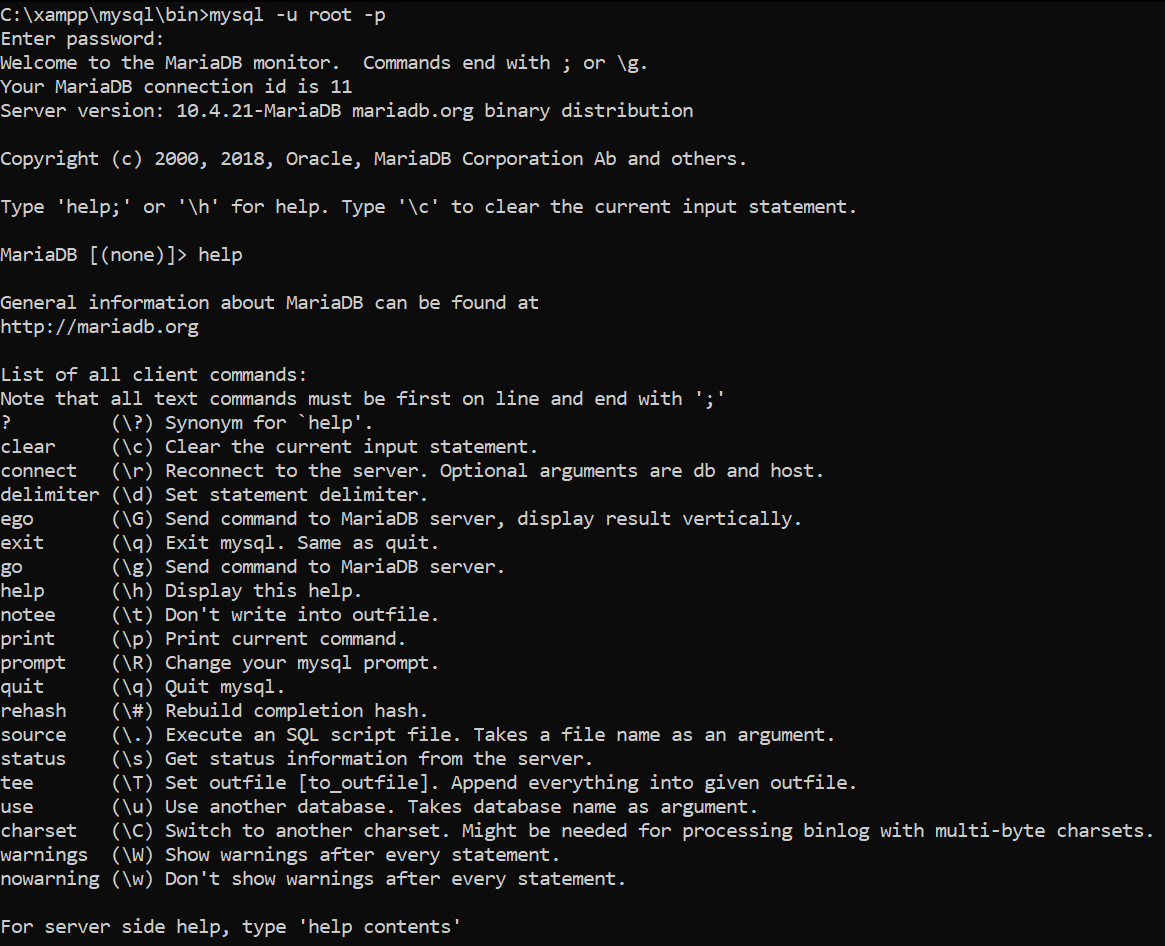


Jika keluar mysqld is alive artinya mysql berjalan, kemudian untuk mengecek versi dari mysql sama halnya dengan memeriksa mysql jalan atau tidak, bedanya jika mengecek versi bukan ping melainkan version seperti ini

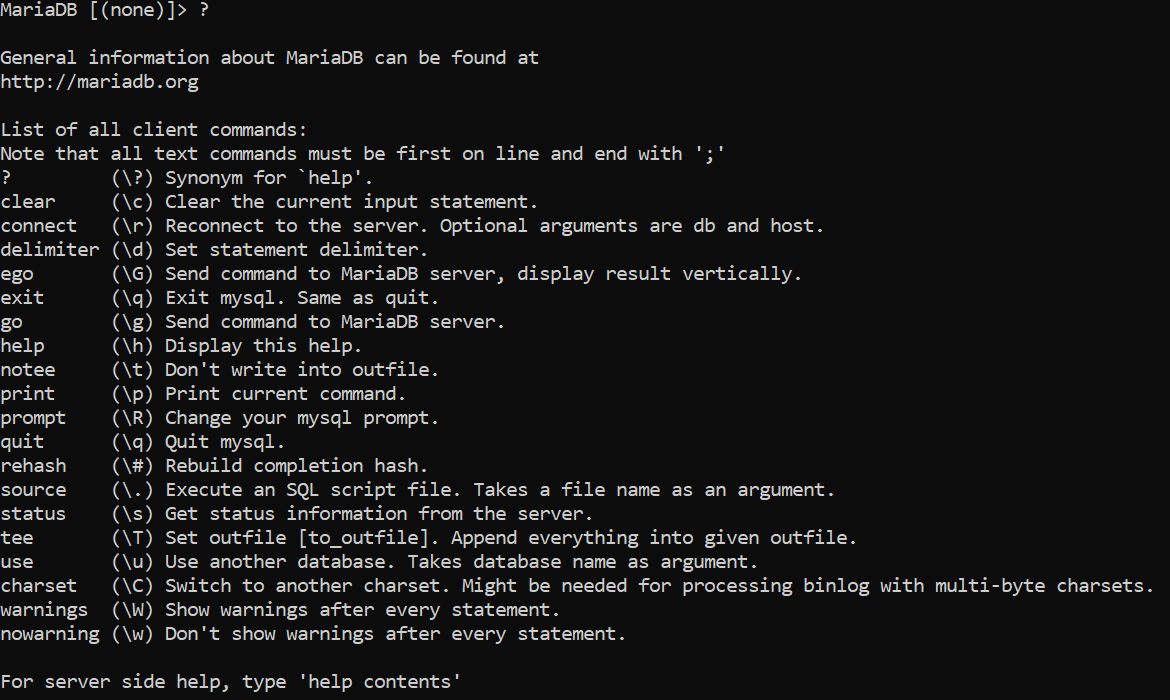


4. Karena banyaknya perintah dalam menjalankan mysql, tentunya sql memiliki fitur help agar bisa melihat perintah mysql yang akan digunakan. Caranya dengan menjalankan mysqlnya dulu seperti tadi yaitu mysql -u root -p kemudian enter, enter code tetap enter sampai keluar MariaDB [(none)]> ketikkan help atau ? seperti ini

Jika help

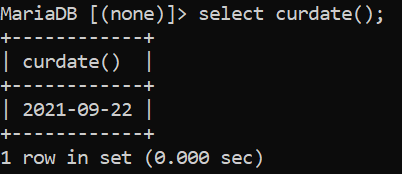


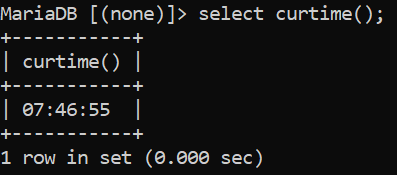
Jika ?



Intinya help dan ? sama.

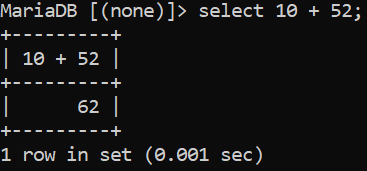
5. Mysql bisa juga digunakan untuk menampilkan tanggal dan waktu, caranya jalankan mysql dulu sampai ada MariaDB [(none)]> lalu ketikkan select curdate(); (harus ada ;) untuk melihat tanggal dan select curtime(); untuk melihat waktu. Hasilnya seperti ini



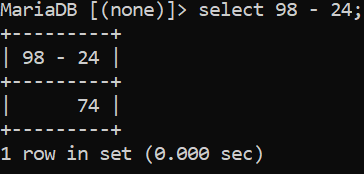


6. Mysql juga dapat digunakan sebagai kalkulator, baik penjumlahan, pengurangan, perkalian, pembagian, modulo (sisa hasil pembagian), dan hasil pembagian integer (utuh). Untuk menggunakan kalkulatornya dengan cara berikut, jalankan mysql sampai ada MariaDB [(none)]> kemudian ketikkan select angka yang akan dioperasikan beserta pengoperasiannya misalkan 10 ditambah 5 berarti 10 + 5 dan tutup dengan ; jadinya seperti ini select 10 + 5;

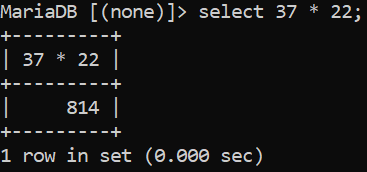
Penjumlahan



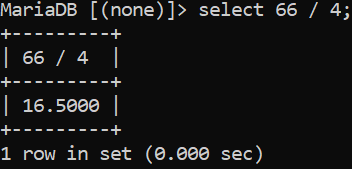
Pengurangan



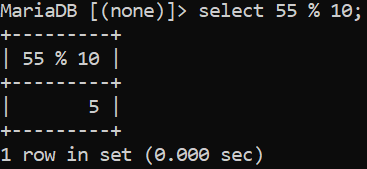
Perkalian



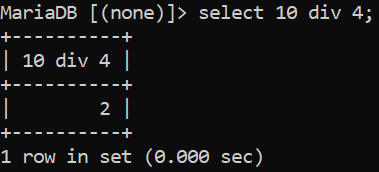
Pembagian



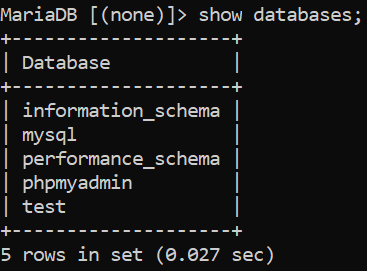
Modulo



Hasil pembagian integer



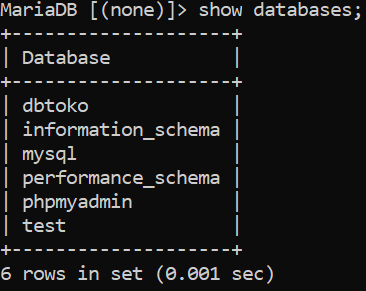
7. DDL merupakan singkatan dari Data Definition Language yang berfungsi untuk pembuatan database, table, dan index. DDL adalah proses pembuatan RUMAH dari data. Data akan disimpan dalam **kolom**, kolom akan disimpan dalam **table**, dan table akan disimpan dalam **database**. Untuk menampilkan semua database yang ada di dalam komputer, di dalam cmd jalankan mysql dulu lalu enter enter sampai keluar MariaDB [(none)]> dan ketikkan show databases; seperti ini



Yang berarti di dalam computer ini ada 5 databases yang tersimpan. Kemudian untuk membuat databases baru melalui cmd caranya dengan mengetikkan create database nama\_databasenya; di dalam MariaDB [(none)]> sampai muncul code berhasil dibuat seperti ini



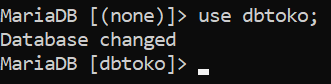
Kemudian jika dicek lagi dalam daftar database akan muncul database baru yang dibuat tadi



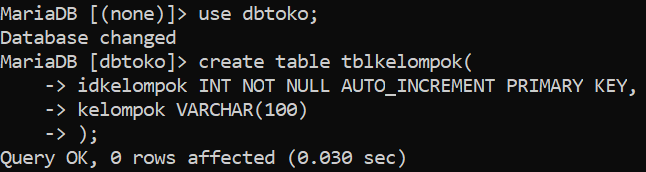
Untuk menghapus database codenya sama dengan ketika membuatnya, bedanya jika membuat create jika menghapus drop. Contoh drop database dbtoko



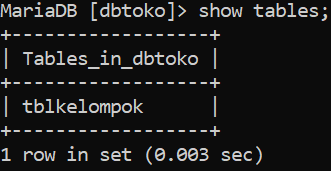
8. Jika sebuah database sudah dibuat, untuk mengaktifkannya agar dapat digunakan caranya dengan mengetikkan use dbtoko di dalam MariaDB [(none)]> sampai MariaDB[(none)] menjadi MariaDB[dbtoko] seperti ini



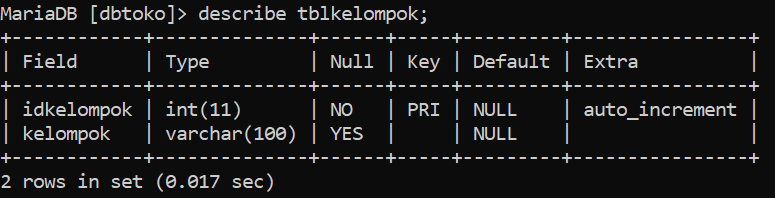
9. Karena database sudah diaktifkan, maka table dapat dibuat dengan cara mengetikkan create table nama\_table (nama\_kolom TIPE\_DATA, nama\_kolom TIPE\_DATA); di dalam MariaDB[dbtoko] yang sudah diaktifkan sebelumnya



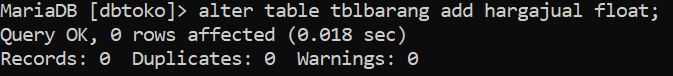
10. Sebuah table sudah dibuat di dalam database dbtoko dengan 2 kolom yaitu idkelompok dan kelompok, untuk melihat apakah table sudah masuk ke dalam dbtoko maka bisa dilihat dengan mengetikkan show tables; di MariaDB[dbtoko]



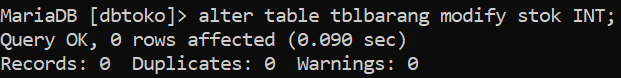
Untuk melihat struktur table atau kolom apa saja yang ada di dalamnya bisa ketik describe nama\_table seperti ini



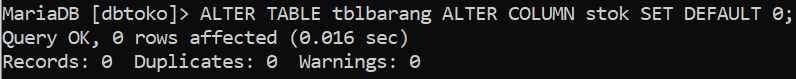
Untuk menambahkan sebuah kolom dari sebuah table yang sudah dibuat caranya dengan mengetikkan alter table nama\_table add nama\_kolom tipe data seperti ini



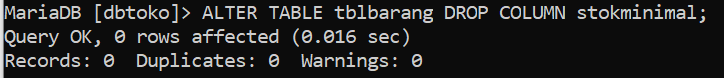
Dengan begitu kolom baru akan masuk ke dalam table yang dituju, untuk mengubah tipe data dari sebuah kolom caranya dengan mengetikkan alter table nama\_tabel modify nama\_kolom tipe data; seperti ini



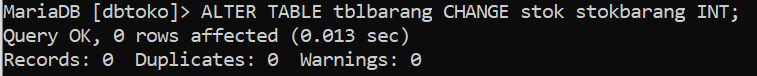
Untuk memberi nilai default pada sebuah kolom caranya dengan mengetikkan ALTER TABLE nama\_table ALTER COLUMN nama\_kolom SET DEFAULT ISI\_NILAI\_DEFAULT\_JIKA\_VARCHAR\_BERI\_TANDA\_PETIK; seperti ini



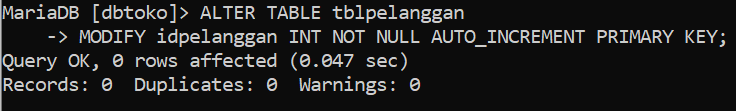
11. Untuk menghapus kolom dari sebuah table caranya dengan mengetikkan di MariaDB[dbtoko] ALTER TABLE nama\_table DROP COLUMN nama\_kolom; seperti ini



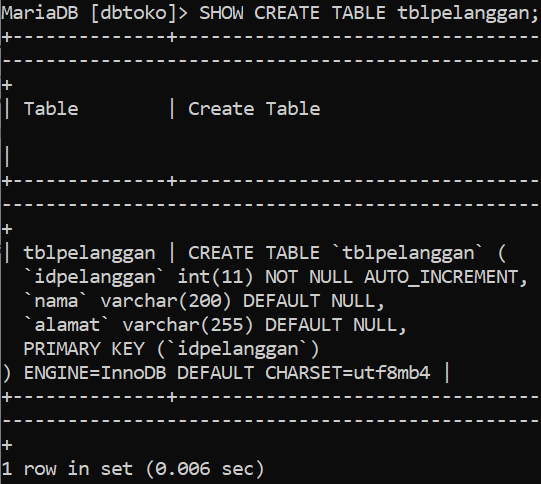
12. Untuk merubah nama sebuah kolom caranya dengan mengetikkan ALTER TABLE nama\_table CHANGE kolom\_lama kolom\_baru TIPE DATA; di dalam MariaDB[dbtoko] seperti ini



13. Untuk menambahkan primary key dari sebuah kolom caranya dengan mengetikkan ALTER TABLE nama\_table MODIFY nama\_kolom TIPE DATA NOT NULL AUTO\_INCREMENT PRIMARY KEY; seperti ini

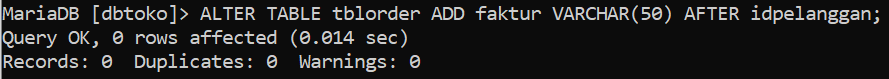


14. Pada pembuatan database terdapat 2 engine atau mesin yang digunakan yaitu MYSIAM dan InnoDB. Untuk proses pembuatan database relasional harus menggunakan InnoDB dan MYISAM tidak bisa menerima penggunaan penggunaan FOREIGN KEY. Untuk mengecek engine apa yang digunakan dalam sebuah table caranya dengan mengetikkan SHOW CREATE TABLE nama\_table; di MariaDB[dbtoko] seperti ini

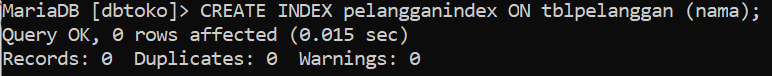


Disitu terdapat ENGINE = InnoDB yang berarti engine yang digunakan dalam tblpelanggan adalah engine InnoDB.

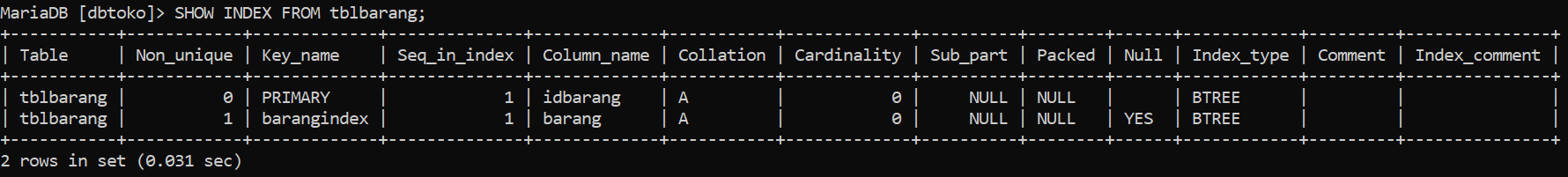
15. Untuk menambahkan sebuah kolom baru dalam sebuah table namun letak kolom ini berada dibawah kolom urutan pertama misalnya dibawah kolom id maka ketikkan ALTER TABLE nama\_table ADD nama\_kolom\_baru TIPE DATA AFTER nama\_kolom\_sebelumnya\_atau\_nama\_kolom\_urutan\_sebelumnya; seperti ini



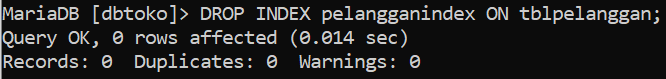
16. Index digunakan untuk mempercepat proses pencarian data terutama pada data yang jumlahnya diatas ribuan. Pada table diatas yang biasa digunakan untuk pencarian data adalah nama barang dan nama pelanggan. Untuk membuat sebuah index caranya dengan mengetikkan CREATE INDEX nama\_index ON nama\_table (kolom); seperti ini



17. Jika sebelumnya membuat index, maka untuk menampilkan sebuah index caranya dengan mengetikkan SHOW INDEX FROM nama\_table; lalu enter maka hasilnya akan seperti ini



18. Untuk menghapus sebuah index caranya dengan mengetikkan DROP INDEX nama\_index ON nama\_table; seperti ini

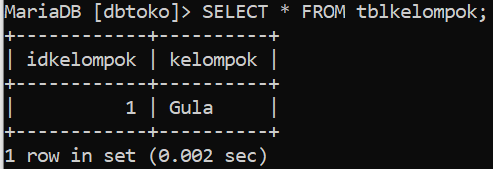


19. DML atau singkatan dari Data Manipulation Language yang berfungsi untuk Insert, Delete, Update, Select, View, Trigger, Procedure, dan Function.

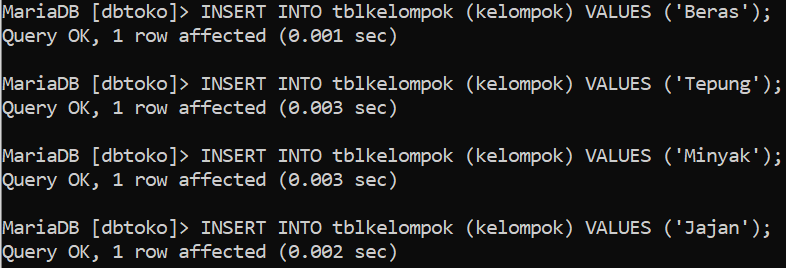
20. Sebagai sebuah percobaan yang pertama yaitu proses insert data dalam tblkelompok, caranya dengan mengetikkan INSERT INTO nama\_table VALUES (kolom, kolom, semua\_kolom) sebagai contoh seperti ini



Yang berarti karena di dalam tblkelompok ada 2 kolom yaitu idkelompok dan kelompok, idkelompok adalah auto increment primary key yang berarti data akan otomatis berurutan dari 1 dan kelompok adalah kolom biasa yang bertipe varchar, dari INSERT INTO artinya tambahkan data ke dalam tblkelompok dengan VALUES atau isi (‘’ untuk kolom ke satu yaitu idkelompok yang nanti data akan otomatis keluar 1 tanpa perlu diberi isi dalam ‘’ nya kemudian koma, ‘Gula’ untuk diletakkan dalam kolom kedua yaitu kelompok lalu enter. Untuk memeriksa apakah data yang diinsert tadi masuk ke dalam tblkelompok caranya dengan mengetikkan SELECT \* FROM nama\_table; seperti ini



21. Jika sebelumnya insert untuk dimasukkan ke dalam semua kolom yang ada dalam sebuah table, berikutnya adalah cara untuk menambahkan data ke dalam kolom tertentu saja dengan mengetikkan INSERT INTO nama\_table (kolom\_yang\_dituju) VALUES (isi\_untuk\_semua\_kolom\_yang\_dituju) contohnya seperti ini

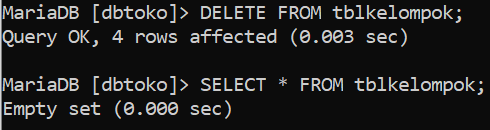


22. Untuk menghapus sebagian record (baris data) yang sudah dibuat sebelumnya caranya dengan mengetikkan DELETE FROM nama\_table WHERE baris\_yang\_akan\_dihapus = dan\_isinya; contohnya



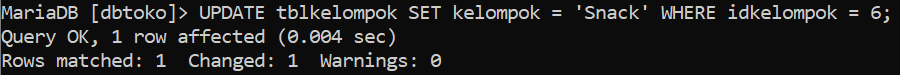
Yang berarti hapus data dari tblkelompok yang idkelompoknya adalah 5.

23. Untuk menghapus semua record (baris data) dari sebuah table caranya dengan mengetikkan DELETE FROM nama\_table; seperti ini



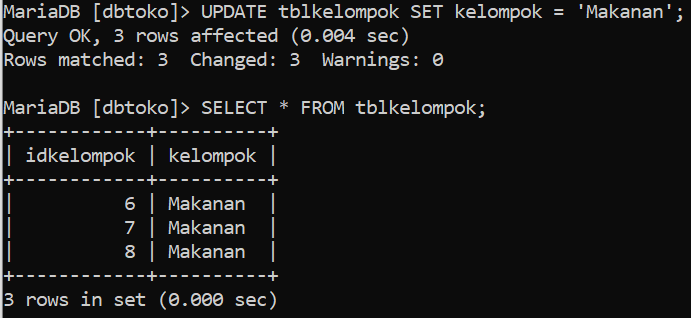
Yang berarti hapus semua data yang ada di tblkelompok kemudian ketika di cek maka data yang ada di tblkelompok akan kosong atau empty.

24. Untuk melakukan update atau mengubah data dari suatu kolom dalam sebuah table caranya dengan mengetikkan UPDATE nama\_table SET nama\_kolom = isi\_kolom WHERE nama\_kolom = baris\_yang\_dipilih\_untuk\_diubah; contohnya seperti ini



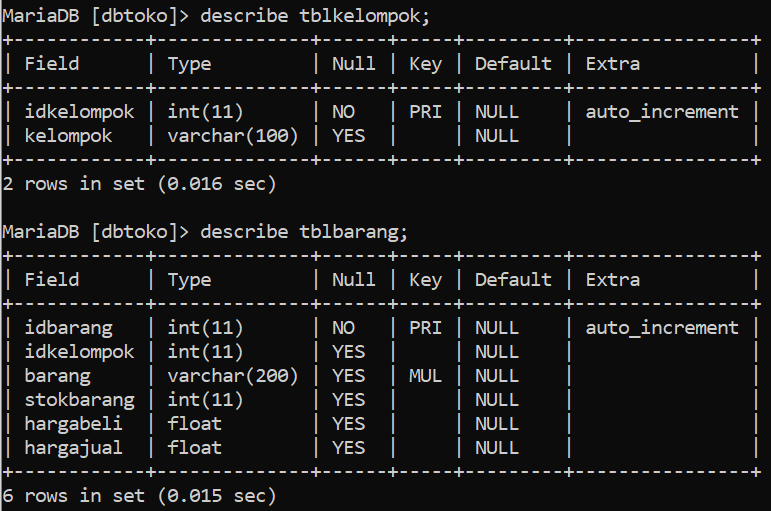
Yang berarti update atau ubah di tblkelompok bagian kolom kelompok untuk diupdate dengan kata ‘Snack’ dengan patokan idkelompoknya adalah 6.

25. Jika sebelumnya update untuk salah satu data saja, maka langkah berikutnya adalah update untuk semua data dari sebuah table dengan mengetikkan UPDATE nama\_table SET nama\_kolom = isi\_kolom; seperti ini



Artinya update atau lakukan perubahan pada tblkelompok dengan mengubah semua data dalam kolom kelompok menjadi ‘Makanan’ semua sehingga ketika dicek tblkelompoknya pada bagian kolom kelompok isinya Makanan semua.

26. Tabel Master adalah table yang Memasuki table lain. Pada contoh tblkelompok dan tblbarang berikut

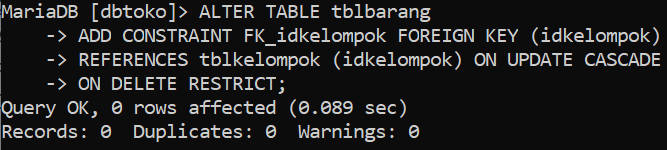


Di dalam tblkelompok dan tblbarang terdapat kolom [idkelompok] yang merupakan primary key pada tblkelompok dan memasuki tblbarang sebagai bukan primary key, maka tblkelompok dinamakan TABEL MASTER. Kemudian ada TABEL TRANSAKSI yang berarti table yang dimasuki table lain atau lawan dari table masternya, jika ada hubungan antara tblkelompok dengan tblbarang lalu tblkelompok adalah TABEL MASTER nya maka tblbarang adalah TABEL TRANSAKSI nya. Terakhir ada PRIMARY KEY dan FOREIGN KEY, pada tblbarang terdapat kolom yang menjadi kunci atau KEY yang digunakan untuk membedakan baris data yang satu dengan yang lain yaitu iDCLarang. Kolom yang berbeda antara baris data yang satu dengan yang lain disebut dengan PRIMARY KEY. Pada tblbarang juga terdapat kolom yang masuk dari table lain yaitu tblkelompok. Kolom yang masuk ke tblbarang adalah idkelompok. Kolom yang masuk dari table lain disebut dengan FOREIGN KEY.

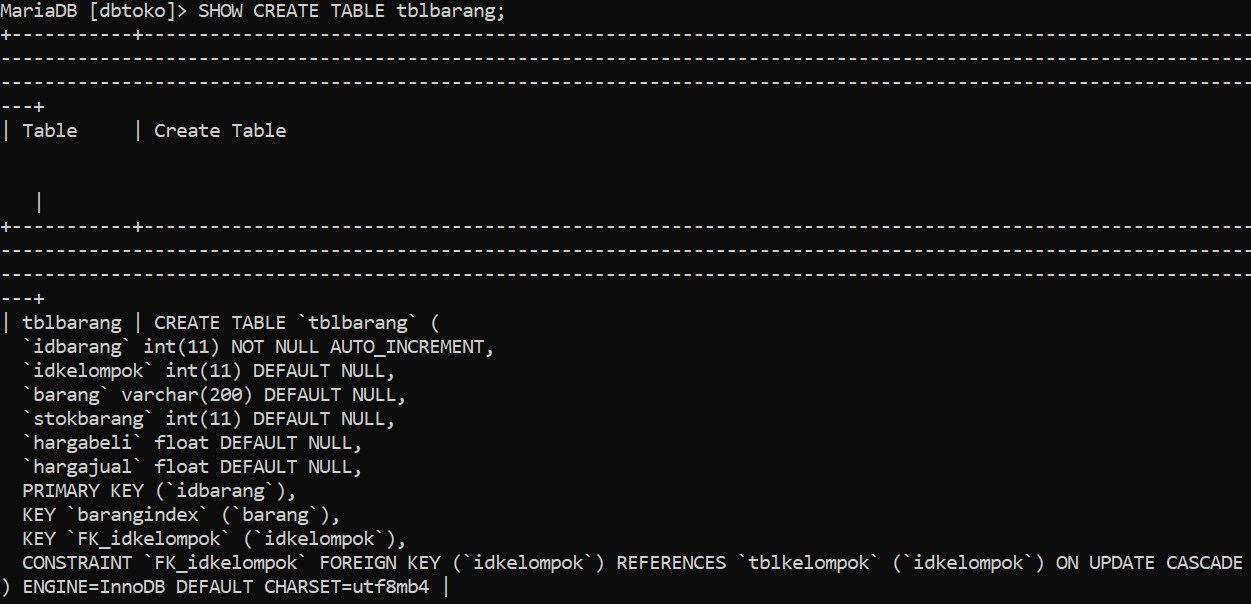
27. Relational Dependencies (Hubungan Ketergantungan), database dibuat agar pengaturan data benar sesuai dengan yang diharapkan. Relational dependencies adalah pengaturan agar table yang satu terhubung dengan table yang lain. Sehingga data yang masuk ke satu table bergantung dari table yang lain dalam bentuk ber-urutan.

28. Proses Bisnis (Alur Kerja), pada relasi dbtoko terdapat alur kerja yang sering disebut dengan PROSES BISNIS. Proses bisnis pada relasi tersebut yaitu sebelum toko dibuka pemilik akan mengelompokkan barang-barang yang akan dijual, kelompok barang tersebut akan disimpan di table kelompok.

29. Membuat relasi antar table dengan cara mengetikkan ALTER TABLE nama\_table\_detail ADD CONSTRAINT FK\_foreign\_key\_pada\_tabel\_detail FOREIGN KEY (foreign\_key) REFERENCES nama\_tabel\_master (primary\_key\_table\_master) ON UPDATE CASCADE ON DELETE RESTRICT; seperti ini



Keterangan ON UPDATE CASCADE adalah jika kolom primary key pada table master maka kolom foreign key pada table transaksi akan berubah. ON DELETE RESTRICT adalah kolom primary key di table master tidak bisa di hapus jika kolom tersebut masih digunakan pada table transaksi. ON DELETE CASCADE jika baris pada kolom master dihapus maka semua baris data yang menggunakan isi table master pada table transaksi akan terhapus. Untuk melihat hasil relasi yang telah dibuat caranya dengan mengetikkan SHOW CREATE TABLE tblbarang; seperti ini

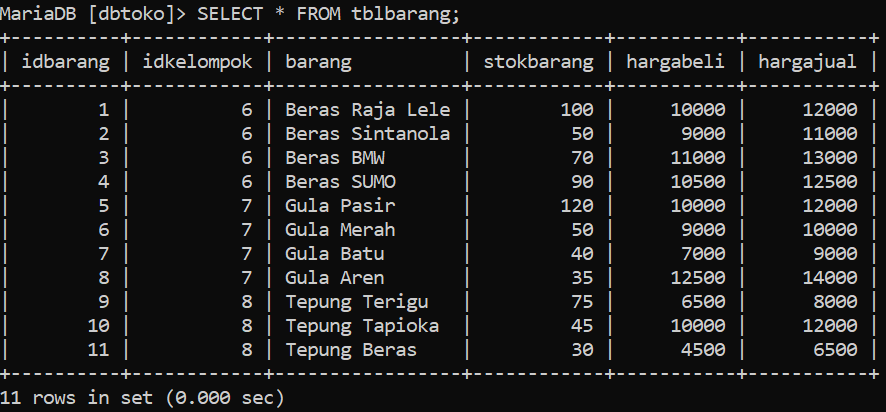


Disitu terdapat CONSTRAINT atau jika dibesarkan



Yang berarti kolom idkelompok dari tblbarang ini merupakan FOREIGN KEY yang berelasi dengan tblkelompok bagian idkelompoknya.

30. Untuk menguji hasil pembuatan relasi bisa dilakukan dengan menginsert data ke dalam tblbarang dengan memilih idkelompoknya sesuai dengan idkelompok yang tersedia di tblkelompok seperti ini

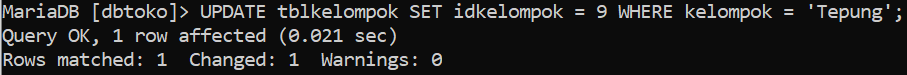


Berdasarkan table diatas, idkelompok yang digunakan ada semua dalam idkelompok di dalam tblkelompok. Apabila idkelompok yang digunakan di dalam tblbarang values nya tidak ada di dalam idkelompok yang ada di tblkelompok maka akan terjadi error ketika proses insert seperti ini (idkelompok di tblkelompok hanya berisi 3 yaitu 6,7,8)



Sesuai dengan keterangannya bahwa FOREIGN KEY dari table master yaitu idkelompok tidak memiliki values 9 yang berarti table yang direlasikan atau table transaksi tidak dapat menggunakan values lain selain dari values yang ada di idkelompok milik table master yaitu tblkelompok.

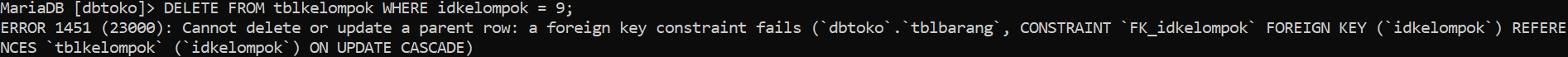
31. Pengujian untuk update dalam table master pada bagian kolom yang digunakan sebagai foreign key atau kolom yang telah direlasikan seperti ini



Yang berarti apabila yang diubah adalah foreign key atau kolom utama yang berada di table master dan dijadikan sebagai relasi untuk table lain dapat diupdate atau diubah serta otomatis mengubah kolom yang terelasikan dengan kolom tersebut.

32. Jika sebelumnya kolom foreign key dapat diupdate, berbeda halnya dengan dihapus karena kolom foreign key tidak dapat dihapus selama kolom ini sudah direlasikan di dalam table transaksi namun jika kolom ini belum digunakan atau belum direlasikan sama sekali maka kolom foreign key dapat dihapus.

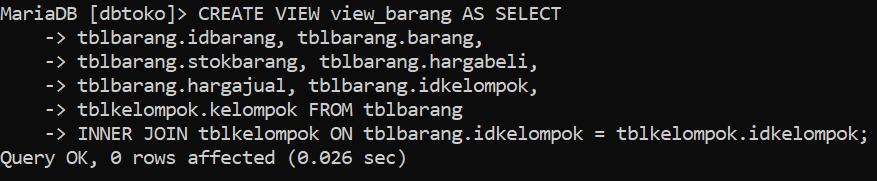
Jika foreign key sudah digunakan ada error



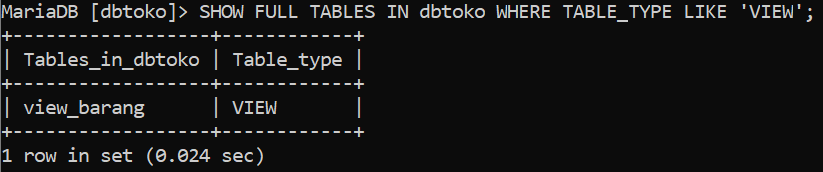
Jika belum digunakan



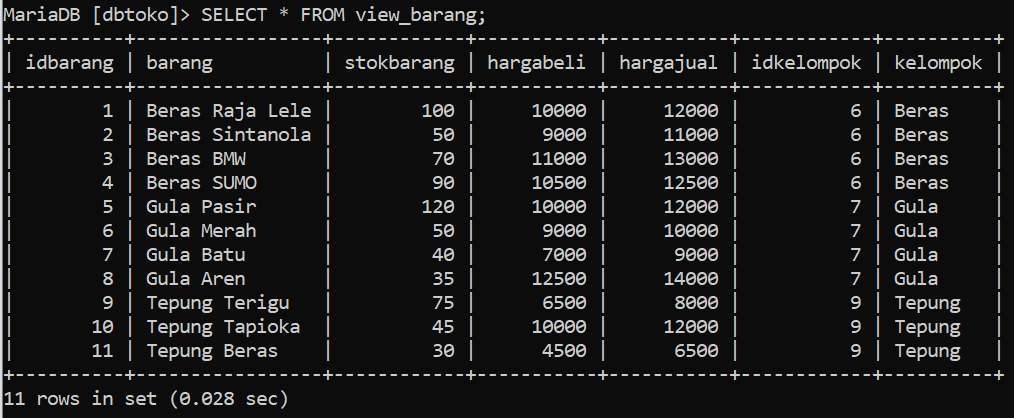
33. Membuat View yang berfungsi untuk menggabungkan table master dan table transaksi menjadi sebuah table baru agar lebih mudah dilihat dan digunakan. Caranya dengan mengetikkan CREATE VIEW nama\_view AS SELECT table\_transaksi.kolom\_transaksi, table\_master.kolom\_master FROM table\_transaksi INNER JOIN table\_master ON table\_transaksi.kolom\_transaksi = table\_master.kolom\_master; seperti ini



34. Untuk menampilkan semua view yang sudah dibuat sebelumnya caranya dengan mengetikkan SHOW FULL TABLES IN nama\_database WHERE TABLE TYPE LIKE ‘VIEW’ seperti ini



Sedangkan untuk melihat isi dari sebuah view caranya dengan mengetikkan SELECT \* FROM nama\_view seperti ini



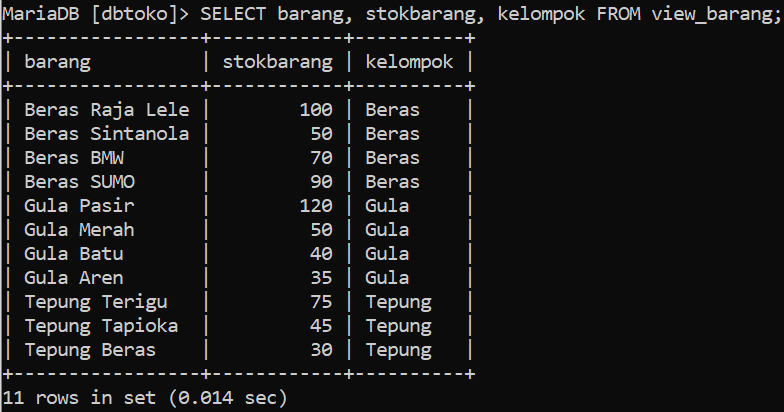
35. Untuk menghapus sebuah view caranya dengan mengetikkan DROP VIEW nama\_view; seperti ini



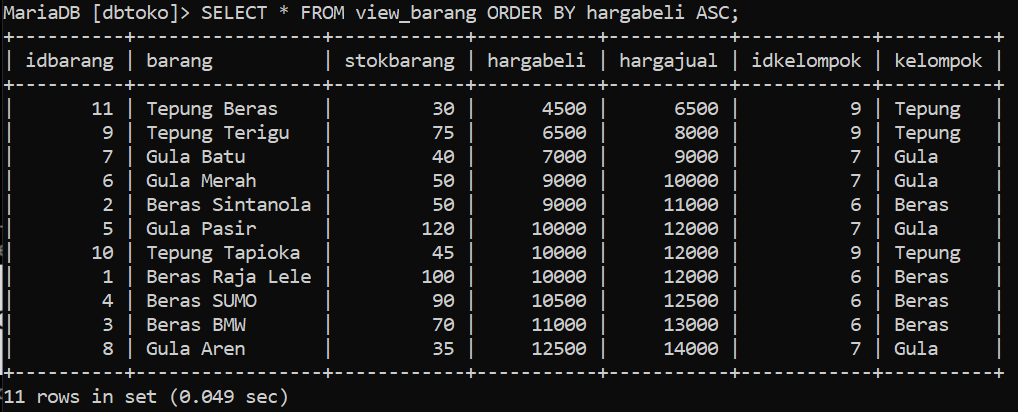
Dan ketika dicek lagi maka view nya sudah tidak ada



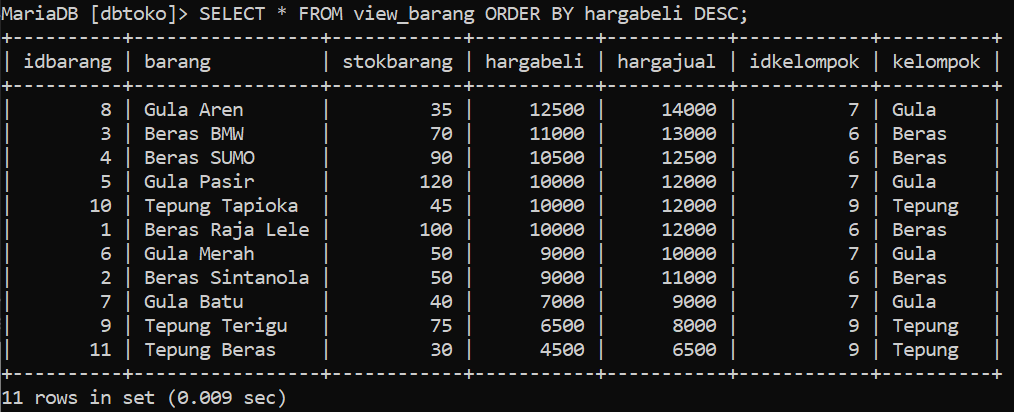
36. Jika sebelumnya untuk menampilkan semua kolom dalam table view menggunakan SELECT \* FROM nama\_table\_atau\_nama\_view; maka untuk menampilkan beberapa kolom caranya dengan mengetikkan SELECT nama\_kolom, nama\_kolom, FROM nama\_table\_atau\_nama\_view; seperti ini



37. SELECT ORDER adalah cara untuk menampilkan data dalam urutan naik atau turun, jika NAIK menggunakan ASC sedangkan jika TURUN menggunakan DESC. Untuk mengurutkan data ASC bisa ketik SELECT \* FROM nama\_tabel\_atau\_view ORDER BY nama\_kolom ASC; dan hasilnya

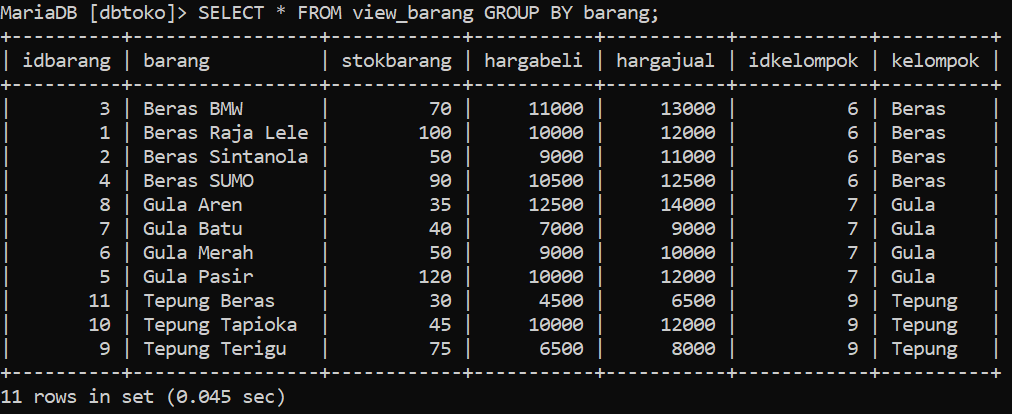


Yang berarti urutkan data yang ada di view barang secara naik atau dari yang terkecil berdasarkan kolom hargabelinya. Sedangkan untuk mengurutkan data DESC bisa ketik SELECT \* FROM nama\_table\_atau\_view ORDER BY nama\_kolom DESC; seperti ini

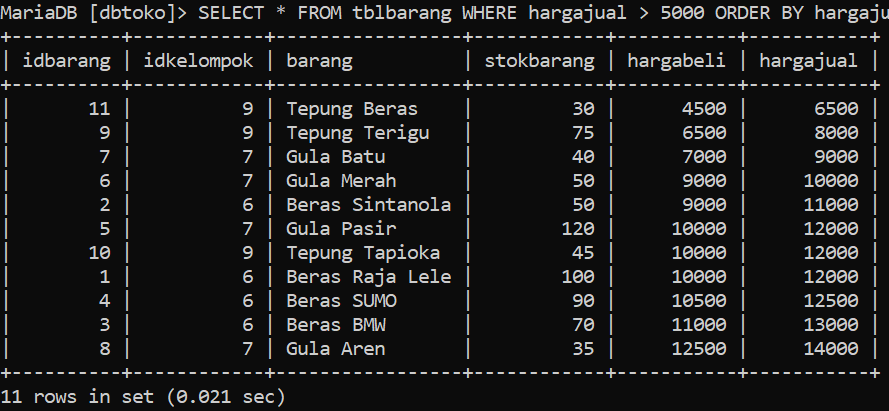


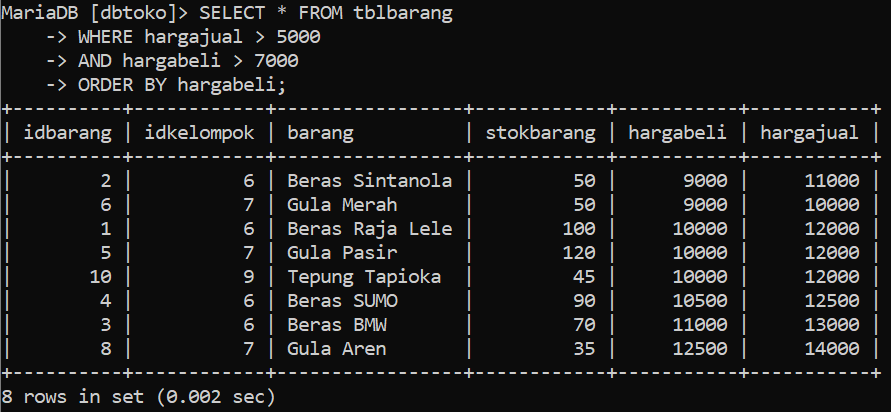
Yang berarti urutkan data yang ada di view barang secara turun atau dari yang terbesar berdasarkan kolom hargabelinya.

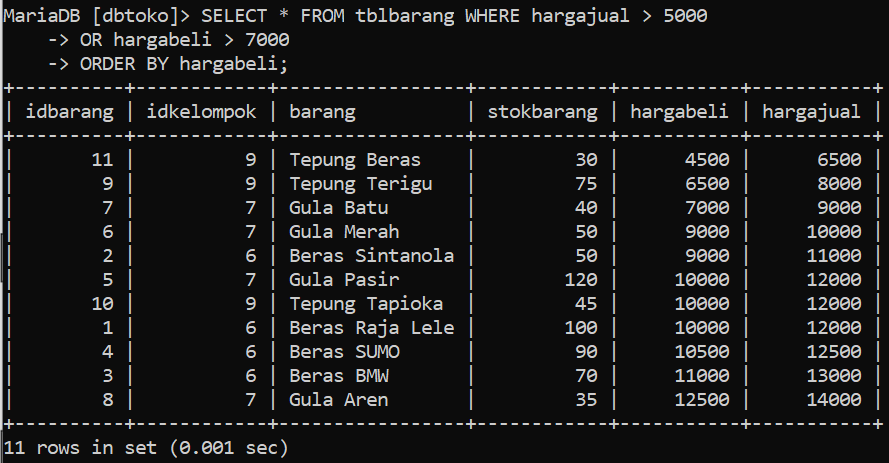
38. SELECT GROUP digunakan untuk mengelompokkan data sesuai dengan kelompok dari kolom yang dipilih. Untuk melakukan select group bisa ketikkan SELECT \* FROM nama\_table\_atau\_view GROUP BY nama\_kolom; seperti ini



39. Pengujian WHERE Pengujian pada SQL yang bisa diletakkan dibelakang SELECT, UPDATE, dan DELETE dengan Pengujian akan menjalankan SELECT, DELETE, UPDATE jika kondisi yang diuji terpenuhi Operator pembanding (=, >, <, <>, >=, <=, LIKE), Operator Logika (AND, OR), dan SELECT. Contoh Pengujian sebagai berikut SELECT \* FROM nama\_table\_atau\_view WHERE pengujian ORDER BY nama\_kolom; seperti ini

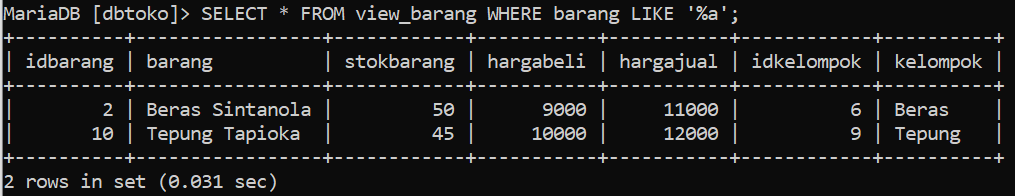




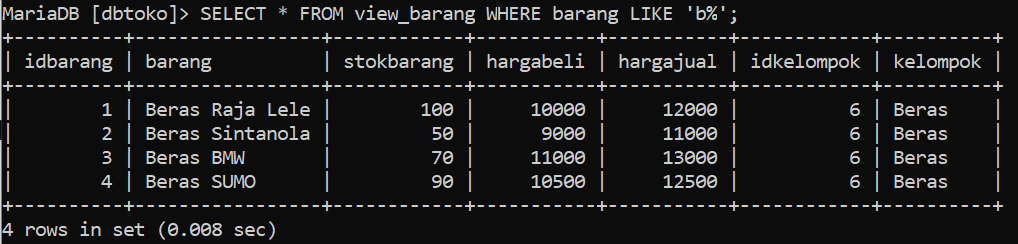


SELECT \* FROM nama\_table\_atau\_view WHERE nama\_kolom LIKE ‘%\_apapun’; dengan keterangan % adalah sebutan untuk sembarang atau apapun,

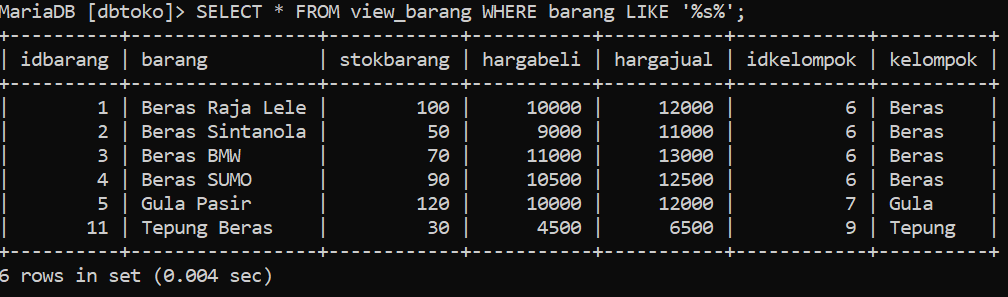
contoh %a artinya awalan sembarang yang penting akhirannya a



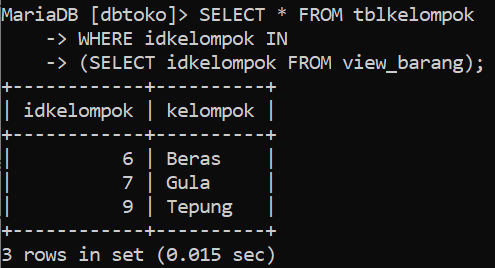
Contoh b% artinya awalan b akhiran sembarang

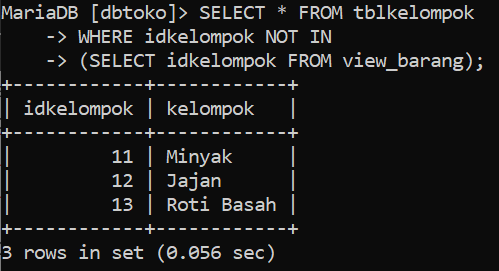


Contoh terakhir %s% artinya awalan sembarang dan artinya sembarang yang tengahnya huruf s

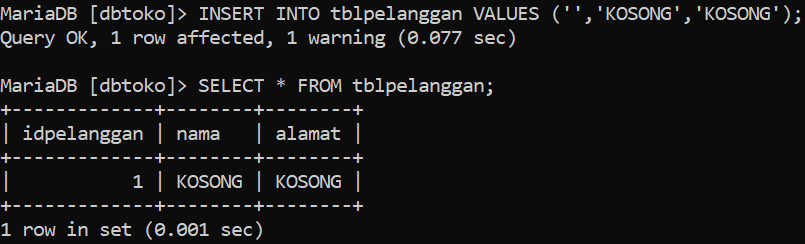


40. SUBQUERY adalah SELECT yang ada di dalam SELECT, untuk menggunakannya bisa dengan mengetik SELECT \* FROM nama\_table WHERE nama\_kolom IN (SELECT nama\_kolom FROM nama\_table); seperti ini contohnya

  
yang berarti perintah tersebut bertujuan untuk menampilkan data yang ada di dalam tblkelompok namun data yang dipilih adalah yang memiliki idkelompok di dalam view\_barang, jadi jika idkelompok dari sebuah data dalam tblkelompok tidak ada di dalam view\_barang maka data tidak akan ditampilkan. Jika SUBQUERY sebelumnya menggunakan IN ada juga NOT IN yang berfungsi sebaliknya seperti ini



41. Dummy Data adalah data yang tidak sebenarnya. Data ini diperlukan jika data sebenarnya tidak bisa diperoleh. Dummy data biasa digunakan selama pengujian database. Contohnya pada nama pelanggan yang umumnya pelanggan mempunyai nama dan jika berbelanja di supermarket atau minimarket tidak akan ditanya tentang nama pelanggan tersebut. Namun hanya ditanya yang akan dibeli oleh pelanggan. Karena itu pada tblpelanggan aplikasi akan diisi dengan dummy data dengan nama pelanggan sebagai berikut. Contohnya insert data ke dalam tblpelanggan seperti ini



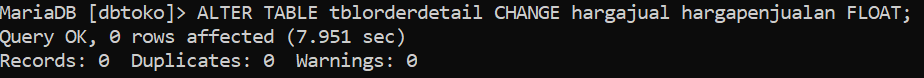
Dan nantinya data ini akan digunakan sebagai dummy data.

42. Konsep trigger adalah perintah INSERT, UPDATE, DELETE, FUNCTION, PROCEDURE yang ditanam pada MySQL yang akan DIJALANKAN pada kejadian berikut:

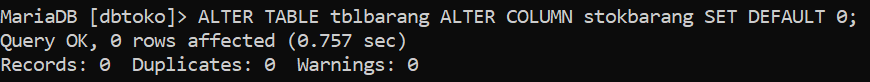
* AFTER INSERT (setelah INSERT) pada table yang dimaksud
* BEFORE INSERT (sebelum INSERT) pada table yang dimaksud
* AFTER DELETE (setelah DELETE) pada table yang dimaksud
* BEFORE DELETE (sebelum DELETE) pada table yang dimaksud
* AFTER UPDATE (setelah UPDATE) pada table yang dimaksud
* BEFORE UPDATE (sebelum UPDATE) pada table yang dimaksud

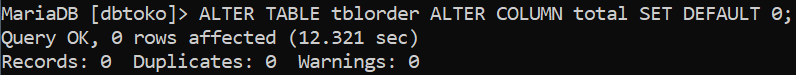
Dalam proses yang terjadi di dbtoko terdapat penyusunan proses bisnis seperti kolom stokbarang pada tblbarang akan berkurang jika terjadi insert data pada tblorderdetail. Kemudian kolom stokbarang pada tblbarang akan bertambah jika terjadi delete data pada tblorderdetail. Kolom tota pada tblorder akan bertambah jika terjadi insert data pada tblorderdetail. Dan kolom total tblorder akan berkurang jika terjadi delete pada tblorderdetail.

Sebelum membuat trigger pastikan tidak ada nama kolom yang sama pada setiap table. Jika melihat table-table yang ada di dbtoko terdapat nama kolom yang sama yaitu hargajual pada tblbarang dan hargajual pada tblorderdetail. Agar berbeda, harus ada yang diubah misalnya hargajual dari tblorderdetail menjadi hargapenjualan seperti ini.



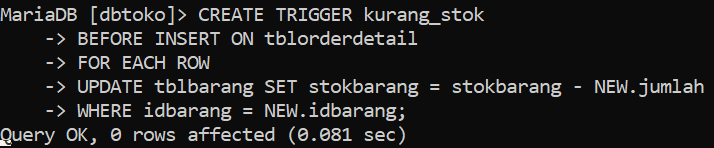
Melakukan pemeriksaan pada kolom yang akan terkena dampak dari proses trigger. Jika yang diproses oleh trigger adalah operasi matematika (penjumlahan, pengurangan, perkalian, pembagian). Pastikan SET DEFAULT VALUE pada kolom table tersebut diisi dengan ANGKA NOL. Jika belum diisi angka NOL lakukan ALTER untuk merubah kolom tersebut. Contohnya stokbarang dan total





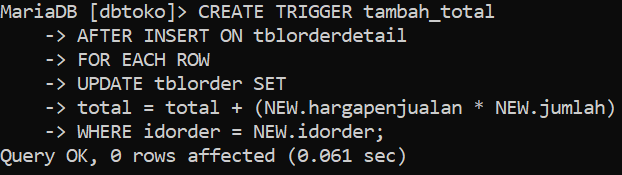
43.Membuat Trigger setelah semua konsep, proses bisnis, dan pemeriksaan table dilakukan. Untuk membuat sebuah table bisa ketikkan CREATE TRIGGER nama\_trigger AFTER INSERT ON nama\_table\_yang\_dipasang trigger FOR EACH ROW ketik\_perintah\_trigger\_disini\_diakhiri\_dengan\_titik\_koma;

Contoh pembuatan trigger yang pertama yaitu kurang\_stok



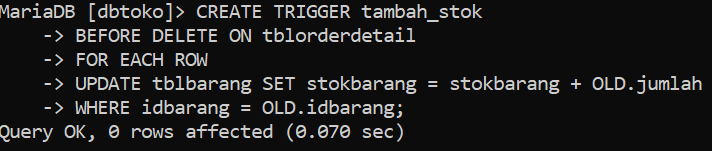
Trigger kurang\_stok digunakan untuk mengurangi stokbarang pada tblbarang pada saat BESFORE INSERT (sebelum insert) pada table (tblorderdetail). NEW.jumlah adalah data atau nilai yang akan masuk pada kolom jumlah di tblotderdetail

Yang kedua yaitu tambah\_total



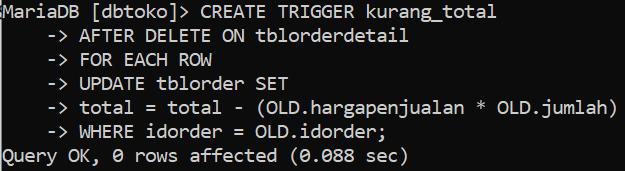
Trigger tambah\_total digunakan untuk menambah nilai total pada table tblorder saat AFTER INSERT (sesudah insert) pada tblorderdetail. (NEW.hargapenjualan \* NEW.jumlah) adalah data jumlah yang akan dijual dan harga penjualan yang akan masuk pada kolom jumlah dan kolom hargapenjualan di tblorderdetail.

Contoh ketiga yaitu tambah stok



Trigger tambah\_stok digunakan untuk menambah stokbarang pada tblbarang pada saat BEFORE DELETE (sebelum delete) pada tblorderdetail. OLD.jumlah adalah data yang sudah ada pada kolom jumlah di tblorderdetail.

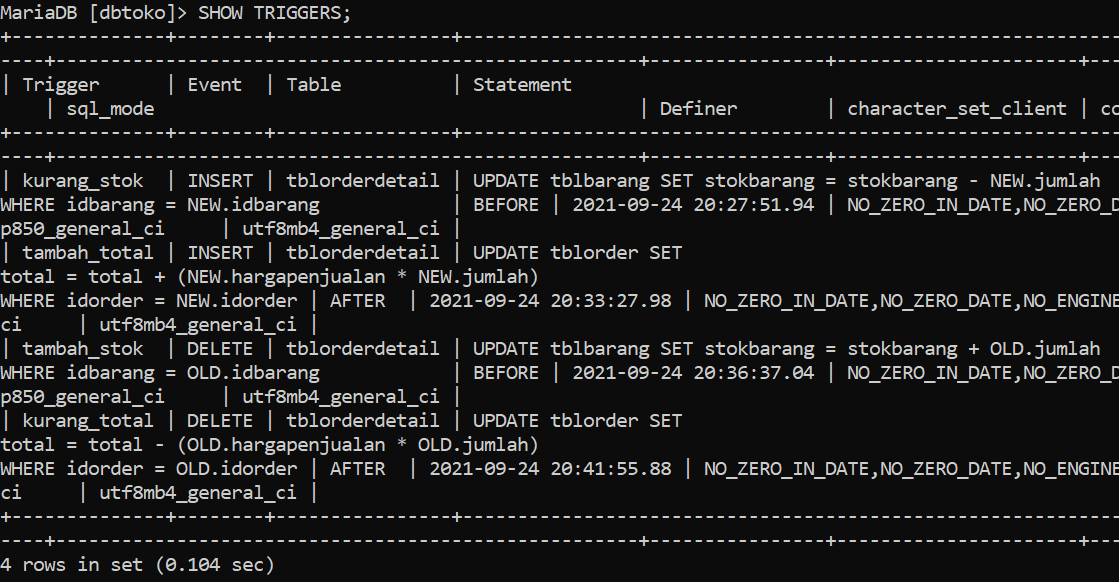
Contoh keempat yaitu kurang total



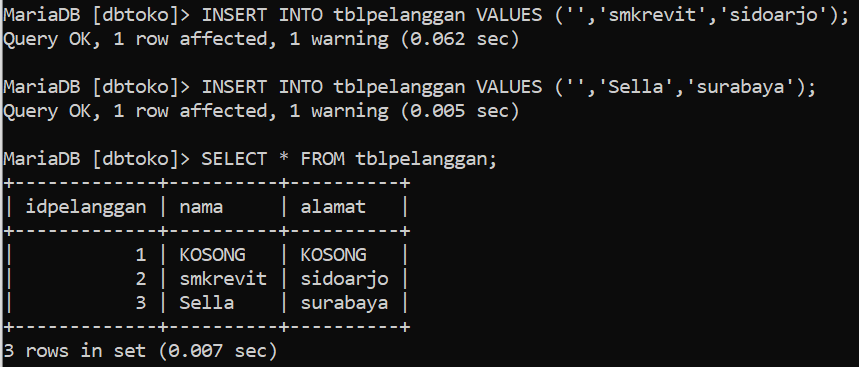
Trigger kurang\_total digunakan untuk mengurangi nilai total pada tblorder saat AFTER DELETE (sesudah delete) pada tblorderdetail.

(OLD.hargapenjualan \* OLD.jumlah) adalah data jumlah dan harga penjualan yang sidah ada pada tblorderdetail.

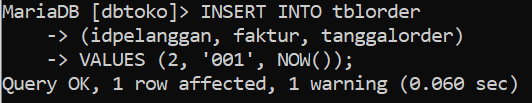
44. Untuk menampilkan sebuah trigger dapat dengan mengetikkan SHOW TRIGGERS; seperti ini



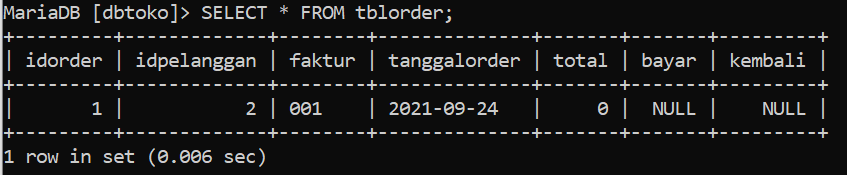
45. Langkah selanjutnya adalah melakukan pengujian trigger, yang pertama dilakukan yaitu persiapan dengan mengecek data yang ada di tblbarang dan tblpelanggan. Karena dalam tblpelanggan hanya memiliki 1 data, maka tambahkan beberapa data lagi seperti ini



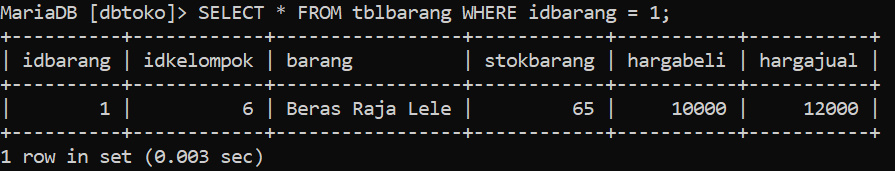
Kedua yang harus dilakukan yaitu pengujian insert pada table (tblorderdetail) dengan membuat order atas nama pelanggan (smkrevit) dengan idpelanggan = 2 pada tblorder dengan cara memasukkan data sebagai berikut



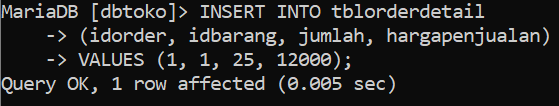
Jika dicek



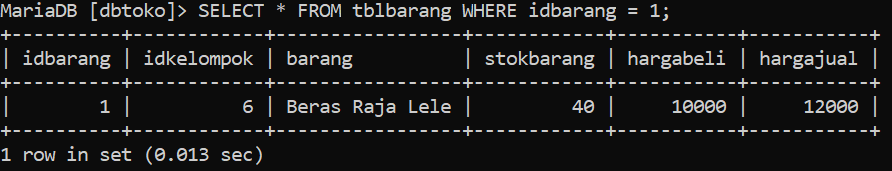
Kemudian buat insert data pada tblorderdetail dengan menggunakan idorder = 1 sesuai yang ada di tblorder dengan iDCLrang = 1 pada tblbarang. Jika sebelumnya stok barang adalah 65



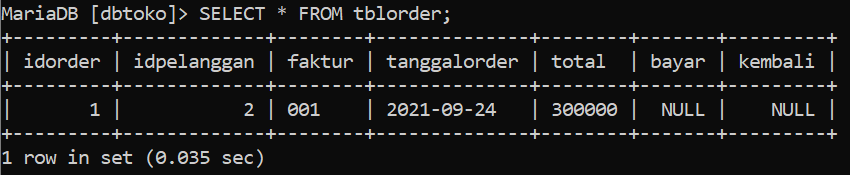
Kemudian dalam idorderdetail ditambahkan sebuah data baru dan mengurangi stokbarang idbarang 1 seperti ini



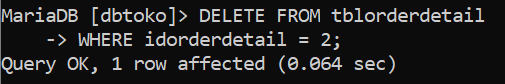
Maka ketika dicek lagi tblbarangnya, stokbarang idbarang 1 akan berkurang sebanyak 25 dan menjadi 40 seperti ini



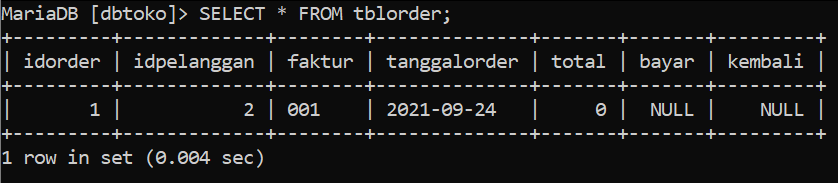
Dan dalam tblordernya akan tertera total dari orderan tadi



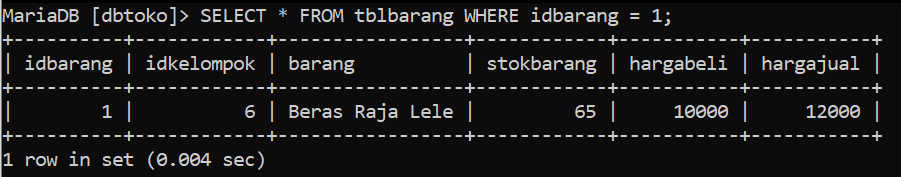
46. Pengujian kedua yaitu delete pada table tblorderdetail seperti ini



Karena idorderdetail ke 2 sudah dihapus, maka ketika di cek di dalam tblorder bagian total akan kembali ke 0



Begitu juga dalam tblbarang ketika tblorderdetail dihapus maka stok barang yang ada di tblbarang kembali ke stok barang semula



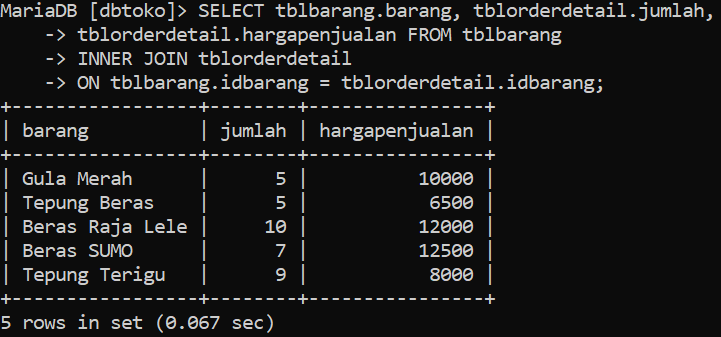
47. Untuk menghapus trigger caranya dengan mengetikkan DROP TRIGGER nama\_trigger; seperti ini



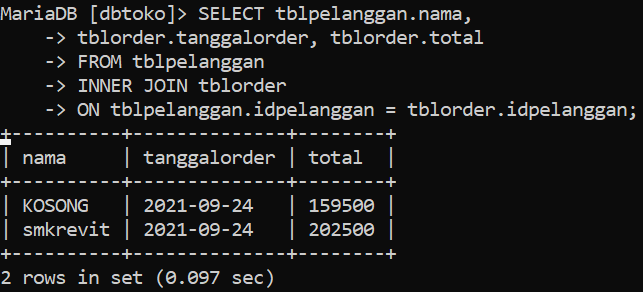
Dengan begitu ketika dicek lagi dalam data semua trigger, trigger kurang\_total sudah tidak ada karena terhapus.

48. Adanya JOIN (Penggabungan Table) bertujuan untuk memudahkan pembacaan pada table apalagi ada beberapa kolom yang sulit dipahami.

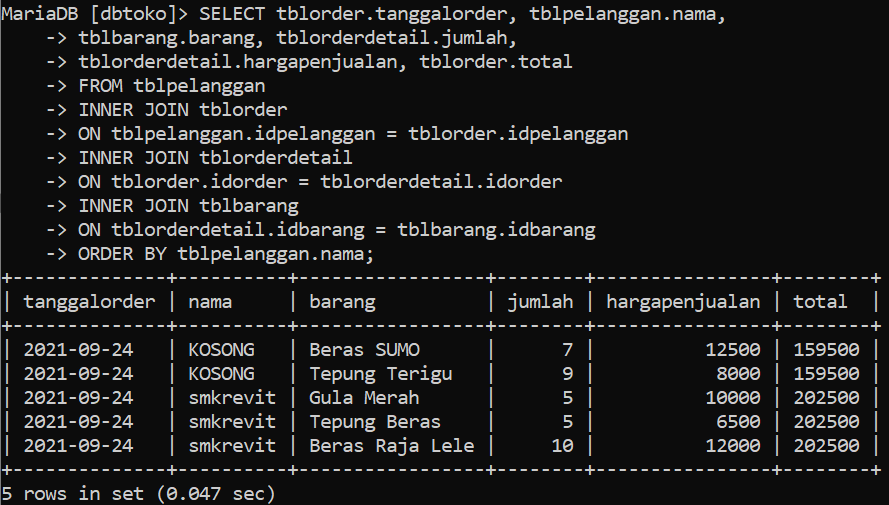
49. Inner Join (mengambil bagian yang ada di table master dan table transaksi) misalkan antara tblbarang dan tblorderdetail. Inner Join juga digunakan untuk mengetahui barang apa yang sudah laku dengan mengetikkan SELECT Table\_master\_kolom\_master, table\_transaksi.kolom\_transaksi FROM table\_master INNER JOIN table\_transaksi ON table\_master.kolom\_master = table\_transaksi.kolom\_transaksi; seperti ini



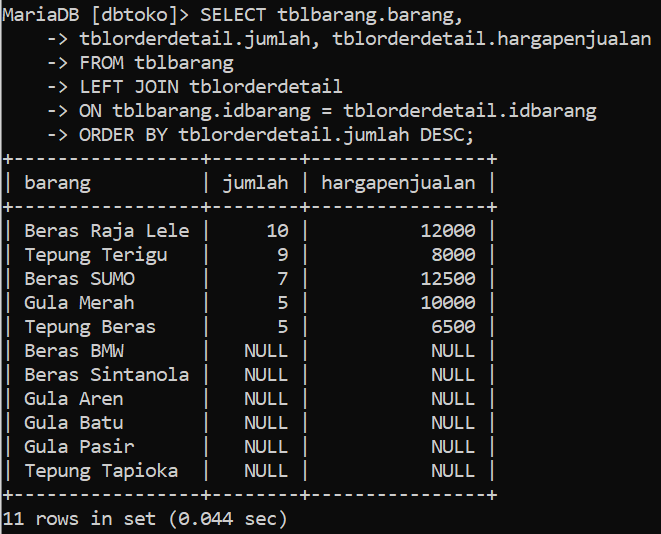
Kemudian ada INNER JOIN untuk mengetahui pelanggan yang melakukan ORDER seperti ini



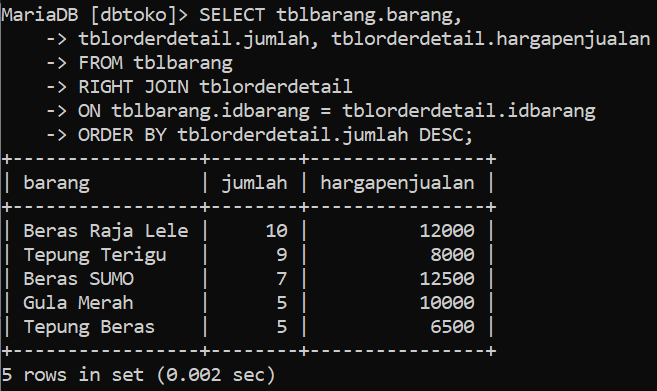
50. Ada juga Inner Join Banyak Table, dengan menggunakanINNER JOIN banyak table bisa diketahui barang yang dibeli pelanggan dan total pembelian oleh pelanggan seperti ini



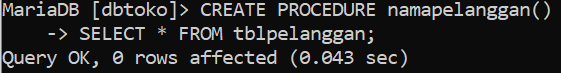
51. Left Join (Menampilkan Table Master), left join digunakan untuk menampilkan semua yang ada di table master dan mengambil Sebagian yang ada di table transaksi seperti ini



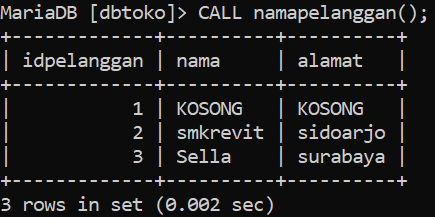
52. RIGHT JOIN (Menampilkan Table Transaksi), Right Join digunakan untuk menampilkan semua yang ada di table transaksi dan sebagian yang ada di table master seperti ini



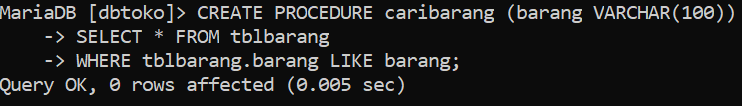
53. Store Procedure adalah blok program yang diletakkan pada MySQL. Blok yang sudah dibuat bisa dipanggil jika diperlukan. Procedure tidak memiliki return, contohnya procedure tanpa parameter dengan mengetikkan CREATE PROCEDURE nama\_procedure isi\_procedure;



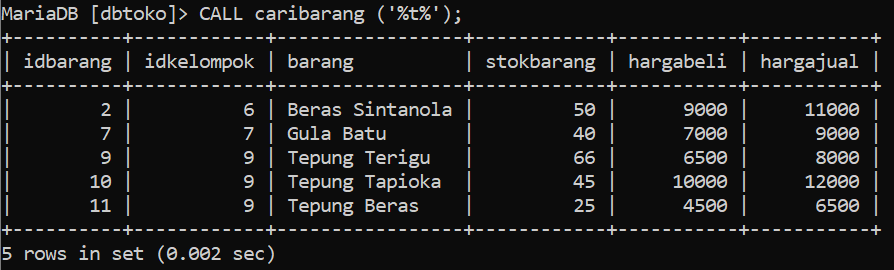
Dan untuk memanggilnya menggunakan CALL nama\_procedure();



Sedangkan procedure dengan parameter seperti ini

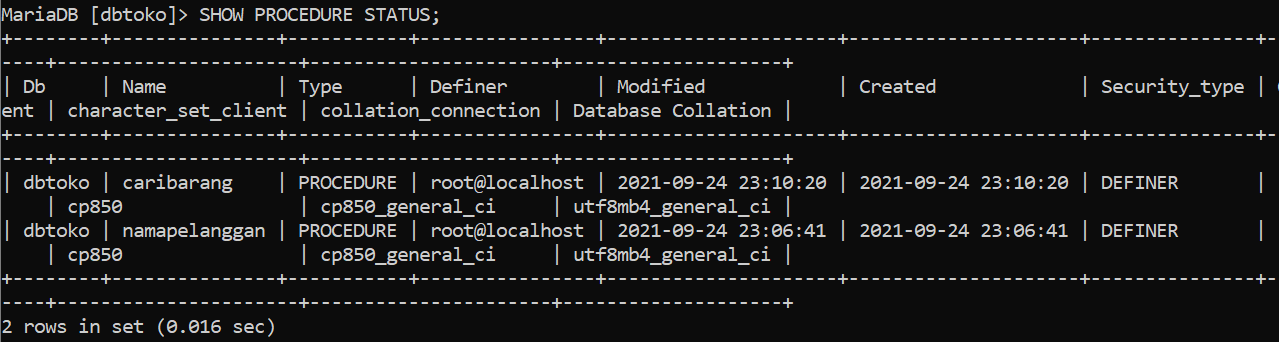


Untuk memanggil procedure yang sudah dibuat dengan parameter yang dimasukkan [‘%t%’] yang menunjukkan nama barang yang mengandung [t]. contohnya



Dengan menggunakan procedure, kita dapat membuat semua QUERY atau SELECT di dalam procedure sehingga memudahkan dalam pembuatan aplikasi.

54. Untuk menampilkan semua procedure karena MySQL dapat menampilkan semua isi procedure dengan mengetikkan SHOW PROCEDURE STATUS; seperti ini

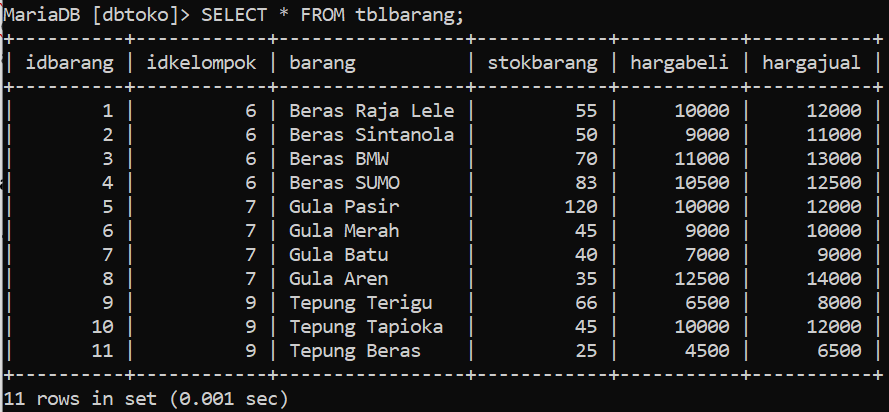


55. Untuk menghapus store procedure caranya sebagai berikut

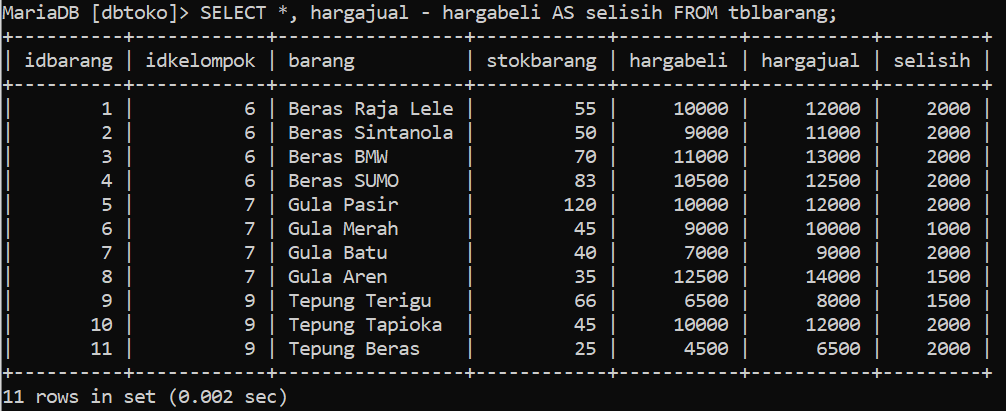


Dengan begitu ketika dicek di SHOW PROCEDURE STATUS sudah terhapus dan tidak ada.

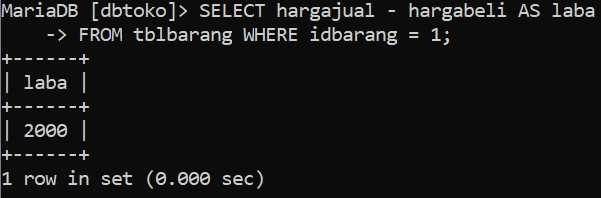
56. FUNCTION adalah blok program yang disimpan di MySQL yang bisa menerima INPUT atau PARAMETER dan memiliki return. Untuk membuat sebuah function, yang pertama harus dilakukan yaitu menyipakan isi function terlebih dahulu agar hasil dari function sesuai dengan yang diharapkan. Contohnya membuat function untuk menampilkan selisi hargajual dan hargabeli pada tblbarang



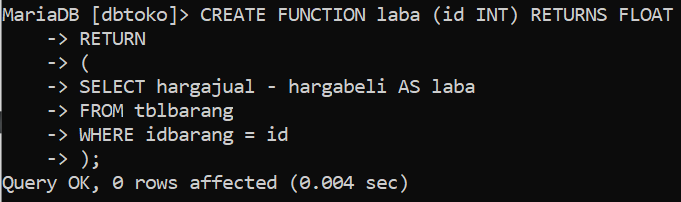
Pada table diatas akan ditambahkan 1 kolom lagi sebagai selisih dari hargajual dikurangi dengan hargabeli. Caranya seperti ini



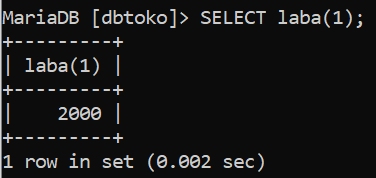
Selisih hargajual dan hargabeli ini disebut dengan laba. Kemudian diambil selisih perbarang untuk dihitung sebagai laba. Sebagai contoh diambil satu barang dengan [iDCLrang = 1] maka selisihnya akan ditampilkan seperti ini



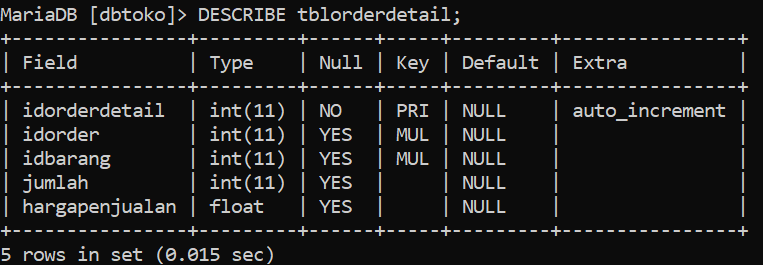
Jika persiapan sudah selesai, maka pembuatan function dapat dilakukan dengan mengetikkan CREATE FUNCTION nama\_function (parameter TIPE DATA) RETURNS TIPE DATA RETURN (isi\_function); seperti ini



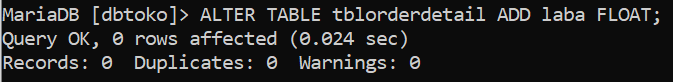
57. Untuk memanggil sebuah function caranya dengan mengetikkan SELECT FUNCTION nama\_function(parameter); seperti ini



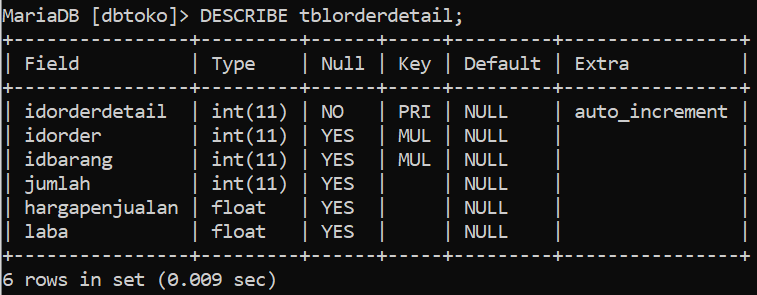
58. Langkah selanjutnya adalah menguji dan menggunakan function yang telah dibuat, pada bagian ini bisa buka atau cek tblorderdetail terlebih dahulu



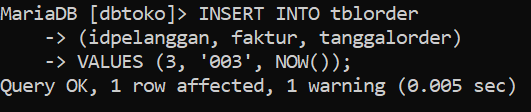
Kemudian tambahkan sebuah kolom baru dengan nama laba



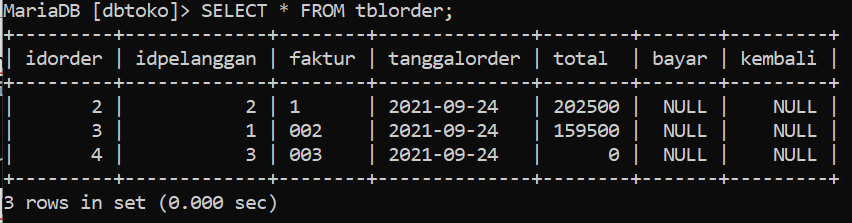
Dan periksa kembali



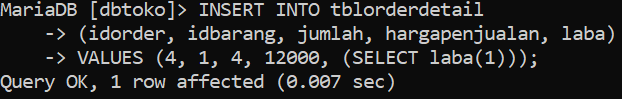
Kemudian lakukan proses order dengan melakukan insert pada tblorder



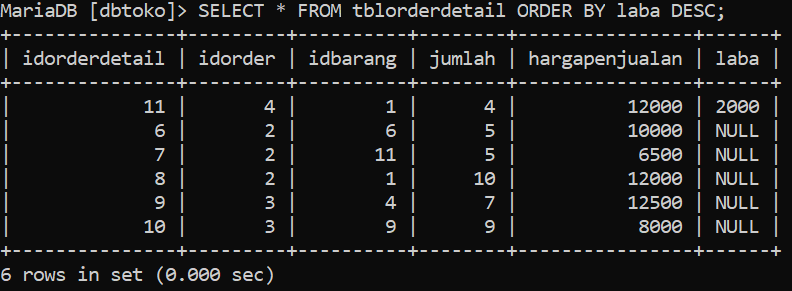
Periksa tblorder



Buat insert pada tblorderdetail

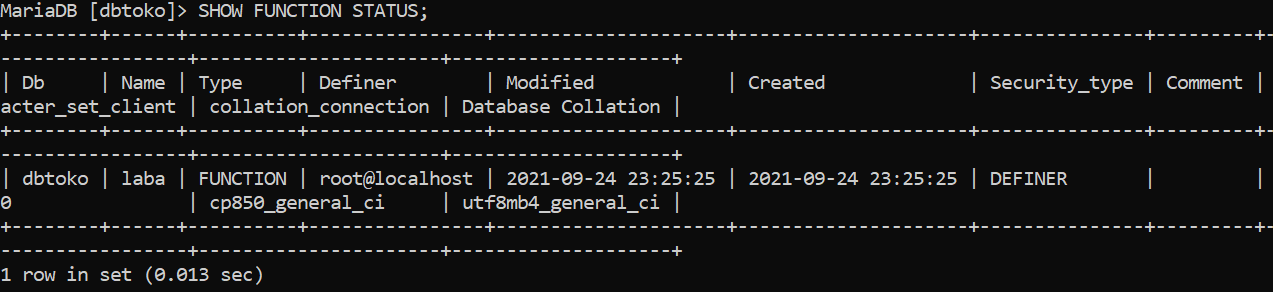


Pada saat insert kolom laba akan diisi dengan function yang menghitung hargajual dikurangi hargabeli dimana iDCLrang dimasukkan sebagai parameter input pada function. Di dalam tblorderdetailnya aka nada laba



Apabila selisih yang ada sesuai dengan selisih dari tblbarang, maka function berhasil digunakan.

59. Untuk menampilkan semua function caranya dengan mengetikkan SHOW FUNCTION STATUS; seperti ini

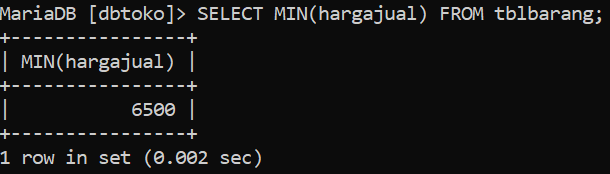


Dan untuk menghapus function caranya dengan mengetikkan DROP FUNCTION seperti ini

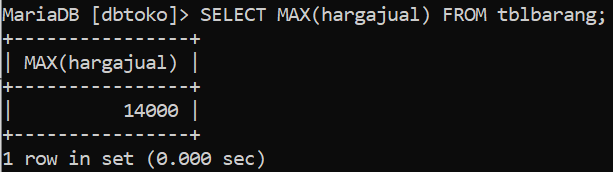


60. Select Aggregate adalah select yang menampilkan function bawaan dari MySQL. Ada beberapa select aggregate yang akan dipelajari yaitu MIN, MAX, SUM, AVG, COUNT, dan COUNT(\*). Untuk menggunakannya bisa mengetikkan SELECT function\_aggregate(nama\_kolom); dengan ketentuan antara nama aggregate dan kurung tidak boleh ada spasi. Contohnya

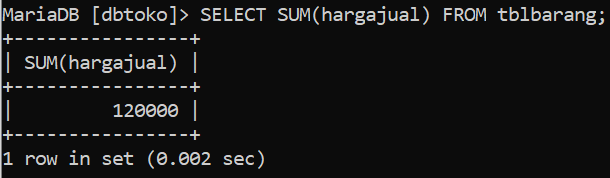
SELECT MIN



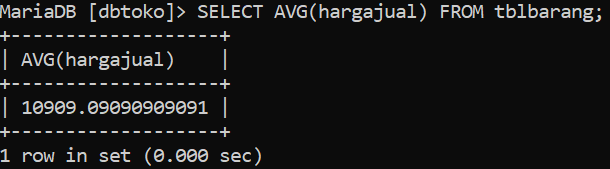
SEELCT MAX



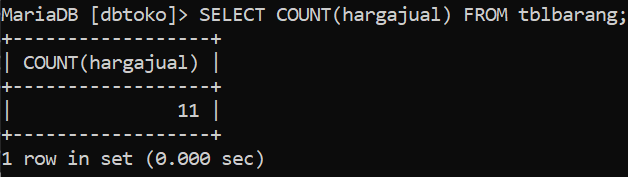
SELECT SUM



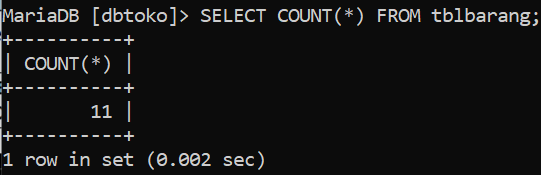
SELECT AVG



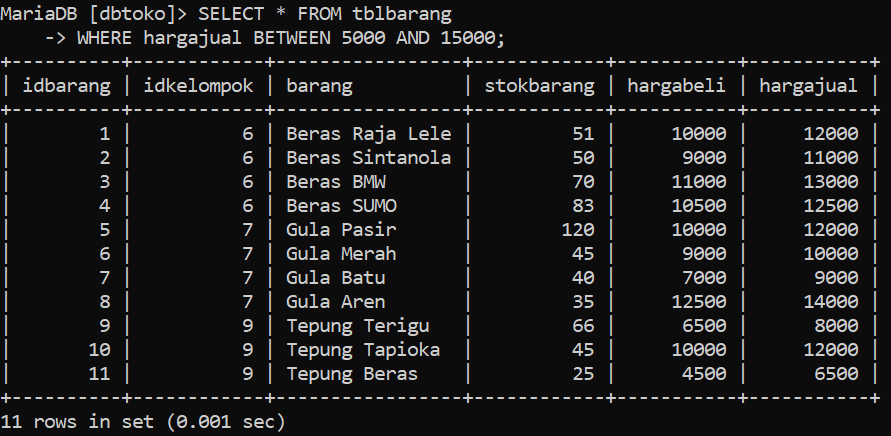
SELECT COUNT



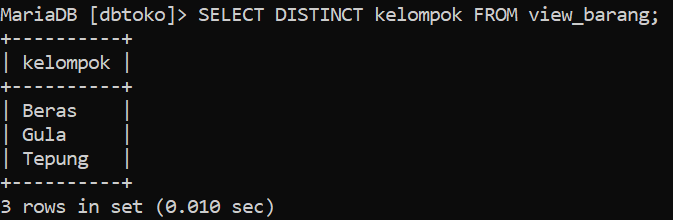
SELECT COUNT(\*)



61. SELECT BETWEEN (Select antara dua nilai) yang digunakan untuk menampilkan data antara dua nilai terendah dan tertinggi, untuk menggunakannya dengan cara mengetikkan SELECT nama\_kolom\_atau\_\* FROM nama\_table WHERE nama\_kolom BETWEEN awal AND akhir seperti ini



62. SELECT DISTINCT (Menampilkan data yang sam ahanya satu kali), penggunaannya dengan mengetikkan SELECT DISTINCT nama\_kolom FROM nama\_table\_view

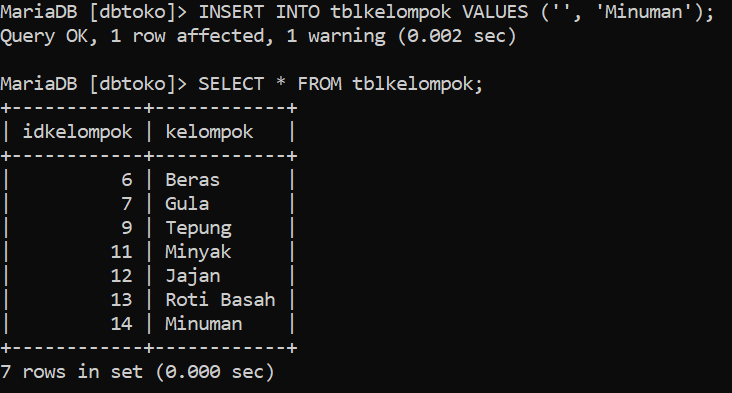


63. START TRANSACTION, COMMIT, DAN ROLLBACK. Setiap petintah di MySQL yang masuk kelompok (INSERT, UPDATE, DELETE, SELECT) bisa dimasukkan ke dalam perintah START TRANSACTION, COMMIT, DAN ROLLBACK. Perintah yang dimulai dengan START TRANSACTION bisa dilakukan ROLLBACK (undo atau pembatalan perintah). COMMIT digunakan agar perintah yang dijalankan tidak bisa di rollback(undo).

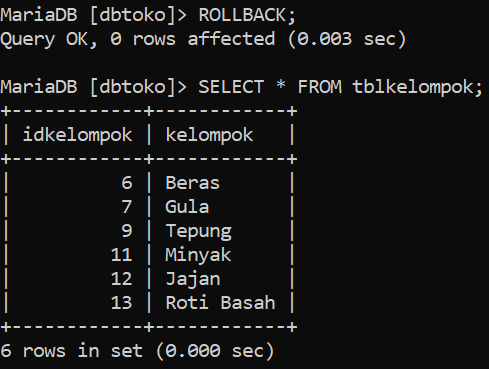
Contoh di dalam tblkelompok dibuka kemudian start transaction seperti ini



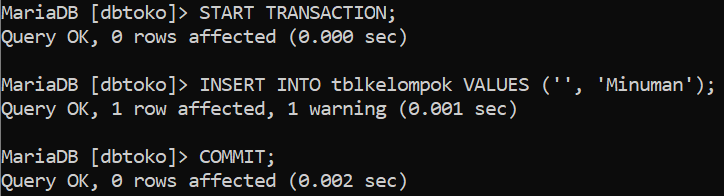
Karena transaction sudah dimulai atau distart, kemudian insert sebuah data ke dalam tblkelompok dan hasilnya akan berada di paling bawah



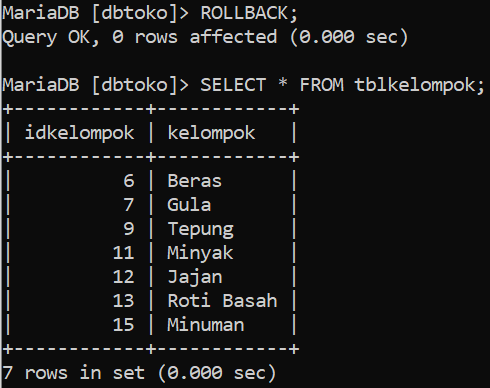
Apabila rollback digunakan dalam proses ini, maka data yang baru saja dimasukkan atau data apapun yang dimasukkan selama transaction berjalan atau sudah distart dapat dirollback atau mengundo data seperti ini



Kemudian untuk proses commit dilakukan seperti tadi yaitu START TRANSACTION dulu kemudian insert data ke dalam tblkelompok lalu commit



Data yang sudah dicommit tidak akan dapat dirollback atau diundo, buktinya seperti ini



Meskipun data yang diinsertkan tadi sudah dirollback atau diundo, namun data tersebut tetap ada di dalam tblkelompok.

64. Kesimpulan START TRANSACTION, COMMIT, dan ROLLBACK. Transaksi adalah perintah (INSERT, UPDATE, DELETE, SELECT). Transaksi yang bisa di ROLLBACK (Undo) adalah transaksi yang dimulai dengan START TRANSACTION dan BELUM DI AKHIRI DENGAN COMMIT. Transaksi yang sudah di COMMIT tidak bisa di ROLLBACK.

Penggunaan START TRANSACTION, COMMIT, dan ROLLBACK ketika database yang dibuat sudah digunakan oleh orang yang membeli program maka database tersebut akan terisi dengan data VALID (BENAR) sesuai dengan yang dimiliki oleh orang tersebut. Kemudian database tersebut dikembalikan untuk proses modifikasi atau penambahan sesuai dengan DATA DUMMY (data coba-coba) karena sedang melakukan pengujian database. Agar data DUMMY tidak tercampur dengan data VALID bisa gunakan perintah START TRANSACTION sebelum melakukan proses TRANSAKSI. Jika pengujian sudah selesai lakukan ROLLBACK dan jangan pernah lakukan commit jika data yang dimasukkan adalah data DUMMY.

**Saya Belum Mengerti**

1.