Tugas Praktikum Sistem Basis Data



Nama : Hanif Abdul A  
NIM : L200160118

Kelas : A

# MODUL 1

1. Jelaskan mengapa dibutuhkan data! Jawab :

Karena data di perlukan dalam segala hal, baik berupa pengukuaran, pencatatan, pengumpulan informasi, maupun pengambilan keputusan.

1. Jelaskan manfaat database dan contohnya! Jawab :
2. mempermudah dalam pencarian data dan penyimpanan data.
3. Sebagai komponen utama dalam Sistem Informasi
4. Menentukan kualitas Informasi
5. Mengatasi kerangkapan data.
6. Menghindari terjadinya inkonsistensi data.
7. Mengatasi kesulitan dalam mengakses data.
8. Menyusun format yang standart bagi sebuah data.
9. Penggunaan oleh banyak pemakai.
10. Melakukan perlindungan dan keamanan data.
11. Agar pemakai mampu menyusun suatu pandangan abstraksi dari data. Contohnya : Database Online Shop, Databes daftar buku di Perpustakaan,

Database Data Penduduk.

3. Untuk menentukan jenis database yang di gunakan, apa yang menjadi acuan dalam pemilihan database tersebut?

Jawab :

1. Pendeskripsian kebutuhan informasi dan data.
2. Spesifikasi data.
3. Pemrosesan yang di perlukan oleh data.
4. Pertimbangan keamanan.
5. Kecocokan dengan tipe aplikasi.
6. Bahasa query.
7. Biaya tak langsung terhadap pemrosesan.
8. Jelaskan istilah atau terminology yang digunakan dalam Database (database, table, filed, record)!

Jawab :

* Database : kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain, sehingga membentuk data untuk menginformasikan suatu perusahaan dan instansi.
* Tables : hal yang paling mendasar dalam hal penyimapanan data yang terdiri dari field dan record.
* Field (kolom): elemen dari table yang berisikan informasi tertentu yang spesifik tentang subjudul table pada sebuah item data.
* Record (baris) : sekumpulan data yang berkaitan tentang sebuah subjek tertentu.

1. Bandingkan perbedaan pengolahan data secara manual dengan menggunakan system database.

Jawab :

|  |  |
| --- | --- |
| Manual | Database |
| 1. Program Oriented | 1. Data Oriented |
| 2. Kaku | 2. Luwes |
| 3. Adanya kerangkapan data | 3. Terkontrolnya kerangkapan |

1. Mengapa dibutuhkan DBMS? Jawab :

Karena DBMS merupakan software yang digunakan untuk membangun sebuah system basis data yang berbasis komputerisasi dan membantu dalam pemeliharaan dan pengolahan kumpulan data dalam jumlah besar.

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

Jawab :

* 1. Terdapat Field “Nama” dengan Type : char(30)

 Maksudnya field “Nama” bertipekan char karena field tersebut nantinya dapat diisi huruf, angka, symbol-symbol lain. Ukuran tipe data pada field “Nama” diatur sepanjang 30, artinya kita dapat memasukan digit max kedalam record yaitu sepanjang 30 digit.

* 1. Terdapat Field “Pass” dengan Type : char(32)

 Maksudnya field “Pass” bertipekan char karena field tersebut nantinya dapat diisi huruf, angka, symbol-symbol lain. Ukuran tipe data pada field “Pass” diatur sepanjang 32, artinya kita dapat memasukan digit max kedalam record yaitu sepanjang 32 digit.

# MODUL 2

**No 1**

* + 1. **Menentukan entitas :**
       - Mahasiswa = Menyimpan semua data pribadi Mahasiswa
       - Dosen = Menyimpan semua data pribadi Dosen
       - Mata Kuliah = Menyimpan semua informasi mata kuliah
       - Ruang kelas = Menyimpan semua informasi mengenai ruang kelas

# Menentukan attribute :

* + - * Mahasiswa :
        + NIM = Nomor Induk Mahasiswa (varchar(10)) PK
        + Nama = Nama lengkap Mahasiswa (varchar(45))
        + Alamat = Alamat lengkap Mahasiswa (varchar(255))
      * Dosen :
        + NIP = Nomor Induk Pegawai (integer(20)) PK
        + Nama = Nama lengkap Dosen (varchar(45))
        + Alamat = Alamat lengkap Dosen (varchar(255))
      * Mata kuliah :
        + Kode Mata Kuliah = Kode untuk Mata Kuliah (varchar(10)) PK
        + Nama Mata Kuliah = Nama dari Mata Kuliah (varchar(45))
      * Ruang kelas :
        + Kode Kelas = Kode masing-masing kelas (varchar(10)) PK
        + Jumlah kursi = Jumlah kursi setiap kelas (integer(3))

# Menentukan Relationship :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Mahasiswa | Dosen | Mata Kuliah | Ruang Kelas |
| Mahasiswa |  |  |  |  |
| Dosen | 1:n |  |  |  |
| Mata Kuliah | n:n | n:n |  |  |
| Ruang Kelas | 1:n | 1:1 | n:1 |  |

Hubungan

* + - * **Dosen** mengajar **Mahasiswa** :
        + Tabel utama = Dosen
        + Tabel kedua = Mahasiswa
        + Relationship = One to many (1:n)
        + Attribute penghubung = NIP, NIM (FK NIM di Mahasiswa)
      * **Mata Kuliah** diambil **Mahasiswa** :
        + Tabel utama = Mata Kuliah, Mahasiswa
        + Tabel kedua = Mata Kuliah\_has\_Mahasiswa
        + Relationship = Many to many (n:n)
        + Attribute penghubung = Kode Mata Kuliah, NIM (FK Kode Mata Kuliah, NIM di Mata Kuliah\_has\_Mahasiswa)
      * **Mata Kuliah** dipilih **Dosen** :
        + Tabel utama = Mata Kuliah, Dosen
        + Tabel kedua = Mata Kuliah\_has\_Dosen
        + Relationship = Many to many (n:n)
        + Attribute penghubung = Kode Mata Kuliah, NIP (FK Kode Mata Kuliah, NIP di Mata Kuliah\_has\_Dosen)
      * **Ruang Kelas** dipakai **Mahasiswa** :
        + Tabel utama = Ruang Kelas
        + Tabel kedua = Mahasiswa
        + Relationship = One to many (1:n)
        + Attribute penghubung = Kode Kelas, NIM (FK Kode Kelas di Mahasiswa)
      * **Ruang Kelas** digunakan **Dosen** :
        + Tabel utama = Ruang Kelas
        + Tabel kedua = Dosen
        + Relationship = One to one (1:1)
        + Attribute penghubung = Kode Kelas, NIP
      * **Mata Kuliah** mengisi **Ruang Kelas**:
        + Tabel utama = Mata Kuliah
        + Tabel kedua = Ruang Kelas
        + Relationship = One to many (1:n)
        + Attribute penghubung = Kode Mata Kuliah, Kode Kelas (FK Kode Mata Kuliah di Ruang Kelas)

# Menggambar ERD Diagram :



Nama

Alamat

NIM

Mahasiswa

Dipakai

Diambil

Kode Mata Kuliah

Kode Kelas

Mengajar

Mata Kuliah

Mengisi

Ruang Kelas

Jumlah Kursi

Nama Mata Kuliah

Dipilih

NIP

Dosen

Mengajar

Nama

Alamat

**No 2**

1. **Menentukan entitas :**
   * Pegawai = Menyimpan semua data pribadi Pegawai
   * Sponsor = Menyimpan semua data Sponsor
   * ProgramTv = Menyimpan semua informasi Program Tv
   * Ruangan = Menyimpan semua informasi mengenai ruangan
   * Jabatan = Menyimpan semua informasi mengenai Jabatan

# Menentukan attribute :

* + Pegawai :
    - Id\_pegawai = Nomor Induk Mahasiswa (varchar(45)) PK
    - Nama\_pegawai = Nama lengkap Mahasiswa (varchar(255))
  + Sponsor :
    - id\_sponsor = Id Sponsor (varchar(45)) PK
    - nama\_sponsor = Nama Sponsor (varchar(255))
    - durasi = durasi waktu (time)
  + Program Tv :
    - Id\_programTv = Kode untuk Mata Kuliah (varchar(45)) PK
    - Nama\_programTv = Nama dari Mata Kuliah (varchar(255))
  + Ruangan :
    - Id\_ruang = Kode masing-masing kelas (varchar(45)) PK
    - Nama\_ruang = Jumlah kursi setiap kelas (varchar(255))
  + Jabatan :
    - Id\_jabatan = id jabatan (integer) PK
    - Nama\_jabatan = nama jabatan(varchar(45))

# Menentukan Relationship :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Pegawai | Sponsor | Program Tv | Ruangan | Jabatan |
| Pegawai |  |  |  |  |  |
| Sponsor |  |  |  |  |  |
| Program Tv | 1:n | n:n |  |  |  |
| Ruang |  |  | N:1 |  |  |
| Jabatan | 1:n |  |  |  |  |

Hubungan

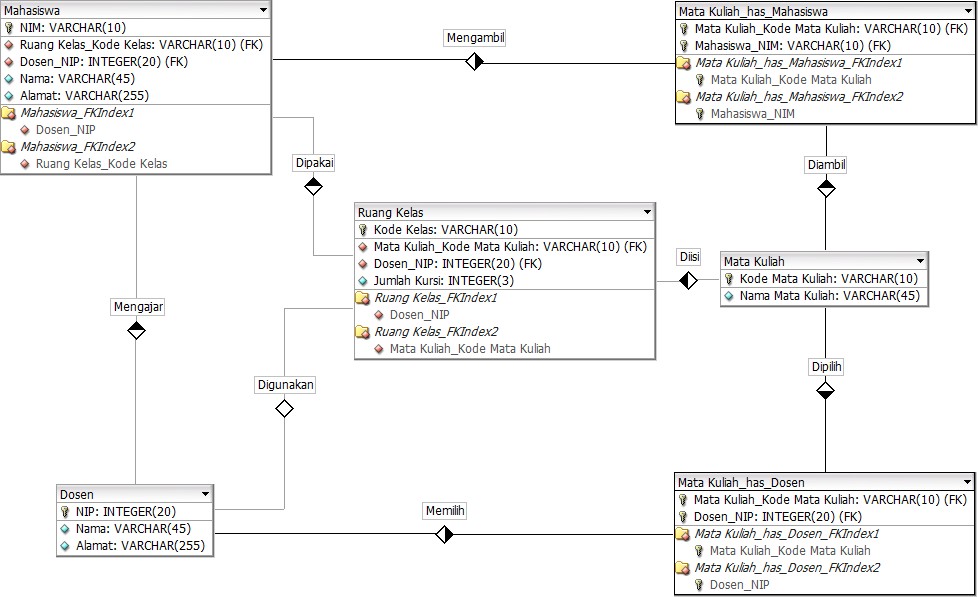
* + **Pegawai** menghendle **Program Tv** :
    - Tabel utama = Pegawai
    - Tabel kedua = Program Tv
    - Relationship = One to many (1:n)
    - Attribute penghubung = id\_pegawai, id\_program (FK id\_program di Program Tv)
  + **Sponsor** mendukung **Program Tv** :
    - Tabel utama = Sponsor, Program Tv
    - Tabel kedua = Sponsor\_has\_Program Tv
    - Relationship = Many to many (n:n)
    - Attribute penghubung = id\_sponsor, id\_program (FK id\_sponsor, id\_program di Sponsor\_has\_Program Tv)
  + **Pegawai** menempati **Jabatan** :
    - Tabel utama = Pegawai
    - Tabel kedua = Jabatan
    - Relationship = One to many (1:n)
    - Attribute penghubung = id\_pegawai, id\_jabatan (FK id\_jabatan di Jabatan)
  + **Program Tv** menempati **Ruang** :
    - Tabel utama = Program Tv
    - Tabel kedua = Ruang
    - Relationship = One to one (1:1)
    - Attribute penghubung = id\_program, id\_ruang

# Menggambar ERDiagram

**MODUL 3**

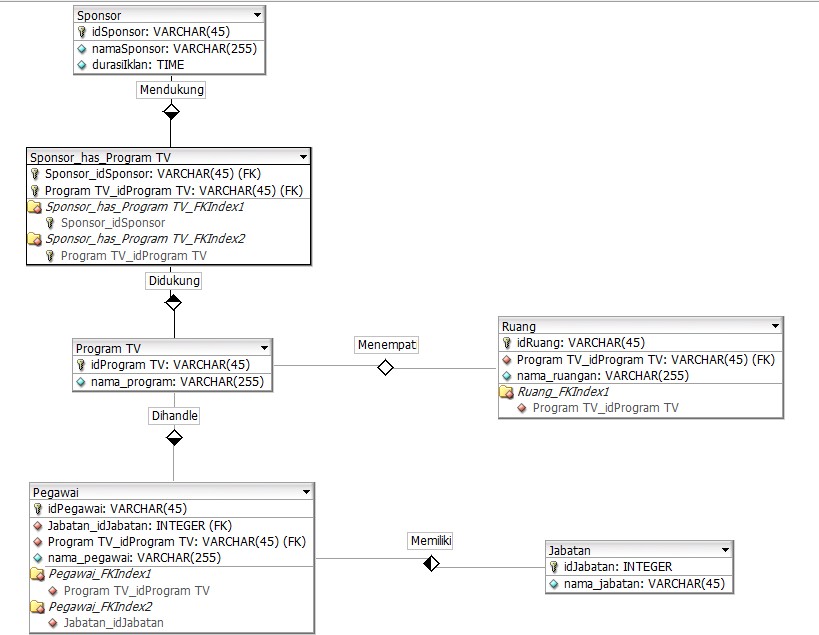
**No 1**

**Tabel relasi Database Perkuliahan**

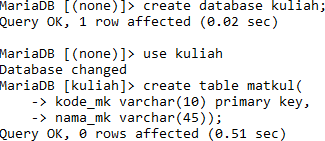


**No 2**

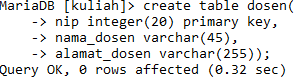
**Tabel relasi Database Stasiun Tv**

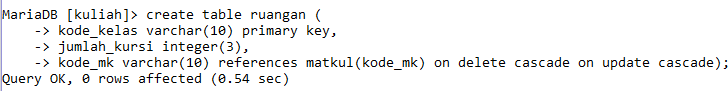


**MODUL 4**

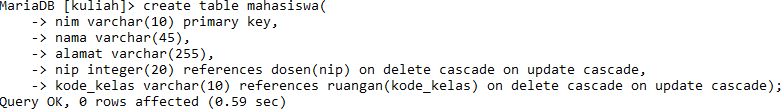
Membuat database kuliah dan membuat table mata kuliah

Membuat tabel dosen



Membuat tabel ruangan

Membuat tabel mahasiswa



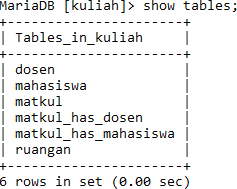
Membuat tabel relasi matakuliah\_has\_dosen



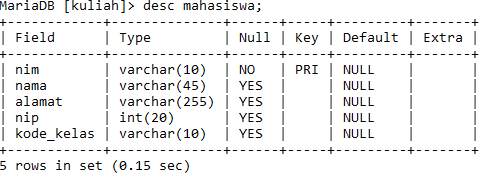
Membuat tabel relasi matakuliah\_has\_mahasiswa



Menampilkan tabel yang sudah di bentuk



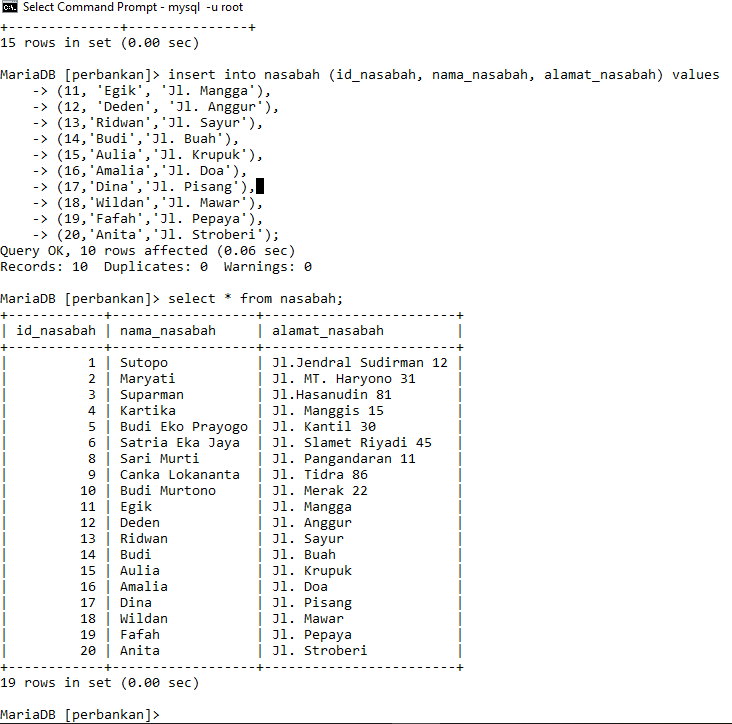
Menampilkan atribut dari suatu tabel mengunakan “Describe” Contoh :



# MODUL 5

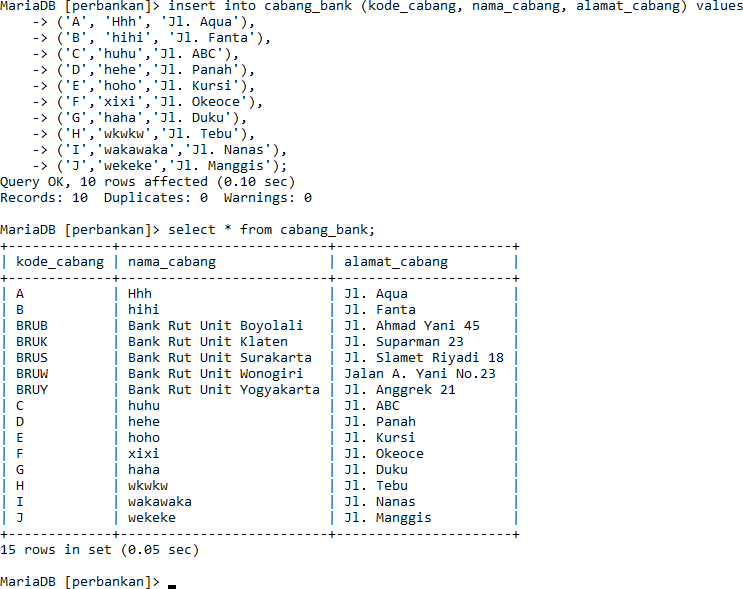
1. Nasabah :

Memasukan 10 record data untuk table Nasabah dan menampilkannya.



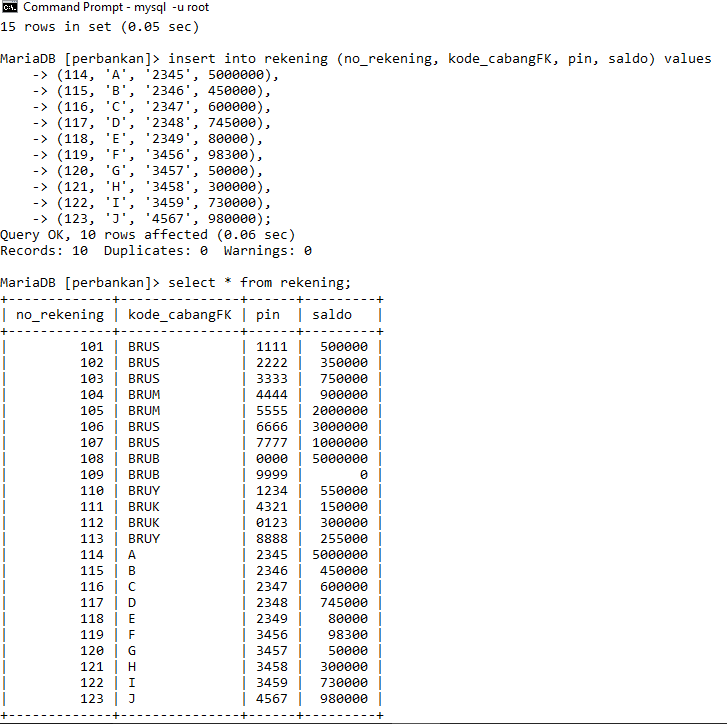
1. Cabang Bank :

Memasukkan 10 record data ke dalam table Cabang Bank dan menampilkan hasilnya.



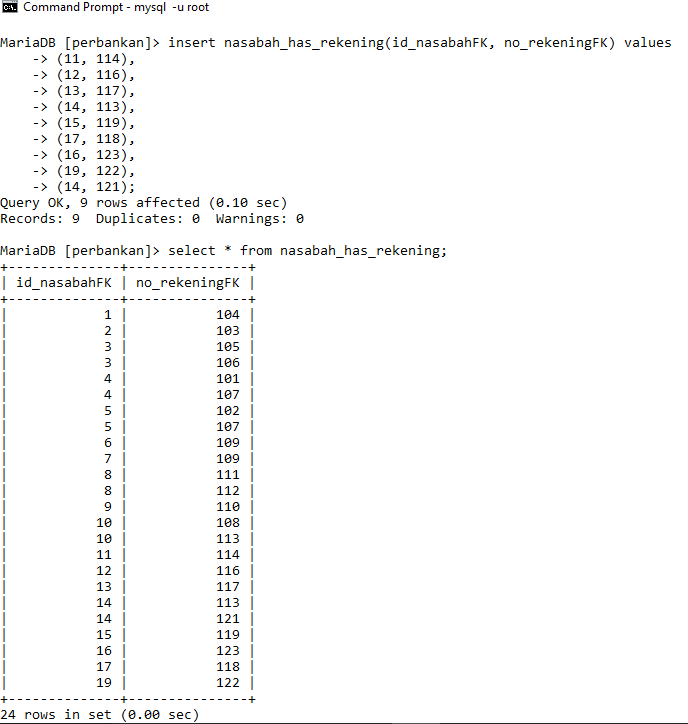
1. Rekening :

Memasukkan 10 record data ke dalam table Rekening dan menampilkan datanya.

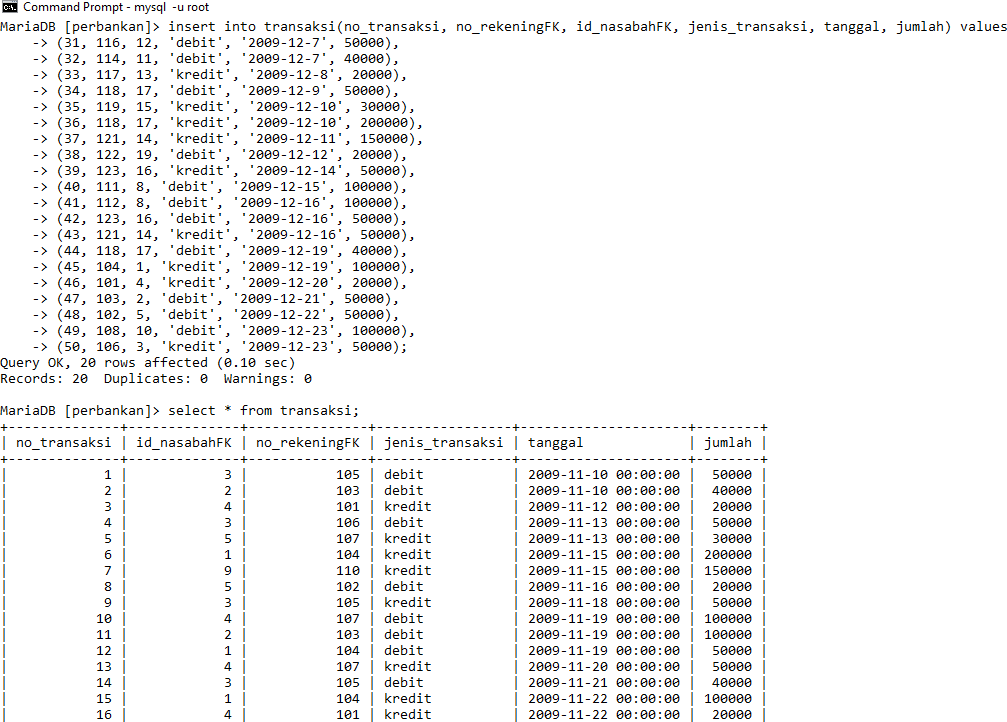


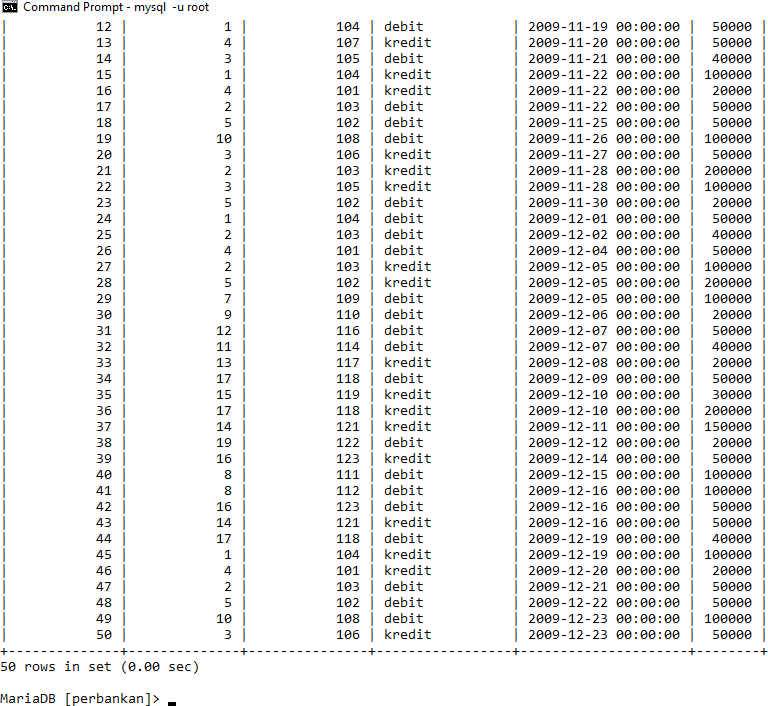
1. Nasabah Has Rekening :

Memasukkan 10 record data kedalam table Nasabah has Rekening dan menampilkan datanya.



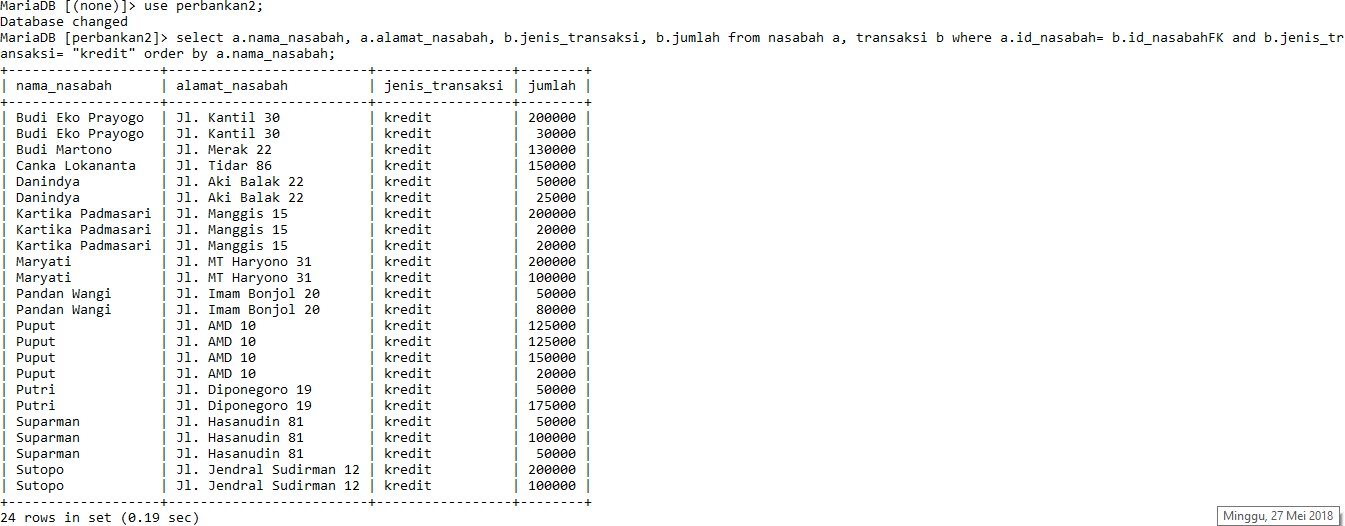
1. Transaksi :

Memasukkan 20 record data ke dalam table Transaksi dan menampilkan hasilnya.



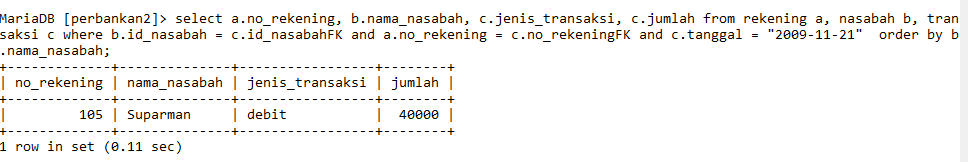
# MODUL 6

No 1

Menampilkan nama nasabah, alamat nasabah, jenis transaksi dan jumlah transaksi dimana jenis transaksinya adalah kredit dan di urutkan berdasarkan nama nasabah.

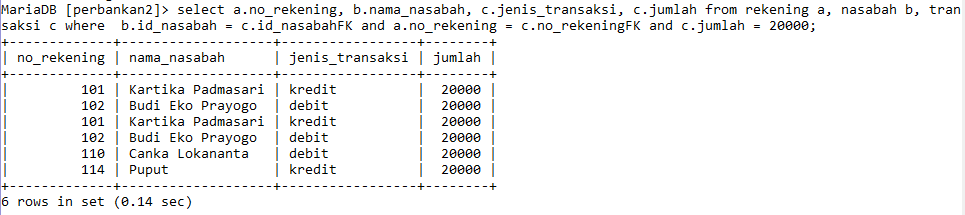
No 2

Menampilkan nomer rekening, nama nasabah, jenis transaksi dan jumlah transaksi yang melakukan transaksi pada tanggal 21 November 2009 dan diurutkan berdasarkan nama nasabah.



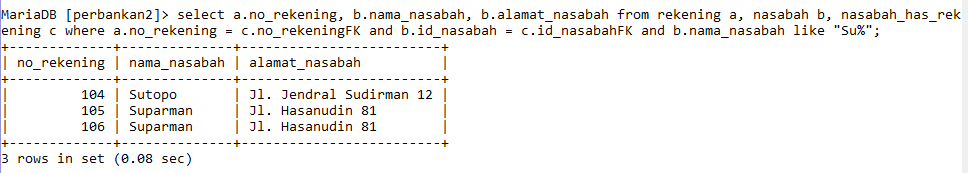
No 3

Menampilkan nomer rekening, nama nasabah, jenis transaksi, dan jumlah transaksi dimana jumlah transaksi = Rp. 20.000



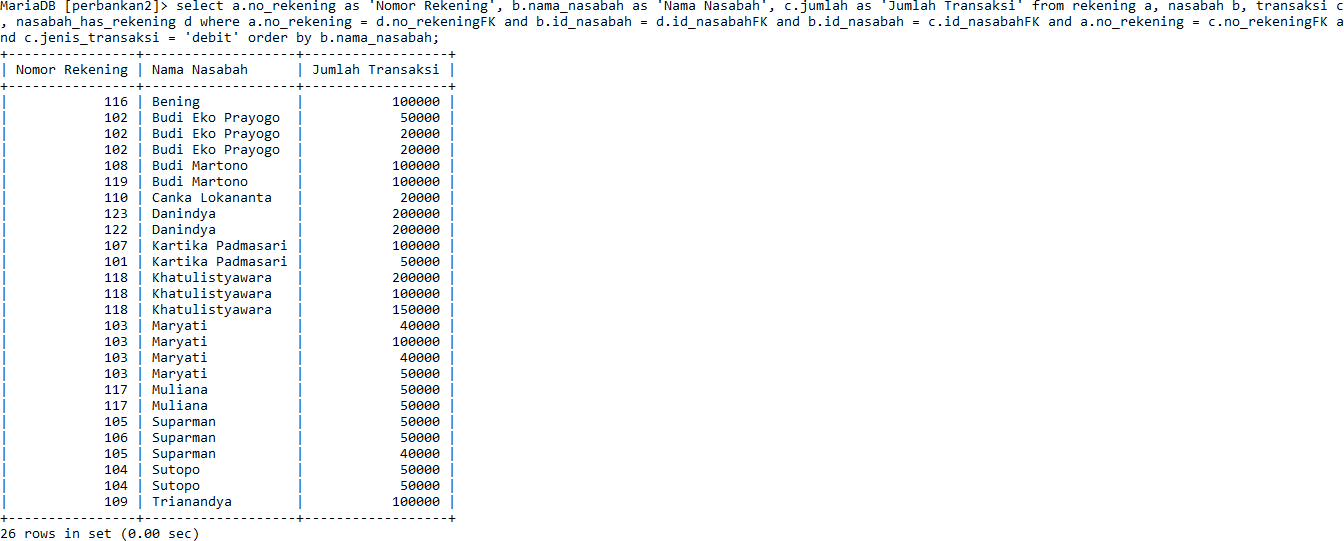
No 4

Menampilkan nomer rekening, nama nasabah dan alamat nasabah dimana nama nasabah di awali dengan kata „Su‟



No 5

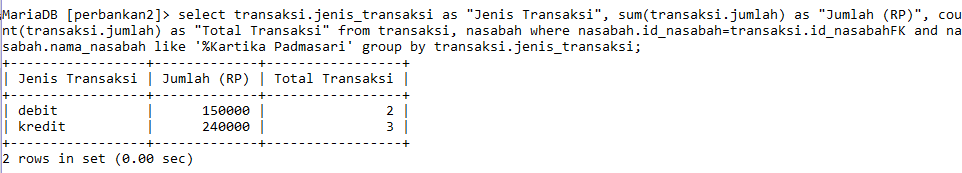
Menampilkan nomer rekening dengan alias „Nomor Rekening‟, nama nasabah dengan alias „Nama Nasabah‟, jumlah transaksi dengan alias „Jumlah Transaksi‟ dimana jenis transaksinya debit dan diurutkan berdasarkan nama \_nasabah.



# MODUL 7

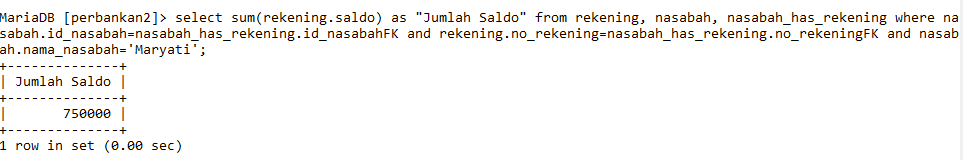
No 1

Menampilkan jenis transaksi, jumlah transaksi dalam Rp dan total transaksi untuk nasabah yang bernama „Kartika Padmasari‟ untuk masing-masing jenis transaksinya



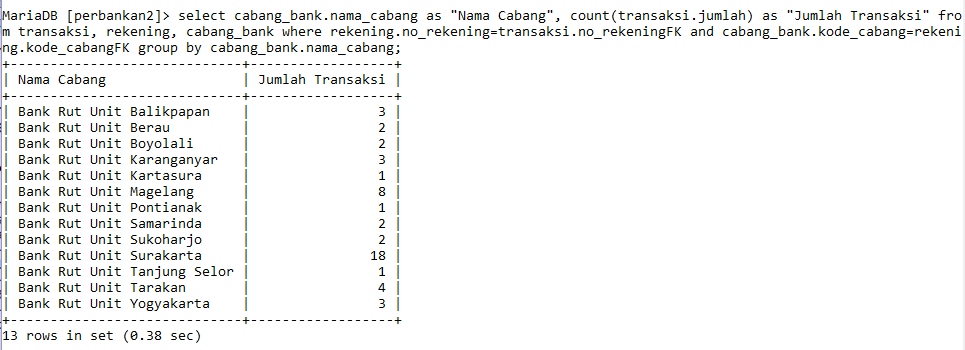
No 2

Menampilkan jumlah total saldo yang dimiliki Maryati



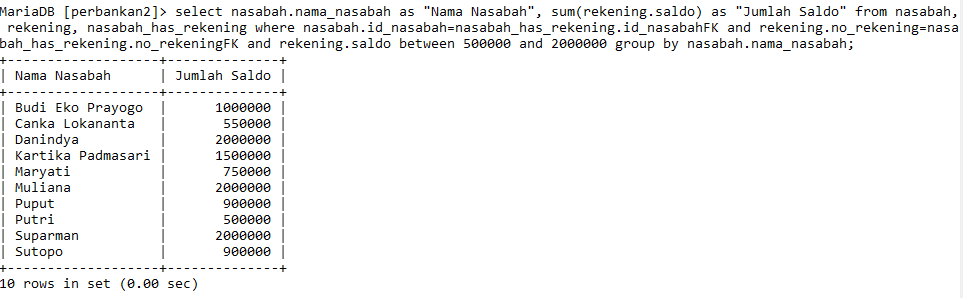
No 3

Menampilkan jumlah transaksi yang di tangani oleh masing-masing cabang bank



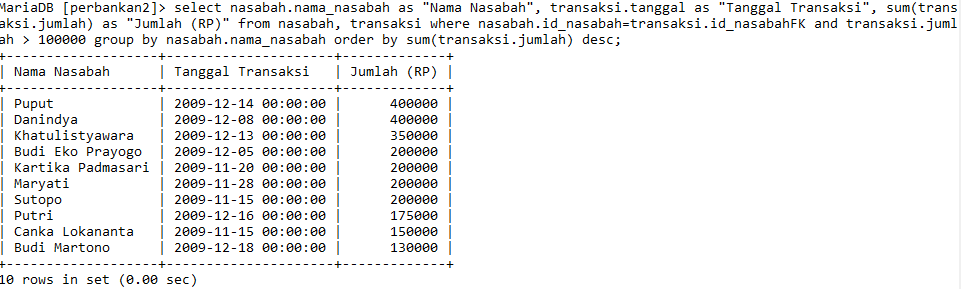
No 4

Menampilkan nama nasabah dan jumlah saldo yang memiliki saldo antara 500.000 sampai 2.000.000



No 5

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



# MODUL 8

No 1

Membuat user baru menggunakan Nama masing-masing



No 2

Memberi privilege untuk user yang telah di buat

* 1. User dapat melakukan Insert, Update , Delete pada tabel nasabah

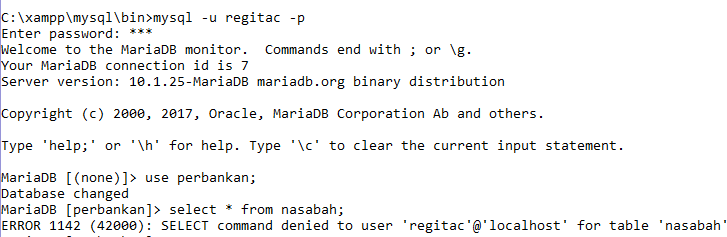


* 1. User hanya dapat melakukan select pada tabel cabang\_bank



No 3

1. Melakukan Login sebagai user yang telah di buat, dan menampilkan data dari mahasiswa;



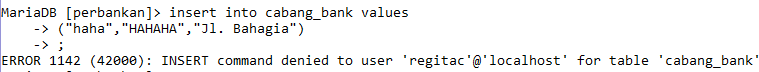
Tidak bisa karena pada user „regitac‟ tidak di berikan hak akses „Select‟ untuk database perbankan pada tabel nasabah.

1. User „regitac‟ melakukan input data ke dalam tabel nasabah



Berhasil karena user „regitac‟ di berikan hak akses „Insert‟ untuk database perbankan pada tabel nasabah.

1. User „regitac‟ melakukan input data ke dalam tabel cabang\_bank



Tidak berhasil, karena user „regitac‟ tidak di berikan hak akses untuk memasukan data kedalam tabel cabang\_bank pada database perbankan.

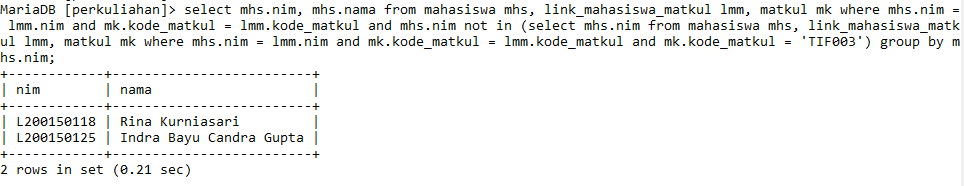
No 4 Analisa :

Untuk sistem keamanan setiap user dapat diatur untuk hak aksesnya, misalnya apabila ia hanya di beri izin untuk mengakses „insert‟ maka ia hanya akan bisa memasukan data kedalam tabel tertentu. Pemberian hak ases dari user satu ke user lainnya juga dapat dilakukan dengan menggunakan perintah „Grant option‟.

# MODUL 9 SUBQUERY

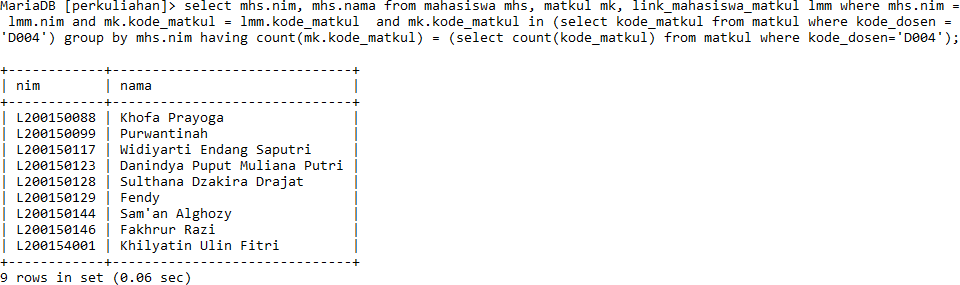
No 1

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



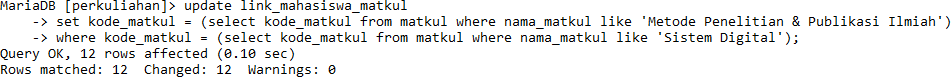
No 2

Menampilkan daftar mahasiswa yang mengambil semua mata kuliah yang diampu oleh salah satu dosen.



No 3

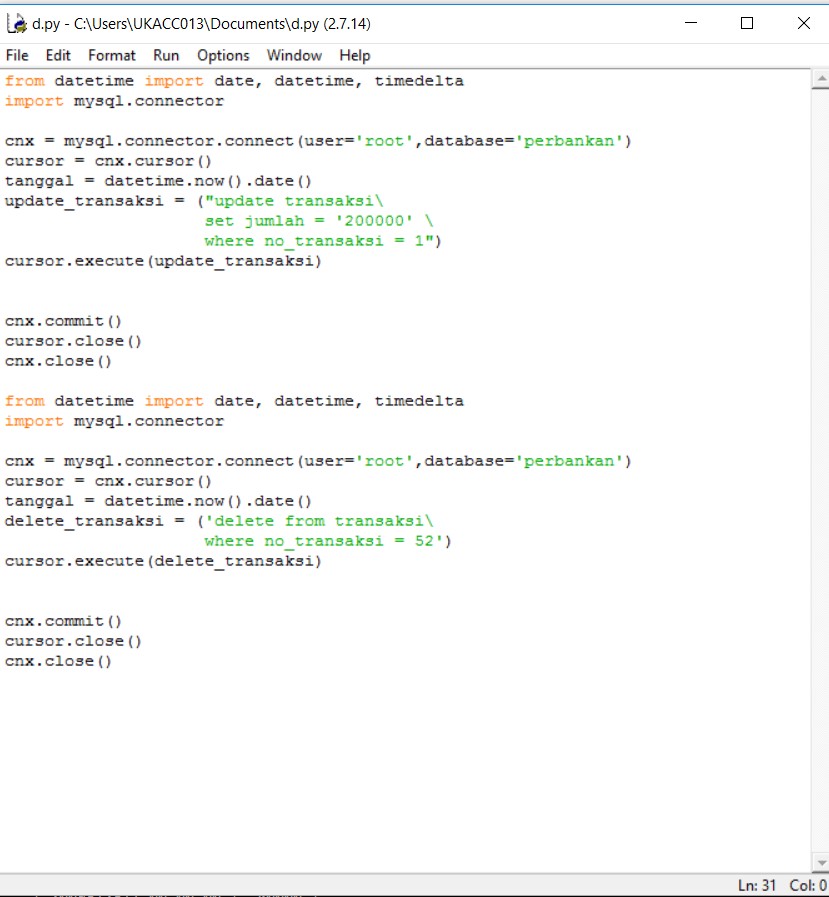
Melakukan update date , karena salah satu makul(A) dihilanhkan, seluruh mahasiswa yang mengambil matakuliah tersebut dipindahkan untuk mengambil matakuliah lain(B).



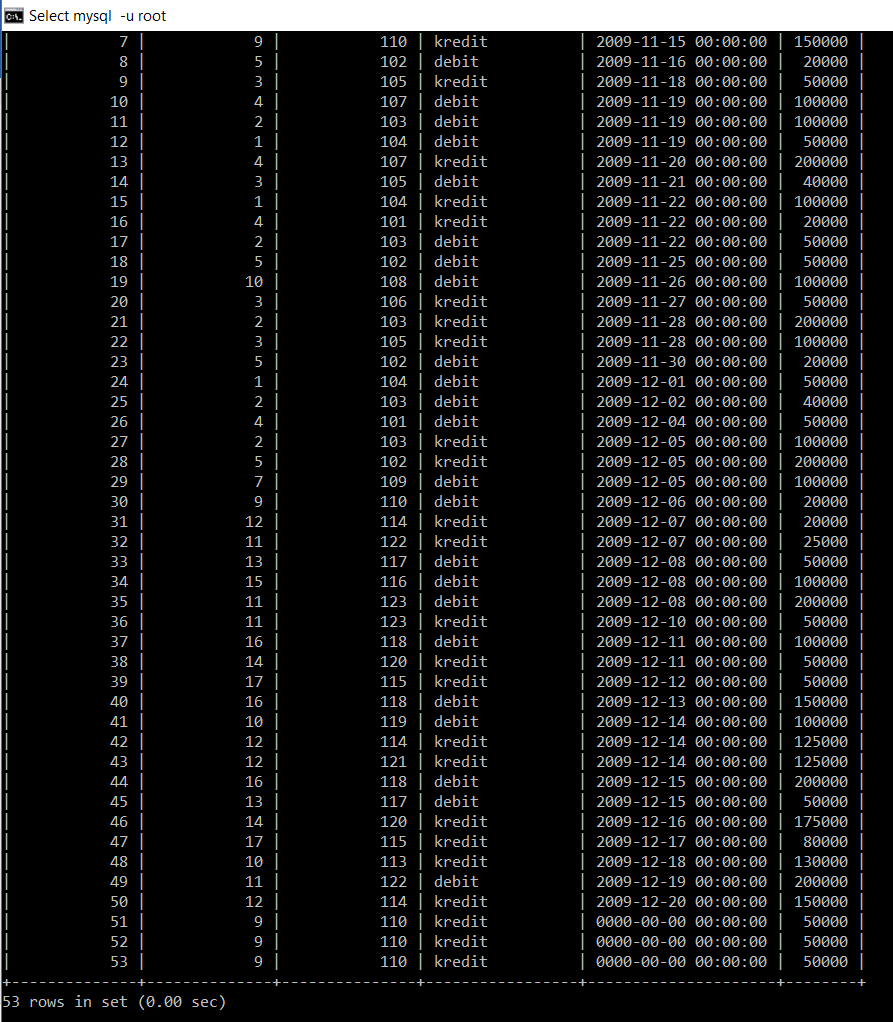
# MODUL 10

**Tugas**

* 1. Kode program Python untuk melakukan INSERT, UPDATE dan DELETE pada data transaksi

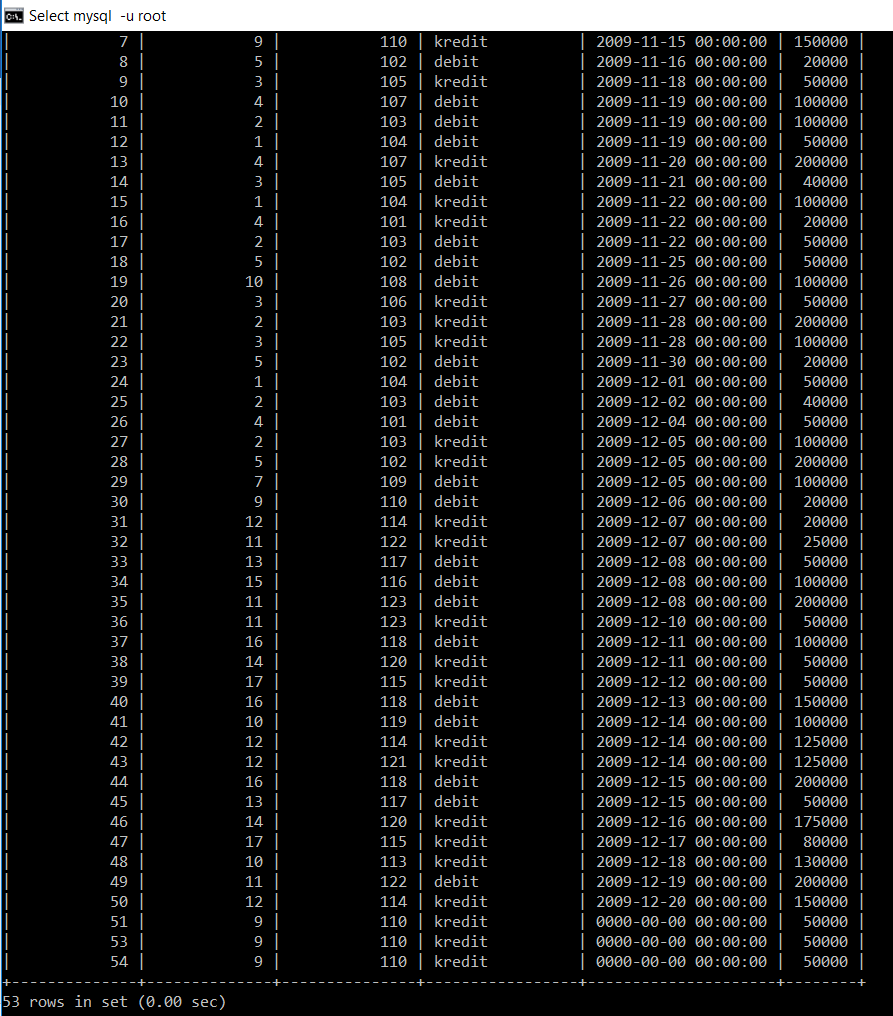


Before :



After :

No transaksi 52 terhapus



* 1. a. Kode untuk mendapatkan data nasabah

b. Kode untuk mendapatkan data nasabah yang melakukan transaksi antara bulan Oktober sampai Desember



