

Laporan Akhir Praktikum Sistem Basis Data



Disusun oleh :
ADNAN SHAFRY ARI PURNAMA AJI (L200170021)

**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 berupa informasi, maka data-data yang lama akan menjadi acuan terhadap data-data yang baru

2. Jelaskan manfaat database dan contohnya!

- a. Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
- b. Menentukan kualitas informasi yaitu cepat, akurat, dan relevan, sehingga informasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- c. Mengatasi kerangkapan data (redundancy data).
- d. Menghindari terjadinya inkonsistensi data.
- e. Mengatasi kesulitan dalam mengakses data.
- f. Menyusun format yang standar dari sebuah data.
- g. Penggunaan oleh banyak pemakai (multiple user). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multiuser).
- h. 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.
- i. 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.

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

- a. Spesifikasi data
- b. Pemrosesan yang diperlukan oleh data
- c. Pertimbangan keamanan
- d. Kecocokan dengan tipe aplikasi
- e. Bahasa query
- f. Biaya tak langsung terhadap pemrosesan

4. 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. Field merepresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Record yaitu kumpulan dari field.

5. Bandingkan perbedaan pengolahan data secara manual dengan menggunakan system database. Pengolahan data manual masih menggunakan alat berupa 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.

6. 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

7. 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

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

- Mahasiswa
 - ✓ nama_mhs : nama lengkap mahasiswa (varchar(45))

- ✓ nim_mhs : nomor id mahasiswa (integer) PK
- ✓ alamat_mhs : alamat lengkap mahasiswa (varchar(225))

● Dosen

- ✓ nama_dosen : nama lengkap dosen (varchar(45))
- ✓ nidn_dosen : nomor id dosen (integer) PK
- ✓ alamat_dosen : alamat lengkap dosen (varchar(225))

● Mata_Kuliah

- ✓ nama_matkul : nama mata kuliah (varchar(45))
- ✓ kode_matkul : nomor kode mata kuliah (integer) PK

● Ruang_Kuliah

- ✓ kode_ruang : nomor ruang kuliah (integer)PK
- ✓ kapasitas_ruang : jumlah muatan setiap ruang (integer)PK
- ✓ lokasi_ruang : alamat suatu ruang (varchar(25))

3. 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) ●

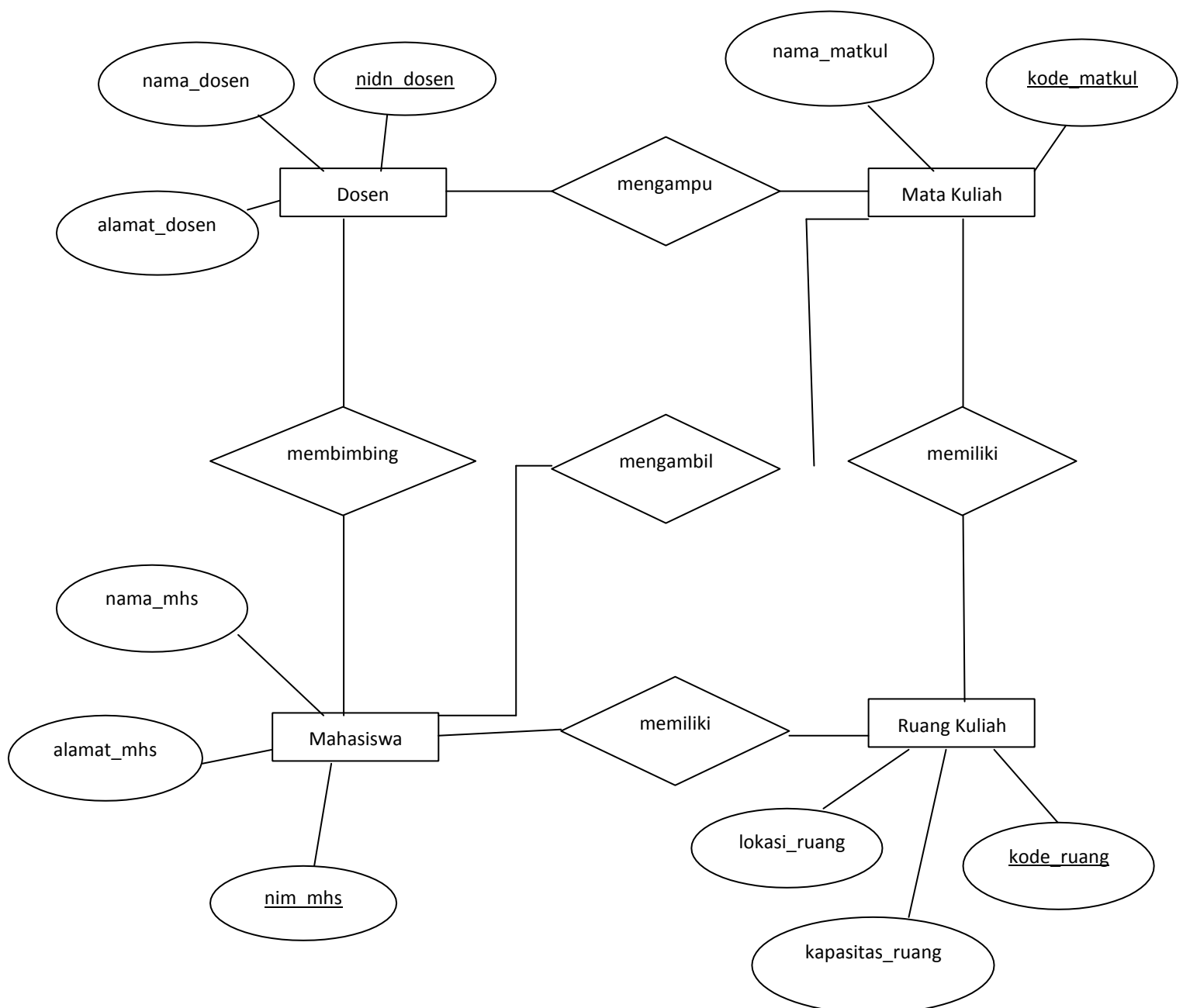
Mahasiswa mengambil mata kuliah

- ✓ 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)

4. Menggambar ERD Diagram



2. 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 (integer) PK
 - ✓ pnrbt_buku : nama penerbit buku (varchar(45))
 - ✓ pngrng_buku : nama lengkap pengarang buku (varchar(45))
- Peminjaman
 - ✓ kode_pinjam : nomor id pinjam (integer) PK
 - ✓ tgl_pinjam : tanggal peminjaman buku (integer) PK
 - ✓ tgl_kembali : tanggal pengembalian buku (integer) PK
- Pengembalian
 - ✓ kode_kembali : nomor id kembali (integer) PK
 - ✓ tgl_kembali : tanggal pengembalian buku (integer) PK
- Anggota
 - ✓ nama_anggota : nama lengkap anggota (varchar(45))
 - ✓ id_anggota : nomor id anggota (integer) PK
 - ✓ alamat_anggota : alamat lengkap anggota (varchar(225))
 - ✓ kontak_anggota : nomor telepon anggota (integer) PK
 - ✓ tgllahir_anggota : tanggal lahir anggota (integer) PK
- Pegawai
 - ✓ nama_pegawai : nama lengkap pegawai (varchar(45))
 - ✓ id_pegawai : nomor id pegawai (integer) PK
 - ✓ alamat_pegawai : alamat lengkap pegawai (varchar(225))
 - ✓ kontak_pegawai : nomor telepon pegawai (integer) PK

3. Menentukan relationship (hubungan) antar entitas

Hubungan

● Anggota meminjam buku

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

✓ 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

✓ 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

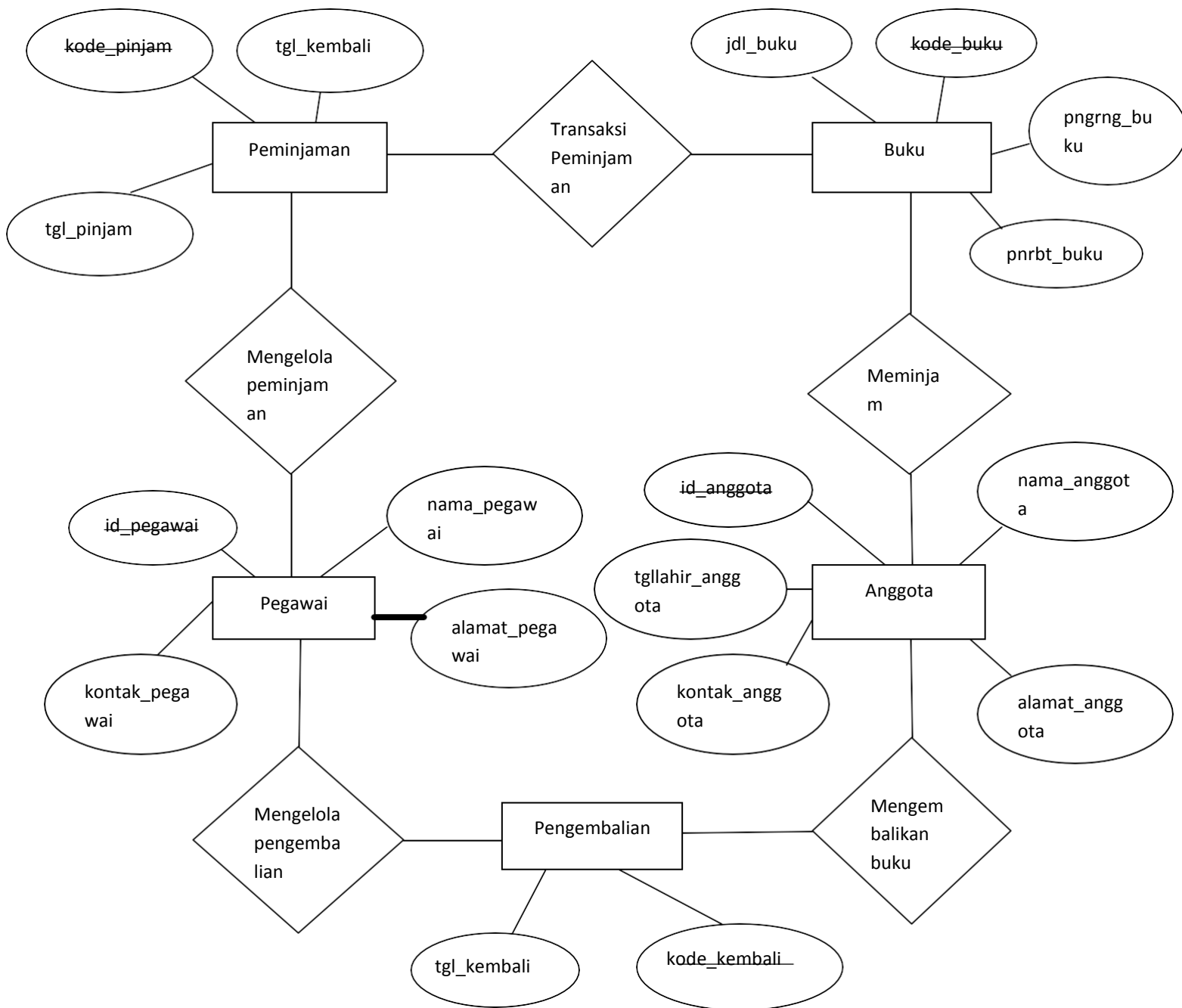
✓ Tabel utama : pengembalian

✓ Tabel kedua : pegawai

✓ Relationship : one-to-many(1:n)

✓ Attribute penghubung : kode_pengembalian(FK kode_pengembalian di pegawai)

4. Menggambar ERD Diagram

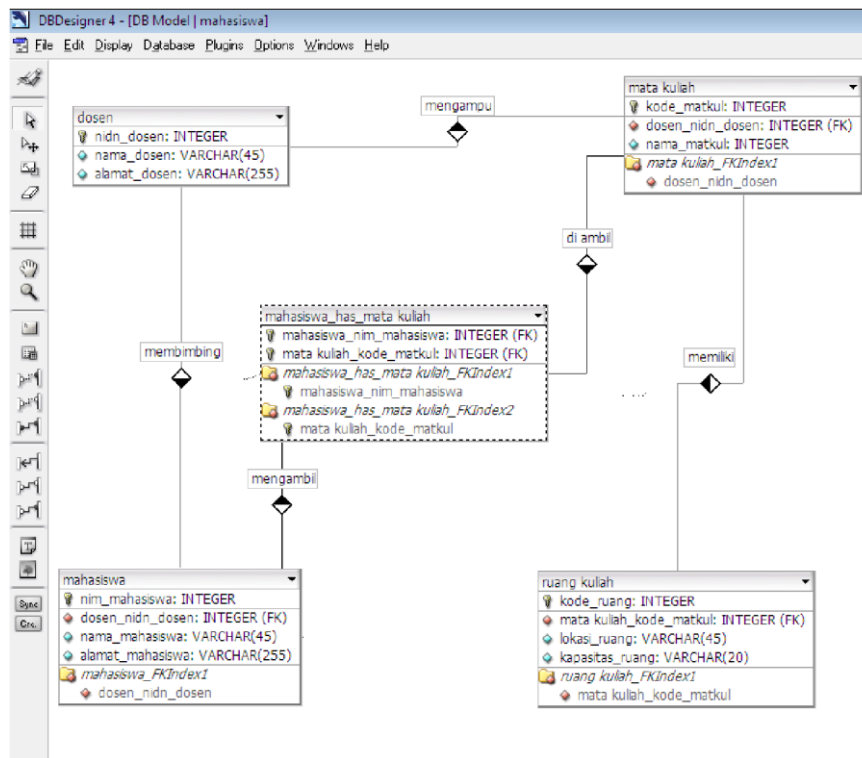


MODUL 3 Perancangan E-R Diagram dengan DBDesigner

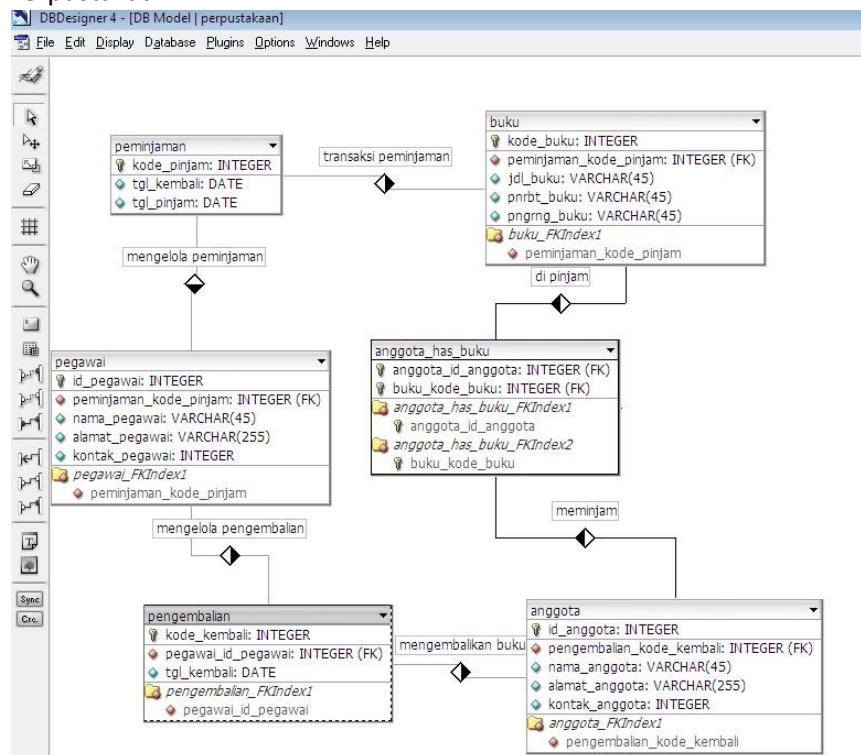
E. Tugas

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.

a. Perkuliahan



b. 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 :

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\PakJ>cd/

C:\>cd xampp/mysql/bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database mahasiswa;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use mahasiswa
Database changed
MariaDB [mahasiswa]> create table dosen(
-> nama_dosen varchar(45) not null,
-> nidn_dosen integer primary key,
-> alamat_dosen varchar(225)
-> );
Query OK, 0 rows affected (1.71 sec)

MariaDB [mahasiswa]> create table mahasiswa(
-> nama_mhs varchar(45) not null,
-> nim_mhs varchar(20) primary key,
-> alamat_mhs varchar(225),
-> nidn_dosenfk integer references dosen(nidn_dosen) on delete cascade on up
date cascade
-> );
Query OK, 0 rows affected (0.41 sec)

MariaDB [mahasiswa]> create table mata_kuliah(
-> nama_matkul varchar(45) not null,
-> kode_matkul integer primary key,
-> nidn_dosenfk integer references dosen(nidn_dosen) on delete cascade on up
date cascade
-> );
Query OK, 0 rows affected (0.51 sec)

MariaDB [mahasiswa]> create table ruang_kuliah(
-> kode_ruang varchar(20) primary key,
-> kapasitas_ruang integer,
-> lokasi_ruang varchar(40),
-> kode_matkulfk integer references mata_kuliah(kode_matkul) on delete casca
de on update cascade
-> );
Query OK, 0 rows affected (0.38 sec)

MariaDB [mahasiswa]> create table mahasiswa_has_mata_kuliah(
-> nim_mhsfk varchar(20) references mahasiswa(nim_mhs) on delete cascade on
```

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

```

MariaDB [mahasiswa]> create table mahasiswa_has_mata_kuliah(
  -> nim_mhsfk varchar(20) references mahasiswa(nim_mhs) on delete cascade on
update cascade,
  -> kode_matkulfk varchar(20) references mata_kuliah(kode_matkul) on delete
ascade on update cascade,
  -> primary key(nim_mhsfk,kode_matkulfk)
  -> );
Query OK, 0 rows affected (0.38 sec)

MariaDB [mahasiswa]> describe mahasiswa
  -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MariaDB server version for the right syntax to use near ')'
at line 2
MariaDB [mahasiswa]> describe mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_mhs | varchar(45) | NO | | NULL | |
| nim_mhs | varchar(20) | NO | PRI | NULL | |
| alamat_mhs | varchar(225) | YES | | NULL | |
| nidn_dosenfk | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.41 sec)

MariaDB [mahasiswa]> describe dosen;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_dosen | varchar(45) | NO | | NULL | |
| nidn_dosen | int(11) | NO | PRI | NULL | |
| alamat_dosen | varchar(225) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [mahasiswa]> describe mata_kuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_matkul | varchar(45) | NO | | NULL | |
| kode_matkul | int(11) | NO | PRI | NULL | |
| nidn_dosenfk | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [mahasiswa]> describe ruang_kuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_ruang | varchar(20) | NO | PRI | NULL | |
| kapasitas_ruang | int(11) | YES | | NULL | |
| lokasi_ruang | varchar(40) | YES | | NULL | |
| kode_matkulfk | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

MariaDB [mahasiswa]> describe mahasiswa_has_mata_kuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim_mhsfk | varchar(20) | NO | PRI | NULL | |
| kode_matkulfk | varchar(20) | NO | PRI | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

MariaDB [mahasiswa]> _

```

```

Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\PakJ>cd/

C:\>cd xampp/mysql/bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database perpustakaan;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use perpustakaan;
Database changed
MariaDB [perpustakaan]> create table anggota(
-> nama_anggota varchar(45) not null,
-> id_anggota integer primary key,
-> alamat_dosen varchar(225),
-> kontak_anggota integer not null,
-> tgllahir_anggota datetime not null default current_timestamp
-> );
Query OK, 0 rows affected (1.42 sec)

MariaDB [perpustakaan]> create table pegawai(
-> nama_pegawai varchar(45) not null,
-> id_pegawai varchar(20) primary key,
-> alamat_pegawai varchar(225),
-> kontak_pegawai integer not null,
-> kode_bukufk varchar(20) references buku(kode_buku) on delete cascade on u
pdate cascade
-> );
Query OK, 0 rows affected (0.42 sec)

MariaDB [perpustakaan]> create table buku(
-> jdl_buku varchar(20) not null,
-> kode_buku varchar(20) primary key,
-> pnrht_buku varchar(45) not null,
-> pngrng_buku varchar(45) not null,
-> kode_pinjamfk varchar(20) references pinjaman(kode_pinjam) on delete ca
scade on update cascade
-> );
Query OK, 0 rows affected (0.37 sec)

MariaDB [perpustakaan]> create table pinjaman(
-> kode_pinjam varchar(20) primary key,
-> tgl_pinjam datetime not null default current_timestamp,
-> tgl_kembali datetime not null default current_timestamp,
-> id_pegawaifk varchar(20) references pegawai(id_pegawai) on delete cascade
on update cascade
-> );
Query OK, 0 rows affected (0.30 sec)

MariaDB [perpustakaan]> create table pengembalian(
-> kode_kembali varchar(20) primary key,
-> tgl_kembali datetime not null default current_timestamp,
-> id_anggotafk integer references anggota(id_anggota) on delete cascade on
update cascade
-> );
Query OK, 0 rows affected (0.33 sec)

MariaDB [perpustakaan]> create table anggota_has_buku(
-> id_anggotafk integer references anggota(id_anggota) on delete cascade on
update cascade,
-> kode_bukufk varchar(20) references buku(kode_buku) on delete cascade on u
pdate cascade,
-> primary key(id_anggotafk,kode_bukufk)
-> );
Query OK, 0 rows affected (0.78 sec)

MariaDB [perpustakaan]> describe anggota;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_anggota | varchar(45) | NO | | NULL | |
| id_anggota | int(11) | NO | PRI | NULL | |
| alamat_dosen | varchar(225) | YES | | NULL | |
| kontak_anggota | int(11) | NO | | NULL | |
| tgllahir_anggota | datetime | NO | | CURRENT_TIMESTAMP | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

MariaDB [perpustakaan]> describe pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nama_pegawai | varchar(45) | NO | | NULL | |
| id_pegawai | varchar(20) | NO | PRI | NULL | |
| alamat_pegawai | varchar(225) | YES | | NULL | |
| kontak_pegawai | int(11) | NO | | NULL | |
| kode_bukufk | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

MariaDB [perpustakaan]> describe pengembalian;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_kembali | varchar(20) | NO | PRI | NULL | |
| tgl_kembali | datetime | NO | | CURRENT_TIMESTAMP | |
| id_anggotafk | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [perpustakaan]> describe pinjaman;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_pinjam | varchar(20) | NO | PRI | NULL | |
| tgl_pinjam | datetime | NO | | CURRENT_TIMESTAMP | |
| tgl_kembali | datetime | NO | | CURRENT_TIMESTAMP | |
| id_pegawaifk | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

```

```
MariaDB [perpustakaan]> describe buku;
```

Field	Type	Null	Key	Default	Extra
jdl_buku	varchar(20)	NO		NULL	
kode_buku	varchar(20)	NO	PRI	NULL	
pnrbt_buku	varchar(45)	NO		NULL	
pngrng_buku	varchar(45)	NO		NULL	
kode_pinjamfk	varchar(20)	YES		NULL	

```
5 rows in set (0.01 sec)
```

```
MariaDB [perpustakaan]> describe anggota_has_buku;
```

Field	Type	Null	Key	Default	Extra
id_anggotafk	int(11)	NO	PRI	NULL	
kode_bukufk	varchar(20)	NO	PRI	NULL	

```
2 rows in set (0.01 sec)
```

```
MariaDB [perpustakaan]> _
```


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 :

```
MariaDB [modul_51] > INSERT INTO nasabah(id_nasabah, nama_nasabah, alamat_nasabah)
-> VALUES(11, 'Budi Murtopo', 'Jl. Perak 20'),
-> (12, 'Budi Subagiyo', 'Jl. Ahmad Yani 10'),
-> (13, 'Sumaryanto', 'Jl. Nanas 2'),
-> (14, 'Sulatan Wakanda', 'Jl. Asia Frika 1'),
-> (15, 'Rudi', 'Jl. Semangka 33'),
-> (16, 'Yerika', 'Jl. Salak 76'),
-> (17, 'Michele', 'Jl. Duku 56'),
-> (18, 'Kevin', 'Jl. Lemon 24'),
-> (19, 'Ahmad', 'Jl. Obama 8'),
-> (20, 'Ahmad Sudarmono', 'Jl. Donald 22');
```

Hasil :

```
MariaDB [modul_51] > select * from nasabah;
```

id_nasabah	nama_nasabah	alamat_nasabah
1	Sutopo	Jl. Jendral Sudirman 12
2	Maryati	Jl. MT. Haryono 31
3	Suparman	Jl. Hasanudin 81
4	Kartika Padmasari	Jl. Manggis 15
5	Budi Eko Prayogo	Jl. Kantil 30
6	Satria Eka Jaya	Jl. Slamet Riyadi 45
8	Sari Murti	Jl. Pangandaran 11
9	Canka Lokananta	Jl. Tidar 86
10	Budi Murtono	Jl. Merak 22
11	Budi Murtopo	Jl. Perak 20
12	Budi Subagiyo	Jl. Ahmad Yani 10
13	Sumaryanto	Jl. Nanas 2
14	Sulatan Wakanda	Jl. Asia Frika 1
15	Rudi	Jl. Semangka 33
16	Yerika	Jl. Salak 76
17	Michele	Jl. Duku 56
18	Kevin	Jl. Lemon 24
19	Ahmad	Jl. Obama 8
20	Ahmad Sudarmono	Jl. Donald 22

```
19 rows in set (0.06 sec)
```

2. Menambah cabang bank. Kode :

```
MariaDB [modul_51] > INSERT INTO cabang_bank(kode_cabang, nama_cabang, alamat_cabang)
-> VALUES('BRUA', 'Bank Rut Unit Aceh', 'Jl. Ahmad Yani 8'),
-> ('BRUC', 'Bank Rut Unit Cilacap', 'Jl. Tentara Pelajar 34'),
-> ('BRUD', 'Bank Rut Unit Demak', 'Jl. Slamet Raharjo 5'),
-> ('BRUBB', 'Bank Rut Unit Bangka Belitung', 'Jl. Kenangan 12'),
-> ('BRUS', 'Bank Rut Unit Surakarta', 'Jl. Slamet Riyadi 18'),
-> ('BRUSE', 'Bank Rut Unit Semarang', 'Jl. Mangka 22'),
-> ('BRUN', 'Bank Rut Unit Ngawi', 'Jl. Slamet 6'),
-> ('BRUSD', 'Bank Rut Unit Sungai Danau', 'Jl. Gethek 2'),
-> ('BRUS', 'Bank Rut Unit Surakarta', 'Jl. Slamet Riyadi 18'),
-> ('BRUSR', 'Bank Rut Unit Solo Raya', 'Jl. Slamet Riyadi 54');
```

Hasil :

```
MariaDB [modul_5]> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUA	Bank Rut Unit Aceh	Jl. Ahmad Yani 8
BRUB	Bank Rut Unit Boyolali	Jl. Ahmad yani 45
BRUBB	Bank Rut Unit Bangka Belitung	Jl. Kenangan 12
BRUC	Bank Rut Unit Cilacap	Jl. Tentara Pelajar 34
BRUD	Bank Rut Unit Demak	Jl. Slamet Raharjo 5
BRUK	Bank Rut Unit Klaten	Jl. Suparman 23
BRUN	Bank Rut Unit Ngawi	Jl. Slamet 6
BRUS	Bank Rut Unit Surakarta	Jl. Slamet Riyadi 18
BRUSD	Bank Rut Unit Sungai Danau	Jl. Gethek 2
BRUSE	Bank Rut Unit Semarang	Jl. Nangka 22
BRUSI	Bank Rut Unit Sidoarjo	Jl. Slamet Riyadi 33
BRUSR	Bank Rut Unit Solo Raya	Jl. Slamet Riyadi 54
BRUSUR	Bank Rut Unit Surabaya	Jl. Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jl. A.Yani No.23
BRUY	Bank Rut Unit Yogyakarta	Jl. Anggrek 21

```
15 rows in set (0.04 sec)
```

3. Menambah rekening.

Kode :

```
MariaDB [modul_5]> INSERT INTO rekening(no_rekening, kode_cabangFK, pin, saldo)
-> VALUES('114', 'BRUC', '1112', '55000'),
-> ('115', 'BRUC', '1113', '35000'),
-> ('116', 'BRUS', '1114', '150000'),
-> ('117', 'BRUS', '1115', '546376'),
-> ('118', 'BRUSR', '1116', '333000'),
-> ('119', 'BRUD', '1117', '220000'),
-> ('120', 'BRUC', '1118', '1000000'),
-> ('121', 'BRUD', '1119', '20000'),
-> ('122', 'BRUD', '1120', '500000'),
-> ('123', 'BRUD', '1121', '12000');
```

Hasil :

```
MariaDB [modul_5]> select * from rekening;
```

no_rekening	kode_cabangFK	pin	saldo
101	BRUS	1111	500000
102	BRUS	2222	350000
103	BRUS	3333	750000
104	BRUM	4444	900000
105	BRUM	5555	2000000
106	BRUS	6666	3000000
107	BRUS	7777	1000000
108	BRUB	0000	5000000
109	BRUB	9999	0
110	BRUY	1234	550000
111	BRUK	4321	150000
112	BRUK	0123	300000
113	BRUY	8888	255000
114	BRUC	1112	55000
115	BRUC	1113	35000
116	BRUS	1114	150000
117	BRUS	1115	546376
118	BRUSR	1116	333000
119	BRUD	1117	220000
120	BRUC	1118	1000000
121	BRUD	1119	20000
122	BRUD	1120	500000
123	BRUD	1121	12000

```
23 rows in set (0.04 sec)
```

4. Menambah nasabah_has_rekening.

Kode :

```

MariaDB [modul_5]> INSERT INTO nasabah_has_rekening(id_nasabahFK, no_rekeningFK)
-> VALUES(11, 114),
-> (12,115),
-> (13,116),
-> (14,117),
-> (15,118),
-> (16,119),
-> (17,120),
-> (18,121),
-> (19,122),
-> (20,123);

```

Hasil :

```

MariaDB [modul_5]> select * from nasabah_has_rekening;
+-----+-----+
| id_nasabahFK | no_rekeningFK |
+-----+-----+
|          1   |          104  |
|          2   |          103  |
|          3   |          105  |
|          3   |          106  |
|          4   |          101  |
|          4   |          107  |
|          5   |          102  |
|          5   |          107  |
|          6   |          109  |
|          7   |          109  |
|          8   |          111  |
|          8   |          112  |
|          9   |          110  |
|         10   |          108  |
|         10   |          113  |
|         11   |          114  |
|         12   |          115  |
|         13   |          116  |
|         14   |          117  |
|         15   |          118  |
|         16   |          119  |
|         17   |          120  |
|         18   |          121  |
|         19   |          122  |
|         20   |          123  |
+-----+-----+
25 rows in set (0.03 sec)

```

5. Menambah transaksi. Kode :

```

MariaDB [modul_5]> INSERT INTO transaksi(no_transaksi, no_rekeningFK, id_nasabahFK, jenis_transaksi, tanggal, jumlah)
-> VALUES(31, 115, 12, 'debit', '2009-11-10', 55000 ),
-> (32, 115, 12, 'debit', '2009-11-10', 355000 ),
-> (33, 116, 13, 'debit', '2009-11-20', 355000 ),
-> (34, 117, 14, 'debit', '2009-11-10', 356000 ),
-> (35, 118, 15, 'debit', '2009-11-20', 356000 ),
-> (36, 119, 16, 'debit', '2009-11-10', 356000 ),
-> (37, 120, 17, 'debit', '2009-11-20', 356000 ),
-> (38, 121, 18, 'debit', '2009-11-10', 356000 ),
-> (39, 122, 19, 'debit', '2009-11-20', 200000 ),
-> (40, 123, 20, 'debit', '2009-11-10', 200000 ),
-> (41, 115, 12, 'kredit', '2009-11-20', 200000 ),
-> (42, 116, 13, 'kredit', '2009-11-22', 200000 ),
-> (43, 117, 14, 'debit', '2009-11-22', 200000 ),
-> (44, 118, 15, 'kredit', '2009-11-22', 500000 ),
-> (45, 119, 16, 'kredit', '2009-11-22', 500000 ),
-> (46, 120, 17, 'kredit', '2009-11-22', 500000 ),
-> (47, 121, 18, 'kredit', '2009-11-25', 500000 ),
-> (48, 122, 19, 'kredit', '2009-11-25', 500000 ),
-> (49, 123, 20, 'kredit', '2009-11-25', 342500 ),
-> (50, 115, 12, 'kredit', '2009-11-25', 342500 );

```

Hasil :


```
MariaDB [modul_51] > select * from transaksi;
```

no_transaksi	id_nasabahFK	no_rekeningFK	jenis_transaksi	tanggal	jumlah
1	3	105	debit	2009-11-10 00:00:00	50000
2	2	103	debit	2009-11-10 00:00:00	40000
3	4	101	kredit	2009-11-12 00:00:00	20000
4	3	106	debit	2009-11-13 00:00:00	50000
5	5	107	kredit	2009-11-13 00:00:00	30000
6	1	104	kredit	2009-11-15 00:00:00	200000
7	9	110	kredit	2009-11-15 00:00:00	150000
8	5	102	debit	2009-11-16 00:00:00	20000
9	3	105	kredit	2009-11-18 00:00:00	50000
10	4	107	debit	2009-11-19 00:00:00	100000
11	2	103	debit	2009-11-19 00:00:00	100000
12	1	104	debit	2009-11-19 00:00:00	50000
13	4	107	kredit	2009-11-20 00:00:00	200000
14	3	105	debit	2009-11-21 00:00:00	40000
15	1	104	kredit	2009-11-22 00:00:00	100000
16	4	101	kredit	2009-11-22 00:00:00	20000
17	2	103	debit	2009-11-22 00:00:00	50000
18	5	102	debit	2009-11-25 00:00:00	50000
19	10	108	debit	2009-11-26 00:00:00	100000
20	3	106	kredit	2009-11-27 00:00:00	50000
21	2	103	kredit	2009-11-28 00:00:00	200000
22	3	105	kredit	2009-11-28 00:00:00	100000
23	5	102	debit	2009-11-30 00:00:00	20000
24	1	104	debit	2009-12-01 00:00:00	50000
25	2	103	debit	2009-12-02 00:00:00	40000
26	4	101	debit	2009-12-04 00:00:00	50000
27	2	103	kredit	2009-12-05 00:00:00	100000
28	5	102	kredit	2009-12-05 00:00:00	200000
29	7	109	debit	2009-12-05 00:00:00	100000
30	9	110	debit	2009-12-06 00:00:00	20000
31	12	115	debit	2009-11-10 00:00:00	55000
32	12	115	debit	2009-11-10 00:00:00	355000
33	13	116	debit	2009-11-20 00:00:00	355000
34	14	117	debit	2009-11-10 00:00:00	356000
35	15	118	debit	2009-11-20 00:00:00	356000
36	16	119	debit	2009-11-10 00:00:00	356000
37	17	120	debit	2009-11-20 00:00:00	356000
38	18	121	debit	2009-11-10 00:00:00	356000
39	19	122	debit	2009-11-20 00:00:00	200000
40	20	123	debit	2009-11-10 00:00:00	200000
41	12	115	kredit	2009-11-20 00:00:00	200000
42	13	116	kredit	2009-11-22 00:00:00	200000
43	14	117	debit	2009-11-22 00:00:00	200000
44	15	118	kredit	2009-11-22 00:00:00	500000
45	16	119	kredit	2009-11-22 00:00:00	500000
46	17	120	kredit	2009-11-22 00:00:00	500000
47	18	121	kredit	2009-11-25 00:00:00	500000
48	19	122	kredit	2009-11-25 00:00:00	500000
49	20	123	kredit	2009-11-25 00:00:00	342500
50	12	115	kredit	2009-11-25 00:00:00	342500

```
50 rows in set (0.05 sec)
```

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!

```
MariaDB [perbankan] > select nama_nasabah,alamat_nasabah from nasabah order by nama_nasabah;
```

nama_nasabah	alamat_nasabah
Andara	Jl. Pertama 26
Bening	Jl. Agatis 10
Budi Eko Prayogo	Jl. Kantil 30
Budi Martono	Jl. Merak 22
Canka Lokananta	Jl. Tidar 86
Cinta	Jl. Ketiga 24
Danindya	Jl. Aki Balak 22
Kartika Padmasari	Jl. Manggis 15
Khatulistyawara	Jl. Ir. Soekarno 12
Maryati	Jl. MT Haryono 31
Muliana	Jl. Jend. Sudirman 97
Pandan Wangi	Jl. Imam Bonjol 20
Puput	Jl. AMD 10
Putri	Jl. Diponegoro 19
Rangga	Jl. Kedua 80
Sari Murti	Jl. Pangandaran 11
Satria Eka Jaya	Jl. Slamet Riyadi 45
Suparman	Jl. Hasanudin 81
Sutopo	Jl. Jendral Sudirman 12
Trianandya	Jl. Sutoyo 5

```
20 rows in set (0.00 sec)
```

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

```
MariaDB [perbankan]> select jenis_transaksi,jumlah from transaksi where jenis_transaksi like "kredit";
```

jenis_transaksi	jumlah
kredit	20000
kredit	30000
kredit	200000
kredit	150000
kredit	50000
kredit	200000
kredit	100000
kredit	20000
kredit	50000
kredit	200000
kredit	100000
kredit	100000
kredit	200000
kredit	20000
kredit	25000
kredit	50000
kredit	50000
kredit	50000
kredit	125000
kredit	125000
kredit	175000
kredit	80000
kredit	130000
kredit	150000

```
14 rows in set (0.00 sec)
```

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

```
MariaDB [perbankan]> select jenis_transaksi,jumlah from transaksi where tanggal = "2009-11-21" order by jumlah;
```

jenis_transaksi	jumlah
debit	400000

```
1 row in set (0.00 sec)
```

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

```
MariaDB [perbankan]> select nama_nasabah,jenis_transaksi, jumlah from nasabah, transaksi where nasabah.id_nasabah = transaksi.id_nasabahFK and jumlah = 20000;
```

nama_nasabah	jenis_transaksi	jumlah
Kartika Padmasari	kredit	20000
Budi Eko Prayogo	debit	20000
Kartika Padmasari	kredit	20000
Budi Eko Prayogo	debit	20000
Canka Lokananta	debit	20000
Puput	kredit	20000

```
6 rows in set (0.00 sec)
```

5. Tampilkan nama nasabah dan alamat nasabah dimana nama nasabah diawali dengan kata 'Su'!

```
MariaDB [perbankan]> select nama_nasabah,alamat_nasabah from nasabah where nama_nasabah like "Su%";
```

nama_nasabah	alamat_nasabah
Sutopo	Jl. Jendral Sudirman 12
Suparman	Jl. Hasanudin 81

```
2 rows in set (0.00 sec)
```

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!

```
Database changed
MariaDB [perbankan2]> select nasabah.nama_nasabah,nasabah.alamat_nasabah,jenis_transaksi,jumlah from nasabah, transaksi
where jenis_transaksi = "kredit" and nasabah.id_nasabah = transaksi.id_nasabahFK order by nama_nasabah;
```

nama_nasabah	alamat_nasabah	jenis_transaksi	jumlah
Budi Eko Prayogo	Jl. Kantil 30	kredit	30000
Budi Eko Prayogo	Jl. Kantil 30	kredit	200000
Budi Murtono	Jl. Merak 22	kredit	250000
Canka Lokananta	Jl. Tidar 86	kredit	150000
Indri Hapsari	Jl. Sutoyo 5	kredit	130000
Kartika Padmasari	Jl. Manggis 15	kredit	20000
Kartika Padmasari	Jl. Manggis 15	kredit	20000
Kartika Padmasari	Jl. Manggis 15	kredit	200000
Maryati	Jl. MT. Haryono 31	kredit	200000
Maryati	Jl. MT. Haryono 31	kredit	100000
Nur	Kalimantan	kredit	50000
Nur	Kalimantan	kredit	50000
Nur	Kalimantan	kredit	50000
Nur	Kalimantan	kredit	50000
Nur	Kalimantan	kredit	50000
Nur	Kalimantan	kredit	50000
Sari Murti	Jl. Pangandaran 11	kredit	357000
Suparman	Jl. Hasanudin 81	kredit	100000
Suparman	Jl. Hasanudin 81	kredit	50000
Suparman	Jl. Hasanudin 81	kredit	50000
Sutopo	Jl. Jendral Sudirman 12	kredit	200000
Sutopo	Jl. Jendral Sudirman 12	kredit	520000
Sutopo	Jl. Jendral Sudirman 12	kredit	100000

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

```
MariaDB [perbankan2]> select rekening.no_rekening,nasabah.nama_nasabah,jenis_transaksi,jumlah from nasabah,rekening,transaksi
where tanggal = "2009-11-21" and nasabah.id_nasabah = transaksi.id_nasabahFK and rekening.no_rekening = transaksi.no_rekeningFK
order by nasabah.nama_nasabah;
```

no_rekening	nama_nasabah	jenis_transaksi	jumlah
105	Suparman	debit	40000

1 row in set (0.06 sec)

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

```
MariaDB [perbankan2]> select no_rekening,nama_nasabah,jenis_transaksi,jumlah from nasabah,rekening,transaksi where jumlah
= 20000 and nasabah.id_nasabah = transaksi.id_nasabahFK and rekening.no_rekening = transaksi.no_rekeningFK;
```

no_rekening	nama_nasabah	jenis_transaksi	jumlah
101	Kartika Padmasari	kredit	20000
102	Budi Eko Prayogo	debit	20000
101	Kartika Padmasari	kredit	20000
102	Budi Eko Prayogo	debit	20000
110	Canka Lokananta	debit	20000

5 rows in set (0.00 sec)

4. Tampilkan nomor rekening, nama nasabah dan alamat nasabah dimana nama nasabah diawali dengan kata 'Su'!

```
MariaDB [perbankan2]> select no_rekening,nama_nasabah,alamat_nasabah from nasabah,rekening,nasabah_has_rekening where na
ma_nasabah like "Su%" and nasabah.id_nasabah = nasabah_has_rekening.id_nasabahFK and rekening.no_rekening = nasabah_has_
rekening.no_rekeningFK;
```

no_rekening	nama_nasabah	alamat_nasabah
104	Sutopo	Jl. Jendral Sudirman 12
105	Suparman	Jl. Hasanudin 81
106	Suparman	Jl. Hasanudin 81

3 rows in set (0.01 sec)

5. 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!

```
MariaDB [perbankan2]> select rekening.no_rekening as "Nomor Rekening",nasabah.nama_nasabah as "Nama Nasabah",jumlah as "Jumlah Transaksi" from nasabah,rekening,transaksi where jenis_transaksi = "debit" and nasabah.id_nasabah = transaksi.id_nasabahFK and rekening.no_rekening = transaksi.no_rekeningFK order by nasabah.nama_nasabah;
```

Nomor Rekening	Nama Nasabah	Jumlah Transaksi
109	Anggit	700000
109	Anggit	700000
109	Anggit	700000
109	Anggit	700000
110	Budi Eko Prayogo	26000
102	Budi Eko Prayogo	20000
102	Budi Eko Prayogo	20000
103	Budi Eko Prayogo	263000
102	Budi Eko Prayogo	50000
108	Budi Murtono	100000
110	Canka Lokananta	20000
109	Indri Hapsari	100000

MODUL 8 Agregasi

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!

```
Command Prompt - mysql -u root -p
MariaDB [perbankan1]> select distinct jenis_transaksi, jumlah as "Jumlah(Rp.)", count(jenis_transaksi) as "Total Transaksi" from transaksi T, nasabah N where T.id_nasabahFK = N.id_nasabah and N.nama_nasabah like "Kartika Padmasari%" group by T.jenis_transaksi order by T.jenis_transaksi;
```

jenis_transaksi	Jumlah(Rp.)	Total Transaksi
debit	100000	2
kredit	20000	3

2 rows in set (0.00 sec)

2. Berapa jumlah total saldo yang dimiliki oleh Maryati?

```
MariaDB [perbankan1]> select sum(transaksi.jumlah) as "Jumlah Saldo" FROM transaksi, nasabah WHERE nasabah.id_nasabah = transaksi.id_nasabahFK AND nasabah.nama_nasabah = "Maryati" AND transaksi.jenis_transaksi = "debit";
```

Jumlah Saldo
777000

1 row in set (0.00 sec)

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

```
Command Prompt - mysql -u root -p
MariaDB [perbankan1]> select distinct kode_cabang,nama_cabang,count(jumlah)as "Jumlah Transaksi" from cabang_bank C,transaksi T,rekening R where C.kode_cabang = R.kode_cabangFK AND T.no_rekeningFK = R.no_rekening GROUP BY C.kode_cabang;
```

kode_cabang	nama_cabang	Jumlah Transaksi
BRUB	Bank Rut Unit Boyolali	1
BRUK	Bank Rut Unit Klaten	2
BRUM	Bank Rut Unit Magelang	4
BRUS	Bank Rut Unit Surakarta	29
BRUW	Bank Rut Unit Wonogiri	4
BRUY	Bank Rut Unit Yogyakarta	10

6 rows in set (0.00 sec)

4. Tampilkan nama nasabah dan jumlah saldo yang memiliki saldo antara Rp. 500.000 sampai Rp. 2.000.000!

```

Command Prompt - mysql -u root -p
MariaDB [perbankan1]> SELECT nasabah.nama_nasabah as "Nama", rekening.saldo as "Jumlah saldo" from nasabah, rekening where
re saldo between 500000 and 1000000 group by id_nasabah;
+-----+-----+
| Nama          | Jumlah saldo |
+-----+-----+
| Sutopo        | 500000       |
| Maryati       | 500000       |
| Suparman      | 500000       |
| Kartika Padmasari | 500000       |
| Budi Eko Prayogo | 500000       |
| Satria Eka Jaya | 500000       |
| Indri Hapsari  | 500000       |
| Sari Murti     | 500000       |
| Canka Lokananta | 500000       |
| Budi Murtono   | 500000       |
| Joko Ndo Kondo | 500000       |
| Jon Koplo     | 500000       |
| Anggit        | 500000       |
| Sutopo        | 500000       |
| Maryati       | 500000       |
+-----+-----+
15 rows in set (0.00 sec)

```

5. 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!

```

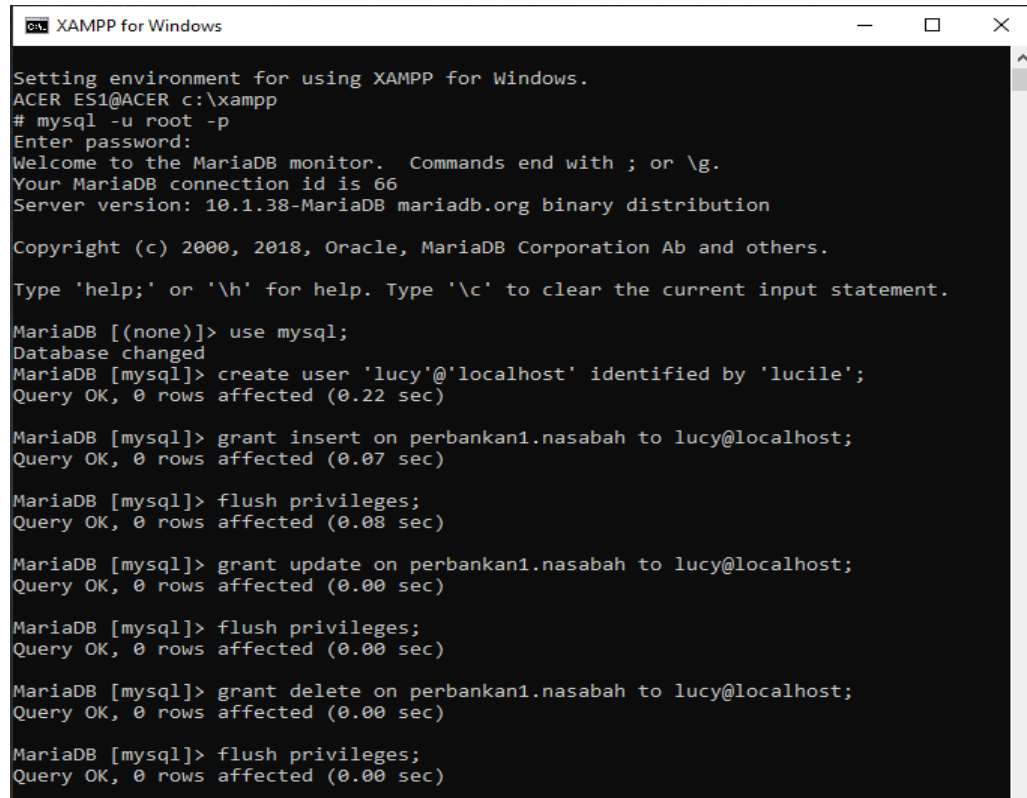
Command Prompt - mysql -u root -p
MariaDB [perbankan1]> select N.nama_nasabah as "Nama Nasabah", T.tanggal as 'Tanggal', T.jumlah as 'Jumlah (Rp.)' from n
asabah N, transaksi T where N.id_nasabah = T.id_nasabahFK and T.jumlah >= 100000 order by T.jumlah DESC;
+-----+-----+-----+
| Nama Nasabah | Tanggal          | Jumlah (Rp.) |
+-----+-----+-----+
| Jon Koplo    | 2017-05-26 00:00:00 | 9700000       |
| Jon Koplo    | 2017-05-26 00:00:00 | 8500000       |
| Anggit       | 2017-06-08 00:00:00 | 700000        |
| Anggit       | 2017-06-08 00:00:00 | 700000        |
| Anggit       | 2017-06-08 00:00:00 | 700000        |
| Anggit       | 2017-05-26 00:00:00 | 700000        |
| Maryati      | 2017-05-24 00:00:00 | 547000        |
| Sutopo       | 2017-05-26 00:00:00 | 520000        |
| Sari Murti   | 2017-05-24 00:00:00 | 357000        |
| Budi Eko Prayogo | 2017-06-14 00:00:00 | 263000        |
| Budi Murtono | 2017-05-24 00:00:00 | 250000        |
| Sutopo       | 2009-11-15 00:00:00 | 200000        |
| Budi Eko Prayogo | 2009-12-05 00:00:00 | 200000        |
| Maryati      | 2009-11-28 00:00:00 | 200000        |
| Kartika Padmasari | 2009-11-20 00:00:00 | 200000        |
| Canka Lokananta | 2009-11-15 00:00:00 | 150000        |
| Indri Hapsari | 2017-05-24 00:00:00 | 130000        |
| Satria Eka Jaya | 2017-05-24 00:00:00 | 125000        |
| Kartika Padmasari | 2009-11-19 00:00:00 | 100000        |
| Maryati      | 2009-11-19 00:00:00 | 100000        |
| Sutopo       | 2009-11-22 00:00:00 | 100000        |
| Budi Murtono | 2009-11-26 00:00:00 | 100000        |
| Suparman     | 2009-11-28 00:00:00 | 100000        |
| Maryati      | 2009-12-05 00:00:00 | 100000        |
| Indri Hapsari | 2009-12-05 00:00:00 | 100000        |
+-----+-----+-----+

```

MODUL 9 Data Control Language

E. Tugas

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



```
XAMPP for Windows
Setting environment for using XAMPP for Windows.
ACER ES1@ACER c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 66
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql;
Database changed
MariaDB [mysql]> create user 'lucy'@'localhost' identified by 'lucile';
Query OK, 0 rows affected (0.22 sec)

MariaDB [mysql]> grant insert on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.07 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.08 sec)

MariaDB [mysql]> grant update on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.00 sec)

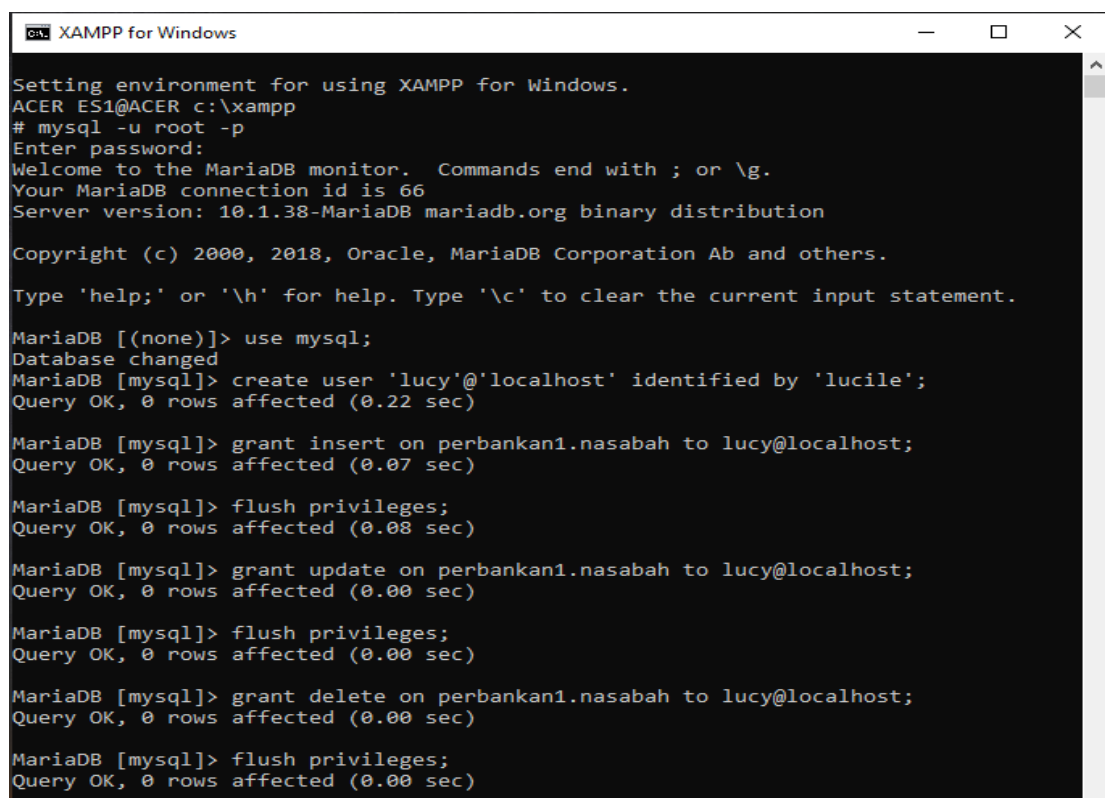
MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> grant delete on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

2. Beri Privilege untuk user tersebut dengan ketentuan sebagai berikut ini.

- a. Dapat melakukan insert,update,dan delete pada tabel nasabah.



```
XAMPP for Windows
Setting environment for using XAMPP for Windows.
ACER ES1@ACER c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 66
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql;
Database changed
MariaDB [mysql]> create user 'lucy'@'localhost' identified by 'lucile';
Query OK, 0 rows affected (0.22 sec)

MariaDB [mysql]> grant insert on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.07 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.08 sec)

MariaDB [mysql]> grant update on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> grant delete on perbankan1.nasabah to lucy@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

- b. Hanya dapat melakukan select pada tabel cabang_bank.

```
MariaDB [mysql]> grant select on perbankan1.cabang_bank to lucy@localhost;
Query OK, 0 rows affected (0.06 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [mysql]> exit
```

3. Coba lakukan kasus berikut ini dengan login sebagai user yang dibuat dan screenshot hasilnya.

- a. Lakukan perintah select pada tabel nasabah.

```
ACER ES1@ACER c:\xampp
# mysql -u lucy -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 68
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use perbankan1
Database changed
MariaDB [perbankan1]> select * from nasabah;
ERROR 1142 (42000): SELECT command denied to user 'lucy'@'localhost' for table 'nasabah'
```

- b. Lakukan perintah insert pada tabel nasabah.

```
MariaDB [(none)]> use perbankan1;
Database changed
MariaDB [perbankan1]> insert into nasabah values (696969,"deemee","mars");
Query OK, 1 row affected (0.43 sec)

MariaDB [perbankan1]>
```

- c. Lakukan perintah insert pada tabel cabang_bank

```
MariaDB [perbankan1]> insert into cabang_bank values ("BYOB", "Bank Yang Orangnya B
ARBAR", "jakarta");
ERROR 1142 (42000): INSERT command denied to user 'lucy'@'localhost' for table 'cabang_bank'
MariaDB [perbankan1]>
```

4. 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.

```
MariaDB [perkuliahan]> select * from mahasiswa where nim not in(select nim from link_mahasiswa_matkul where kode_matkul = 'TIF001');
+-----+-----+-----+-----+
| nim      | nama                               | alamat   | tgl_lahir |
+-----+-----+-----+-----+
| L200150125 | Indra Bayu Candra Gupta           | Wonogiri | 1993-08-28 |
+-----+-----+-----+-----+
1 row in set (0.08 sec)
```

```
MariaDB [perkuliahan]> select * from mahasiswa where nim not in(select nim from link_mahasiswa_matkul where kode_matkul = 'TIF004');
+-----+-----+-----+-----+
| nim      | nama                               | alamat   | tgl_lahir |
+-----+-----+-----+-----+
| L200150118 | Rina Kurniasari                  | Wonogiri | 1997-11-11 |
| L200150125 | Indra Bayu Candra Gupta           | Wonogiri | 1993-08-28 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Satu dosen dapat mengampu lebih dari satu mata kuliah dan satu mahasiswa dapat mengambil lebih dari satu mata kuliah. Tampilkan daftar mahasiswa yang mengambil semua mata kuliah yang diampu oleh salah satu dosen!

```
MariaDB [perkuliahan]> select mahasiswa.nim, mahasiswa.nama from mahasiswa , link_mahasiswa_matkul, matkul where mahasiswa.nim = link_mahasiswa_matkul.nim and matkul.kode_matkul = link_mahasiswa_matkul.kode_matkul and matkul.kode_dosen = (select kode_dosen from dosen where kode_dosen = 'D002');
+-----+-----+
| nim      | nama                               |
+-----+-----+
| L200150088 | Khofa Prayoga                     |
| L200150099 | Purwantinah                       |
| L200150112 | Dwi Pratiwi Putri                 |
| L200150113 | Ismi Kamelia Najib Putri          |
| L200150118 | Rina Kurniasari                   |
| L200150123 | Danindya Puput Muliana Putri      |
| L200150125 | Indra Bayu Candra Gupta            |
| L200150127 | Lyon                              |
| L200150128 | Sulthana Dzakira Drajat           |
| L200150129 | Fendy                             |
| L200150146 | Fakhrur Razi                      |
| L200154001 | Khilyatin Ulin Fitri              |
+-----+-----+
12 rows in set (0.00 sec)
```


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

```
MariaDB [perkuliahan] > update link_mahasiswa_matkul set kode_matkul = 'TIF002' where kode_matkul = (select kode_matkul from matkul where kode_matkul = 'TIF002');
Query OK, 0 rows affected (0.01 sec)
Rows matched: 14  Changed: 0  Warnings: 0

MariaDB [perkuliahan] > select * from link_mahasiswa_matkul;
```

nim	kode_matkul	nilai
L200150083	TIF001	70
L200150083	TIF002	60
L200150083	TIF003	70
L200150083	TIF004	88
L200150083	TIF005	79
L200150088	TIF001	80
L200150088	TIF002	80
L200150088	TIF003	80
L200150088	TIF004	80
L200150088	TIF005	80
L200150088	TIF006	80
L200150088	TIF007	84
L200150088	TIF008	80
L200150088	TIF010	80
L200150099	TIF001	70
L200150099	TIF002	80
L200150099	TIF003	70
L200150099	TIF004	70
L200150099	TIF005	90
L200150099	TIF006	80
L200150099	TIF007	74
L200150099	TIF008	70
L200150099	TIF010	70
L200150112	TIF001	90
L200150112	TIF002	70
L200150112	TIF003	70
L200150112	TIF004	80
L200150112	TIF005	70
L200150112	TIF008	70
L200150112	TIF010	70
L200150113	TIF001	70
L200150113	TIF002	60
L200150113	TIF003	70
L200150113	TIF004	80
L200150113	TIF005	70
L200150113	TIF008	70
L200150113	TIF010	80
L200150117	TIF001	80
L200150117	TIF002	80
L200150117	TIF003	70
L200150117	TIF004	88
L200150117	TIF005	89
L200150117	TIF006	80
L200150117	TIF007	81
L200150118	TIF001	80
L200150118	TIF002	80
L200150118	TIF006	80

L200150125	TIF007	78
L200150125	TIF008	78
L200150125	TIF010	82
L200150127	TIF001	90
L200150127	TIF002	85
L200150127	TIF003	94
L200150127	TIF004	96
L200150127	TIF005	80
L200150127	TIF007	85
L200150127	TIF008	80
L200150127	TIF010	92
L200150128	TIF001	80
L200150128	TIF002	80
L200150128	TIF003	78
L200150128	TIF004	80
L200150128	TIF005	76
L200150128	TIF006	80
L200150128	TIF007	74
L200150128	TIF008	85
L200150128	TIF010	80
L200150129	TIF001	80
L200150129	TIF002	90
L200150129	TIF003	88
L200150129	TIF004	80
L200150129	TIF005	96
L200150129	TIF006	90
L200150129	TIF007	94
L200150129	TIF008	85
L200150129	TIF010	90
L200150144	TIF001	80
L200150144	TIF002	80
L200150144	TIF003	70
L200150144	TIF004	78
L200150144	TIF005	79
L200150144	TIF006	80
L200150144	TIF007	81
L200150146	TIF001	90
L200150146	TIF002	90
L200150146	TIF003	90
L200150146	TIF004	90
L200150146	TIF005	90
L200150146	TIF006	90
L200150146	TIF007	94
L200150146	TIF008	90
L200150146	TIF010	90
L200154001	TIF001	70
L200154001	TIF002	80
L200154001	TIF003	90
L200154001	TIF004	80
L200154001	TIF005	70
L200154001	TIF006	80
L200154001	TIF007	94
L200154001	TIF008	80
L200154001	TIF010	70

115 rows in set (0.00 sec)

E. Tugas

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

a. Perintah insert

```
*ekstensi.py - C:/Users/dessyna/AppData/Local/Programs/Python/Python37-32/ekstensi.py (3.7.3)*
File Edit Format Run Options Window Help

from datetime import date,datetime,timedelta
import mysql.connector

cnx = mysql.connector.connect (user = 'root', database = 'perbankan2')
cursor = cnx.cursor()
tanggal = datetime.now().date()
tambah_transaksi = ("INSERT INTO transaksi(id_nasabahFK,no_rekeningFK,jenis_transaksi,tanggal,jumlah)VALUES(%s,%s,%s,%s,%s)")
data_transaksi = ('9','110','kredit',tanggal,'40000')
cursor.execute(tambah_transaksi,data_transaksi)

cnx.commit()
cursor.close()
cnx.close()
```

85	13	109	debit	2017-05-26 00:00:00	700000	
86	19	110	kredit	2017-05-30 00:00:00	50000	
87	19	110	kredit	2017-06-06 00:00:00	50000	
88	19	110	kredit	2017-06-06 00:00:00	50000	
89	13	109	debit	2017-06-08 00:00:00	700000	
90	13	109	debit	2017-06-08 00:00:00	700000	
91	13	109	debit	2017-06-08 00:00:00	700000	
92	5	103	debit	2017-06-14 00:00:00	263000	
93	9	110	kredit	2019-06-03 00:00:00	40000	

51 rows in set (0.00 sec)

b. Perintah update

```
from datetime import date,datetime,timedelta
import mysql.connector

cnx = mysql.connector.connect (user = 'root', database = 'perbankan2')
cursor = cnx.cursor()
tanggal = datetime.now().date()
query = ("UPDATE transaksi SET jenis_transaksi = 'debit' WHERE no_transaksi = 93")
cursor.execute(query)

cnx.commit()
cursor.close()
cnx.close()
```

85	13	109	debit	2017-05-26 00:00:00	700000	
86	19	110	kredit	2017-05-30 00:00:00	50000	
87	19	110	kredit	2017-06-06 00:00:00	50000	
88	19	110	kredit	2017-06-06 00:00:00	50000	
89	13	109	debit	2017-06-08 00:00:00	700000	
90	13	109	debit	2017-06-08 00:00:00	700000	
91	13	109	debit	2017-06-08 00:00:00	700000	
92	5	103	debit	2017-06-14 00:00:00	263000	
93	9	110	debit	2019-06-03 00:00:00	40000	

51 rows in set (0.00 sec)

c. Perintah delete

```
from datetime import date,datetime,timedelta
import mysql.connector

cnx = mysql.connector.connect (user = 'root', database = 'perbankan2')
cursor = cnx.cursor()
tanggal = datetime.now().date()
hapus_transaksi = ("DELETE FROM transaksi WHERE no_transaksi = 93")
cursor.execute(hapus_transaksi)

cnx.commit()
cursor.close()
cnx.close()
```

85	13	109	debit	2017-05-26 00:00:00	700000	
86	19	110	kredit	2017-05-30 00:00:00	50000	
87	19	110	kredit	2017-06-06 00:00:00	50000	
88	19	110	kredit	2017-06-06 00:00:00	50000	
89	13	109	debit	2017-06-08 00:00:00	700000	
90	13	109	debit	2017-06-08 00:00:00	700000	
91	13	109	debit	2017-06-08 00:00:00	700000	
92	5	103	debit	2017-06-14 00:00:00	263000	

50 rows in set (0.00 sec)

2. Buatlah kode program pyhton untuk mendapatkan :

a. Data nasabah

```

from datetime import date,datetime,timedelta
import mysql.connector

cnx = mysql.connector.connect (user = 'root', database = 'perbankan2')
cursor = cnx.cursor()
tanggal = datetime.now().date()
query = ("SELECT * FROM nasabah")
cursor.execute(query)
for (id_nasabah,nama_nasabah,alamat_nasabah) in cursor:
    print ("ID: {} Nama: {} Alamat: {}".format(id_nasabah,nama_nasabah,alamat_nasabah))
cursor.close()
cnx.close()

```

RESTART: C:/Users/dessyna/AppData/Local/Programs/Python/Python37-32/ekstensi.py

```

ID: 1 Nama: Sutopo Alamat: Jl. Jendral Sudirman 12
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 6 Nama: Satria Eka Jaya Alamat: Jl. Slamet Riyadi 45
ID: 7 Nama: Indri Hapsari Alamat: Jl. Sutoyo 5
ID: 8 Nama: Sari Murti Alamat: Jl. Pangandaran 11
ID: 9 Nama: Canka Lokananta Alamat: Jl. Tidar 86
ID: 10 Nama: Budi Murtono Alamat: Jl. Merak 22
ID: 11 Nama: Joko Ndo Kondo Alamat: Jl. Bareng jadian kagak
ID: 12 Nama: Jon Koplo Alamat: Jl. Angin Besar 12
ID: 13 Nama: Anggit Alamat: Solo
ID: 19 Nama: Nur Alamat: Kalimantan
ID: 1000 Nama: Sutopo Alamat: Jl. Jendral Sudirman
ID: 2000 Nama: Maryati Alamat: Jl. MT Haryono 31
>>>

```

b. Data nasabah yang melakukan transaksi antara oktober sampai desember

```

from datetime import date,datetime,timedelta
import mysql.connector

cnx = mysql.connector.connect (user = 'root', database = 'perbankan2')
cursor = cnx.cursor()
tanggal = datetime.now().date()
query = ("SELECT nasabah.* FROM nasabah,transaksi\
        WHERE nasabah.id_nasabah = transaksi.id_nasabahFK AND\
        transaksi.tanggal BETWEEN '2009-10-1' AND '2009-12-31'")
cursor.execute(query)
for (id_nasabah,nama_nasabah,alamat_nasabah) in cursor:
    print ("ID: {} Nama: {} Alamat: {}".format(id_nasabah,nama_nasabah,alamat_nasabah))
cursor.close()
cnx.close()

```


RESTART: C:/Users/dessyna/AppData/Local/Programs/Python/Python37-32/ekstensi.py

```
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 1 Nama: Sutopo Alamat: Jl. Jendral Sudirman 12
ID: 9 Nama: Canka Lokananta Alamat: Jl. Tidar 86
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 1 Nama: Sutopo Alamat: Jl. Jendral Sudirman 12
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 1 Nama: Sutopo Alamat: Jl. Jendral Sudirman 12
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 10 Nama: Budi Murtono Alamat: Jl. Merak 22
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 3 Nama: Suparman Alamat: Jl. Hasanudin 81
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 1 Nama: Sutopo Alamat: Jl. Jendral Sudirman 12
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 4 Nama: Kartika Padmasari Alamat: Jl. Manggis 15
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 5 Nama: Budi Eko Prayogo Alamat: Jl. Kantil 30
ID: 7 Nama: Indri Hapsari Alamat: Jl. Sutoyo 5
ID: 9 Nama: Canka Lokananta Alamat: Jl. Tidar 86
>>>
```