**Laporan Akhir Praktikum Sistem Basis Data**



Disusun oleh :

Dimas Kurniawan S (L200170032)

**Program Studi Informatika**

**Universitas Muhammadiyah Surakarta**

# MODUL 1 Pengenalan dan Instalasi MySQL

## E. Tugas

1. Jelaskan mengapa dibutuhkan data!

Data merupakan informasi yang isinya relevan yang merupakan hasil dari pengamatan, pengukuran, pencatatan. Data sangat dibutuhkan sebagai referensi ataupun sebagai perbandingan tentang suatu hal yang baru. Artinya jika ada suatu hal yang baru yang masih berua informasi, makan data-data yang lama akan menjadi acuan terhadap data-data yang baru

1. Jelaskan manfaat database dan contohnya!
   1. Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
   2. Menentukan kualitas informasi yaitu cepat, akurat, dan relevan, sehingga infromasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkanya.
   3. Mengatasi kerangkapan data (redundancy data).
   4. Menghindari terjadinya inkonsistensi data.
   5. Mengatasi kesulitan dalam mengakses data.
   6. Menyusun format yang standar dari sebuah data.
   7. Penggunaan oleh banyak pemakai (multiple user). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multiuser).
   8. Melakukan perlindungan dan pengamanan data. Setiap data hanya bisa diakses atau dimanipulasi oleh pihak yang diberi otoritas dengan memberikan login dan password terhadap masing-masing data.
   9. Agar pemakai mampu menyusun suatu pandangan (view) abstraksi dari data. Hal ini bertujuan menyederhanakan interaksi antara pengguna dengan sistemnya dan database dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer dan administratornya.

Contohnya: database dosen, database mahasiswa, database rumah sakit, database karyawan supermarket, dll.

1. Untuk menentukan jenis database yang digunakan, apa yang menjadi acuan dalam pemilihan database tersebut?
   1. Spesifikasi data
   2. Pemrosesan yang diperlukan oleh data
   3. Pertimbangan keamanan
   4. Kecocokan dengan tipe aplikasi
   5. Bahasa query
   6. Biaya tak langsung terhadap pemrosesan
2. Jelaskan istilah atau terminology yang digunakan dalam Database (database,table,field,record)

Database merupakan kumpulan dari file / tabel membentuk suatu database. Tabel merupakan Tempat untuk menyimpan data, tabel terdiri dari field dan record. Filed merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Record yaitu kumpulan dari field.

1. Bandingkan perbedaan pengolahan data secara manual dengan menggunakan system database. Pengolahan data manual masih menggunakan alat berupapa pensil, bolpoin, kertas, dll. Sehingga membutuhkan banyak berkas dan penyimpanan untuk menyimpan berbagai data dan informasi. Keamanannya pun lebih rentan, data terbatas untuk dibagi, integritas data kurang. Bersifat konkrit. Sedangkan pengolahan data pada sistem database lebih praktis, dapat meminimalkan duplikasi data, integritas data tinggi, keamanan lebih terjamin, dan sharing data lebih mudah.
2. Mengapa dibutuhkan DBMS?

DBMS berfungsi untuk mendeskripsikan data serta relasi, mendokumentasikan susunan serta pendefinisian data, mengorganisasikan dan menaruh data untuk akses yang selektif/diambil secara efektif, interaksi antara user dengan sumber daya data, menjamin perlindungan dan keamanan sumber daya data, memisahkan persoalan logical dan physical, dan memastikan sharing data pada beberapa user yang terhubung pada sumber daya data. Performance yang didapat dengan penyimpanan dalam bentuk DBMS cukup besar, sangat jauh berbeda dengan performance data yang disimpan dalam bentuk flat file. Disamping memiliki unjuk kerja yang lebih baik, juga akan didapatkan efisiensi penggunaan media penyimpanan dan memori

1. Pada percobaan diatas ada beberapa field yang tipe data dan ukurannya berbeda.Jelaskan!

Pada tabel Mhs, terdapat 6 field yaitu: NIM, NAMA, TEMPAT\_LAHIR, TANGGAL\_LAHIR, ALAMAT, TELEPON. Masing-masing field diatur sesuai tipe data yang dibutuhkan, pada field NIM diatur tipe data text, berarti pada record dapat diisikan data berupa teks huruf, angka, dan simbol-simbol lain. Ukuran tipe data pada field NIM diatur sepanjang 10, artinya digit maksimum yang dapat dimasukkan ke dalam record adalah 10 digit. Jika data yang dimasukkan lebih dari 10 digit, maka digit ke-11 dan berikutnya tidak akan terbaca. Untuk pengaturan maksimum defaultnya sendiri adalah sepanjang 255 digit. Kemudian pada field NAMA, sama-sama diatur tipe data text, namun panjang ukurannya berbeda, yaitu 30, artinya kita dapat memasukkan digit maksimum ke dalam record yaitu sepanjang 30 digit, melebihi 30 data ke-31 dan seterusnya tidak akan terbaca.

# MODUL 2 Perancangan Basis Data

## E. Tugas

1. Buatlah rancangan database untuk menangani data-data kuliah. Data – data yang akan ditanganinya adalah: data pribadi mahasiswa,data pribadi mengenai dosen,data mata kuliah dan data ruang kelas. Mahasiswa boleh mengambil lebih dari satu mata kuliah, dan satu mata kuliah boleh diambil oleh lebih dari satu mahasiswa sekaligus (*joint account*). Buatlah ER Diagram manual untuk kasus tersebut dari tahap 1 sampai tahap 4!

1. Menentukan entities (object-object dasar) yang perlu ada di database

* + Mahasiswa : menyimpan semua data pribadi semua mahasiswa
  + Dosen : menyimpan semua data pribadi semua dosen
  + Mata Kuliah : menyimpan informasi tentang semua mata kuliah
  + Ruang Kelas : menyimpan informasi tentang semua ruang kuliah

1. Menentukan attributes (sifat - sifat) masing-masing entity sesuai kebutuhan database

 Mahasiswa

* + - nama\_mhs : nama lengkap mahasiswa (varchar(45))
    - nim\_mhs : nomor id mahasiswa (interger) PK
    - alamat\_mhs : alamat lengkap mahasiswa (varchar(225))

 Dosen

* + - nama\_dosen : nama lengkap dosen (varchar(45))
    - nidn\_dosen : nomor id dosen (interger) PK
    - alamat\_dosen : alamat lengkap dosen (varchar(225))

 Mata\_Kuliah

* + - nama\_matkul : nama mata kuliah (varchar(45))
    - kode\_matkul : nomor kode mata kuliah (interger) PK

 Ruang\_Kuliah

* + - kode\_ruang : nomor ruang kuliah (interger)PK
    - kapasitas\_ruang :jumlah muatan setiap ruang (interger)PK
    - lokasi\_ruang :alamat suatu ruang (varchar(25))

1. Menentukan relationship (hubungan) antar entitas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Mahasiswa | Dosen | Mata\_Kuliah | Ruang\_Kuliah |
| Mahasiswa | - | 1 : n | m : n | - |
| Dosen |  | - | 1 : n | - |
| Mata Kuliah |  |  | - | 1 : n |
| Ruang Kelas |  |  |  | - |

Hubungan

 Dosen membimbing Mahasiswa

* + - Tabel utama : dosen
    - Tabel kedua : mahasiswa
    - Relationship : One-to-many (1 : n)
    - Attribute penghubung : nidn\_dosen (FK nidn\_dosen di mahasiswa)  Mahasiswa mengambil mata kuliah
    - Tabel utama : mahasiswa, mata\_kuliah
    - Tabel kedua : mahasiswa\_has\_mata\_kuliah
    - Relationship : many-to-many (m:n)
    - Attribute penghubung : kode\_matkul, nim\_mhs

(FK nim\_mhs,kode\_matkul di mahasiswa\_has\_mata\_kuliah)

 Dosen mengampu mata kuliah

* + - Tabel utama : dosen
    - Tabel kedua : mata\_kuliah
    - Relationship : one-to-many(1:n)
    - Attribute penghubung : nidn\_dosen (FK nidn\_dosen di mata kuliah)

 Mata kuliah memiliki ruang kuliah

* + - Tabel utama : mata kuliah
    - Tabel kedua : ruang kuliah
    - Relationship : one-to-many(1:n)
    - Attribute penghubung : kode\_matkul,kode\_ruang(FK kode\_matkul

di ruang kuliah)

1. Menggambar ERD Diagram

nama\_

dosen

nidn\_dosen

nama\_matkul

kode\_matkul

mengampu

alamat\_dosen

membimbing

mengambil

memiliki

nama\_mhs

memiliki

alamat\_mhs

lokasi\_ruang

kode\_ruang

nim\_mhs

kapasitas\_ruang

Ruang Kuliah

Mahasiswa

Mata Kuliah

Dosen

1. Ambil contoh sembarang database (harus berbeda untuk setiap mahasiswa). Buatlah rancangan ER Diagram manual database tersebut dari tahap 1 sampai 4, dengan ketentuan database minimal 4 buah entitas.
   1. Menentukan entities (object-object dasar) yang perlu ada di database
      * Buku : menyimpan semua data buku
      * Peminjaman : menyimpan informasi tentang semua peminjaman buku
      * Pengembalian : menyimpan informasi tentang semua pengembalian buku
      * Anggota : menyimpan semua data anggota
      * Pegawai :menyimpan semua data pegawai
   2. Menentukan attributes (sifat - sifat) masing-masing entity sesuai kebutuhan database
      * Buku
        + jdl\_buku : judul buku (varchar(45))
        + kode\_buku : nomor id buku (interger) PK
        + pnrbt\_buku : nama penerbit buku (varchar(45))
        + pngrng\_buku :nama lengkap pengarang buku(varchar(45))
      * Peminjaman
        + kode\_pinjam : nomor id pinjam (interger)PK
        + tgl\_pinjam : tanggal peminjaman buku (interger) PK
        + tgl\_kembali : tanggal pengembalian buku (interger)PK
      * Pengembalian
        + kode\_kembali : nomor id kembali (interger)PK
        + tgl\_kembali : tanggal pengembalian buku (interger) PK
      * Anggota
        + nama\_anggota : nama lengkap anggota (varchar(45))
        + id\_anggota :nomor id anggota (interger)PK
        + alamat\_anggota :alamat lengkap anggota (varchar(225))
        + kontak\_anggota :nomor telepon anggota (interger)PK
        + tgllahir\_anggota :tanggal lahir anggota (interger)PK
      * Pegawai
        + nama\_pegawai :nama lengkap pegawai (varchar(45))
        + id\_pegawai :nomor id pegawai (interger)PK
        + alamat\_pegawai :alamat lengkap pegawai (varchar(225))
        + kontak\_pegawai :nomor telepon pegawai (interger)PK
2. Menentukan relationship (hubungan) antar entitas

Hubungan

 Anggota meminjam buku

Tabel utama : anggota,buku

Tabel kedua : anggota\_has\_buku

Relationship : many-to-many (m:n)

Attribute penghubung : id\_anggota,kode\_buku (FK id\_anggota,kode\_buku di anggota\_has\_buku )

 Anggota melakukan transaksi peminjaman

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Buku | Peminjaman | Pengembalian | Anggota | Pegawai |
| Buku | - | 1:n | 1:n | m:n | - |
| Peminjaman |  | - | - | - | 1:n |
| Pengembalian |  |  | - | 1:n | - |
| Anggota |  |  |  | - | - |
| Pegawai |  |  |  |  | - |

Tabel utama : anggota

Tabel kedua : peminjaman

Relationship : one-to-many (1:n)

Attribute penghubung : id\_anggota (FK id\_anggota di peminjaman)

 Pegawai mengelola peminjaman

Tabel utama : peminjaman

Tabel kedua : pegawai

Relationship : one-to-many(1:n)

Attribute penghubung :kode\_peminjaman (FK kode\_peminjaman di pegawai)

 Anggota melakukan pengembalian

Tabel utama : anggota

Tabel kedua : pengembalian

Relationship : one-to-many(1:n)

Attribute penghubung : id\_anggota (FK id\_anggota di pengembalian)

 Pegawai mengelola pengembalian

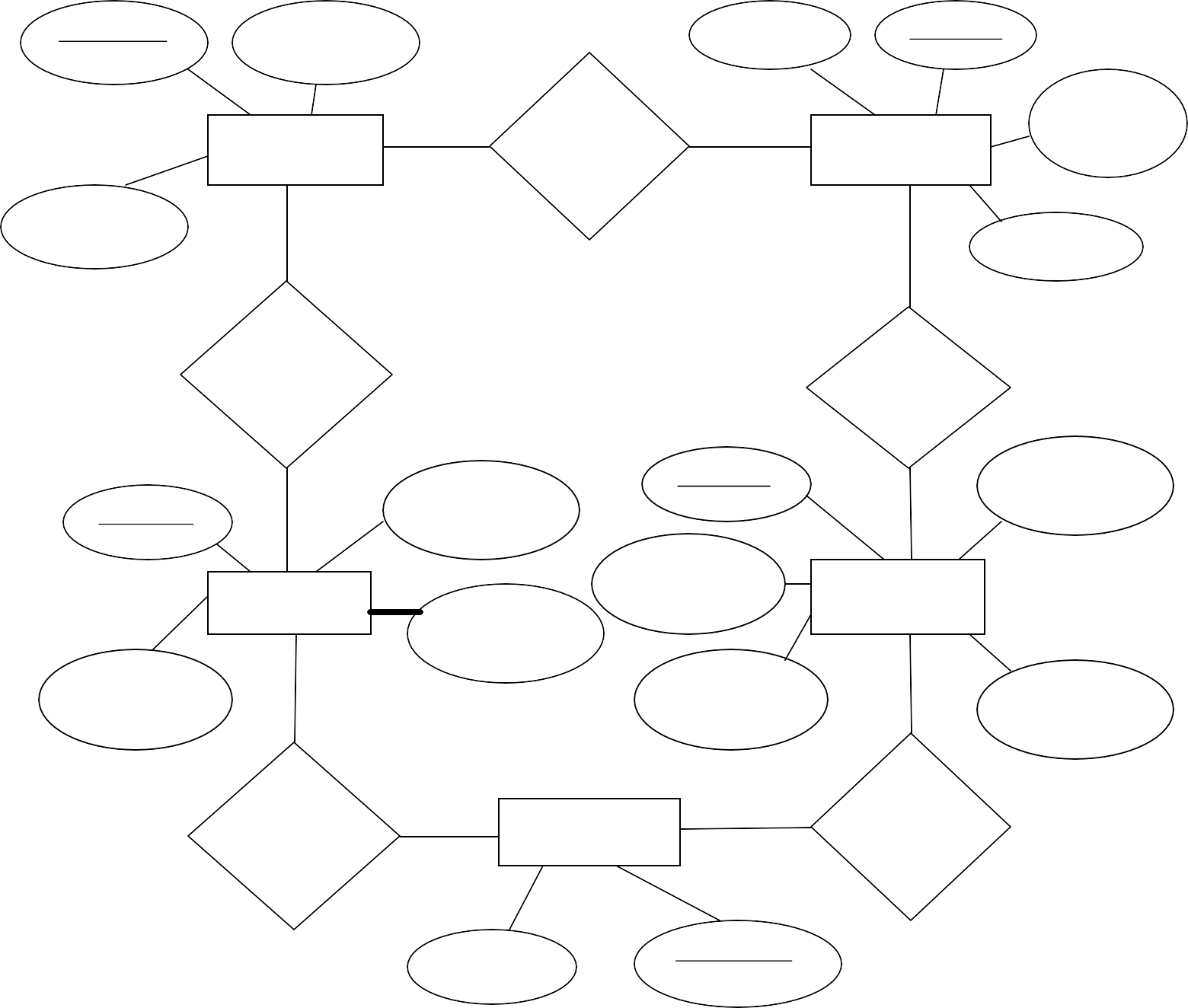
Tabel utama : pengembalian

Tabel kedua : pegawai

Relationship : one-to-many(1:n)

Attribute penghubung : kode\_pengembalian(FK kode\_pengembalian di pegawai)

1. Menggambar ERD Diagram



kode\_pinjam

tgl\_kembali

jdl\_buku

kode\_buku

Peminjaman

Transaksi

Peminjam

an

Buku

pngrng\_bu

ku

tgl\_pinjam

pnrbt\_buku

Mengelola

peminjam

an

Meminja

m

id\_pegawai

nama\_pegaw

ai

id\_anggota

nama\_anggot

a

kontak\_pega

wai

Pegawai

alamat\_pega

wai

tgllahir\_angg

ota

kontak\_angg

ota

Anggota

alamat\_angg

ota

Mengelola

pengemba

lian

Pengembalian

Mengem

balikan

buku

tgl\_kembali

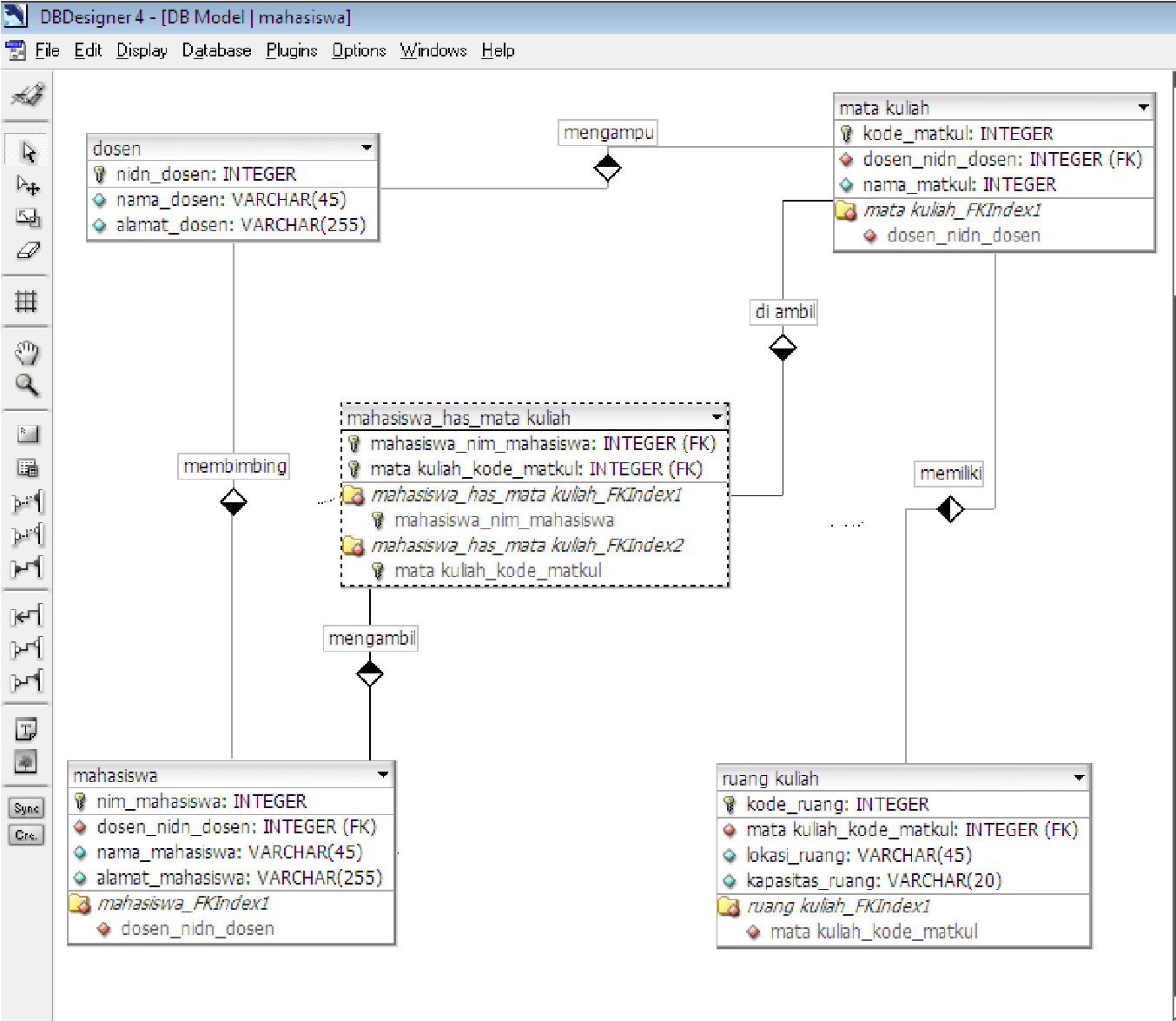
kode\_kembali

# MODUL 3 Perancangan E-R Diagram dengan DBDesigner

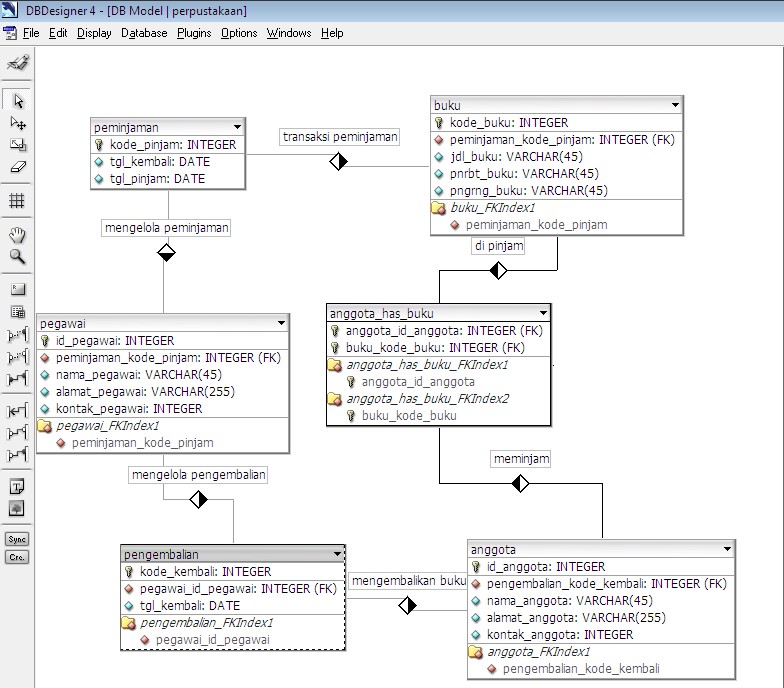
## E. Tugas

Dari tugas pada modul 3 soal nomer 1 dan 2, buatlah rancangan basis data dari tahap 1 sampai tahap 4. Gunakan program DB Designer untuk membuat ER Diagram.

1. Perkuliahan



1. Perpustakaan



# Modul 4 Data Definition Language(DDL)

## E. Tugas

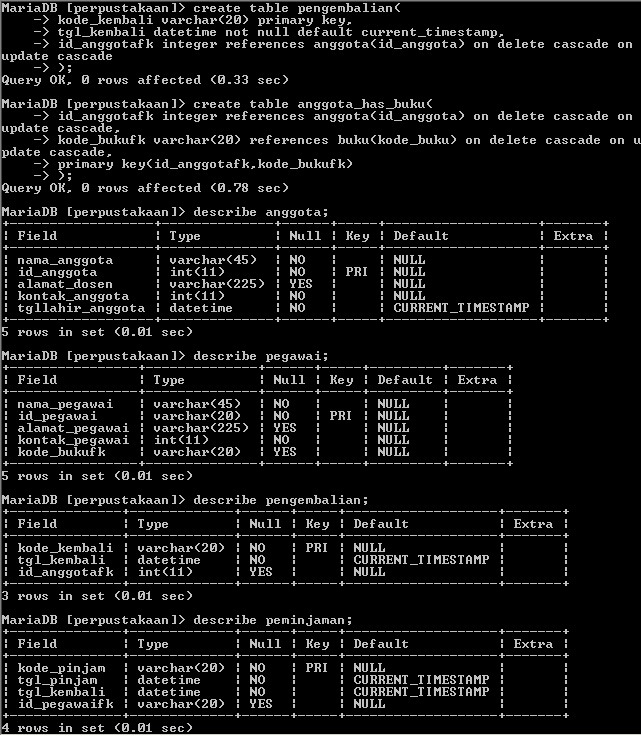
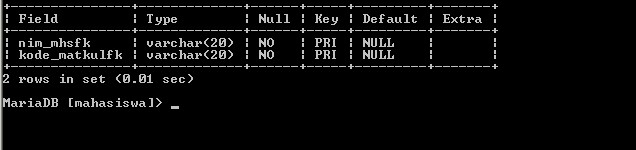
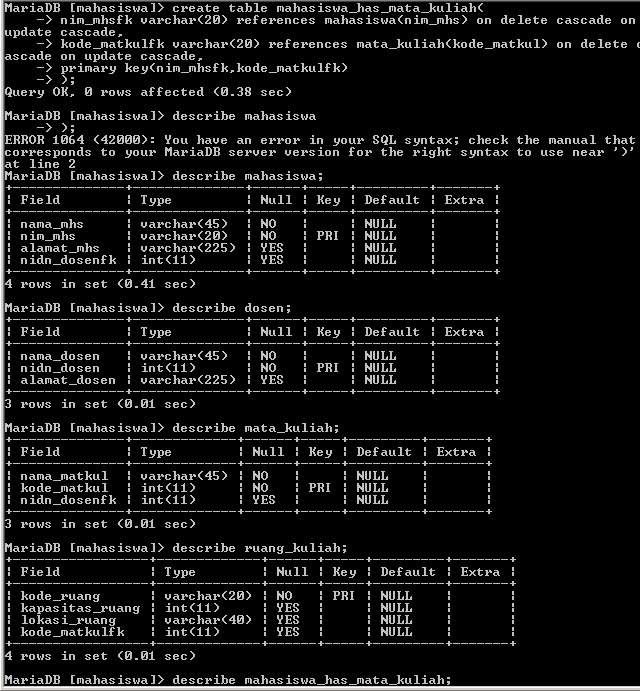
Implementasikan hasil rancangan database yang menangani data kuliah pada tugas modul 2 ke dalam program mysql.

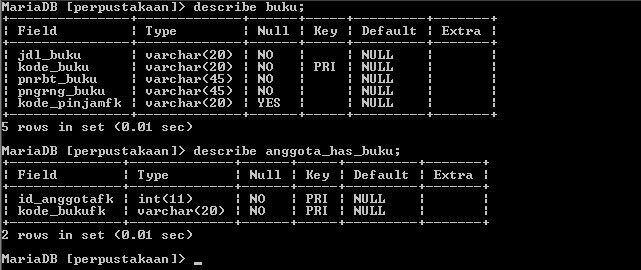
## Langkah – langkah Praktikum

1. Jalankan XAMPP Control Panel
2. Jalankan server Apache dan MySQL
3. Buka Command Prompt dan login sebagai root ke MySQL
4. Buat database baru dengan perintah create database mahasiswa;
5. Hubungkan ke dalam database yang telah dibuat dengan perintah berikut. Sehingga akan muncul pemberitahuan “database changed”. Use mahasiswa;
6. Membuat tabel dosen,mahasiswa,mata kuliah, ruang kuliah dengan script berikut :



1. Untuk melihat struktur tiap tabel dapat dilakukan dengan perintah describe. Misalkan untuk melihat struktur tabel mahasiswa dapat dilakukan dengan perintah describe mahasiswa;





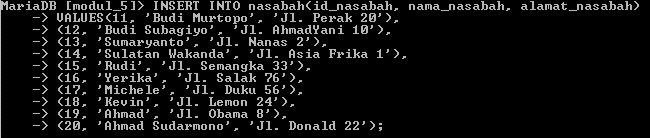
# MODUL 5 Data Manipulation Language (DML)

## E. Tugas

Masukkan beberapa record ke setiap tabel dalam database yang telah anda buat pada modul 4. Print out hasil implementasi rancangan tersebut dan analisa hasilnya. (Masing-masing 10 record untuk tabel master dan 20 record untuk tabel transaksi)

1. Menambah nasabah.

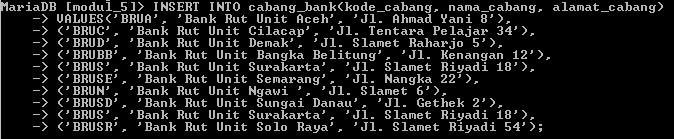
Kode :



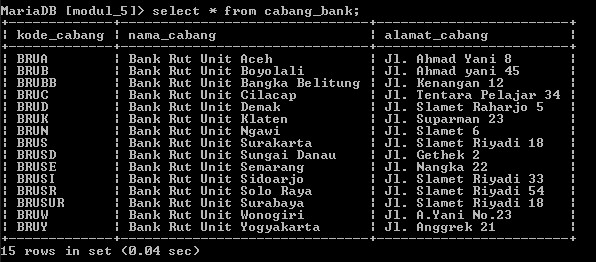
Hasil :



1. Menambah cabang bank. Kode :

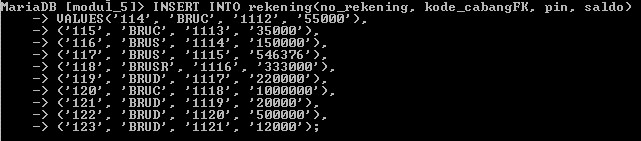


Hasil :

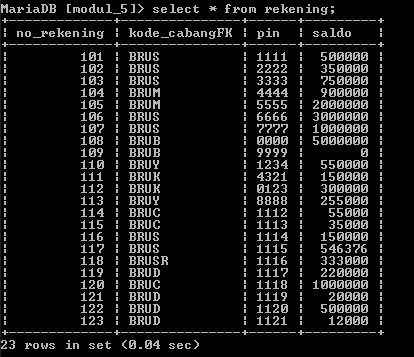


1. Menambah rekening.

Kode :



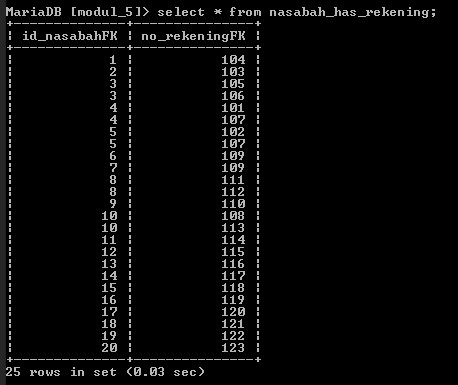
Hasil :



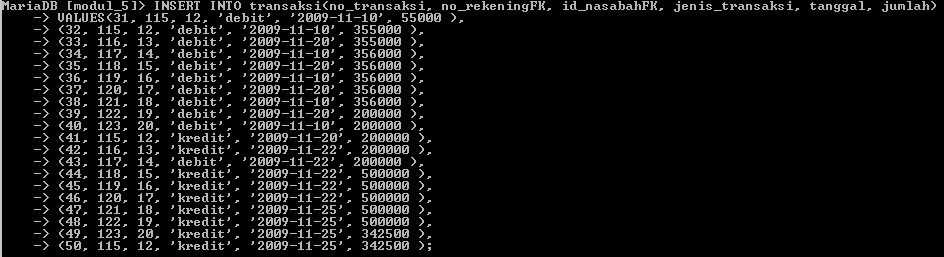
1. Menambah nasabah\_has\_rekening. Kode :



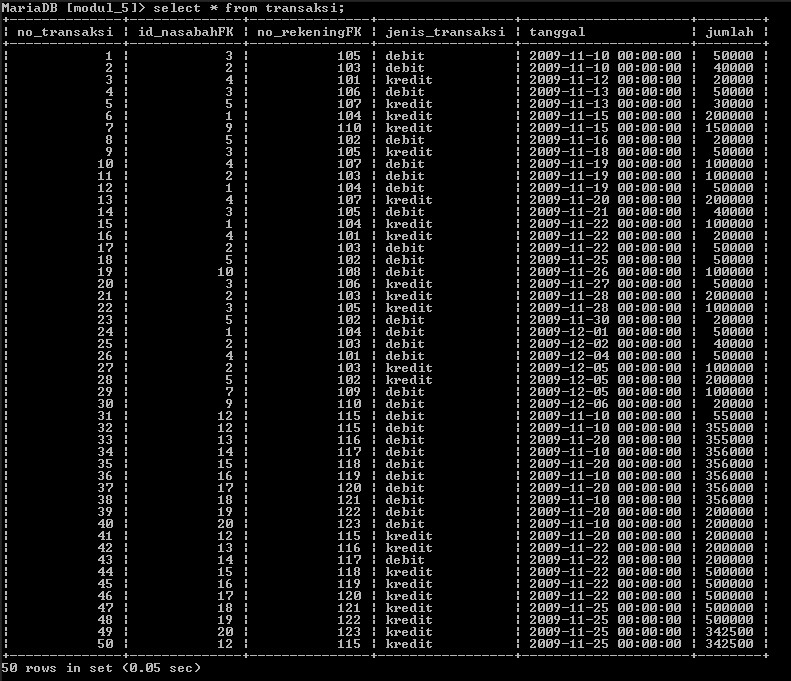
Hasil :



1. Menambah transaksi. Kode :



Hasil :

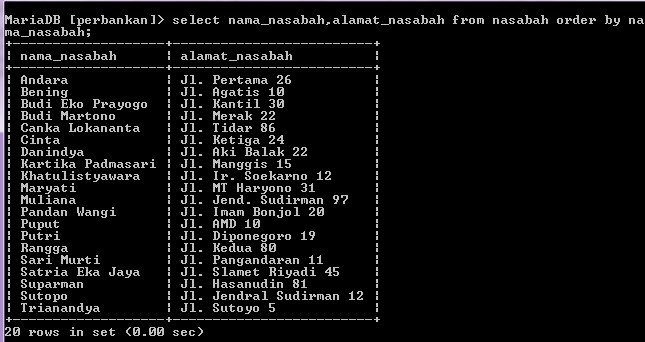


kesimpulan dari percobaan diatas adalah saat kita memasukkan record ke dalam database maka secara otomatis data di dalam database bertambah dan sesuai pada tabel yang ada

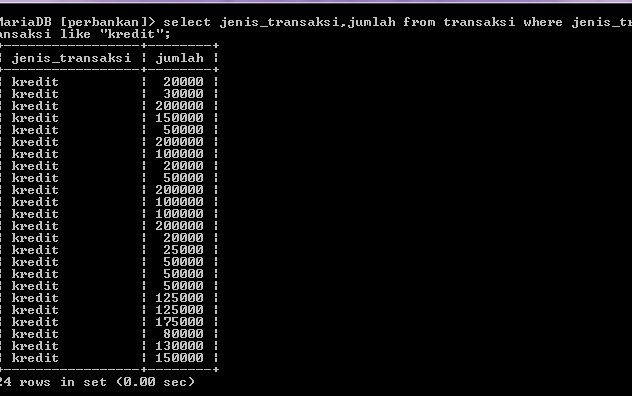
# MODUL 6 Query Standart dan Variasinya

## E. Tugas

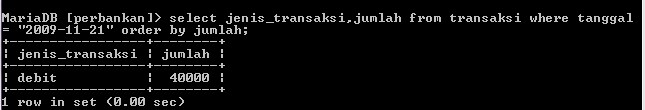
1. Tampilkan nama nasabah dan alamat nasabah diurutkan berdasarkan nama nasabah!



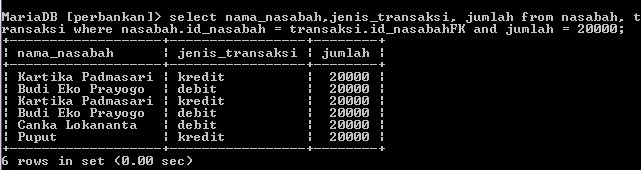
1. Tampilkan jenis transaksi dan jumlah transaksi dimana jenis transaksinya adalah kredit!



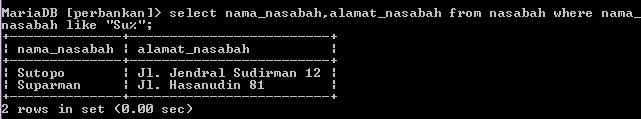
1. Tampilkan jenis transaksi dan jumlah transaksi yang melakukan transaksi pada tanggal 21 November 2009 dan diurutkan berdasarkan jumlah transaksi!



1. Tampilkan nama nasabah, jenis transaksi dan jumlah transaksi dimana jumlah transaksi = Rp 20.000!



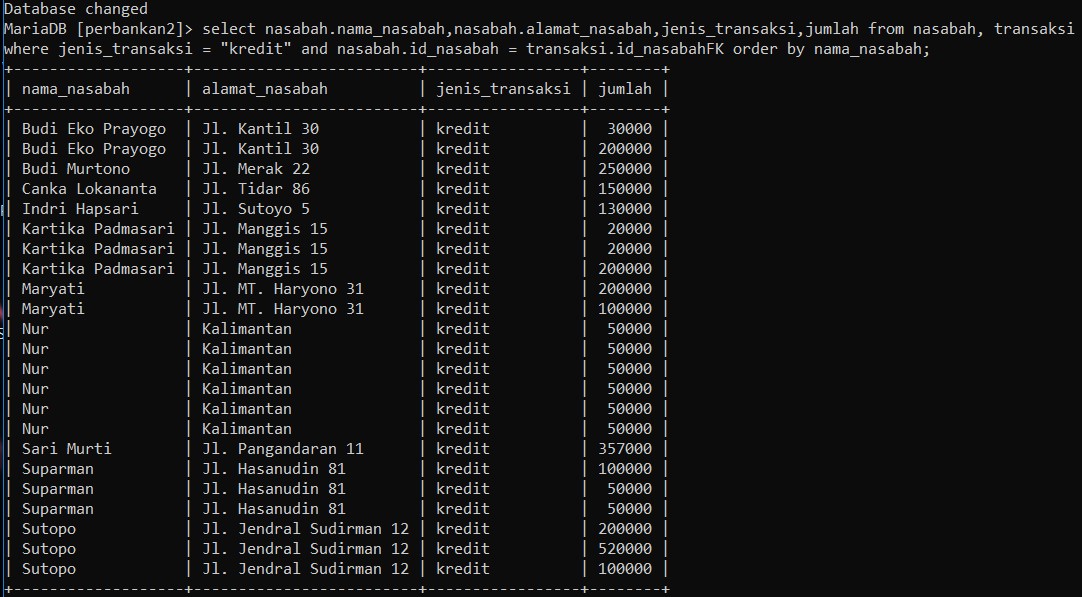
1. Tampilkan nama nasabah dan alamat nasabah dimana nama nasabah diawali dengan kata ‘Su’!



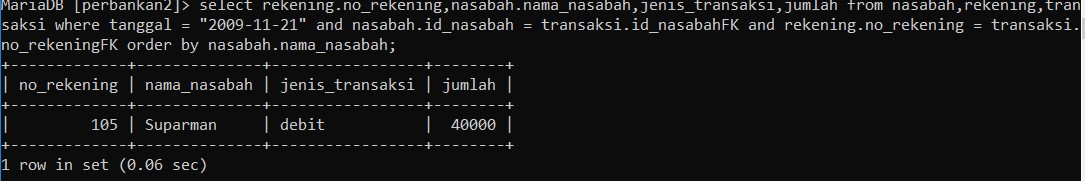
# MODUL 7 JOIN

## E. Tugas

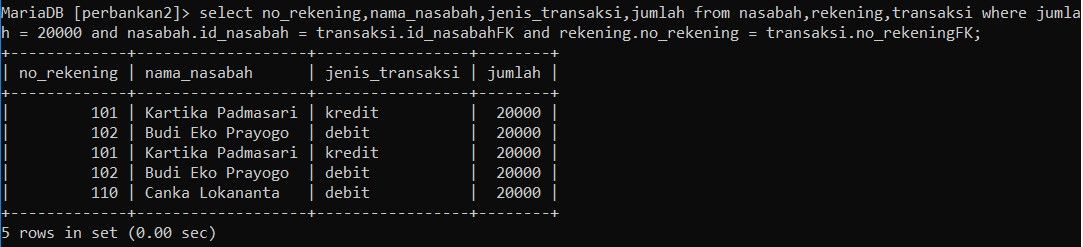
1. Tampilkan nama nasabah, alamat nasabah, jenis transaksi dan jumlah transaksi dimana jenis transaksinya adalah kredit dan diurutkan berdasarkan nama nasabah!



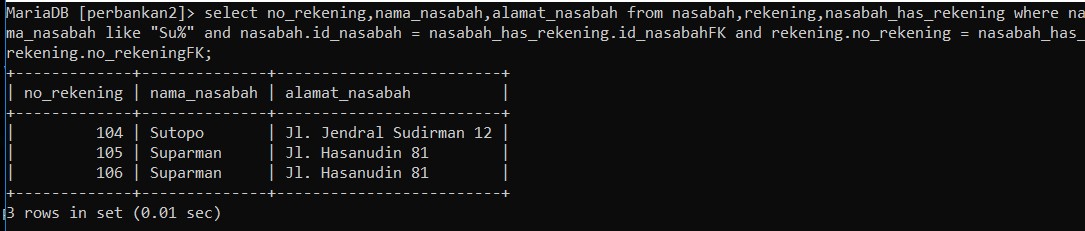
1. Tampilkan nomor rekening , nama nasabah, jenis transaksi dan jumlah transaksi yang melakukan transaksi pada 21 November 2009 dan diurutkan berdasarkan nama nasabah!



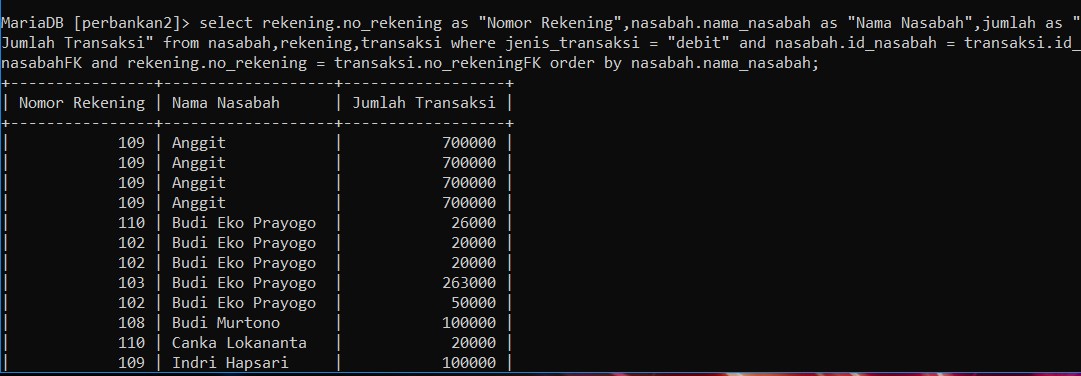
1. Tampilkan nomor rekening, nama nasabah, jenis transaksi dan jumlah transaksi dimana jumlah transaksi = Rp. 20.000!



1. Tampilkan nomor rekening, nama nasabah dan alamat nasabah dimana nama nasabah diawali dengan kata ‘Su’!



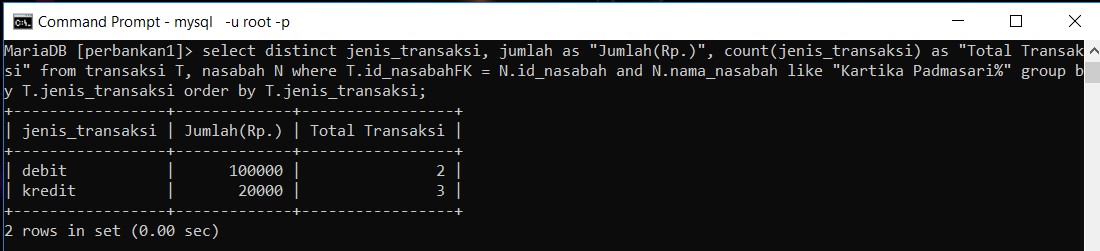
1. Tampilkan nomor rekening dengan alias ‘Nomor Rekening’, nama nasabah dengan alias’Nama Nasabah’, jumlah transaksi dengan alias ’Jumlah Transaksi’ dimana jenis transaksinya adalah debit! Urutkan berdasarkan nama nasabah!



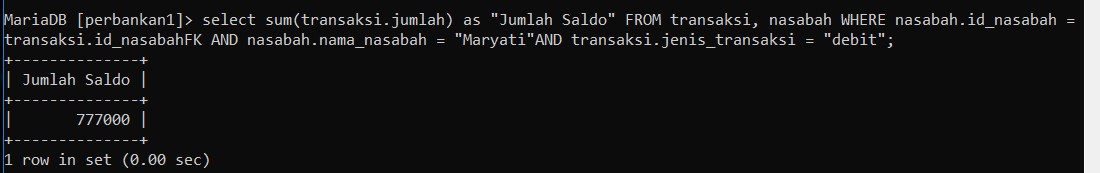
# MODUL 8 Aggregasi

## E. Tugas

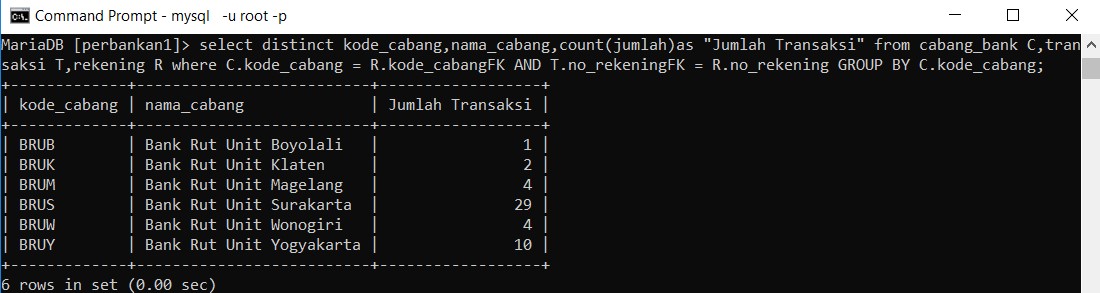
1. Tampilkan jenis transaksi, jumlah transaksi dalam Rp dan total transaksi untuk nasabah yang bernama akhiran ‘Kartika Padmasari’ untuk masing-masing jenis transaksi!



1. Berapa jumlah total salso yang dimiliki oleh Maryati?



1. Tampilkan jumlah transaksi yang ditangani oleh masing-masing cabang bank!

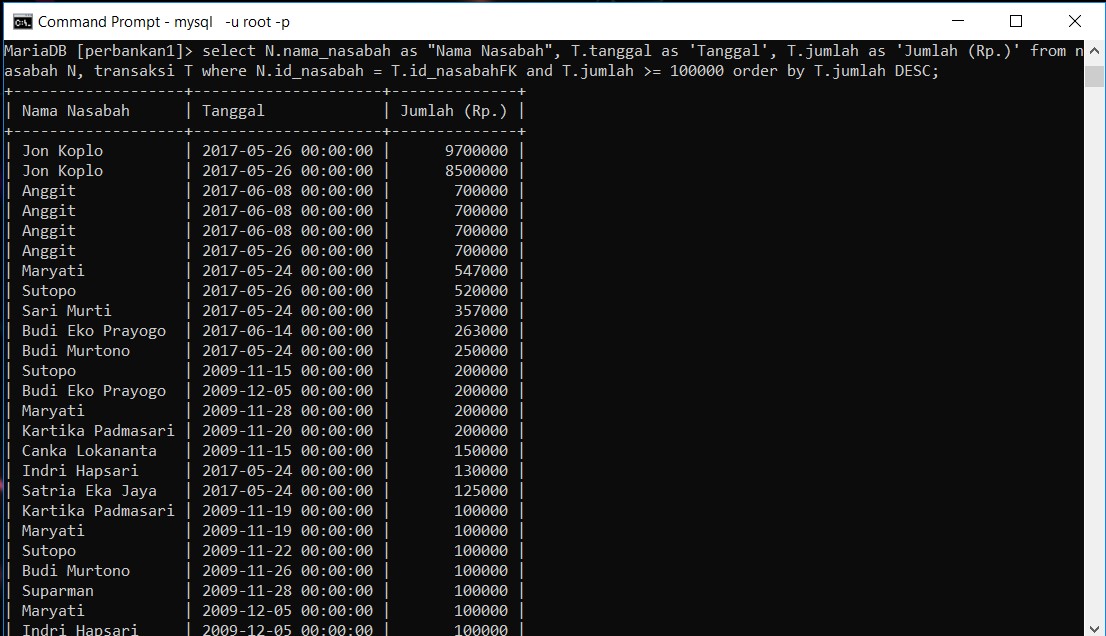


1. Tampilkan nama nasabah dan jumlah saldo yang memiliki saldo antara Rp. 500.000 sampai Rp.

2.000.000!



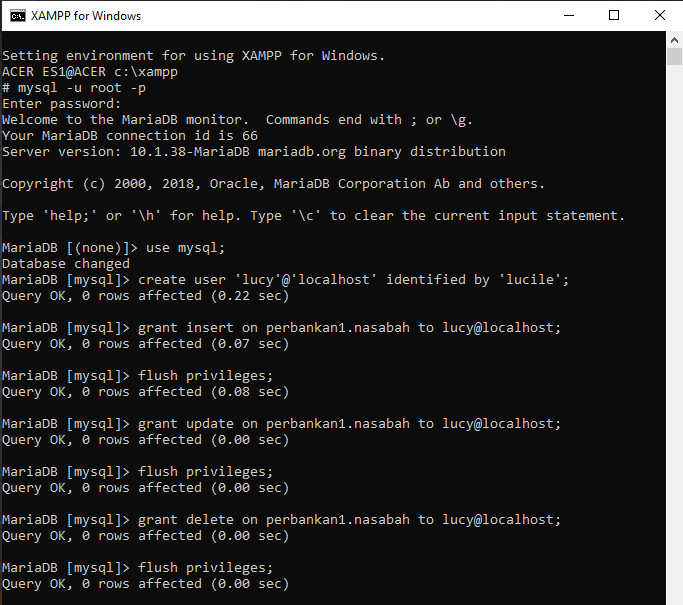
1. Tampilkan nama nasabah,tanggal transaksi dan jumlah transaksi dalam Rp dimana jumlah transaksi di atas Rp 100.000 dan urutkan berdasarkan jumlah transaksi dari yang besar ke yang kecil!



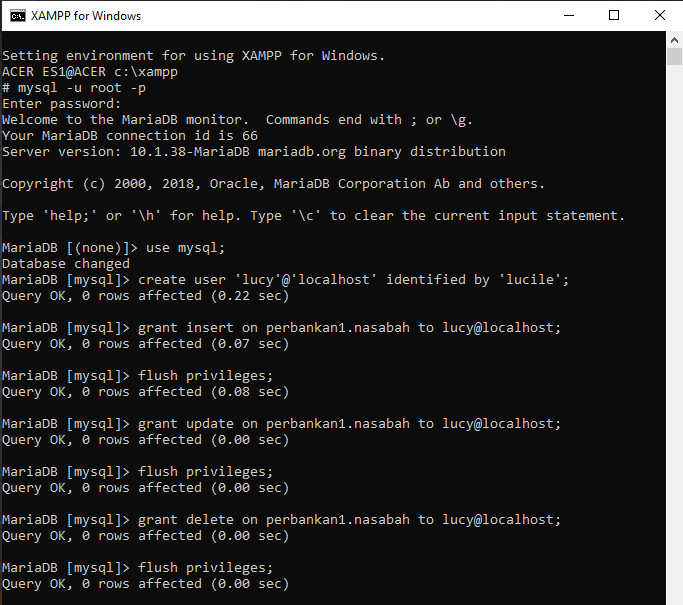
# MODUL 9 Data Control Language

## E. Tugas

1. Buat user baru sesuai dengan nama anda masing - masing



1. Beri Privilege untuk user tersebut dengan ketentuan sebagai berikut ini.
   1. Dapat melakukan insert,update,dan delete pada tabel nasabah.



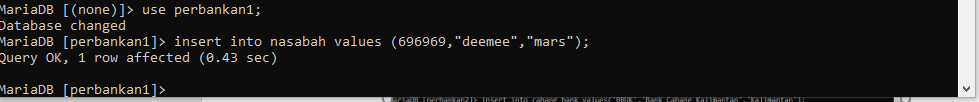
* 1. Hanya dapat melakukan select pada tabel cabang\_bank.



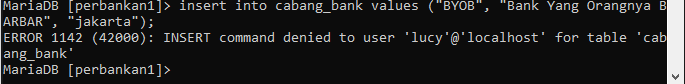
1. Coba lakukan kasus berikut ini dengan login sebagai user yang dibuat dan screenshot hasilnya.
   * 1. Lakukan perintah select pada tabel nasabah.



* + 1. Lakukan perintah insert pada tabel nasabah.



* + 1. Lakukan perintah insert pada tabel cabang\_bank



1. Analisa hasil percobaan kalian tersebut

Grant memiliki fungsi untuk membuat user baru dan memberikan hak istimewa. Grant merupakan privilege untuk tabel. Dimana dalam percobaan ini apabila sebelum melakukan hak akses pada sebuah tabel maka harus membuat GRANT dengan privilege yang dibutuhkan terlebih dahulu. Apabila tidak membuat grant dengan privilege terlebih dahulu maka hasilnya akan eror. Revoke digunakan untuk menghapus batasan akses yang telah diatur menggunakan perintah GRANT.

# MODUL 10 SubQuery

## E. Tugas

Gunakan sub query untuk mendapatkan data-data berikut dari database data kuliah yang telah anda bangun pada modul 4 dan 5

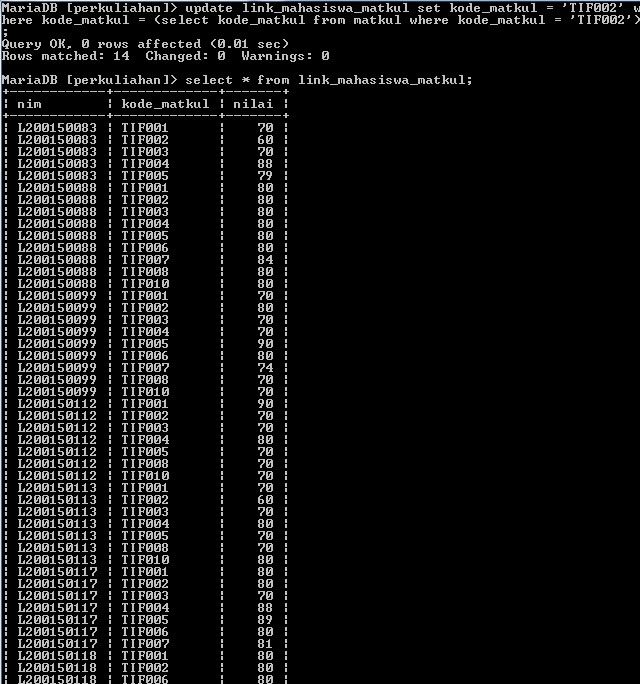
1. Ambil salah satu mata kuliah. Tampilkan daftar mahasiswa yang tidak mengambil mata kuliah tersebut.

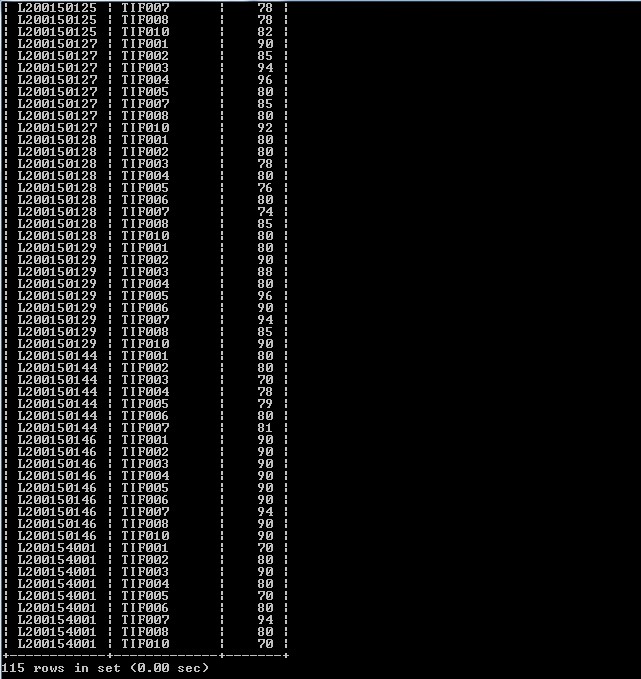


1. Satu dosen dapat mengmpu lebih dari satu mata kuliah dan satu mahasiswa dapat mengambil lebih dari datu mata kuliah. Tampilkan daftar mahasiswa yang mengambil semua mata kuliah yang diampu oleh saah satu dosen!



1. Karena salah satu mata kuliah(A) dihilangkan, seluruh mahasiswa yang mengambil mata kuliah tersebut dipindahkan untuk mengambil mata kuliah lain (B). Lakukan update data menggunnakan sub query!





# MODUL 11 MySQL dan Python

## E. Tugas

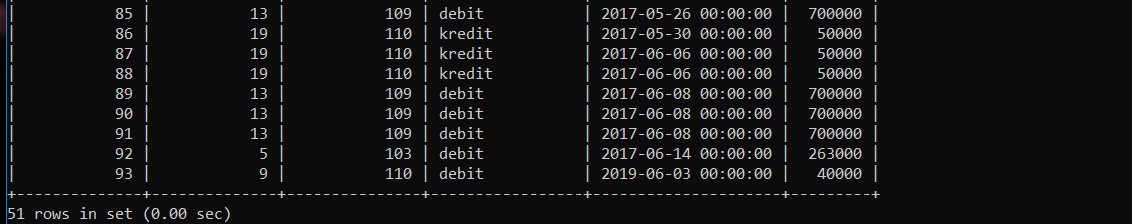
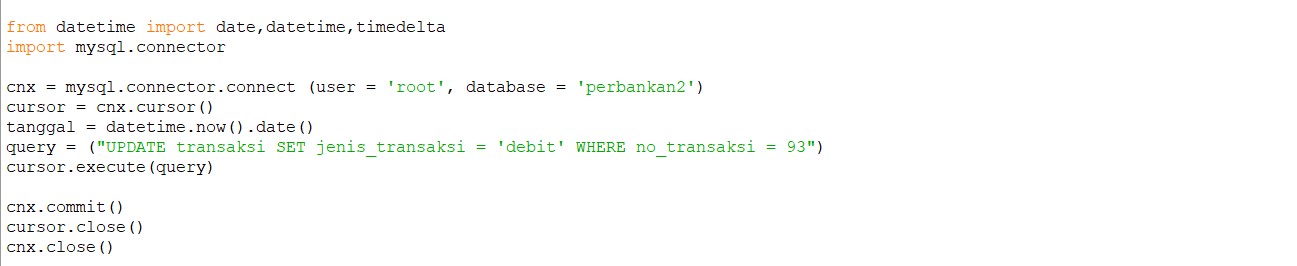
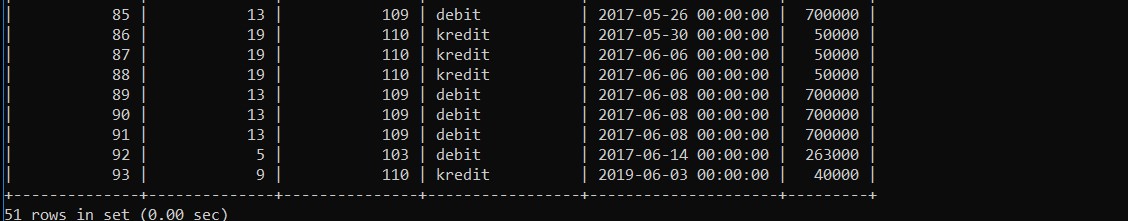
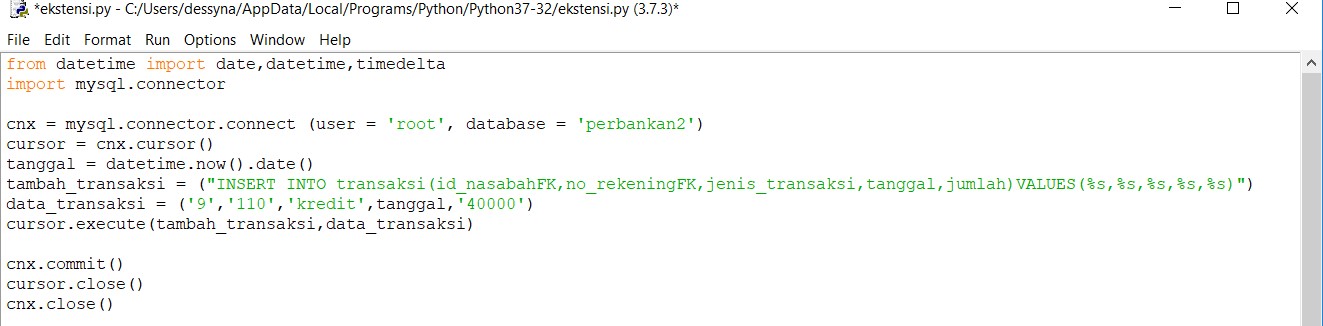
1. Buat kode program python untuk melakukan perintah INSERT, UPDATE, dan DELETE pada data transaksi.

a.

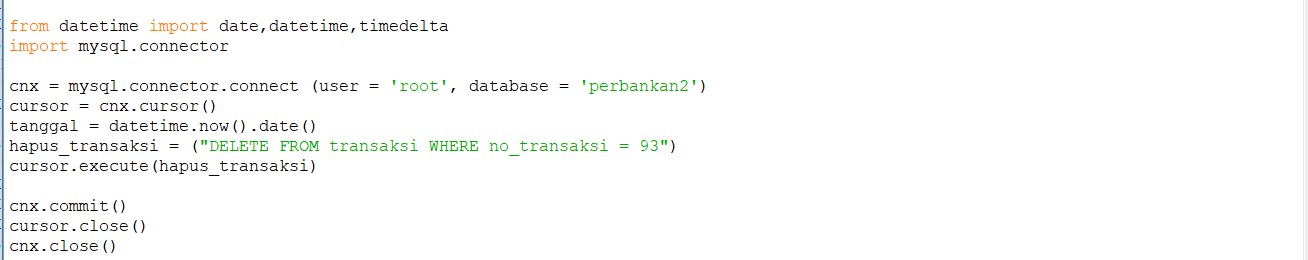
Perintah insert

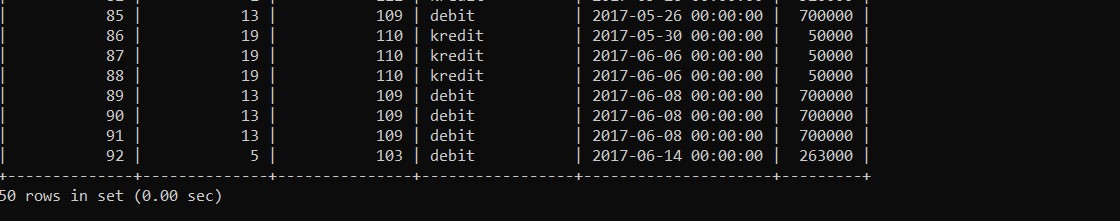
b.

Perintah update

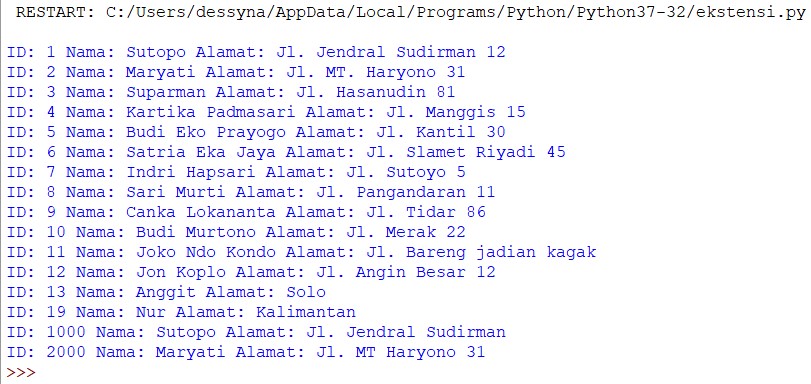


c. Perintah delete





1. Buatlah kode program pyhton untuk mendapatkan :
   1. Data nasabah



* 1. Data nasabah yang melakukan transaksi antara oktober sampai desember

