|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Belajar Mysql Dari Nol | 80 |

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

1. Database merupakan inti daripada sebuah aplikasi (ibarat seperti jantung dan rumah dari suatu data). Jadi awal dari pembuatan aplikasi ialah dimulai dari pembuatan database terlebih dahulu (baik tidaknya database ini nantinya akan berpengaruh pada aplikasinya).

2. Data, ialah suatu bagian terkecil yang digunakan untuk menyusun suatu aplikasi.

3. Bisa mendownload aplikasi XAMPP dan menjalankannya sesuai dengan yang ada di buku.

4. Command Prompt (CMD), digunakan untuk pengujian MySQL yang telah di install sebelumnya.

5. Bisa membuka dan MySQL dengan cmd.

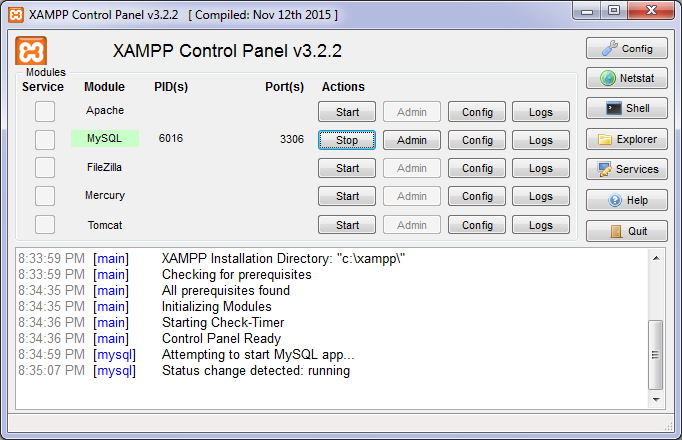
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_eyTEXkPMII.png

6. Bisa menjalankan MySQL dengan mengetikkan mysql –u root –p.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_9o9G1b5sXo.png

7. Pada bagian Enter Password: akan kosong dan tidak terisi apa-apa karena sudah default atau bawaan dari XAMPP nya, jadi langsung di klik enter saja.

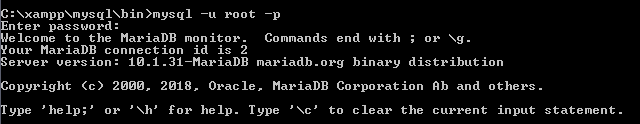
8. Tidak lupa untuk menyalakan MySQL di dalam control panel dari xampp nya terlebih dahulu. Bila tidak, maka akan terjadi error pada saat menekan dan klik enter di bagian Enter Password: nya.



(Bila tidak dinyalakan maka akan terjadi error seperti ini).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_9kB8N1dr6r.png

9. Berhasil menjalankan MySQL yang ada di dalam cmd setelah mengklik enter pada bagian Enter Password: dan menyalakan MySQL di dalam control panel dari xampp nya.



10. Bila ingin keluar dari MySQL nya, pada bagian bawah dari Type ‘help’ …. Input statement, terdapat tulisan MariaDB [<none>]>. Setelah tanda > terakhirnya, bisa diketikkan perintah quit untuk keluar dari MySQL nya dan akan memunculkan tulisan Bye.

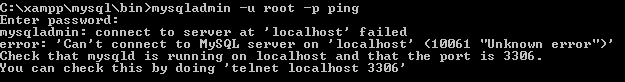
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_nvg4q0gUpN.png

11. Bisa memeriksa apakah suatu MySQL yang digunakan sudah berjalan atau tidak. Dengan cara mengetikkan mysqladmin –u root –p ping, yang kemudian akan memunculkan Enter Password: kembali, dan klik enter. Pastikan sudah keluar dari MySQL nya terlebih dahulu (dengan perintah quit).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ftsE7HMMsJ.png

(sudah bisa berjalan bila bertuliskan mysqld is alive).

Bila MySQL yang ada di dalam control panel dari XAMPP nya belum dinyalakan ataupun mati. Maka akan tampil seperti ini.



12. Bisa mengecek versi dari MySQL yang sedang digunakan dengan mengetikkan mysqladmin –u root –p version, yang kemudian akan memunculkan Enter Password: kembali, dan klik enter. Pastikan sudah keluar dari MySQL nya terlebih dahulu (dengan perintah quit).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_EBhImkNpUI.png

Untuk hasil pengecekkannya seperti ini.



13. Bisa menampilkan fitur help pada MySQL, yang digunakan untuk mencari sesuatu seperti perintah yang ada di dalam MySQL (dikarenakan perintahnya banyak dan bilamana ada lupa atau tidak ingatnya). Dengan mengetikkan mysql –u root –p. Hasil keluarnya nanti akan sama seperti pada saat pertama kali menjalankan MySQL (hasil Enter Password: nya).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_Ai9PifkjeE.png

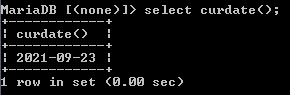
Bisa mengerti bilama sudah ada di dalam MySQL dengan

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_HZUgaT7YEk.png

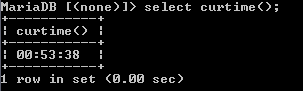
Untuk menampilkan fitur help nya, bisa dengan mengetikkan help maupun simbol tanda tanya (?) setelah MariaDB [<none>]> , seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\L0Bd4TuVRc.png atau C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_aPYbH5d2pc.png

14. Bisa menampilkan tanggal di dalam MySQL dengan mengetikkan select curdate();. Pastikan agar tidak lupa memberikan titik koma (;) pada akhir kalimat. Mengetikkannya setelah setelah MariaDB [<none>]>.

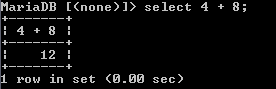


15. Bisa menampilkan waktu atau jam di dalam MySQL dengan mengetikkan select curtime();. Pastikan agar tidak lupa memberikan titik koma (;) pada akhir kalimat seperti pada saat ingin menampilkan tanggal. Mengetikkannya setelah setelah MariaDB [<none>]>.

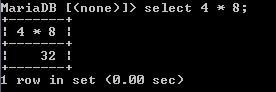


16. Bisa menggunakan kalkulator dengan perintah penjumlahan (+), perkalian (\*), pengurangan (-), pembagian (/), modulo (sisa hasil pembagian (%)), dan hasil pembagian integer (utuh (div)). Dengan mengetikkan Select (angka pertama) …. (angka kedua); (titik-titik tersebut berisikan salah satu perintah kalkulator yang akan digunakan). Mengetikkannya setelah setelah MariaDB [<none>]>.

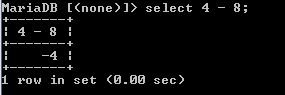
=> Penjumlahan



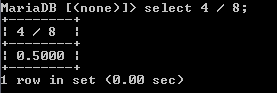
=> Perkalian



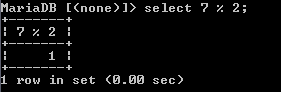
=> Pengurangan



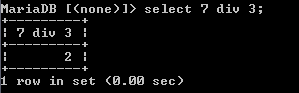
=> Pembagian



=> Modulo (sisa hasil pembagian)



=> Hasil pembagian integer (utuh)

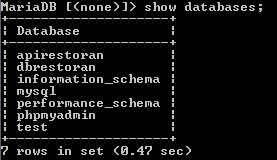


17. Data Definition Languange (DDL), adalah suatu proses pembuatan rumah dari sebuah data. Data tersebut nantinya akan disimpan di dalam kolom, kolom tersebut akan disimpan di dalam tabel, dan tabel sendiri akan disimpan di dalam database.

18. Bisa menampilkan semua database MySQL yang ada di komputer saya dengan menggunakan perintah show databases; (tidak lupa untuk memberikan petik koma (;) diakhirnya).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\lama\ShareX_PemRFvQvkQ.png

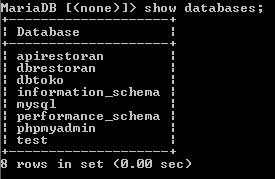
Hasilnya adalah



19. Bisa membuat database dengan nama dbtoko untuk latihan, dengan menggunakan perintah create database (nama database, misalkan dbtoko); (tidak lupa untuk memberikan petik koma (;) diakhirnya).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ho95xJmyVu.png

Kemudian untuk memeriksa apakah database yang baru saja dibuat sudah masuk atau belum, bisa mengeceknya kembali dengan menggunakan perintah show databases;



20. Bisa menghapus database dengan menggunakan perintah drop database (nama database, misalkan dbtoko); (tidak lupa untuk memberikan petik koma (;) diakhirnya). Penghapusan database tersebut digunakan agar semisal ada database yang tidak bisa digunakan bisa dihapuskan.  
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_PONENOwT2M.png

21. Bisa mengaktifkan database sebelum digunakan untuk menjadi tempat penyimpanan suatu tabel dengan menggunakan perintah use (nama database, misalkan dbtoko); (tidak lupa untuk memberikan petik koma (;) diakhirnya).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_Ruv9viQ8la.png

Bila sudah diaktifkan, maka akan terjadi perubahan yang awalnya di dalamnya berupa kata none, maka setelah pengaktifan tersebut akan berubah menjadi nama database yang sudah diaktifkan sebelumnya tadi (misalkan dbtoko).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_2RjMNBfEen.png

22. Tipe data, digunakan untuk membedakan suatu data dengan data yang lainnya pada database. Dapat disesuaikan dengan bentuk dari datanya tersebut (nanti dapat ditentukan suatu data ini dapat menggunakan tipe ini dan seterusnya). Disarankan agar dapat memanfaatkan keseluruhan dari tipe data yang tersedia, agar tiap data tersebut dapat berjalan dengan tipe yang seharusnya.

23. Tipe data numerik, berisi sekumpulan tipe data yang dapat menangani data-data numerik.

24. TINYINT, merupakan suatu tipe yang berbentuk numerik paling kecil dalam menangani data di dalamnya. Dapat menangani data mulai dari angka -128 sampai 127.

25. SMALLINT, memiliki kemampuan untuk menyimpan data lebih besar dari TNIYINT. Mampu untuk menangani data mulai dari -32768 sampai dengan 32767.

26. MEDIUMINT, mampu untuk menangani data mulai dari -8388608 sampai dengan 8388607.

27. INT, merupakan tipe yang sangat sering digunakan dalam pembuatan suatu database (karena tipe ini sangat cukup menampung dalam menangani suatu data). Mampu menyimpan data mulai dari -2147483648 sampai 2147483647.

28. BIGINT, merupakan bentuk terbesar dalam menangani suatu data Numerik. Mampu menangani data mulai dari -9223372036854775808 sampai 9223372036854775807.

29. Tipe data penanggalan dan waktu, untuk menangani data-data yang berkaitan dengan waktu dan penanggalan .

30. DATETIME, dapat menyimpan dua buah bentuk tipe data sekaligus, yaitu penanggalan dan waktu. Bentuk yang dapat diciptakan oleh DATETIME adalah ‘0000-00-00 (dimulai dari tahun, bulan dan tanggal) 00:00:00 (tempat menyimpan waktu atau jam)’. Misalnya: ‘2004-06-22 08:15:45’.

31. DATE, digunakan untuk menyimpan data penanggalan saja (bentuk penulisannya ‘0000-00-00’). Mirip seperti tipe DATETIME, namun yang ditampilkan hanya penggalannya saja (misalkan ‘2004-06-22’).

32. TIMESTAMP, ditampilkan berjajar tanpa ada pembatasnya, bentuk penulisannya 00000000000000.

33. TIME, hanya digunakan untuk menyimpan data dengan bentuk jam saja. Dimulai dari tahun yang dibaca dari dua karakter terakhir dan selanjutnya diikuti bulan dan tanggal. Bentuk penulisannya ’00:00:00’. Pada protoype pewaktuan tersebut dapat dijabarkan bahwa 00 perma adalah jam yang diikuti menit dan detik (misal seperti 08:35:55).

34. YEAR, merupakan bentuk yang paling sederhana yang hanya menyimpan data berupa tahunnya saja. Dituliskan secara lengkap 4 digit (misalnya 2004 atau 1999).

35. Tipe data string, disesuaikan oleh MySQL karena banyaknya data dengan berbagai macam tipe yang ada.

36. TINYTEXT, merupakan bentuk terkecil dari data String. Mampu menangani data sampai dengan 2^8-1 data.

37. TINYBLOB, berbentuk sama dengan tipe TINYTEXT. Mampu menangani data sampai dengan 2^8-1 data.

38. TEXT, salah satu bentuk tipe String yang mampu menangani data sampai dengan berukuran 2^16-1 (64k-1) data.

39. BLOB, memiliki kemampuan yang sama dengan tipe TEXT, yaitu mampu menangani data sampai dengan berukuran 2^16-1 (64k-1) data.

40. MEDIUMTEXT, dapat menyimpan data dengan ukuran yang cukup besar, mampu menangani data sampai dengan 2^24-1 (16M-1) data.

41. MEDIUMBLOB, tipe bentuk dari MEDIUMTEXT dapat diganti dengan tipe MEDIUMBLOB ini. Mampu menyimpan data sampai dengan 2^24-1 (16M-1) data.

42. LONGBLOB, adalah bentuk tipe data yang paling besar dalam menangani data. Data yang disimpan sampai dengan berukuran Giga Byte. Memiliki batasan penyimpan sampai dengan 2^32-1 (4G-1) data.

43. Ada beberapa Tipe Data String yang sering digunakan seperti VARCHAR, CHAR, ENUM, dan SET.

44. VARCHAR, dapat menyimpan data sampaid engan 225 karakter. Tipe ini dapat digunakan apabila data yang dimasukkan tidak lebih dari batasan tersebut.

45. CHAR, bentuk tipe ini hampir sama dengan bentuk dari tipe VARCHAR. Keduanya sangat signifikan dalam menyimpan data, misalkan membuat kolom dengan tipe VARCHAR(25), meskipun memasukkan datanya kurang dari 25, akan tetapi nantinya tetap dibaca sebanyak digit yang dimasukkan pada awalnya (25). Namun untuk Tipe data CHAR, misalkan membuat dengan kolom CHAR(4) dan memasukkan digitnya kurang dari 4, maka akan tetap dibaca 4 digit. Mampu menangani sampai dengan 225 karakter.

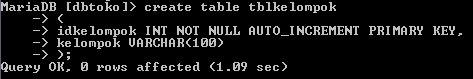
46. ENUM, digunakan untuk validasi. Untuk tipe data seperti ini, biasanya kolom akan ditentukan terlebih dahulu (misal pada pembuatan kolom yang isinya tentang golongan darah A, B, AB, dan O, maka bentuk penulisannya adalah ENUM(‘A’, ‘B’, ‘AB’, ‘O’)). Jika bila nantinya pada saat memasukkan suatu datanya tidak sesuai dengan kriteria, maka akan terjadi suatu kesalahan atau tidak bisa dibaca (kosong).

47. SET, sebenarnya memiliki fungsi yang sama dengan tipe ENUM, yaitu dengan mendeklarasikan anggota dari isi kolom yang mungkin akan menjadi anggotanya.

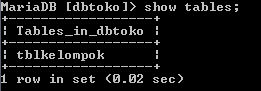
48. Binary Large Object (BLOB), merupakan tipe data yang biasa digunakan untuk menyimpan data berbentuk biner.

49. LONGBLOB, tipe data yang dapat digunakan untuk menyimpan gambar.

50. Bisa membuat tabel baru berdasarkan desain yang sudah dibuat sebelumnya (tabel pertama yang dibuat ialah tblkelompok) dengan menggunakan perintah create table (nama tabelnya, misal tblkelompok), kemudian berpindah baris menggunakan tombol enter dan mengetikkan nama kolomnya (idkelompok) tipe data (INT NOT NULL AUTO\_INCREMENT PRIMARY KEY), kolom tipe data; (kelompok VARCHAR(100)); (menggunakan VARCHAR karena tipe data pada pembuatan sebelumnya menggunakan tipe text)). (Pastikan sebelum membuat suatu tabel pada database sudah dipilih dengan benar terlebih dahulu).



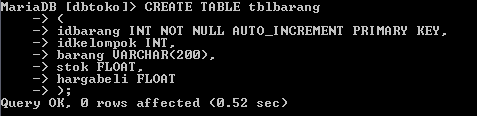
50. Bisa menampilkan tabel yang sebelumnya telah dibuat dengan menggunakan perintah show tables;.



51. Bisa menampilkan struktur tabel dari tabel yang sebelumnya telah dibuat dengan menggunakan perintah describe (nama tabel, misal tblkelompok);.



52. Bisa membuat tabel tblbarang sesuai dengan yang ada di buku.

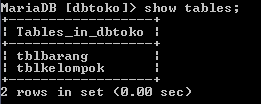


Karena pada MySQL tidak terdapat tipe data Currency, maka bisa menggunakan tipe data FLOAT sebagai gantinya.

Periksa apakah strukturnya sudah benar



Menampilkan tabel yang sudah dibuat sebelumnya dengan menggunakan perintah



53. Bisa menambah kolom dari tabel yang sudah dibuat di dalam database sebelumnya (misalkan ada kekurangannya dapat ditambahkan sewaktu-waktu). Dengan menggunakan perintah alter table (nama tabel, misal tblbarang) add (nama kolom, misal hargajual) (tipe datanya, misal float);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\b8JWxLLHih.png

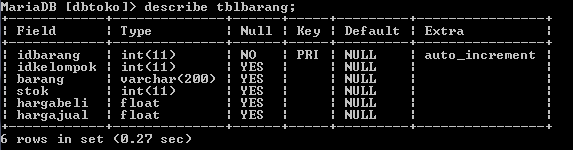
Setelah penambahan kolom tersebut, bisa dilakukan pengecekan kembali dari struktur tabelnya.



54. Bisa mengubah tipe data dari dalam kolom tabel yang telah dibuat sebelumnya (misalkan ada suatu kesalahan dan ingin mengganti tipe datanya). Dengan menggunakan perintah alter table (nama tabel, misal tblbarang) modify (nama kolom, misal stok) (tipe data yang ingin diubah, misal INT);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_UBv5PsQUn8.png

Setelah melakukan pengubahan dari tipe data tersebut, bisa dilakukan pengecekan kembali dari struktur tabelnya.



55. Bisa memberi nilai default pada suatu kolom. Pada kolom stok di dalam tblbarang akan diberi nilai DEFAULT [0] atau nilai awal [0]. Dapat menggunakan perintah ALTER TABLE (nama tabel, misal tblbarang) ALTER COLUMN (nama kolom, misal stok) SET DEFAULT (bilamana ada isi dari nilai tersebut maka diisikan, misal 0. Namun bila merupakan VARCHAR berikan tanda petik);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_RCuhUqZm2a.png

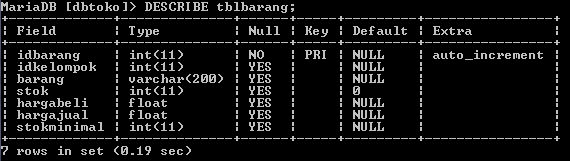
Setelah memberikan nilai default pada suatu kolom tersebut, bisa dilakukan pengecekan kembali dari struktur tabelnya.



56. Bisa melakukan penghapusan kolom dengan menggunakan perintah ALTER TABLE (nama tabel, misal tblbarang) DROP COLUMN (nama kolom yang akan dihapus di dalam suatu tabel tersebut, misal stokminimal);. Namun sebelumnya membuat terlebih dahulu kolom stokminimal.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_JXvilsUtfe.png

Periksa tabelnya



Lalu lakukan penghapusan (DROP COLUMN)

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_OT00r0iHF7.png

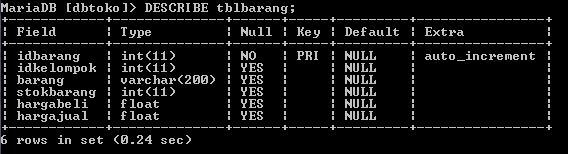
Setelah menghapus suatu kolom tersebut, bisa dilakukan pengecekan kembali dari struktur tabelnya. (Untuk kolomnya (stokminimal) dibuat dulu pertama (untuk latihan, agar tidak menghapus yang sudah diatur dan dibuat sebelumnya)).



57. Bisa merubah nama kolom (bilamana ada suatu kesalahan atau apapun dapat sewaktu-waktu langsung diganti) dengan menggunakan perintah ALTER TABLE (nama tabel, misalkan tblbarang) CHANGE (nama kolom yang lama, misal stok) (nama kolom yang baru (diganti menjadi ini), misal stokbarang) (tipe datanya, misal INT);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_Bcs4JEpqY9.png

Setelah mengubah nama kolom pada suatu kolom tersebut, bisa dilakukan pengecekan kembali dari struktur tabelnya.



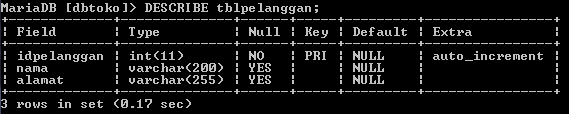
58. Dapat membuat tabel baru, yaitu tabel pelanggan (tblpelanggan). (sesuai dengan yang ada di buku).



59. Menambahkan primary key dengan merubah tipe data pada kolomnya, kemudian menambahkan PRIMARY KEY; (dengan menggunakan perintah ALTER TABLE tblpelanggan INT NOT NULL AUTO\_INCREMENT PRIMARY KEY);

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_KQbEmHJ6JH.png

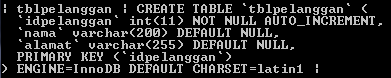
Setelah menambahkan primary key ke dalam suatu tipe data yang awalnya belum terdapat hal tersebut pada suatu kolomnya, bisa dilakukan pengecekan kembali dari struktur tabelnya.



60. Dapat menampilkan engine (mesin) yang digunakan pada saat pembuatan database dengan menggunakan perintah SHOW CREATE TABLE (nama tabelnya, seperti tblpelanggan);.

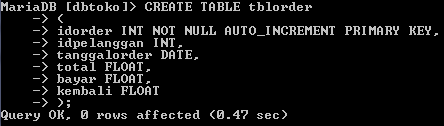
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_jnngI74ITm.png

Hasil yang ditampilkan ialah

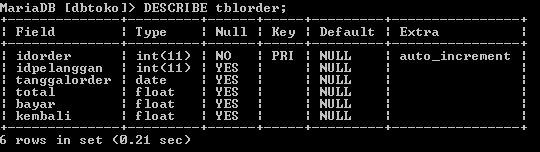


Pada saat pembuatan suatu database, terdapat dua (2) engine yang digunakan, yaitu MYISAM dan InnoDB. Dan untuk pembuatan suatu database relasional harus menggunakan engine InnoDB.

61. Dapat membuat tabel baru, yaitu tabel order (tblorder). (sesuai dengan yang ada di buku).



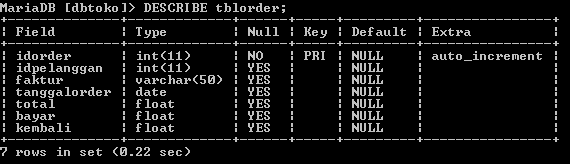
Struktur tabelnya



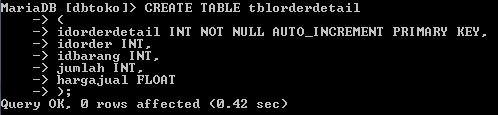
62. Dapat menambahkan suatu kolom baru setelah kolom yang lama (sudah dibuat sebelumnya). Dengan menggunakan perintah ALTER TABLE (nama tabel, seperti tblorder) ADD (nama kolom yang akan ditambahkan atau baru, seperti faktur) (tipe datanya, seperti VARCHAR(50)) AFTER (nama kolom yang sebelumnya, seperti idpelanggan (karena akan diletakkan setelah idpelanggan)).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_PV876SLpIo.png

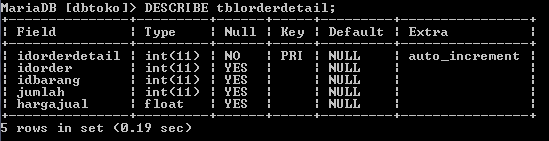
Hasilnya ialah



63. Dapat membuat tabel baru, yaitu tabel order detail (tblorderdetail). (sesuai dengan yang ada di buku).



Struktur tabelnya



64. Index, digunakan untuk mempercepat proses pencarian data terutama pada data yang jumlahnya diatas ribuan. Pada tabel yang sudah dibuat sebelumnya, biasanya yang paling banyak dicari untuk pencarian data adalah nama barang dan nama pelanggan.

65. Bisa membuat index dengan menggunakan perintah CREATE INDEX (nama idex, seperti pelangganindex) ON (nama tabel, seperti tblpelanggan) (nama kolomnya, seperti nama);.

Untuk tabel pelanggan

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_dBjM8CpRoA.png

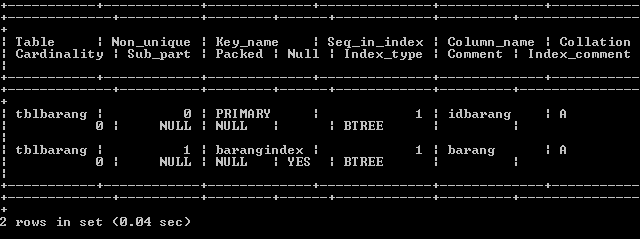
Untuk tabel barang

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_RYBkb8nG7I.png

66. Bisa menampilkan index yang sudah dibuat sebelumnya dengan perintah SHOW INDEX FROM (nama tabel, seperti tblbarang);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_JdOaDmgwhc.png

Hasilnya ialah

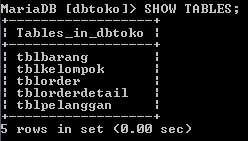


67. Bisa menghapus suatu index bilamana sudah tidak bisa digunakan dengan perintah DROP INDEX (nama indexnya, seperti pelangganindex) ON (nama tabel, seperti tblpelanggan);.

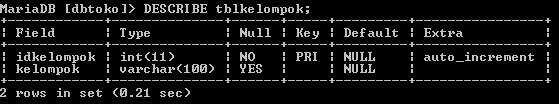
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_cAfU5hK1Zi.png

68. Data Manipulation Language (DML),

69. Bisa menginsertkan semua kolom yang digunakan untuk memasukkan data pada semua kolom tabel (pastikan untuk memeriksa tabel yang akan digunakan terlebih dahulu).



Lihat dahulu kolom dari tblkelompok

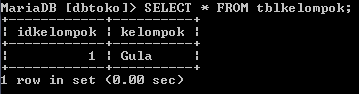


Menggunakan dua (2) kolom dari tblkelompok yang akan di INSERT datanya, yaitu idkelompok dan kelompok.

Menggunakan perintah INSERT INTO (nama tabel, seperti tblkelompok) VALUES (kolom, kolom, semua kolomnya, seperti ‘’, ‘Gula’);

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ELG3Pvl0ps.png

70. Untuk memeriksa apakah suatu data yang di INSERTKAN tadi sudah berhasil digunakan dengan menggunakan perintah SELECT \* FROM (nama tabel, seperti tblkelompok);



71. Bisa menginsertkan sebagian kolom yang digunakan untuk INSERT hanya pada kolom yang dipilih atau disebutkan. Dengan menggunakan perintah INSERT INTO (nama tabel, seperti tblkelompok) (kolom yang disebutkan, seperti kelompok) VALUES (isi yang akan digunakan untuk semua kolom yang sudah disebutkan, seperti ‘Beras’);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_i1ctP32Dlj.png

Hasilnya ialah (setelah menginsertkan berbagai macam hal)



72. Bisa menghapus (delete) sebagian dari record (baris data) dengan menggunakan perintah DELETE FROM (nama tabel, seperti tblkelompok) WHERE (urutan baris mana yang akan dihapus, seperti idkelompok=5);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\yqUhNIbwtl.png

Hasilnya setelah ada yang sudah dihapus



73. Bisa menghapus (delete) semua baris record yang telah dibaut sebelumnya dengan menggunakan perintah DELETE FROM (nama tabelnya, seperti tblkelompok);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_Byr0Vc2LRi.png

Karena semua datanya telah dihapus, maka yang tampil nanti adalah kosong.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ckpGWkqUcr.png

74. Bisa mengupdate (mengubah atau mengganti) sebagian record dengan menggunakan perintah UPDATE (nama tabel, seperti tblkelompok) SET (nama kolom=isi kolomnya, seperti kelompok=’Snek’ (awalnya bernamakan jajan)) WHERE (nama kolom=baris yang akan dipilih untuk diubah, seperti idkelompok=6);.

Menambahkan beberapa isian kembali ke dalam tabelnya (tblkelompok)



Untuk mengupdatenya

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_jIaRq02aPN.png

Hasilnya ialah



75. Bisa mengupdate (mengubah atau mengganti) semua record dengan menggunakan perintah UPDATE (nama tabel, seperti tblkelompok) SET (nama kolom=’isi kolomnya’, seperti kelompok=’Makanan’);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_0myy9cp4PO.png

Hasilnya ialah



Kemudian mengupdatenya lagi sehingga hasilnya menjadi

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_epcqfgu3nd.png

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_v1gf038keL.png

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_DUdJbIS0Jb.png

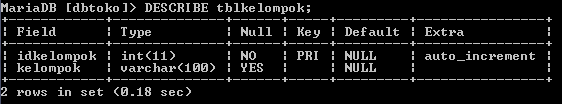


76. Tabel Master, merupakan tabel yang memasuki tabel lain. Seperti pada kolom idkelompok dari tabel tblkelompok yang memasuki tabel tblbarang. Sehingga tblkelompok tersebut menjadi TABEL MASTER.

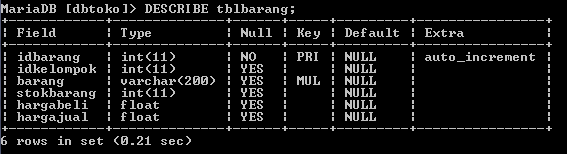
77. Tabel Transaksi, merupakan tabel yang dimasuki tabel lain. Seperti tabel tblbarang yang dimasuki kolom idkelompok yang berasal dari tabel tblkelompok. Sehingga tabel tblbarang tersebut menjadi TABEL TRANSAKSI.

=> Ini merupakan hasil dari TABEL MASTER DAN TABEL TRANSAKSI (DETAIL)

tblkelompok



tblbarang



78. Primary Key, merupakan kolom yang berbeda antara baris data yang satu dengan yang lainnya.

79. Foreign Key, merupakan kolom yang masuk dari tabel lain.

80. Relational Dependencies (hubungan ketergantungan), merupakan suatu pengaturan agar tabel yang satu terhubung dengan tabel yang lainnya. Data yang masuk ke satu tabel itu pun akan bergantung dari tabel yang lain dalam bentuk yang ber-urutan.

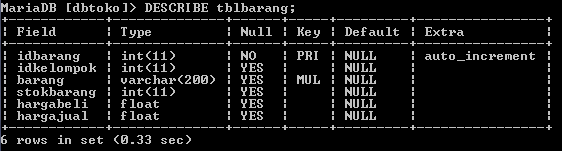
81. Proses Bisnis (Alur Kerja), pada relasinya akan terdapat suatu alur kerja yang sering disebut dengan proses bisnis. Proses bisnis pada relasi tersebut yaitu sebelum suatu toko akan dibuka oleh pemiliknya, pemilik tersebut akan mengelompokkan terlebih dahulu barang-barang apa saja yang akan dijual nantinya. Kelompok barang tersebut akan disimpan di dalam tabel kelompok.

Misalkan toko tersebut akan menjual Beras, Gula, dan Tepung



Kemudian pada kelompok tersebut akan dibuatkan detail dari isinya. Kemudian pada saat pembuatan database, setiap kelompok akan di wakili oleh kolom PRIMARY KEY yang akan dimasukkan ke dalam tabel yang lain untuk menunjukkan bahwa kolom tersebut merupakan suatu anggota dari tabel yang lain juga.

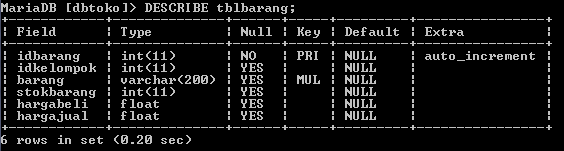
Kemudian pada kelompok tersebut akan dibuat DETAIL isinya, misalkan seperti kelompok Beras yang isi kelompoknya Beras Rojo Lele, Beras Cianjur, Beras Medium, Beras Mahal. Kelompok Gula yang isi kelompoknya Gula Putih, Gula Merah, Gula Batu, Gula Aren. Kelompok Tepung yang isi kelompoknya Tepung Terigu, Tepung Tapioka, Tepung Kanji.



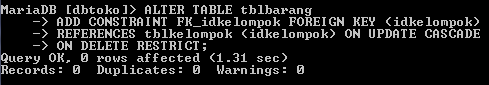
(Pada tabel tblbarang diatas, idkelompok akan menunjukkan bahwa kolom tersebut adalah isi dari tabel tblkelompok).

82. Bisa membuat relasi antar tabel sesuai dengan yang ada di buku, dengan perintah ALTER TABLE (nama tabel detail, tblbarang) lalu di enter ke bawahnya dan mengetikkan ADD CONSTRAINT FK\_(foreign key pada tabel detail, FK\_idkelompok) FOERIGN KEY (foreign key, idkelompok) lalu di enter ke bawahnya lagi dan mengetikkan REFRENCES (nama tabel master, tblkelompok) (primary key dari tabel master, (idkelompok)) ON UPDATE CASCADE dan di enter ke bawah kembali, kemudian mengetikkan ON DELETE RESTRICT;

Lihat dahulu struktur tabel yang akan dibuat relasinya



Untuk membuat relasinya

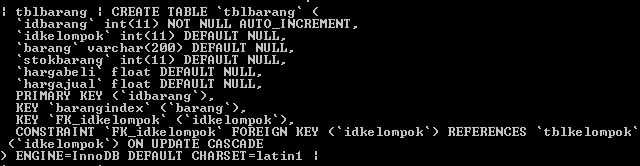


(ON UPDATE CASCADE, jika kolom primary key pada tabel master maka kolom foreign key pada tabel transaksi akan berubah. ON DELETE RESTRICT, kolom pada primary key di tabel master tidak akan bisa di hapus jika kolom tersebut masih digunakan pada tabel transaksi. ON DELETE CASCADE, jika baris pada kolom master dihapus, maka semua baris data yang menggunakan isi tabel master pada tabel transaksi akan terhapus juga).

Untuk melihat hasil pembuatan relasinya dapat menggunakan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_yjRj9vYjwm.png

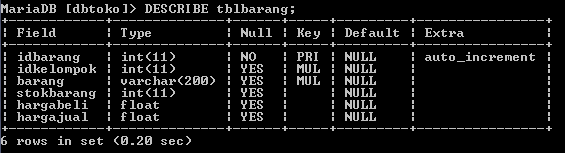
Tampilan bila relasinya sudah berhasil dibuat



83. Bisa menguji hasil dari pembuatan relasi, melakukan INSERT data pada tabel master terlebih dahulu, baru kemudian ke tabel transaksi.



Tampilkan struktur tabel agar memudahkan proses pada saat insert data



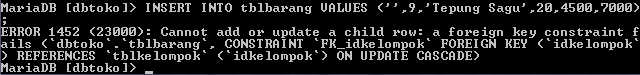
Isi tabel transaksu tersebut sesuai dengan idkelompok yang akan digunakan, karena iDCLrang AUTO\_INCREMENT, maka kolom bisa diganti dengan menggunakan petik kosong (“)

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_XcuPJXohiV.png

Hasil dari melakukan (banyak) insert data, sehingga muncul tabel seperti



84. Bisa melakukan pengujian Insert, insert pada tabel tblbarang hanya bisa dilakukan dengan menggunakan data yang tersedia pada tabel masternya saja. Idkelompok pada tabel master hanya tersedia [6,7,8]….. Pada pengujian INSERT idkelompok yang digunakan adalah 9 yang tidak tersedia pada tabel masternya (tabel tblkelompok).



85. Bisa melakukan pengujian update, dengan menggunakan perintah UPDATE tblkelompok SET idkelompok=9 WHERE kelompok=’Tepung’; (mengubah idkelompok tepung dari 8 menjadi 9).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ryLfJ1yycK.png

Kemudian periksa perubahannya pada tblkelompok



Periksa juga pada tblbarang



(Pada ON UPDATE CASCADE, jika kolom tabel master berubah, maka kolom pada tabel transaksi juga akan ikut berubah).

86. Bisa melakukan pengujian delete dengan menghapus tabel master.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_wgeZvWL4HW.png

Karena idkelompok 9 sudah digunakan pada tabel transaksi maka data tersebut tidak bisa dihpaus sesuai dengan rekasi yang dibuat yaitu ON DELETE RESTRICT. Data yang bisa dihpaus hanya yang belum digunakan pada tabel transaksi. Lalu menginsertkan data baru pada tabel masternya.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_MnEumEvVDW.png

Setelah itu periksa lagi data yang sudah dimasukkan



Karena kolom idkelompok 10 belum digunakan pada tabel transaksi, maka kolom tersebut bisa dihapus. Lakukan penghapusan dengan menggunakan perintah DELETE FROM tblkelompok WHERE idkelompok=10;

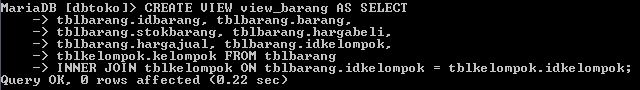
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_c5uZ6VtMyo.png

Periksa lagi tabel yang sudah dihapus tadi

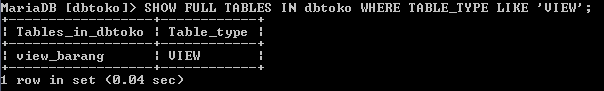


87. View, digunakan untuk menggabungkan tabel master dan tabel transaksi menjadi sebuah tabel baru agar lebih mudah dilihat dan digunakan.

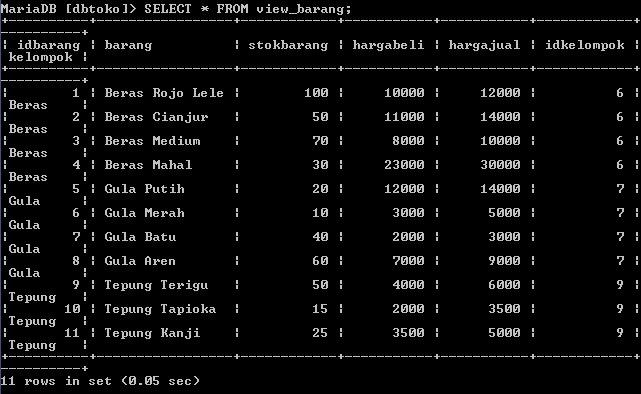
88. Bisa membuat view dengan menggunakan perintah CREATE VIEW (nama\_view, view\_barang) AS SELECT lalu di enter, (tabel transaksi.kolom transaksi, misalkan seperti tblbarang.idbarang), (tabel master.kolom master, misalkan seperti tblbarang.barang) FROM (tabel transaksi, tblbarang) INNER JOIN (tabel master, tblkelompok) ON (tabel transaksi.kolom transaksi = tabel master.kolom master, tblbarang.idkelompok = tblkelompok.idkelompok);



89. Bisa menampilkan semua view yang sudah dibuat dengan menggunakan perintah SHOW FULL TABLES IN (nama database, dbtoko) WHERE TABLE\_TYPE LIKE ‘VIEW’;



90. Bisa melihat isi view dengan menggunakan perintah SELECT \* FROM (nama\_view, view\_barang);



Melihat isi viewnya sama dengan melihat isi dari tabelnya. (cmd saya tidak bisa lebih dilebarkan lagi, sehingga pada kolom kelompok menjadi gabung satu tempat dengan idbarang).

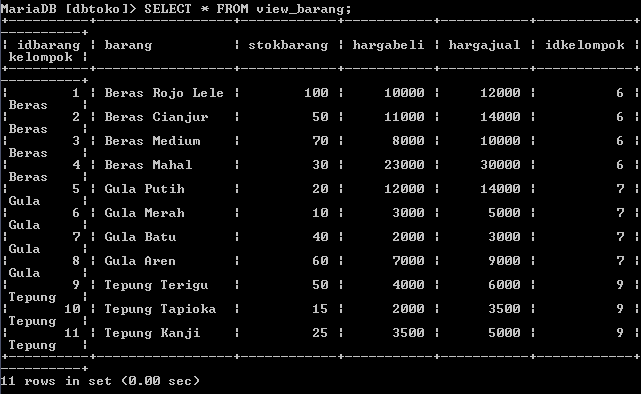
91. Bisa menghapus view dengan menggunakan perintah DROP VIEW (nama\_view, view\_barang);

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_HOhVGkhLH1.png

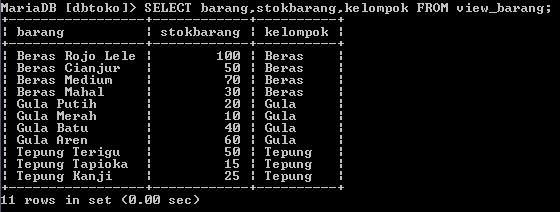
Kemudian periksa kembali daftar viewnya

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ws53V2bMhv.png

92. Bisa men-select semua kolom (\*), select semua ialah menampilkan semua kolom dari tabel atau VIEW, dengan menggunakan perintah SELECT \* FROM (nama tabel atau nama\_viewnya, view\_barang; (kali ini menggunakan nama viewnya)).



93. Bisa men-select sebagian kolom, select sebagian kolom adalah menampilkan hanya pada kolom yang dipilih, dengan menggunakan perintah SELECT (nama kolom, barang), (nama kolom, stokbarang), (nama kolom, kelompok) FROM (nama tabel atau nama\_view, view\_barang; (kali ini menggunakan nama viewnya)).

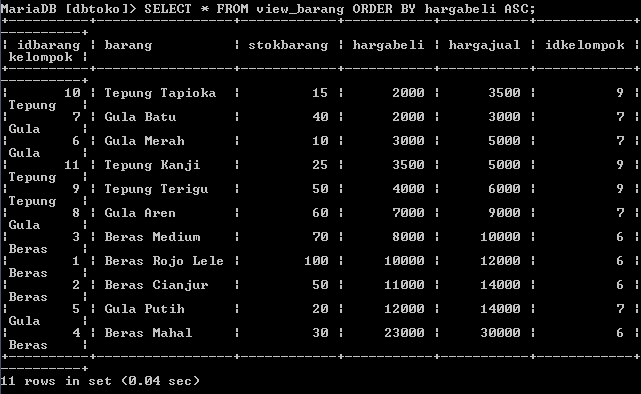


94. Select Order, merupakan cara untuk menampilkan data dalam urutan naik ataupun turun. Jika naik menggunakan ASC, dan bila turun menggunakan DESC.

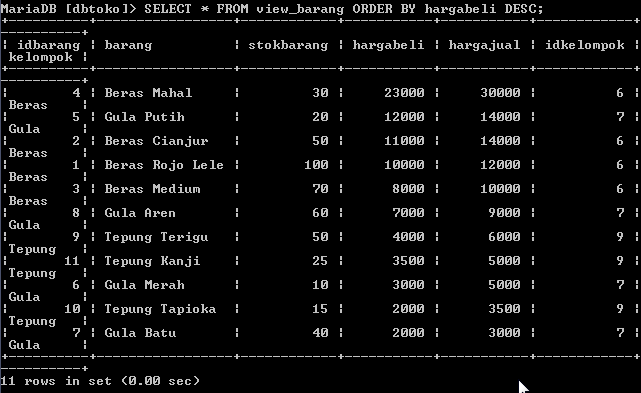
95. Bisa men-select suatu order dengan menggunakan perintah SELECT \* FROM (nama tabel atau nama\_view, view\_barang; (kali ini menggunakan nama view)) ORDER BY (nama kolomnya, hargabeli) ASC;

**C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_dea79O8LPK.png**

Hasilnya (dengan menggunakan ASC) ialah

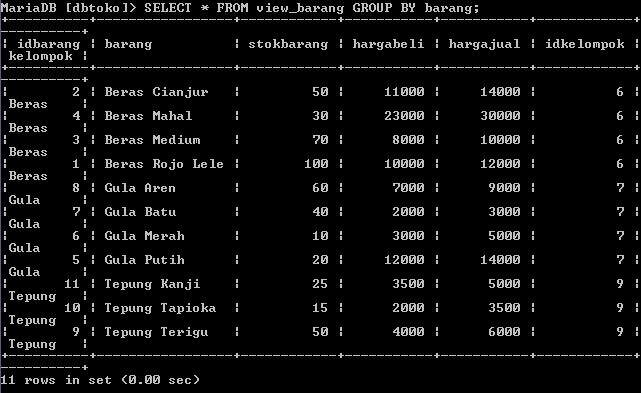


Hasilnya (dengan menggunakan DESC) ialah



96. Select Group, digunakan untuk mengelompokkan data sesuai dengan kelompok dari kolom yang dipilih.

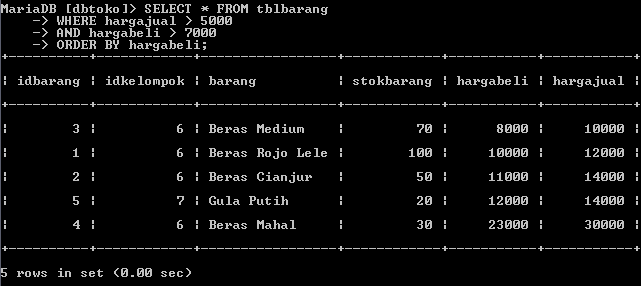
97. Bisa men-select group dengan menggunakan perintah SELECT \* FROM (nama tabel atau nama\_view, view\_barang; (kali ini menggunakan nama viewnya)) GROUP BY (nama kolom, barang);



98. Bisa melakukan pengujian where sesuai dengan yang ada di buku. Pengujian WHERE pada SQL bisa diletakkan di belakang yaitu SELECT, DELETE, dan UPDATE. Pengujian akan menjalankan SELECT, DELETE, UPDATE jika kondisinya yang diuji sudah terpenuhi, seperti operator pembanding [=, >, <, <>, >=, =>, LIKE], operator logika (AND, OR), dan SELECT.

Contoh pengujiannya dengan menggunakan perintah SELECT \* FROM (nama tabel atau nama\_view, tblbarang (kali ini menggunakan nama tabel)) WHERE (nama kolom, hargajual) (pengujian, > 5000) ORDER BY (nama kolom, hargajual);

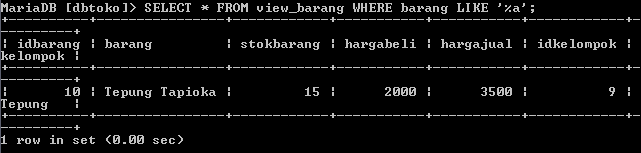




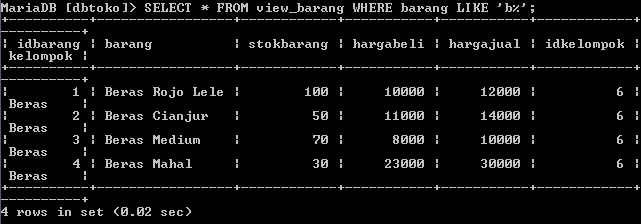


SELECT \* FROM (nama tabel atau nama\_view) WHERE (nama kolom) LIKE ‘%\_apapun’;. Persen (%) adalah sebutan untuk sembarang atau apapun. Contohnya seperti

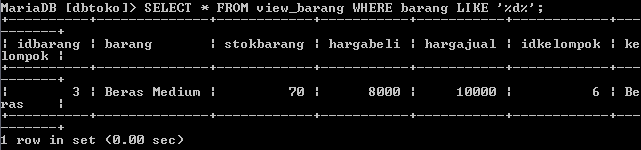
%a yang artinya awalan sembarang yang penting akhirannya a



b% artinya awalan b akhiran sembarang

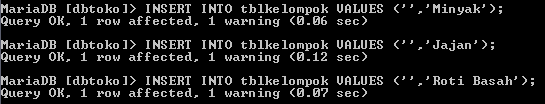


%d% artiny awalan sembarang dan akhiran sembarang yang tengahnya ada huruf d



99. Subquery (SELECT IN SELECT), adalah SELECT yang ada di dalam SELECT.

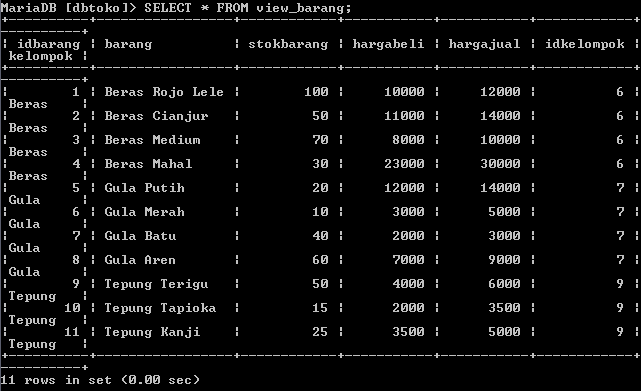
100.. Bisa melakukan Subquery (SELECT IN SELECT). Sebelumnya menambahkan data terlebih dahulu pada tblkelompok



Sehingga datanya akan tampil seperti



Periksa [view\_barang]



Pada [view\_barang] ada kelompok yang sudah menggunakan data dari tabel [tblkelompok]. Bila ingin mengetahui berapa banyak data yang sudah maupun belum saat digunakan pada VIEW, bisa menggunakan SUBQUERY untuk mengetahuinya.

Menggunakan oerintah SELECT \* FROM (nama tabel, tblkelompok) WHERE (nama kelompok, idkelompok) IN ((SELECT (nama kolom, idkelompok) FROM (nama tabel, view\_barang)));



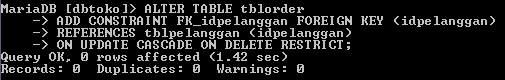
SELECT \* FROM (nama tabel, tblkelompok) WHERE (nama kelompok, idkelompok) NOT IN ((SELECT (nama kolom, idkelompok) FROM (nama tabel, view\_barang)));



(Penggunaan tanda \* pada SELECT IN hanya bisa dilakukan pada select yang pertama)

101. Bisa membuat relasi antar tabel,

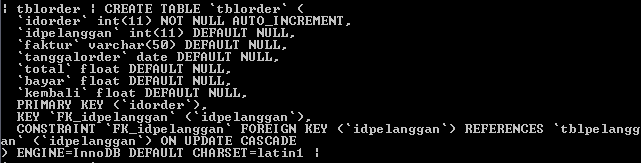
=> Proses Bisnis, pelanggan yang akan membeli akan dicatat di dalam tabel order. Hanya pelanggan yang sudah terdaftar pada tabel pelanggan yang bisa melakukan order.



Periksa kembali hasil pembuatan RELASI

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\skPgTPwTMZ.png

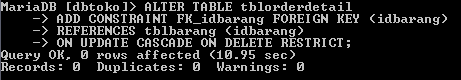
Hasil pembuatan relasinya



102. Relasi 2 Tabel Master dan 1 Tabel Transaksi, pada tabel tblorderdetail terdapat relasi antara tabel tblorder dan tblbarang. Buat dahulu relasi antara tabel tblorderdetail dan tabel tblorder



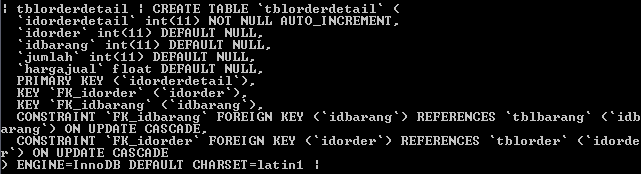
Buat relasi antara tabel tblorderdetail dan tabel tblbarang



Periksa kembali hasil saat membuay relasi antara 2 tabel master [tblorder] [tblbarang] dan 1 tabel transaksi [tblorderdetail]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_XqHfnIo4fB.png

Hasil pembuatan relasinya



103. Dummy Data, ialah data yang tidak sebenarnya. Digunakan selama pengujian database. Data ini diperlukan jika data sebenarnya tidak bisa diperoleh.

104. Pada tabel tblpelanggan aplikasi akan di isi dengan dummy data dengan dummy data dengan nama pelanggan seperti



(Tampilkan struktur tabel terlebih dahulu)

Isikan nama pelanggan dengan dummy data

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_nZ9GmeNH2y.png

Periksa hasil INSERT datanya



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

105. Trigger, ialah perintah INSERT, UPDATE, DELETE, FUNCTION, PROCEDURE yang ditanam pada MySQL dan akan dijalankan pada kejadian seperti

=> AFTER INSERT (setelah INSERT) pada tabel yang dimaksud.

=> BEFORE INSERT (sebelum INSERT) pada tabel yang dimaksud.

=> AFTER DELETE (setelah DELETE) pada tabel yang dimaksud.

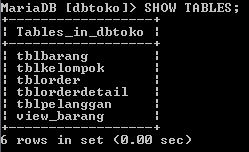
=> BEFORE DELETE (sebelum DELETE) pada tabel yang dimaksud.

=> AFTER UPDATE (setelah UPDATE) pada tabel yang dimaksud.

=> BEFORE UPDATE (sebelum UPDATE) pada tabel yang dimaksud.

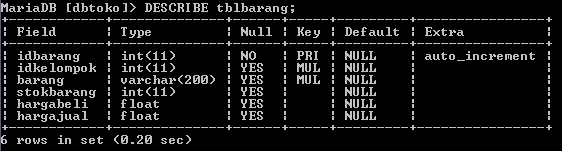
106. Sebagai latihan, buka terlebih dahulu semua database yang sudah dibuat sebelumnya.

1). Tampilkan semua tabelnya

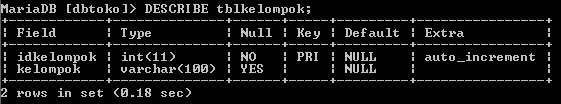


2). Lihat struktur semua tabelnya

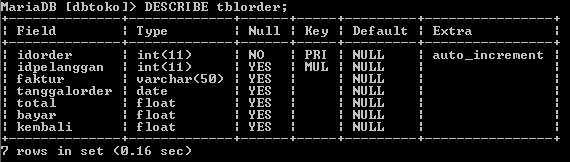
[tblbarang]



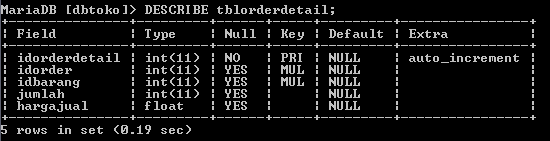
[tblkelompok]



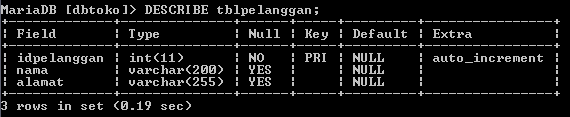
[tblorder]



[tblorderdetail]



[tblpelanggan]



3). Penyusunan proses bisnis

A. Kolom [stokbarang] pada tabel [tblbarang] akan BERKURANG jika terjadi INSERT data pada tabel [tblorderdetail].

A. Kolom [stokbarang] pada tabel [tblbarang] akan BERTAMBAH jika terjadi DELETE data pada tabel [tblorderdetail].

A. Kolom [total] pada tabel [tblorder] akan BERTAMBAH jika terjadi INSERT data pada tabel [tblorderdetail].

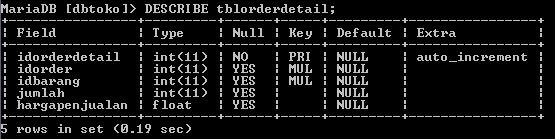
A. Kolom [total] pada tabel [tblorder] akan BERKURANG jika terjadi DELETE data pada tabel [tblorderdetail].

4). Pemeriksaan nama kolom yang sama

Sebelum membuat TRIGGER, pastikan terlebih dahulu bahwa tidak ada kolom yang sama pada setiap tabelnya. Bila terjadi kesamaan, diubah terlebih dahulu (seperti kolom hargajual pada tabel tblbarang dan tblorderdetail. Yang diubah ialah hargajual pada tblorderdetail menjadi hargapenjualan).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\NAKVXFfrr8.png

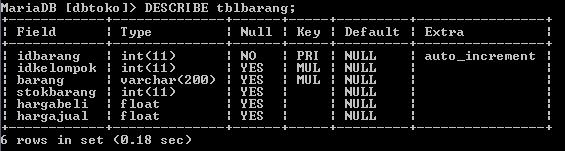
Lihat hasil perubahan nama kolomnya



5). Pemerikdaan kolom untuk perubahan hasil trigger

Diperiksa terlebih dahulu kolom yang terkena dampak dari proses trigger tersebut. Jika yang diproses oleh trigger adalah operasi matematika (penjumlahan, pengurangan, pembagian, dan 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).

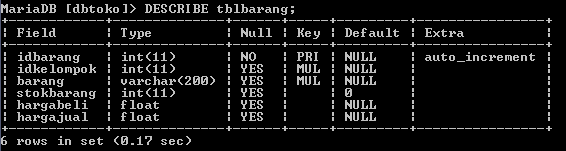
Kolom [stokbarang] belum di SET DEFAULT dengan ANGKA NOL



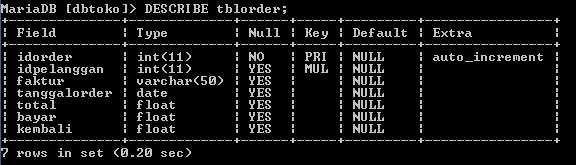
Perintah ALTER pada tabel [tblbarang]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_8r92A1OtGf.png

Hasil perubahannya



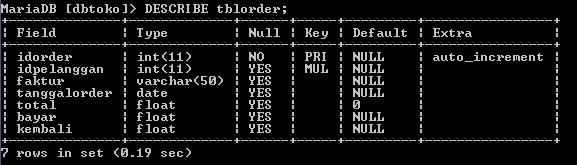
Pemeriksaan kolom [total] pada tabel [tblorder]. Belum di set DEFAULT ANGKA NOL



Lakukan perubahan dengan ALTER. Perintah ALTER pada tabel [tblorder]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_wCJre9vvFa.png

Hasil perubahan kolom [total] pada [tblorder]



107. Pembuatan TRIGGER

=> kurang\_stok (BEFORE INSERT ON), digunakan untuk mengurangi stokbarang pada tabel tblbarang sebelum insert pada tabel tblorderdetail.

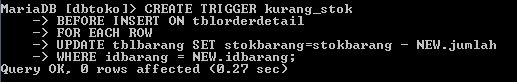
=> tambah\_total (AFTER INSERT ON), digunakan untuk menambahkan total pada tabel tblorder sesudah insert pada tabel tblorderdetail.

=> tambah\_stok (BEFORE DELETE ON), digunakan untuk menambahkan stokbarang pada tabel tblbarang sebelum selete pada tabel tblorderdetail.

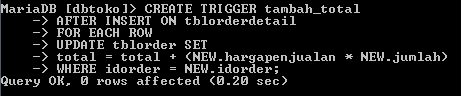
=> kurang\_total (AFTER DELETE ON), digunakan untuk mengurangi total pada tabel tblorder sesudah delete pada tabe; tblorderdetail.

Menggunakan perintah CREATE TRIGGER (nama triggernya, misal kurang\_stok) lalu enter dan ketikkan BEFORE INSERT ON (nama tabel yang akan dipasang trigger, tblorderdetail) enter kembali dan ketikkan FOR EACH ROW lalu enter dan ketikkan (perintah triggernya lalu akhiri dengan titik koma, UPDATE tblbarang SET stokbarang=stokbarang – NEW.jumlah lalu di enter dan ketikkan WHERE idbarang = NEW.idbarang);.

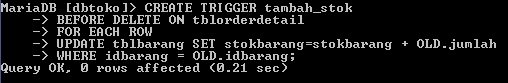
=> kurang\_stok



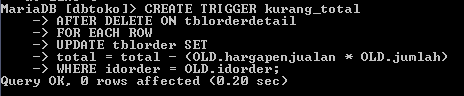
=> tambah\_total



=> tambah\_stok



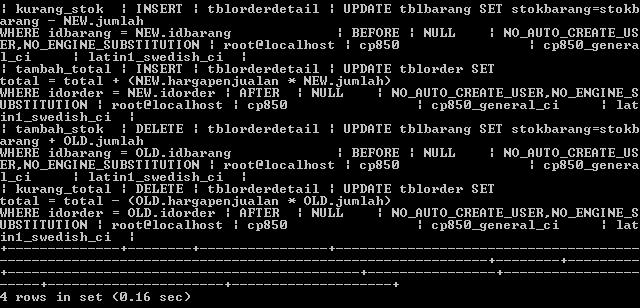
=> kurang\_total



108. Menampilkan trigger yang sudah dibuat dengan

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_wOQnatlkHf.png

Tampilan hasil triggernya



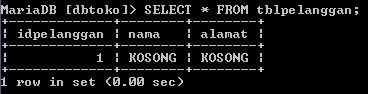
109. Pengujian trigger,

1). Persiapan

Tampilkan tabel [tblbarang]



Tampilkan tabel [tblpelanggan]

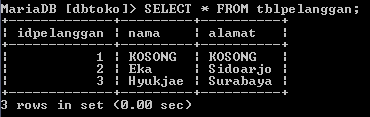


Karena data pelanggan hanya satu, tambahkan dua (2) pelanggan lagi untuk pengujian

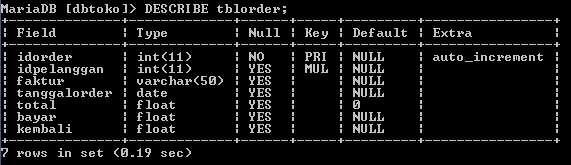
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_o4HtO7nbDO.png

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_xTru4oOev7.png

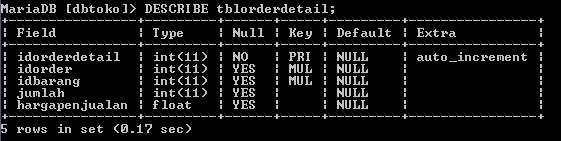
Tampilkan kembali data pelanggannya



Tampilkan struktur tabel [tblorder]



Tampilkan struktur tabel [tblorderdetail]

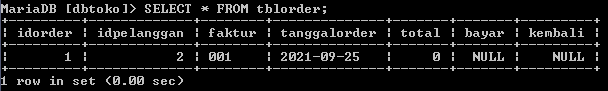


2). Pengujian insert pada tabel [tblorderdetail]

Buat order atas nama pelanggan [eka] dengan [idpelanggan = 2] pada tabel [tblorder] dengan cara memasukkan data seperti

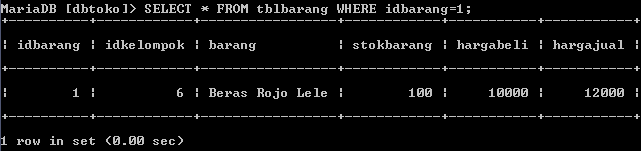
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_csKTu5FNOM.png

Periksa dengan perintah seperti



(Buat INSERT data pada tabel tblorderdetail dengan menggunakan idorder = 1 sesuai dengan yang ada di tabel tblorder dengan iDCLrang = 1 pada tabel tblbarang)

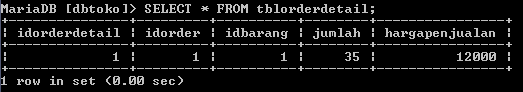
Tabel sebelum proses insert pada tabel tblorderdetail



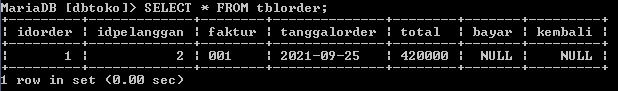
INSERT DATA pada tabel tblorderdetail

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_sJdm10Mlj7.png

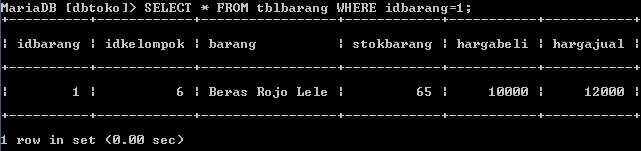
Hasil pada tabel tblorderdetail



Hasil pada tabel tblorder



Hasil pada tabel tblbarang

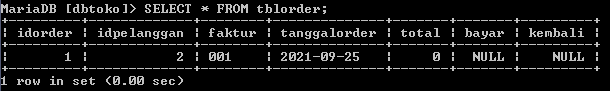


3). Pengujian delete pada tabel [tblorderdetail]

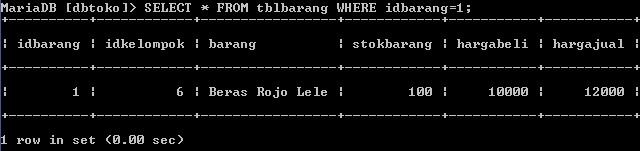
Melakukan DELETE pada tabel [tblorderdetail] dengan perintah seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_wwV6sSVBEa.png

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



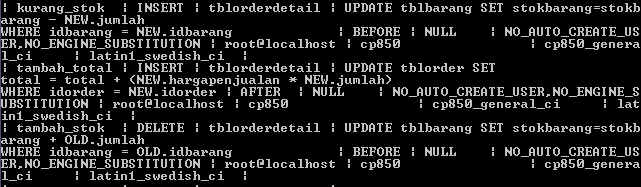
110. Menghapus trigger, bisa dengan menggunakan cara seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_2jCiE4lKSI.png

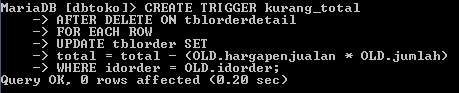
Periksa triggernya setelah dihapus

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_vd5ytohRfr.png

Hasil pemeriksaannya



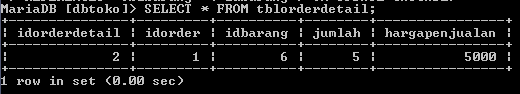
111. Join (Gabungan Tabel), sebelum belajar JOIN buat kembali Trigger yang telah dihapus



Setelah TRIGGER dibuat kembali, lakukan insert data kembali dengan data seperti



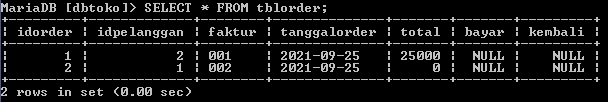
Periksa tabel [tblorderdetail]



Buat terlebih dahulu INSERT data pada tabel [tblorder] lagi dengan data seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_E6H2ErqFBD.png

Periksa tabel [tblorder]

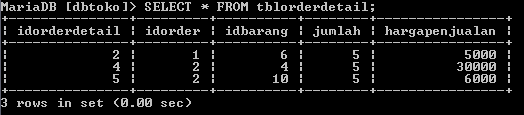


Melakukan insert data pada tabel [tblorderdetail]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\MNnmPv2Pvd.png

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_20W0sPMZMR.png

Periksa kembali tabel [tblorderdetail]



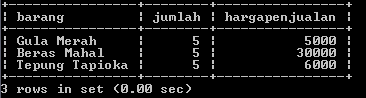
(Dari tabel diatas, yang masuk ke dalam kolom adalah kode yang sulit dipahami. Untuk memudahkan pembacaan pada tabel maka bisa dibuat gabungan tabel (JOIN) untuk memudahkan pembacaannya).

112. Inner Join, digunakan untuk mengambil bagian yang ada di tabel master dan tabel transaksi serta agar mengetahui barang apa saja yang sudah laku.

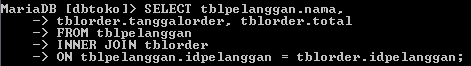
113. Pada saat Inner Join, menggunakan perintah SELECT (tabel master.kolom master, tblbarang.barang, tblorderdetail.jumlah), (tabel transaksi.kolom transaksi, tblorderdetail.hargapenjualan) FROM (table master, tblbarang) INNER JOIN (tabel transaksi, tblorderdetail) ON (tabel master.kolom master = tabel transaksi.kolom transaksi, tblbarang.idbarang = tblorderdetail.idbarang);. (Join harus dimulai dari tabel master).

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_7SCeFw0vwO.png

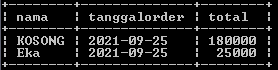
Hasil dari perintah tersebut ialah



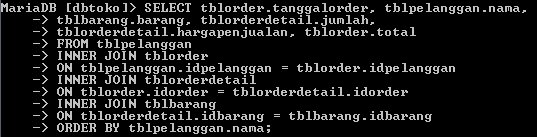
INNER JOIN untuk mengetahui pelanggan yang melakukan ORDER



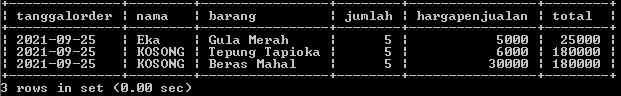
Nama pelanggan yang melakukan order



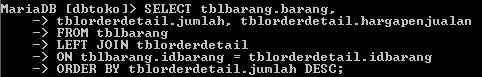
114. Inner Join Banyak Tabel, dengan menggunakan INNER JOIN banyak tabel bisa diketahui barang yang dibeli pelanggan dan total pembelian oleh pelanggan.



Hasil INNER JOIN dengan banyak tabel



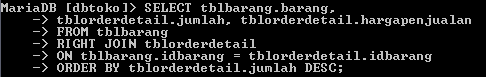
115. Left Join, digunakan untuk menampilkan semua yang ada di dalam tabel master dan mengambil sebagian yang ada di dalam tabel transaksi.



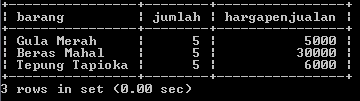
Hasil dari LEFT JOIN



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



Hasil RIGHT JOIN

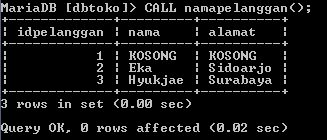


117. Store Procedure, ialah blok program yang diletakkan pada MySQL. Blok yang sudah dibuat bisa dipanggil jika diperlukan. Procedure sendiri tidak memiliki return.

=> Procedure Tanpa Parameter, dengan menggunakan perintah CREATE PROCEDURE (nama procedure(), namapelanggan()) ditekan enter kemudian ketik (Isi procedurenya, SELECT \* FROM tblpelanggan);.

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_4gfSHQ58uX.png

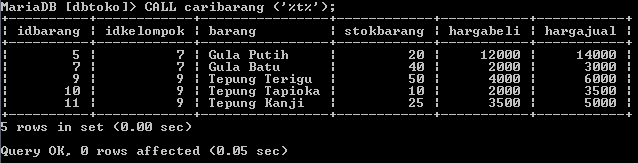
=> Pemanggilan Procedure, dengan menggunakan perintah CALL (nama procedurenya(), namapelanggan());



=> Procedure dengan parameter

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_0Z6Ii9XVqJ.png

Memanggil procedure yang sudah dibuat dengan parameter yang dimasukkan [‘%t%’] yang menunjukkan nama barang yang mengandung huruf [t].

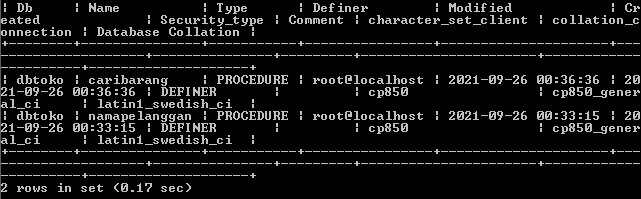


(Dengan menggunakan procedure, bisa membuat semua QUERY atau SELECT di dalam procedure sehingga memudahkan dalam pembuatan aplikasi).

118. Menampilkan semua procedure, MySQL bisa menampilkan semua isi procedure seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_BTw9R9SWfq.png

Procedure yang sudah dibuat bisa dilihat pada



119. Menghapus store procedure, untuk menghapus store procedure bisa menggunakan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_xHnfrUWyw0.png

Periksa kembali procedurenya dengan perintah



120. Function, ialah blok program yang disimpan di MySQL yang bisa menerima INPUT atau PARAMATER dan MEMILIKI RETURN.

Menyiapkan isi function, sebelum membuat function harus disiapkan terlebih dahulu code yang akan dibuat sebagai isi function sehingga hasil dari function sesuai dengan yang diharapkan. Kali ini akan membuat function menampilkan selisih [hargajual] dan [hargabeli] pada tabel [tblbarang].

Tampilkan tabel [tblbarang]



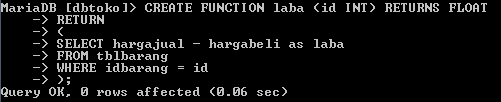
Pada tabel diatas akan ditambahkan satu (1) kolom lagi sebagai selisih dari [hargajual] dikurangi dengan [hargabeli]



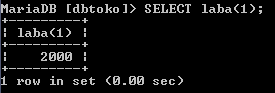
Selisih [hargajual] dan [hargabeli] ini disebut dengan [laba].



Setelah isinya dibuat, saatnya membuat function. Caranya dengan menggunakan perintah CRETAE FUNCTION (nama function, laba) ((parameter TIPE DATA, id INT)) RETURNS (TIPE DATA RETURN, FLOAT RETURN) ((isi function, SELECT hargajual – hargabeli as laba FROM tblbarang WHERE idbarang = id));.



Memanggil function, dengan menggunakn perintah SELECT FUNCTION (nama function (parameter), laba(1));.



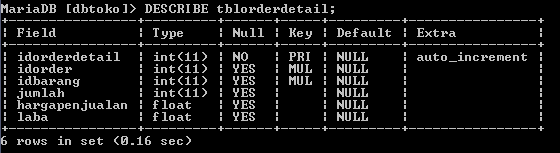
Penggunaan function, pertama bisa dilihat sahulu struktur tabelnya [tblorderdetail]



Menambahkan sebuah kolom dengan nama [laba]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_2jBgy78k36.png

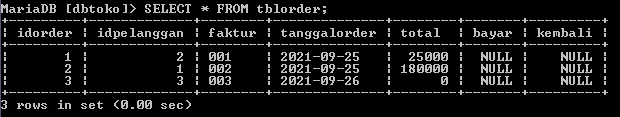
Periksa kembali dengan describe



Lakukan proses order dengan melakukan insert pada tabel [tblorder]

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_SHYK1SUSGE.png

Periksa tabel [tblorder]

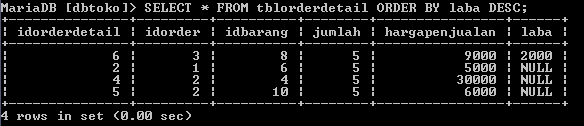


Membuat insert pada tabel [tblorderdetail]

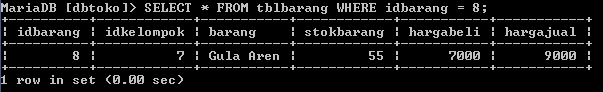
C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ckqkm5BQPs.png

(Pada saat INSERT kolom [laba] akan di isi dengan function yang menghitung [hargajual] dikurangi [hargabeli] dimana [iDLCrang] dimasukkan sebagai parameter input pada function).

Periksa kembali tabel [tblorderdetail]



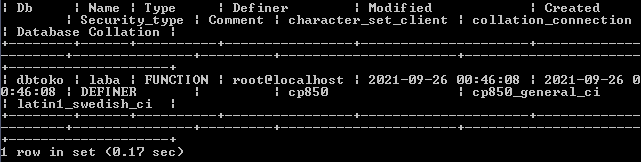
Periksa selisih [hargajual] - [hargabeli] pada tabel [tblbarang]. Jika sesuai berate function sudah berjalan sesuai dengan yang sudah diharapkan.



121. Menampilkan semua function, bisa menggunakan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_4QDiPQZNTe.png

Tampilan function yang telah dibuat



122. Menghapus function, bisa menggunakan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_gp07fn2YCr.png

Periksa kembali function yang telah dibuat

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_9EbBMk6u8O.png

123. Select Aggregate, ialah SELECT yang menampilkan function bawaan dari MySQL. Ada beberapa macam, yaitu

=> MIN, menampilkan nilai TERKECIL pada kolom yang dimaksud.

=> MAX, menampilkan nilai TERBESAR pada kolom yang dimaksud.

=> SUM, menampilkan nilai PENJUMLAHAN pada kolom yang dimaksud.

=> AVG, menampilkan nilai RATA - RATA pada kolom yang dimaksud.

=> COUNT, menampilkan nilai JUMLAH BARIS pada kolom yang dimaksud.

=> COUNT(\*), menampilkan nilai JUMLAH BARIS pada kolom yang dimaksud.

Menjalankannya bisa dengan menggunakan perintah SELECT (function\_aggregate(nama kolom));

(Antara nama AGGREGATE dan KURUNG tidak boleh ada SPASI).

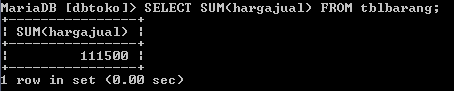
=> SELECT MIN



=> SELECT MAX



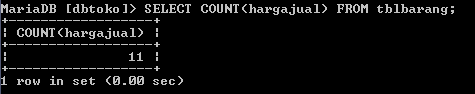
=> SELECT SUM



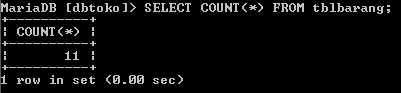
=> SELECT AVG



=> SELECT COUNT



=> SELECT COUNT(\*)



124. Select Beween (Select antara dua nilai), digunakan untuk menampilkan data antara dua nilai yang terendah dan yang tertinggi. Pada saat digunakan menggunakan perintah SELECT (nama kolom atau) \* FROM (nama tabel) WHERE (nama kolom) BETWEEN (awal) AND (akhir). Misalkan seperti



125. Select Distinct, digunakan untuk menampilkan data yang sama namun hanya satu kali saja. Pada saat digunakan menggunakan perintah SELECT DISTINCT (nama kolom, kelompok) FROM (nama tabel view, view\_barang);



126. Start Transaction, Commit, dan Rollback

Setiap perintah di dalam MySQL yang masuk ke dalam kelompok (INSERT, UPDATE, DELETE, SELECT) bisa dimasukkan ke dalam peirntah START TRANSACTION, COMMIT, dan ROLLBACK. Perintah yang dimulai dengan START TRANSACTION bisa dilakukan ROLLBACK (undo atau pembatalan suatu perintah). COMMIT sendiri digunakan agar perintah yang dijalankan tidak bisa di ROLLCBACK (undo).

Buka kembali tabel [tblkelompok]



Lakukan perintah berikut untuk memulai

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_B0bdv0HGm6.png

Berikan perintah INSERT seperti

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_ggrqaUqviP.png

Periksa data yang sudah dimasukkan



Berikan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_qH6IbyDY8B.png

Periksa kembali data yang sudah dimasukkan sebelumya



Ulangi kembali perintah INSERT sebelumnya



Dan ikuti dengan perintah

C:\Users\Owner\Documents\ShareX\Screenshots\2021-09\cmd_rKtgSiULyp.png

Periksa data yang sudah dimasukkan



Lakukan ROLLBACK dan periksa hasilnya.



(Transaksi yang sudah di commit tersebut sudah tidak bisa di ROLLBACK (undo)).

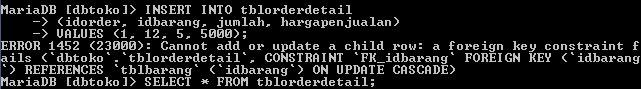
* Kesimpulannya
* Start Transaction merupakan function yang digunakan untuk memasukkan suatu data ke dalamnya (seperti data minuman di dalam tabel kelompok). Data yang dimasukkan tersebut akan dapat hilang bila dilakukan function Rollback (menjadi seperti tidak permanen).
* Commit merupan function yang digunakan untuk mengunci (menjadi seperti permanen) dari apa yang sudah dibuat oleh Start Transaction sebelumnya (jadi tidak akan bisa dihilangkan dengan Rollback).
* Rollback merupakan function yang digunakan untuk menghapus satu data yang baru saja dibuat sebelumnya (seperti data minuman pada tabel kelompok). Data tersebut akan bisa hilang pada saat dilakukan Rollback bilamana hanya menggunakan Start Transaction saja pada saat membuatnya. Namun bilamana sudah ditambahkan Commit, maka data tersebut tidak akan pernah bisa hilang bila dilakukan Rollback sekalipun.
* Transaksi ialah suatu perintah (INSERT, UPDATE, DELETE, dan SELECT).
* Transaksi yang bisa di ROLLBACK (undo), ialah transaksi yang dimuali dengan START TRANSACTION dan BELUM di akhiri dengan COMMIT.
* Transaksi yang sudah di COMMIT SUDAH TIDAK BISA di ROLLBACK.
* Penggunaan START TRANSACTION, COMMIT, dan ROLLBACK

Ketika database yang sudah 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 akan dikembalikan untuk pemrosesan modifikasi atau penambahan, sesuai dengan kebutuhan orang tersebut. Maka data yang VALID dari database tersebut TIDAK BOLEH dicampur dengan DATA DUMMY (data yang coba – coba), kaena sedang dilakukannya PENGUJIAN pada database tersebut. Agar DATA DUMMY tidak bercampur dengan data VALID, digunakanlah perintah START TRANSACTION sebelum melakukan suatu proses pada TRANSAKSI. Jika pengujiannya sudah selesai, lakukanlah ROLLBACK. Dan jangan pernah melakukan COMMIT jika data yang dimasukkan tersebut ialah DATA DUMMY.

**Saya Belum Mengerti**

1. Belum terlalu mengerti maksud dari bytes yang ada di dalam suatu tipe data.

2. Belum bisa menginsertkan idorder, idbarang, jumlah, hargapenjulan untuk yang kedua atau seterusnya pada tblorderdetail (tiba-tiba pada idorderdetailnya sudah terisikan angka 2, tidak berurutan mulai dari 1). (Namun pada saat menambahkannya untuk yang pertama kalinya bisa dan sudah masuk atau ada pada saat di cek).



3.