

LAPORAN PRAKTIKUM SISTEM BASIS DATA



Disusun oleh :

Nama : Arlin Widya Rahayu

Nim : L200170014

Kelas : A

PROGRAM STUDI INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

SURAKARTA

2019

Modul 1

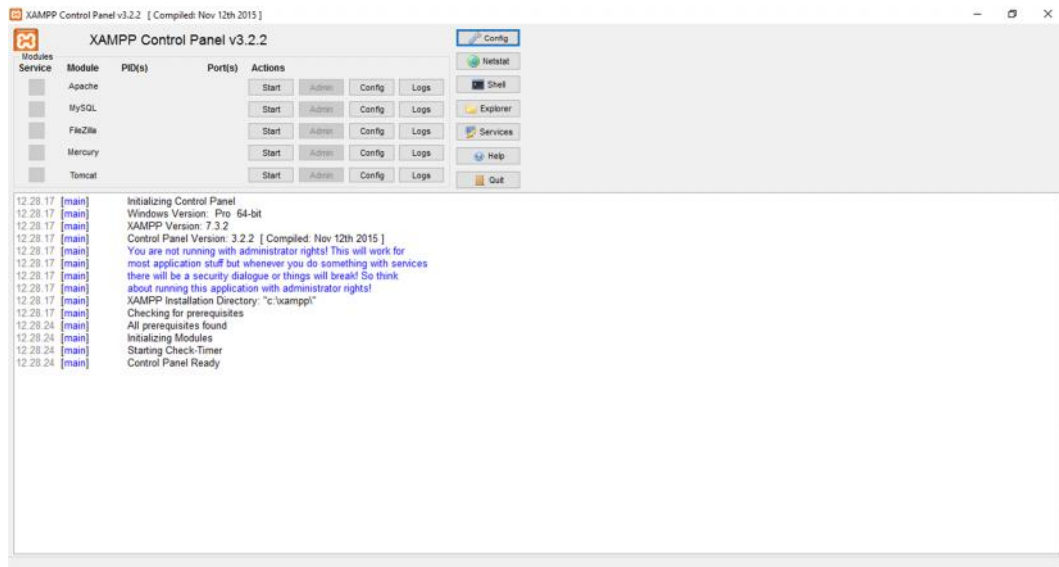
Pengenalan dan Instalasi MySQL

D. Langkah-langkah Praktikum

1. Mengakses PhpMyAdmin

➤ Langkah 1.

Membuka XAMPP Control Panel.

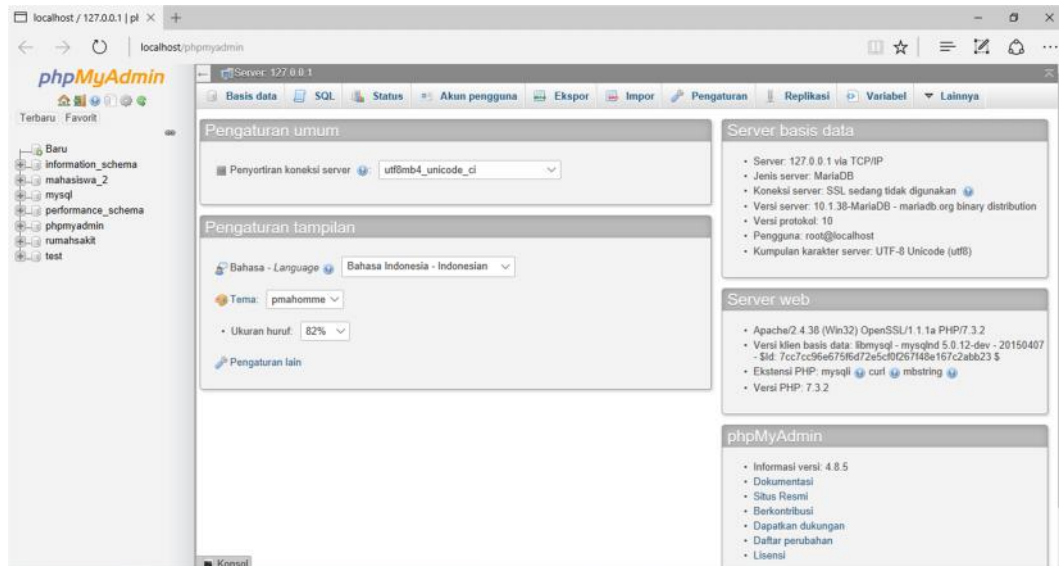


➤ Langkah 2.

Menjalankan Apache Server dan MySQL Server dengan menekan tombol “Start”. Menunggu hingga muncul warna hijau pada nama Module.



- Langkah 3.
- Membuka web browser dan menetikkan <http://localhost/phpmyadmin/>.



- Langkah 4.
- Pada halaman PhpMyAdmin ini kita dapat melakukan pembangunan basis data dan juga melakukan manipulasi isi basis data dengan MySQL. PhpMyAdmin merupakan halaman GUI administratif MySQL server yang saat ini paling banyak dipakai dalam pengembangan aplikasi berbasis web.

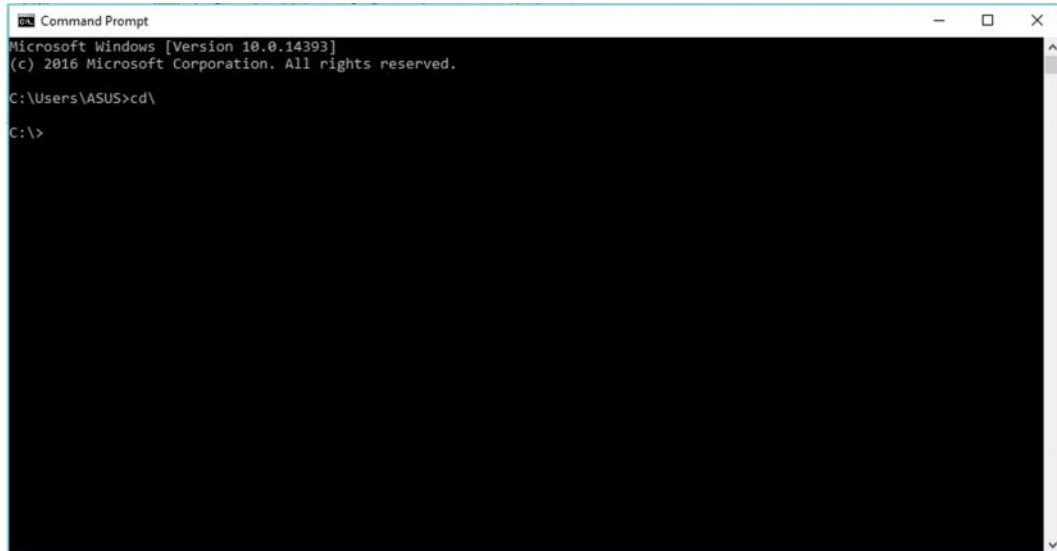
2. Mengakses MySQL lewat command prompt.

- Langkah 1.
- Memastikan bahwa server MySQL telah berjalan.



➤ Langkah 2.

Membuka command prompt dan ketik 'cd\' dan tekan "Enter". Sehingga berada di direktori (C:\).

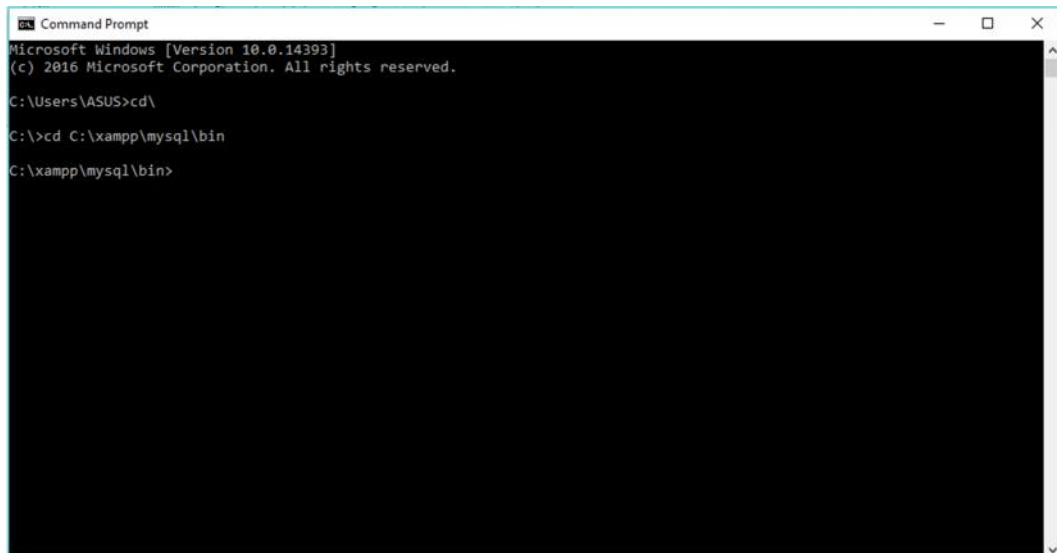


```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd\
C:\>
```

➤ Langkah 3.

Setelah itu mengarahkan ke folder C:\xampp\mysql\bin, caranya ketik 'cd C:\xampp\mysql\bin' tanpa(' ') kemudian tekan tombol Enter.



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd\
C:\>cd C:\xampp\mysql\bin
C:\xampp\mysql\bin>
```

➤ Langkah 4.

Setelah berada di dalam folder C:\xampp\mysql\bin, dapat mengakses mysql. Mengetikkan : 'mysql -u root -p' (tanpa ' ') kemudian tekan tombol Enter. Memasukkan password (jika ada) kemudian klik tombol Enter lagi (secara default tidak ada password untuk root).

```
Command Prompt - mysql -u root -p
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd\

C:\>cd C:\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 17
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)]>
```

➤ Langkah 5.

Setelah tampilan seperti pada langkah 4 berarti sudah berhasil masuk ke MySQL sebagai root user. Untuk melihat database yang ada pada server dapat mengetikkan 'show databases;'. Jangan lupa untuk selalu mengakhiri command dengan ';'.

```
Command Prompt - mysql -u root -p
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd\

C:\>cd C:\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 56
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)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mahasiswa_2        |
| mysql              |
| performance_schema |
| phpmyadmin          |
| rumahsakit         |
| test                |
+-----+
7 rows in set (0.00 sec)

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

MariaDB [(none)]> use mahasiswa;
Database changed
MariaDB [mahasiswa]> create table datadiri(
  -> nim varchar(20) primary key,
  -> nama varchar(255),
  -> alamat varchar(255));
Query OK, 0 rows affected (0.16 sec)
```

```
Command Prompt - mysql -u root -p
MariaDB [mahasiswa]> describe datadiri;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | varchar(20) | NO | PRI | NULL | |
| nama  | varchar(255) | YES | | NULL | |
| alamat | varchar(255) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [mahasiswa]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mahasiswa |
| mahasiswa_2 |
| mysql |
| performance_schema |
| phpmyadmin |
| rumahakit |
| test |
+-----+
8 rows in set (0.00 sec)

MariaDB [mahasiswa]>
```

E. Tugas

1. Jelaskan mengapa dibutuhkan data!

Jawab:

Data adalah informasi yang mengandung arti. Maka dari itu data sangat dibutuhkan karena informasi yang ada akan memberikan arti yang sangat penting baik untuk saat ini maupun untuk akan datang.

2. Jelaskan manfaat database dan contohnya!

Jawab:

-) Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
-) 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.
-) Mengatasi kerangkapan data (redundancy data).
-) Menghindari terjadinya inkonsistensi data.
-) Mengatasi kesulitan dalam mengakses data.
-) Menyusun format yang standar dari sebuah data.
-) Penggunaan oleh banyak pemakai (multiple user). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multiuser).
-) 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.

-) 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 rumah sakit, database akademik, database perusahaan, database bank, dll.

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

Jawab:

-) Deskripsikan kebutuhan informasi dan data
 -) Spesifikasi data
 -) Pemrosesan yang diperlukan oleh data
 -) Pertimbangan keamanan
 -) Kecocokan dengan tipe aplikasi
 -) Bahasa query
 -) Biaya tak langsung terhadap pemrosesan
4. Jelaskan istilah atau terminology yang digunakan dalam Database (database, table, field, record)!

Jawab:

-) Database: kumpulan data yang saling berhubungan satu dengan yang lainnya yang diaorganisasikan sesuai struktur tertentu dan disimpan dengan baik.
 -) Table : Tempat untuk menyimpan data, tabel terdiri dari field dan record..
 -) Field : menyatakan data terkecil yang memiliki makna. Field juga disebut atribut.
 -) Record : disebut juga dengan baris, yaitu satu bagian informasi yang disimpan dalam tabel, misal data seorang mahasiswa akan disimpan dalam satu record yang terdiri dari beberapa kolom/field.
5. Bandingkan perbedaan pengolahan data secara manual dengan menggunakan system database.

Jawab:

Pengolahan data secara manual lebih mengandalkan operasi dengan bantuan tangan dan alat seperti pensil, pulpen, kertas, dll. Maka dari itu pengolahan ini lebih membutuhkan banyak berkas dan penyimpanan untuk menyimpan berbagai data dan informasi, keamanannya pun lebih rentan, data terbatas untuk dibagi, integritas data kurang. Sifatnya 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?

Jawab:

Database Manajement System (DBMS) merupakan software yang digunakan untuk membangun sebuah sistem basis data yang berbasis komputerisasi. DBMS membantu dalam pemeliharaan dan pengolahan kumpulan data dalam jumlah besar. Sehingga dengan menggunakan DBMS tidak menimbulkan kekacauan dan dapat digunakan oleh pengguna sesuai dengan kebutuhan. DBMS merupakan perantara bagi pemakai dengan basis data.

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

Jawab:

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. Pada field TANGGAL_LAHIR diatur tipe date/time yang berarti pada record dapat diisikan data berupa angka dan symbol. Pada field TELEPHONE diatur tipe integer, berarti pada record dapat diisikan angka saja.

Modul 2

Perancangan Basis Data

D. Langkah-langkah Praktikum

Merancang basis data untuk permasalahan berikut ini.

Suatu perusahaan software diminta membuatkan basis data yang akan menangani data-data perbankan. Data-data yang akan ditanganinya adalah: data pribadi mengenai nasabah, data account deposit yang dimiliki oleh nasabah, cabang bank dimana nasabah membuka depositnya, dan data transaksi yang dilakukan nasabah. Nasabah boleh mempunyai lebih dari satu account deposit, dan satu account deposit boleh dimiliki oleh lebih dari satu nasabah sekaligus (*joint account*).

Langkah-langkah perncangan database perbankan:

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

-) nasabah : menyimpan semua data pribadi semua nasabah
-) rekening : menyimpan informasi semua rekening yang telah dibuka
-) cabang_bank : menyimpan informasi tentang semua cabang bank
-) transaksi : menyimpan informasi tentang semua transaksi yang telah terjadi

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

-) nasabah :
 - o id_nasabah : nomor id untuk nasabah (integer) PK
 - o nama_nasabah : nama lengkap nasabah (varchar(45))
 - o alamat_nasabah : alamat lengkap nasabah (varchar(255))
-) rekening :
 - o no_rekening : nomor rekening (integer) PK
 - o pin : personal identification number (varchar(10))
 - o saldo : jumlah saldo rekening dalam Rp (integer)
-) cabang_bank :
 - o kode_cabang : kode untuk cabang bank (varchar(10)) PK
 - o nama_cabang : nama lengkap cabang bank (varchar(20))
 - o alamat_cabang : alamat lengkap cabang bank (varchar(255))
-) transaksi :
 - o no_transaksi : nomer transaksi (integer) PK
 - o jenis_transaksi : kredit atau debit (varchar(10))
 - o tanggal : tanggal terjadinya transaksi (date)
 - o jumlah : besarnya transaksi dalam Rp (integer)

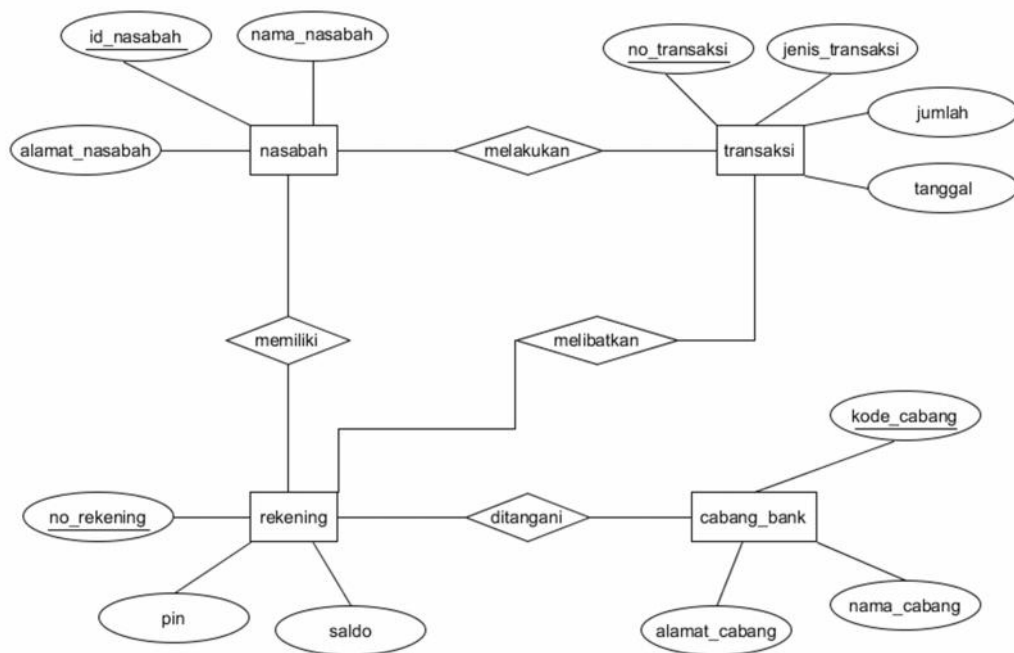
3. Menentukan relationship (hubungan) antar entitas

	nasabah	rekening	cabang_bank	transaksi
nasabah	-	m:n	-	1:n
rekening		-	n:1	1:n
cabang_bank			-	-
transaksi				-

Hubungan

-) nasabah memiliki rekening :
 - o tabel utama : nasabah, rekening
 - o tabel kedua : nasabah_has_rekening
 - o relationship : many-to-many (m:n)
 - o attribute penghubung : id_nasabah, no_rekening (FK id_nasabah, no_rekening di nasabah_has_rekening)
-) nasabah melakukan transaksi :
 - o tabel utama : nasabah
 - o tabel kedua : transaksi
 - o relationship : one-to-many(1:n)
 - o attribute penghubung : id_nasabah (FK id_nasabah di transaksi)
-) cabang_bank menangani rekening :
 - o tabel utama : cabang_bank
 - o tabel kedua : rekening
 - o rekening : one-to-many (1:n)
 - o attribute penghubung : kode_cabang (FK kode_cabang di rekening)
-) rekening terlibat dalam transaksi :
 - o tabel utama : rekening
 - o tabel kedua : transaksi
 - o relationship : one-to-many (1:n)
 - o attribute penghubung : no_rekening (FK no_rekening di transaksi)

4. Menggambar ERD Diagram



E. Tugas

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

Jawab:

- Menentukan entities (object-object dasar) yang perlu ada di database.
 -) mahasiswa : menyimpan semua data pribadi semua mahasiswa
 -) dosen : menyimpan semua data pribadi semua dosen
 -) matakuliah : menyimpan informasi tentang semua mata kuliah
 -) kelas : menyimpan informasi tentang data semua ruang kelas
- Menentukan attributes (sifat-sifat) masing-masing *entity* sesuai kebutuhan database
 -) mahasiswa :
 - o nim : nim untuk mahasiswa (varchar(15)) PK
 - o nama_mhs : nama lengkap mahasiswa (varchar(75))
 - o alamat_mhs : alamat lengkap mahasiswa (varchar(255))
 - o tgllahir_mhs : tanggal lahir mahasiwa (date)
 -) dosen :
 - o nip_dosen : nip untuk dosen (varchar(15)) PK
 - o nama_dosen : nama lengkap dosen (varchar(75))

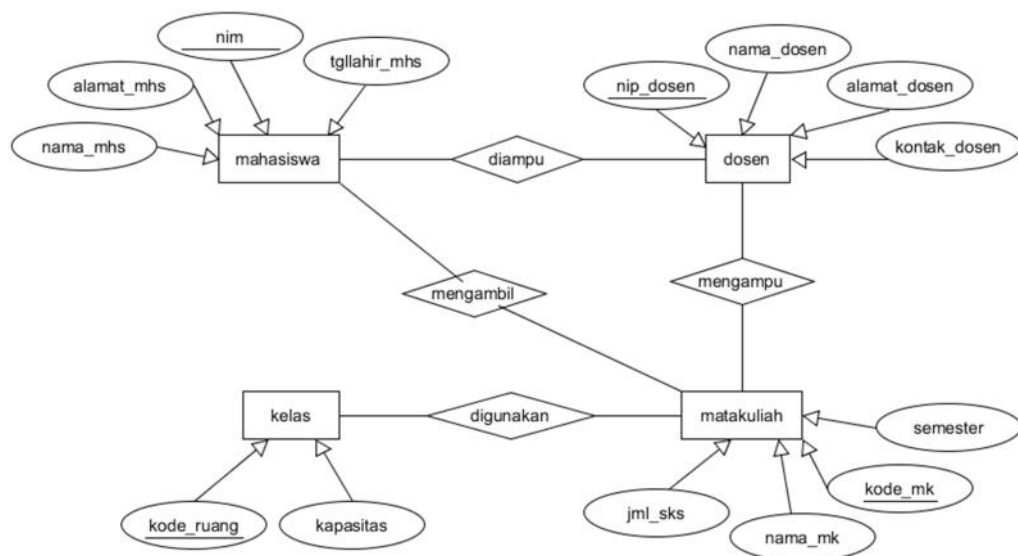
- o alamat_dosen : alamat lengkap dosen (varchar(255))
- o kontak_dosen : nomor telephone dosen (integer(12))
-) matakuliah :
 - o kode_mk : kode untuk mata kuliah (varchar(10)) PK
 - o nama_mk : nama mata kuliah (varchar(50))
 - o jml_sks : jumlah sks mata kuliah (integer)
 - o semester : semester mata kuliah (integer)
-) kelas :
 - o kode_ruang : kode ruang kelas (varchar(10)) PK
 - o kapasitas : kapasitas ruang kelas (integer)

3. Menentukan relationship (hubungan) antar entitas

	mahasiswa	dosen	matakuliah	kelas
mahasiswa	-	1:n	m:n	-
dosen		-	1:n	-
matakuliah			-	1:1
kelas				-

Hubungan

-) mahasiswa diampu dosen :
 - o tabel utama : dosen
 - o tabel kedua : mahasiswa
 - o relationship : many-to-one (n:1)
 - o attribute penghubung : nip_dosen (FK nip_dosen di dosen)
-) dosen mengampu matakuliah :
 - o tabel utama : matakuliah
 - o tabel kedua : dosen
 - o relationship : many-to-one(n:1)
 - o attribute penghubung : kode_mk (FK kode_mk di matakuliah)
-) mahasiswa mengambil matakuliah :
 - o tabel utama : mahasiswa
 - o tabel kedua : mahasiswa_has_matakuliah
 - o rekening : many-to-many (m:n)
 - o attribute penghubung : nim, kode_mk (FK nim, kode_mk di mahasiswa_has_matakuliah)
-) kelas digunakan untuk matakuliah :
 - o tabel utama : kelas
 - o tabel kedua : matakuliah
 - o relationship : one-to-one (1:1)
 - o attribute penghubung : kode_ruang (FK kode_ruang di matakuliah)
-) Menggambar ERD Diagram



2. Ambil contoh sembarang database (harus berbeda untuk setiap mahasiswa).
Buatlah rancangan ER Diagram manual database tersebut dari tahap 1 sampai
tahap 4, dengan ketentuan database minimal mengandung 4 buah entitas.

Jawab:

Database Rumah Sakit

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

-) pasien : menyimpan semua data pribadi semua pasien
-) dokter : menyimpan semua data pribadi semua dokter
-) receptionist : menyimpan semua data pribadi semua receptionist
-) kamar : menyimpan informasi tentang data semua kamar

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

) mahasiswa :

- o id_pasien : id untuk pasien (varchar(15)) PK
- o nama_pasien : nama lengkap pasien (varchar(75))
- o alamat_pasien : alamat lengkap pasien (varchar(255))
- o umur_pasien : umur pasien (integer)

) dokter :

- o id_dokter : id untuk dokter (varchar(15)) PK
- o nama_dokter : nama lengkap dokter (varchar(75))
- o alamat_dokter : alamat lengkap dokter (varchar(255))
- o spesialis : spesialis dokter (varchar(20))

) receptionist :

- o id_receptionist : id untuk receptionist (varchar(10)) PK
- o nama_receptionist : nama receptionist (varchar(50))
- o alamat_receptionist : alamat lengkap receptionist (varchar(255))
- o jam_jaga : jam jaga receptionist (varchar(10))

-) kamar :
 - o no_kamar : nomer kamar (varchar(10)) PK
 - o kelas_kamar : kelas kamar (integer)

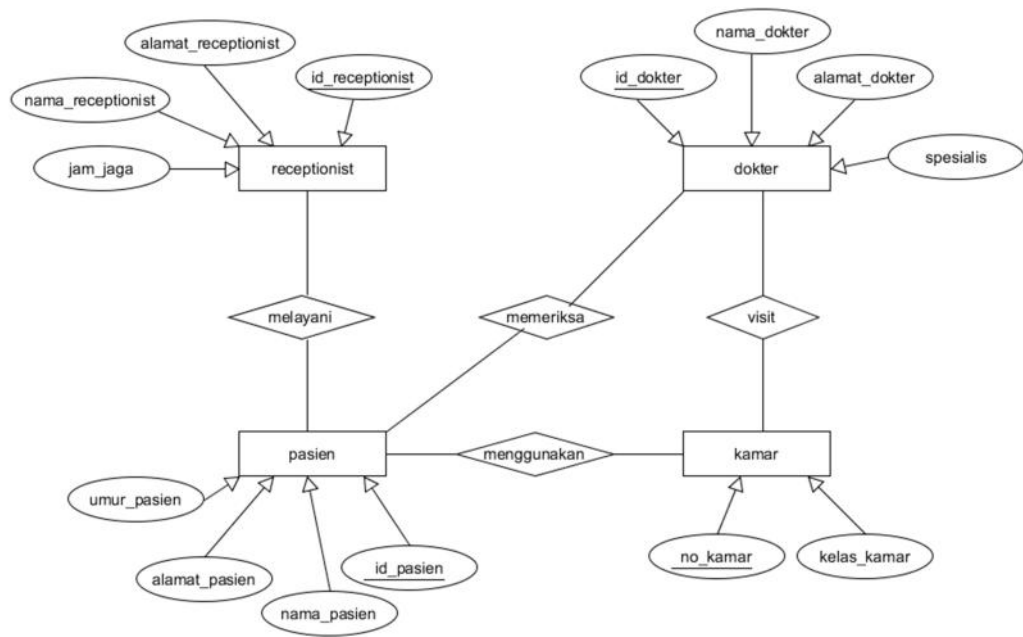
3. Menentukan relationship (hubungan) antar entitas

	pasien	dokter	receptionist	kamar
Pasien	-	-	-	1:1
dokter	1:n	-	-	1:n
receptionist	m:n	-	-	-
kamar	-	-	-	-

Hubungan

-) Dokter memeriksa pasien :
 - o tabel utama : dokter
 - o tabel kedua : pasien
 - o relationship : one-to-many (1:n)
 - o attribute penghubung : id_dokter (FK id_dokter di pasien)
-) receptionist melayani pasien :
 - o tabel utama : receptionist, pasien
 - o tabel kedua : receptionist_has_pasien
 - o relationship : many-to-many(m:n)
 - o attribute penghubung : id_receptionist, id_pasien (FK id_receptionist, id_pasien di receptionist_has_pasien)
-) pasien menggunakan kamar :
 - o tabel utama : pasien
 - o tabel kedua : kamar
 - o rekening : one-to-one (1:1)
 - o attribute penghubung : id_pasien(FK id_pasien di kamar)
-) dokter visit kamar :
 - o tabel utama : dokter
 - o tabel kedua : kamar
 - o relationship : one-to-many (1:n)
 - o attribute penghubung : id_dokter (FK id_dokter di kamar)

4. Menggambar ERD Diagram



Modul 3

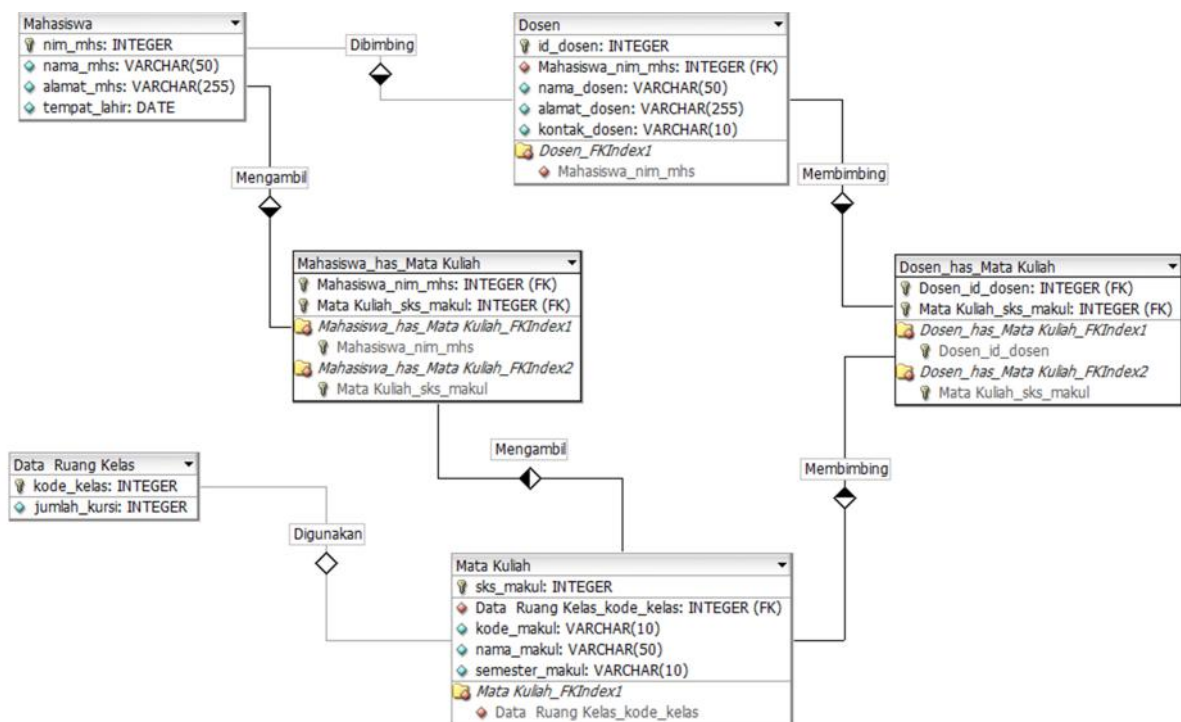
Perancangan E-R Diagram dengan DBDesigner

E. Tugas

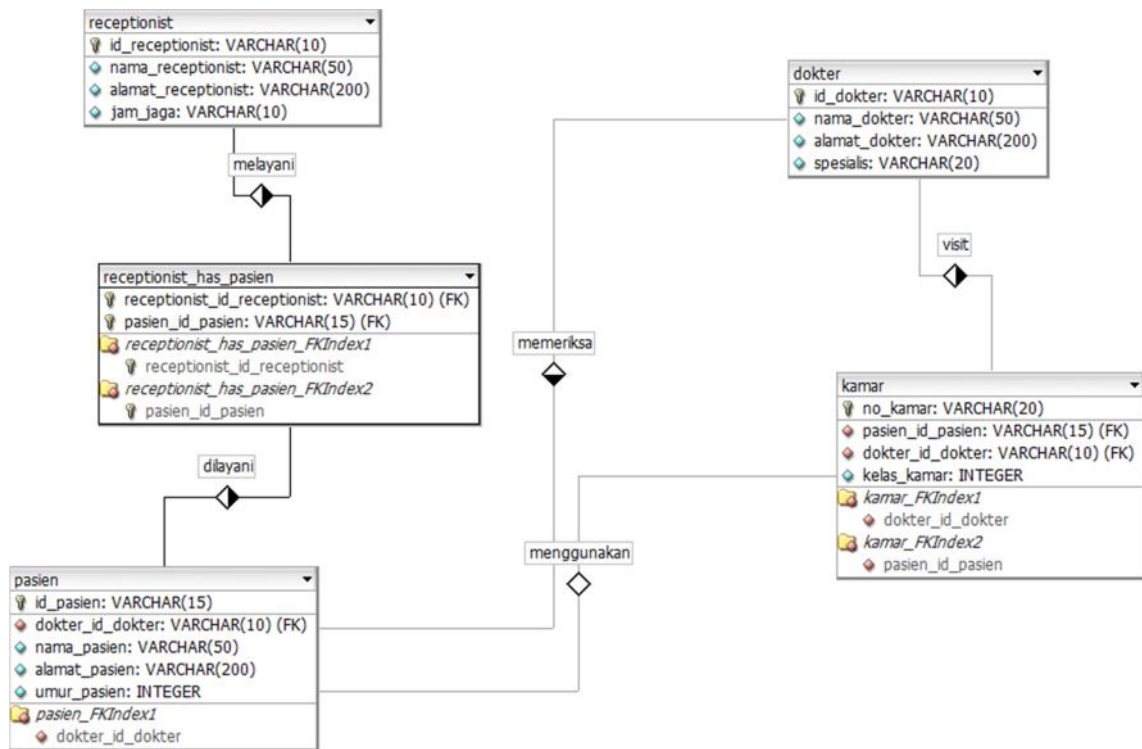
Dari tugas pada modul 3 soal nomer 1 dan 2, buatlah rancangan basis data dari tahap 1 sampai tahap 4. Gunakan program DB Designer untuk membuat ER Diagram.

Jawab:

1. ER Diagram Perkuliahan



2. ER Diagram Rumah Sakit



Modul 4

Data Definition Language (DDL)

E. Tugas

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

Jawab:

```
C:\Windows\system32\cmd.exe - mysql -u root
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\LABSI-10>cd C:/xampp/mysql/bin

C:\xampp>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.1.37-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 mahasiswa;
ERROR 1113 (42000): A table must have at least 1 column
MariaDB [mahasiswa]> create table mahasiswa(
  -> nim varchar(15) primary key,
  -> nama_mhs varchar(75) not null,
  -> alamat_mhs varchar(200) not null,
  -> tgllahir_mhs varchar(30) not null
  -> );
Query OK, 0 rows affected (0.11 sec)

MariaDB [mahasiswa]> describe mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | varchar(15) | NO | PRI | NULL |  |
| nama_mhs | varchar(75) | NO |  | NULL |  |
| alamat_mhs | varchar(200) | NO |  | NULL |  |
| tgllahir_mhs | varchar(30) | NO |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

MariaDB [mahasiswa]> create table dosen(
  -> nip_dosen varchar(15) primary key,
  -> nama_dosen varchar(75) not null,
  -> alamat_dosen varchar(200) not null,
  -> kontak_dosen varchar(12) not null
  -> );
Query OK, 0 rows affected (0.12 sec)

MariaDB [mahasiswa]> describe dosen;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip_dosen | varchar(15) | NO | PRI | NULL |  |
| nama_dosen | varchar(75) | NO |  | NULL |  |
| alamat_dosen | varchar(200) | NO |  | NULL |  |
| kontak_dosen | varchar(12) | NO |  | NULL |  |
+-----+-----+-----+-----+-----+-----+
```

```

C:\Windows\system32\cmd.exe - mysql -u root
4 rows in set (0.01 sec)

MariaDB [mahasiswa]> create table matakuliah(
  -> kode_mk varchar(10) primary key,
  -> nama_mk varchar(50) not null,
  -> jml_sks integer not null,
  -> semester integer not null
  -> );
Query OK, 0 rows affected (0.12 sec)

MariaDB [mahasiswa]> describe matakuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_mk | varchar(10) | NO | PRI | NULL | |
| nama_mk | varchar(50) | NO | | NULL | |
| jml_sks | int(11) | NO | | NULL | |
| semester | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

MariaDB [mahasiswa]> create table kelas(
  -> kode_ruang varchar(10) primary key,
  -> kapasitas integer not null
  -> );
Query OK, 0 rows affected (0.11 sec)

MariaDB [mahasiswa]> describe kelas;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_ruang | varchar(10) | NO | PRI | NULL | |
| kapasitas | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [mahasiswa]> create table mahasiswa_has_matakuliah(
  -> ninfk varchar(15) references mahasiswa(nim) on delete cascade on update cascade,
  -> kode_mk varchar(10) references matakuliah(kode_mk) on delete cascade on update cascade,
  -> primary key(ninfk, kode_mkfk)
  -> );
ERROR 1072 (42000): Key column 'kode_mkfk' doesn't exist in table
MariaDB [mahasiswa]> create table mahasiswa_has_matakuliah(
  -> ninfk varchar(15) references mahasiswa(nim) on delete cascade on update cascade,
  -> kode_mkfk varchar(10) references matakuliah(kode_mk) on delete cascade on update cascade,
  -> primary key(ninfk, kode_mkfk)
  -> );
Query OK, 0 rows affected (0.14 sec)

```

MariaDB [mahasiswa]> describe mahasiswa_has_matakuliah;

Field	Type	Null	Key	Default	Extra
nimfk	varchar(15)	NO	PRI	NULL	
kode_mkfk	varchar(10)	NO	PRI	NULL	

2 rows in set (0.01 sec)

MariaDB [mahasiswa]> alter table mahasiswa add nip_dosenfk varchar(15) references mahasiswa(nim) on delete cascade on update cascade;

Query OK, 0 rows affected (0.28 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mahasiswa]> describe mahasiswa;

Field	Type	Null	Key	Default	Extra
nim	varchar(15)	NO	PRI	NULL	
nama_mhs	varchar(75)	NO		NULL	
alamat_mhs	varchar(200)	NO		NULL	
tgllahir_mhs	varchar(30)	NO		NULL	
nip_dosenfk	varchar(15)	YES		NULL	

5 rows in set (0.00 sec)

MariaDB [mahasiswa]> alter table dosen add nip_dosen varchar(15) references matakuliah(kode_mk) on delete cascade on update cascade;

ERROR 1060 (42S21): Duplicate column name 'nip_dosen'

MariaDB [mahasiswa]> alter table dosen add nip_dosenfk varchar(15) references matakuliah(kode_mk) on delete cascade on update cascade;

Query OK, 0 rows affected (0.16 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mahasiswa]> describe matakuliah;

Field	Type	Null	Key	Default	Extra
kode_mk	varchar(10)	NO	PRI	NULL	
nama_mk	varchar(50)	NO		NULL	
jml_sks	int(11)	NO		NULL	
semester	int(11)	NO		NULL	

4 rows in set (0.01 sec)

MariaDB [mahasiswa]> describe dosen;

Field	Type	Null	Key	Default	Extra
nip_dosen	varchar(15)	NO	PRI	NULL	
nama_dosen	varchar(75)	NO		NULL	
alamat_dosen	varchar(200)	NO		NULL	
kontak_dosen	varchar(12)	NO		NULL	
nip_dosenfk	varchar(15)	YES		NULL	

C:\Windows\system32\cmd.exe - mysql -u root

rows in set (0.00 sec)

MariaDB [mahasiswa] alter table dosen drop nip_dosenfk varchar(15);
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 'var
char(15)' at line 1

MariaDB [mahasiswa] alter table dosen drop nip_dosenfk;

Query OK, 0 rows affected (0.23 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mahasiswa] describe matakuliah

-> ;

Field	Type	Null	Key	Default	Extra
kode_mk	varchar(10)	NO	PRI	NULL	
nama_mk	varchar(50)	NO		NULL	
jml_sks	int(11)	NO		NULL	
semester	int(11)	NO		NULL	

rows in set (0.00 sec)

MariaDB [mahasiswa] alter table matakuliah add nip_dosen varchar(15) references
dosen(nip_dosen) on delete cascade on update cascade;

Query OK, 0 rows affected (0.16 sec)

Records: 0 Duplicates: 0 Warnings: 0

MariaDB [mahasiswa] describe matakuliah;

Field	Type	Null	Key	Default	Extra
kode_mk	varchar(10)	NO	PRI	NULL	
nama_mk	varchar(50)	NO		NULL	
jml_sks	int(11)	NO		NULL	
semester	int(11)	NO		NULL	
nip_dosen	varchar(15)	YES		NULL	

rows in set (0.01 sec)

MariaDB [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)

Jawab:

1. Tabel nasabah

```
MariaDB [modul_5] > 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');
```

Hasilnya:

```
MariaDB [modul_5] > 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. Tabel cabang_bank

```
MariaDB [modul_5] > 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 Denak', '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. Nangka 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');
```

Hasilnya:


```
MariaDB [modul_51]> 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 Denak	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. Tabel rekening

```
MariaDB [modul_51]> 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');
```

Hasilnya:

```
MariaDB [modul_51]> 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. Tabel nasabah_has_rekening

```
MariaDB [modul_51]> 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);
```

Hasilnya:

```
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. Tabel transaksi

```
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 );
```

Hasilnya:

MariaDB [modul 5]> 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)

Modul 6

Query Standart dan Variasinya

E. Tugas

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

Jawab:

```
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. Iman 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!

Jawab:

```
MariaDB [perbankan]> select jenis_transaksi, jumlah from transaksi where jenis_transaksi = "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	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

24 rows in set (0.00 sec)

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

Jawab:

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

jenis_transaksi	jumlah
debit	40000

1 row in set (0.00 sec)

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

Jawab:

```
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

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

Jawab:

```
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!

```
MariaDB [perbankan]> select nasabah.nama_nasabah, nasabah.alamat_nasabah, transaksi.jenis_transaksi, transaksi.jumlah from nasabah, transaksi where nasabah.id_nasabah= transaksi.id_nasabahfk and transaksi.jenis_transaksi= 'kredit' order by nasabah.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	200000
Maryati	Jl. MT. Haryono 31	kredit	100000
Maryati	Jl. MT. Haryono 31	kredit	200000
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	520000
Sutopo	Jl. Jendral Sudirman 12	kredit	100000
Sutopo	Jl. Jendral Sudirman 12	kredit	200000

17 rows in set (0.00 sec)

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

Jawab:

```
MariaDB [perbankan]> select rekening.no_rekening, nasabah.nama_nasabah, transaksi.jenis_transaksi, transaksi.jumlah from rekening, nasabah, transaksi where nasabah.id_nasabah= transaksi.id_nasabahfk and rekening.no_rekening= transaksi.no_rekeningfk and transaksi.tanggal= '2009-11-21' order by nasabah.nama_nasabah;
```

no_rekening	nama_nasabah	jenis_transaksi	jumlah
105	Suparman	debit	40000

1 row in set (0.00 sec)

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

Jawab:

```
MariaDB [perbankan]> select distinct rekening.no_rekening, nasabah.nama_nasabah, transaksi.jenis_transaksi, transaksi.jumlah from rekening, nasabah, transaksi where nasabah.id_nasabah= transaksi.id_nasabahfk and rekening.no_rekening= transaksi.no_rekeningfk and transaksi.jumlah= 20000 order by nasabah.nama_nasabah;
```

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

3 rows in set (0.00 sec)

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

Jawab:

```
MariaDB [perbankan]> select distinct rekening.no_rekening, nasabah.nama_nasabah,
nasabah.alamat_nasabah from rekening, nasabah, transaksi where nasabah.id_nasabah=
transaksi.id_nasabahfk and rekening.no_rekening= transaksi.no_rekeningfk and
nasabah.nama_nasabah like 'Su%' order by nasabah.nama_nasabah;
```

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

4 rows in set (0.00 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!

Jawab:

```
MariaDB [perbankan]> select distinct no_rekening as 'Nomor Rekening', nama_nasabah
as 'Nama Nasabah', jumlah as 'Jumlah Transaksi' from nasabah, rekening, transaksi
where nasabah.id_nasabah= transaksi.id_nasabahfk and rekening.no_rekening=t
ransaksi.no_rekeningfk and jenis_transaksi='debit' order by nasabah.nama_nasabah;
```

Nomor Rekening	Nama Nasabah	Jumlah Transaksi
109	Anggit	700000
102	Budi Eko Prayogo	50000
103	Budi Eko Prayogo	263000
110	Budi Eko Prayogo	26000
102	Budi Eko Prayogo	20000
108	Budi Murtono	100000
110	Canka Lokananta	20000
109	Indri Hapsari	100000
101	Jon Koplo	9700000
101	Jon Koplo	8500000
107	Kartika Padmasari	100000
101	Kartika Padmasari	50000
103	Maryati	50000
103	Maryati	100000
103	Maryati	40000
110	Maryati	547000
106	Satria Eka Jaya	125000
105	Suparman	50000
106	Suparman	50000
105	Suparman	40000
104	Sutopo	50000

21 rows in set (0.00 sec)

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!

```
MariaDB [perbankan]> select transaksi.jenis_transaksi as "Jenis Transaksi", sum(jumlah) as "Jumlah (Rp)", count(jumlah) as "Jumlah Transaksi" from nasabah, transaksi where nasabah.id_nasabah = transaksi.id_nasabahFK and nasabah.nama_nasabah like '%Kartika Padmasari' group by transaksi.jenis_transaksi order by jenis_transaksi;
```

Jenis Transaksi	Jumlah (Rp)	Jumlah Transaksi
debit	150000	2
kredit	240000	3

2 rows in set (0.00 sec)

2. Berapa jumlah total saldo yang dimiliki oleh Maryati?

```
MariaDB [perbankan]> select sum(rekening.saldo) as "Total Saldo" from nasabah, rekening, nasabah_has_rekening where nasabah.id_nasabah = nasabah_has_rekening.id_nasabahfk and rekening.no_rekening = nasabah_has_rekening.no_rekeningfk and nasabah.nama_nasabah in ('Maryati');
```

Total Saldo
750000

1 row in set (0.04 sec)

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

```
MariaDB [perbankan]> select distinct kode_cabang, nama_cabang, count(jumlah) as "Jumlah Transaksi" from cabang_bank, transaksi, rekening where cabang_bank.kode_cabang = rekening.kode_cabangfk and rekening.no_rekening = transaksi.no_rekeningfk group by kode_cabang;
```

kode_cabang	nama_cabang	Jumlah Transaksi
BRUB	Bank Rut Unit Boyolali	11
BRUC	Bank Rut Unit Cikarang	2
BRUJ	Bank Rut Unit Jakarta	2
BRUK	Bank Rut Unit Klaten	4
BRUL	Bank Rut Unit Lamongan	2
BRUM	Bank Rut Unit Magelang	8
BRUS	Bank Rut Unit Surakarta	15
BRUT	Bank Rut Unit Tangerang	2
BRUY	Bank Rut Unit Yogyakarta	2

9 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!

```
MariaDB [perbankan]> select distinct nasabah.nama_nasabah as "Nama Nasabah", rekening.saldo as "Jumlah Saldo" from nasabah, rekening, transaksi where rekening.saldo between '500000' and '2000000' and transaksi.id_nasabahfk = nasabah.id_nasabah and rekening.no_rekening = transaksi.no_rekeningfk;
```

Nama Nasabah	Jumlah Saldo
Parman	2000000
Maryati	750000
Kartika Padmasari	500000
Budi Eko Prayoga	1000000
Sutopo	900000
Kartika Padmasari	1000000
Andi Setiawan	1000000
Riska	1000000
Riska	990000
Supardi	750000

10 rows in set (0.00 sec)

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

```
MariaDB [perbankan]> select distinct nasabah.nama_nasabah, transaksi.tanggal, transaksi.jumlah as "Jumlah Transaksi(Rp)" from nasabah, transaksi where transaksi.jumlah>=100000 and nasabah.id_nasabah = transaksi.id_nasabahfk order by transaksi.jumlah desc;
```

nama_nasabah	tanggal	Jumlah Transaksi(Rp)
Budi Eko Prayoga	2019-05-26 10:56:14	200000
Andi Setiawan	2019-05-26 10:45:16	200000
Riska	2019-05-26 10:45:16	200000
Maryati	2019-05-26 10:56:13	200000
Kartika Padmasari	2009-11-20 00:00:00	200000
Sutopo	2009-11-15 00:00:00	200000
Supardi	2019-05-26 10:45:16	150000
Canka Lokananta	2009-11-15 00:00:00	150000
Budi Murtono	2019-05-26 10:56:13	100000
Kartika Padmasari	2009-11-19 00:00:00	100000
Maryati	2009-11-19 00:00:00	100000
Riska	2019-05-26 10:45:16	100000
Parman	2019-05-26 10:56:14	100000
Sri Maryati	2019-05-26 10:45:16	100000
Ahmad	2019-05-26 10:45:16	100000
Maryati	2019-05-26 10:56:14	100000
Sutopo	2019-05-26 10:56:13	100000
Listiani	2019-05-26 10:45:16	100000

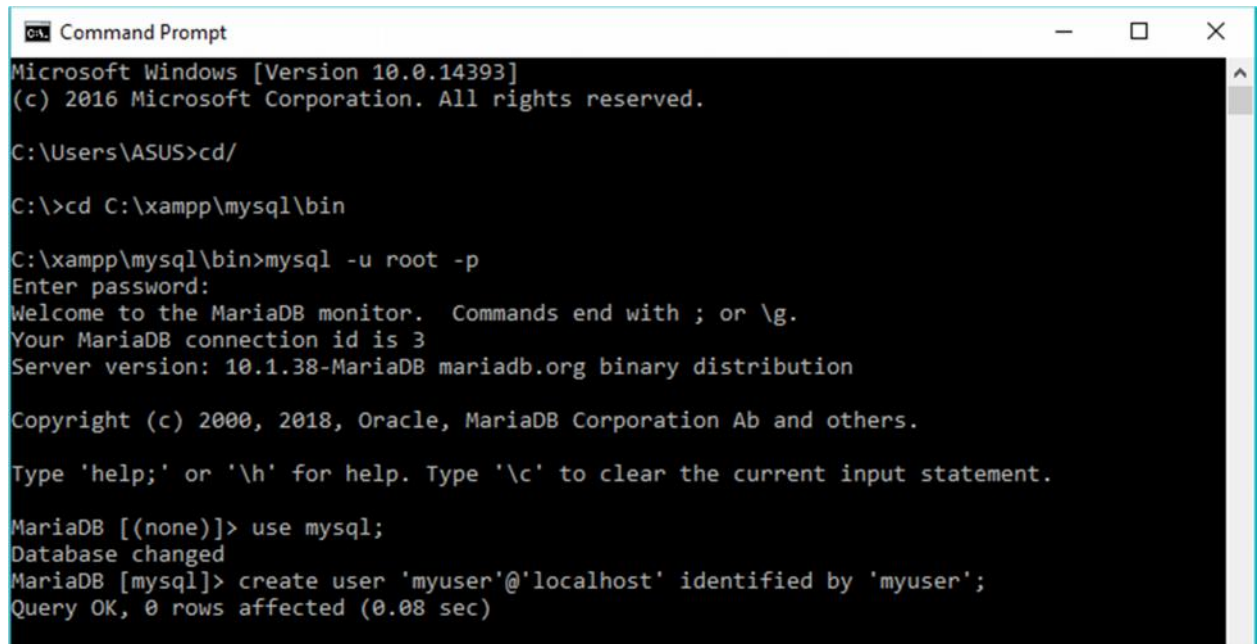
18 rows in set (0.00 sec)

Modul 9

Data Control Language

D. Langkah-langkah Praktikum

- >> membuka command prompt dan login sebagai root ke mySQL
- >> melakukan perintah 'use mysql' hingga muncul 'database changed'
- >> memasukan user baru serta password dengan nama 'myuser'



```
CA. Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd/

C:\>cd C:\xampp\mysql\bin

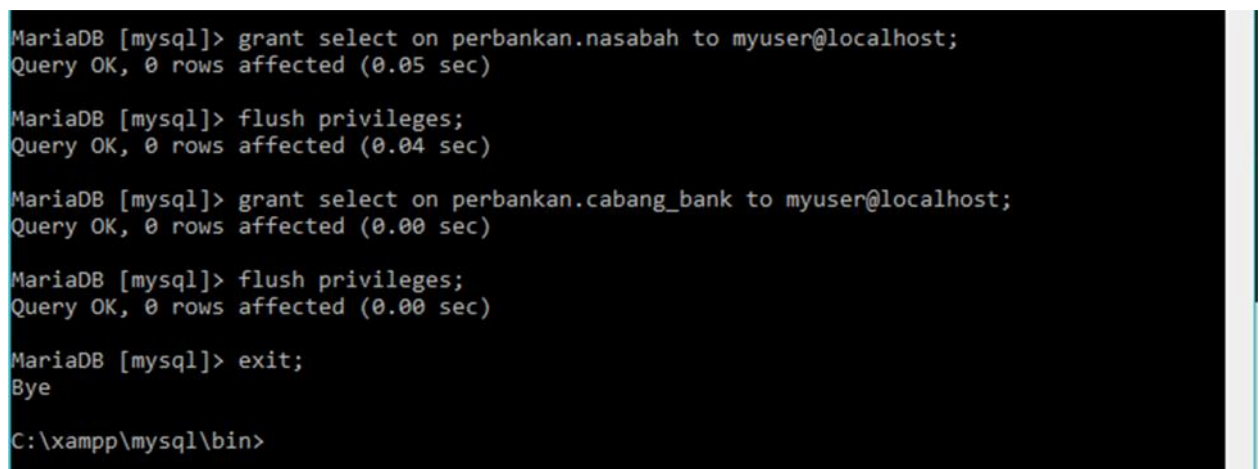
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.38-MariaDB mariadb.org binary distribution

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 'myuser'@'localhost' identified by 'myuser';
Query OK, 0 rows affected (0.08 sec)
```

- >>memberi hak akses pada user 'myuser'
- >> keluar sebagai root dengan syntax 'exit'



```
MariaDB [mysql]> grant select on perbankan.nasabah to myuser@localhost;
Query OK, 0 rows affected (0.05 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.04 sec)

MariaDB [mysql]> grant select on perbankan.cabang_bank to myuser@localhost;
Query OK, 0 rows affected (0.00 sec)

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

MariaDB [mysql]> exit;
Bye

C:\xampp\mysql\bin>
```

- >>login sebagai 'myuser'
- >>mengubah database dengan syntax 'use perbankan' hingga muncul 'database changed'

>>mencoba melakukan select pada table nasabah >>muncul hasil karena sudah ada grant select sebelumnya

```
Command Prompt - mysql -u myuser -p
C:\xampp\mysql\bin>mysql -u myuser -p
Enter password:
ERROR 1045 (28000): Access denied for user 'myuser'@'localhost' (using password: NO)

C:\xampp\mysql\bin>mysql -u myuser -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)]> use perbankan;
Database changed
MariaDB [perbankan]> select * from nasabah;
```

id_nasabah	nama_nasabah	alamat_nasabah
1	Sutopo	Jl. Jend Sudirman 12
2	Maryati	Jl. M.T Haryanto 31
3	Parman	Jl. Hassanudin 81
4	Kartika Padmasari	Jl. Manggis 15
5	Budi Eko Prayoga	Jl. Kantil 30
6	Satria Eka Jaya	Jl. Slamt Riyadi 45
8	Sari Murti	Jl. Pangandaran 1
9	Canka Lokananta	Jl. Tidar 86
10	Budi Murtono	Jl. Merak 22
11	Ahmad	Jl. Jend Sudirman 20
12	Sri Maryati	Jl. M.T Haryanto 11
13	Supardi	Jl. Hassanudin 31
14	Riska	Jl. Manggis 5
15	Andi Setiawan	Jl. Kantil 10
16	Eko Atmaja	Jl. Slamt Riyadi 5
17	Listiani	Jl. Sutoyo 25
18	Yuliana	Jl. Pangandaran 31
19	Candra Aji	Jl. Tidar 6
20	Asriana	Jl. Merak 2

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

>>menghilangkan salah satu privileges untuk user myuser dengan 'revoke'

```
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 6
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)]> revoke select on perbankan.nasabah from myuser@localhost;
Query OK, 0 rows affected (0.29 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit;
Bye
```

>> masuk sebagai myuser

>> mengubah database perbankan dengan 'use perbankan' hingga muncul 'database changed'

>> melakukan perintah select pada table nasabah >> disini hasilnya error karena sebelumnya belum member hak akses select pada table nasabah

```
C:\xampp\mysql\bin>mysql -u myuser -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 10.1.38-MariaDB mariadb.org binary distribution

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

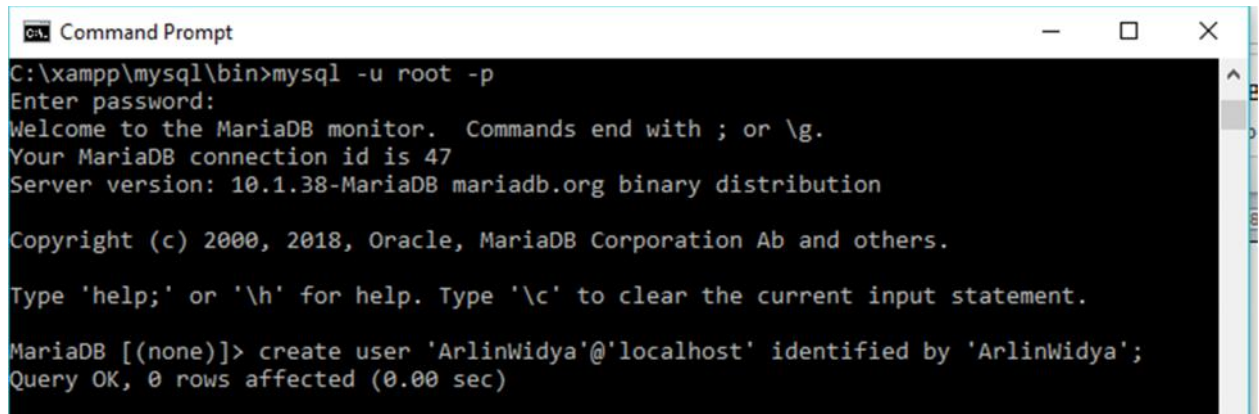
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

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

MariaDB [perbankan]>
```

E. Tugas

>>Membuat user baru dengan nama 'ArlinWidya'



```
Command Prompt
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 47
Server version: 10.1.38-MariaDB mariadb.org binary distribution

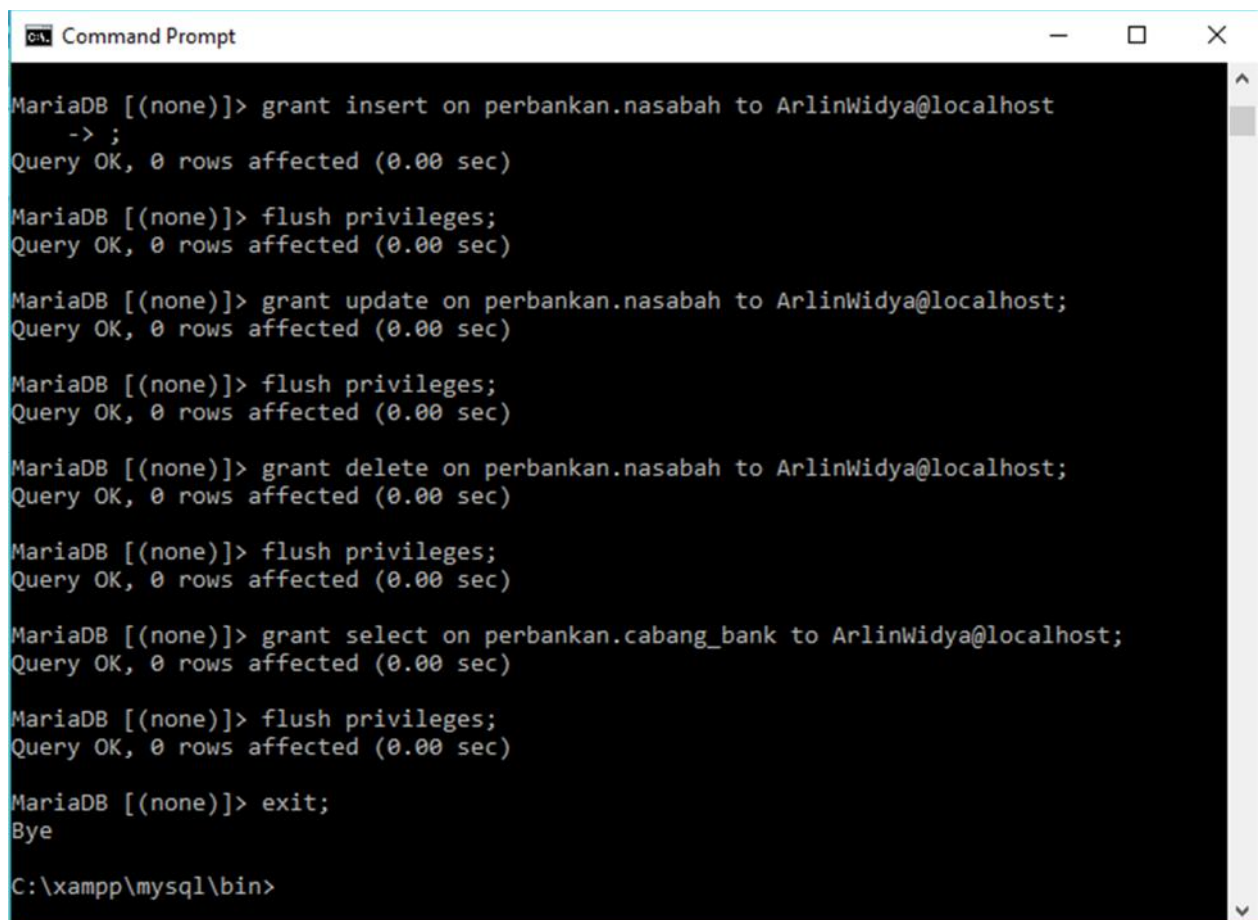
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create user 'ArlinWidya'@'localhost' identified by 'ArlinWidya';
Query OK, 0 rows affected (0.00 sec)
```

>>Memberi privileges dengan insert, update dan delete pada table nasabah

>>Melakukan select pada table cabang_bank >>> pada saat melakukan select hasilnya error karena sebelumnya tidak membuat grant select pada cabang_bank.



```
Command Prompt

MariaDB [(none)]> grant insert on perbankan.nasabah to ArlinWidya@localhost
-> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant update on perbankan.nasabah to ArlinWidya@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant delete on perbankan.nasabah to ArlinWidya@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant select on perbankan.cabang_bank to ArlinWidya@localhost;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit;
Bye

C:\xampp\mysql\bin>
```

-) Perintah select pada table nasabah >>> hasilnya error karena tidak membuat grant dengan hak akses select
-) Perintah insert pada table nasabah >>> hasilnya tidak error/memiliki hasil karena sebelumnya sudah membuat grant insert
-) Perintah insert pada table cabang_bank >>> hasilnya error karena tidak membuat grant select pada cabang_bank

```
C:\xampp\mysql\bin>mysql -u ArlinWidya -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 50
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 perbankan;
Database changed
MariaDB [perbankan]> select * from nasabah;
ERROR 1142 (42000): SELECT command denied to user 'ArlinWidya'@'localhost' for table 'nasabah'
MariaDB [perbankan]> insert into nasabah (id_nasabah, nama_nasabah, alamat_nasabah) values (41, 'Lisa', 'Solo');
Query OK, 1 row affected (0.08 sec)

MariaDB [perbankan]> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values ("BRUJS", "Bank Rut Unit Jakarta Selatan", "Jl. Merak 23");
ERROR 1142 (42000): INSERT command denied to user 'ArlinWidya'@'localhost' for table 'cabang_bank'
MariaDB [perbankan]>
```

>>>Analisa percobaan : 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 error. Revoke digunakan untuk menghapus batasan hak 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 mahasiswa.nim not in (select distinct 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 (0.02 sec)
```

```
MariaDB [perkuliahan]> select * from mahasiswa where mahasiswa.nim not in (select distinct nim from link_mahasiswa_matkul where kode_matkul = 'TIF002');
+-----+-----+-----+-----+
| nim      | nama                | alamat    | tgl_lahir |
+-----+-----+-----+-----+
| L200150125 | Indra Bayu Candra Gupta | Wonogiri  | 1993-08-28 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [perkuliahan]> select * from mahasiswa where mahasiswa.nim not in (select distinct nim from link_mahasiswa_matkul where kode_matkul = 'TIF003');
+-----+-----+-----+-----+
| 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)
```

```
MariaDB [perkuliahan]> select * from mahasiswa where mahasiswa.nim not in (select distinct nim from link_mahasiswa_matkul where kode_matkul = 'TIF005');
+-----+-----+-----+-----+
| nim      | nama                | alamat    | tgl_lahir |
+-----+-----+-----+-----+
| L200150118 | Rina Kurniasari      | Wonogiri  | 1997-11-11 |
+-----+-----+-----+-----+
1 row 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 [perkuliahan1] > 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='D001');
+-----+-----+
| nim      | nama      |
+-----+-----+
| L200150083 | Diyan Bima Saputra |
| L200150088 | Khofa Prayoga      |
| L200150099 | Purwantinah        |
| L200150112 | Dwi Pratiwi Putri  |
| L200150113 | Ismi Kamelia Najib Putri |
| L200150117 | Widiyarti Endang Saputri |
| L200150123 | Danindya Puput Muliana Putri |
| L200150125 | Indra Bayu Candra Gupta |
| L200150127 | Lyon               |
| L200150128 | Sulthana Dzakira Drajat |
| L200150129 | Fendy              |
| L200150144 | Sam'an Alghozy     |
| L200150146 | Fakhrur Razi       |
| L200154001 | Khilyatin Ulin Fitri |
+-----+-----+
14 rows in set (0.00 sec)

MariaDB [perkuliahan1] > 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 matakuliah tersebut dipindahkan untuk mengambil mata kuliah lain (B). Lakukan update data menggunakan sub query!

```

ca. Command Prompt - mysql -u root -p
MariaDB [perkuliahan1]> update link_mahasiswa_matkul set kode_matkul = 'TIF002' w
here 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 [perkuliahan1]> 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    |

```

```

c:\ Command Prompt - mysql -u root -p
+-----+-----+-----+
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)

```


Modul 11

MySQL dan Python

E. Tugas

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

➤ 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)

➤ 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)

➤ 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 python 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()

```

```

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 bulan 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()

```

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

>>>

Modul 12

Aplikasi Dengan Database

C. Tugas

1. Membuat sebuah aplikasi (berbasis Python atau berbasis Java) pada salah satu diantara studi kasus berikut:

Perhotelan, dengan minimal entitas yang ada, antara lain:

-) Tamu
-) Kamar
-) Transaksi

2. Pada aplikasi yang dipilih, buatlah penjelasan studi kasus serta ER Diagramnya!

) Entitas tamu, memiliki attribute:

- o id_tamu
- o nama_tamu
- o alamat_tamu
- o no_hp

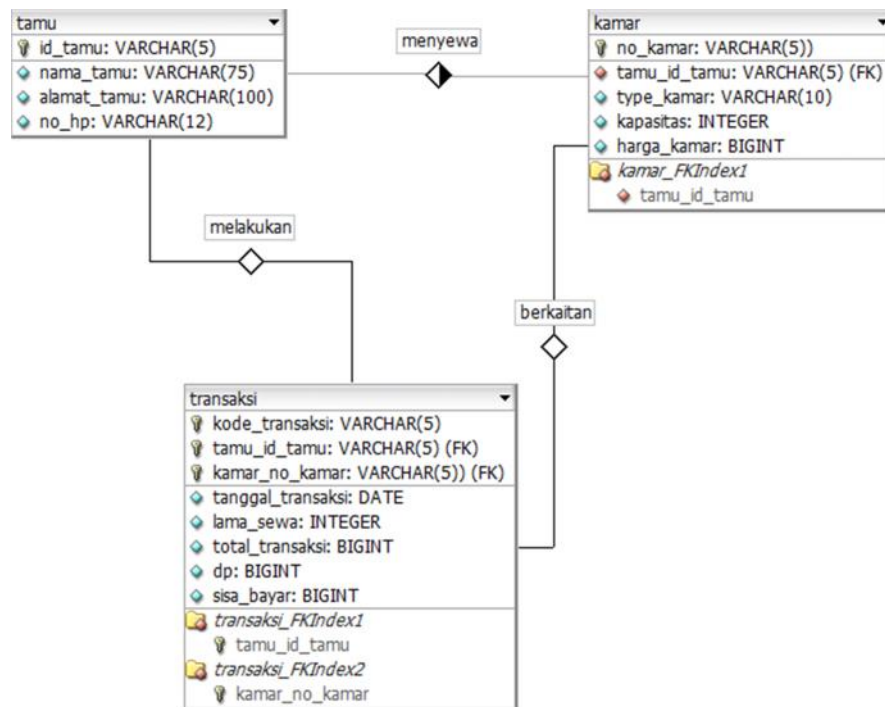
) Entitas kamar, memiliki attribute:

- o no_kamar
- o type_kamar
- o kapasitas_kamar
- o harga_kamar

) Entitas transaksi, memiliki attribute:

- o kode_transaksi
- o tanggal_transaksi
- o lama_sewa
- o total_transaksi
- o dp
- o sisa_bayar

ER Diagram:



3. Terapkan pada aplikasi yang dipilih, beberapa fungsi berikut untuk semua entitas:

TAMU

ID Tamu

Nama

Alamat

No HP

TAMBAH DATA **SIMPAN** **EDIT** **BATAL** **HAPUS** **KELUAR**

No.	ID Tamu	Nama	Alamat	No. HP
1	0001	Wahyu	Solo	089773666111
2	0002	Vely	Jakarta	085677344123
3	0003	Yusi	Bogor	081322566777
4	0004	Rahmad	Surabaya	085464123456
5	0005	Yulia	Jakarta	081322133222
6	0006	Arlin	Solo	089665777111

- a. Proses view (select):


```

private void txtid_tamuKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    idtamu=txtid_tamu.getText();
    int tekanenter=evt.getKeyCode();
    if (tekanenter==10){
        try {
            sql= "select * from tamu where id_tamu='"+ idtamu +"'";
            st= con.createStatement();
            rstamu= st.executeQuery(sql);
            while (rstamu.next()){
                txtnama_tamu.setText(rstamu.getString("nama_tamu"));
                txtalamat_tamu.setText(rstamu.getString("alamat_tamu"));
                txtno_hp.setText(rstamu.getString("no_hp"));
                btn_tambah.setEnabled(false);
                btn_simpan.setEnabled(false);
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Data Tidak Ketemu" + e.getMessage());
            txtnama_tamu.requestFocus();
        }
    }
}

```

b. Insert (memasukkan data):

```

private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    idtamu=String.valueOf(txtid_tamu.getText());
    namatamu=String.valueOf(txtnama_tamu.getText());
    alamattamu=String.valueOf(txtalamat_tamu.getText());
    nohp=String.valueOf(txtno_hp.getText());
    try {
        sql= "insert into tamu (id_tamu, nama_tamu, alamat_tamu, no_hp)"
            + "values"
            + "(" + idtamu + ", " + namatamu + ", " + alamattamu + ", " + nohp + ")";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
        kosongkan_form();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data Gagal Disimpan \n"+e.getMessage());
    }
}

```

c. Update (mengubah data)

```

private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    idtamu=String.valueOf(txtid_tamu.getText());
    namatamu=String.valueOf(txtnama_tamu.getText());
    alamattamu=String.valueOf(txtalamat_tamu.getText());
    nohp=String.valueOf(txtno_hp.getText());
    try {
        sql= "update tamu set id_tamu='"+ idtamu +"', nama_tamu='"+ namatamu +"', alamat_tamu='"+ alamattamu +"',
            + " no_hp='"+ nohp +"' where id_tamu='"+txtid_tamu.getText()+"'";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Diubah");
        kosongkan_form();
    } catch (HeadlessException | SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

d. Delete (menghapus data)

```
private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    idtamu=String.valueOf(txtid_tamu.getText());
    namatamu=String.valueOf(txtnama_tamu.getText());
    alamattamu=String.valueOf(txtalamat_tamu.getText());
    nohp=String.valueOf(txtno_hp.getText());
    try {
        sql = "delete from tamu where id_tamu = '"+idtamu+"'";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");
    } catch (HeadlessException | SQLException e) {
        JOptionPane.showMessageDialog(null, "Data Gagal Dihapus \n" + e.getMessage());
    }
}
```

TRANSAKSI

Kode Transaksi	<input type="text"/>	ID Tamu	<input type="text" value="Pilih"/>
Tanggal Sewa	<input type="text"/>	Nama	<input type="text"/>
Lama Inap	<input type="text"/>	No Kamar	<input type="text" value="Pilih"/>
Total Transaksi	<input type="text"/>	Type Kamar	<input type="text"/>
DP	<input type="text"/>	Harga Kamar	<input type="text"/>
Sisa Bayar	<input type="text"/>		

No.	Kode Tra...	Tanggal ...	Lama Inap	Total Tra...	DP	Sisa Bayar	ID Tamu	No Kamar
1	T0001	2019-06-...	3	750000	300000	450000	0002	14
2	T0002	2019-06-...	3	3000000	850000	2150000	0004	45
3	T0003	2019-06-...	3	1200000	300000	900000	0003	22

a. Proses View (Select)


```

private void cmbno_kamarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        String Sql = "select * from kamar where no_kamar='" + cmbno_kamar.getSelectedItem()+"'";
        st=con.createStatement();
        rskamar = st.executeQuery(Sql);
        while(rskamar.next()){
            txttype_kamar.setText(rskamar.getString("type_kamar"));
            txtharga_kamar.setText(rskamar.getString("harga_kamar"));
        }
    } catch (Exception e) {
    }
}

```

b. Insert (memasukkan data)

```

private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    kodetrans=String.valueOf(txtkode_trans.getText());
    tanggal=String.valueOf(datetanggal.getText());
    idtamu=cmbid_tamu.getItemAt(cmbid_tamu.getSelectedIndex()).toString();
    nomerkamar=cmbno_kamar.getItemAt(cmbno_kamar.getSelectedIndex()).toString();
    lamainap=Integer.parseInt(txtlama_inap.getText());
    total=Integer.parseInt(txttotal.getText());
    dp=Integer.parseInt(txtdp.getText());
    sisabayar=Integer.parseInt(txtsisabayar.getText());
    try {
        sql= "insert into transaksi (kode_transaksi, tanggal_transaksi, lama_sewa, "
            + "total_transaksi, dp, sisa_bayar, id_tamu, no_kamar)"
            + "values ("
            + "'" + kodetrans + "', '" + tanggal + "', '" + lamainap + "', '" + total + "', '" + dp + "', '"
            + sisabayar + "', '" + idtamu + "', '" + nomerkamar + "')";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Disimpan");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data Gagal Disimpan \n"+e.getMessage());
    }
}

```

c. Update (mengubah data)

```

private void btn_editActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    kodetrans=String.valueOf(txtkode_trans.getText());
    tanggal=String.valueOf(datetanggal.getText());
    idtamu=cmbid_tamu.getItemAt(cmbid_tamu.getSelectedIndex()).toString();
    nomerkamar=cmbno_kamar.getItemAt(cmbno_kamar.getSelectedIndex()).toString();
    lamainap=Integer.parseInt(txtlama_inap.getText());
    total=Integer.parseInt(txttotal.getText());
    dp=Integer.parseInt(txtdp.getText());
    sisabayar=Integer.parseInt(txtsisabayar.getText());
    try {
        sql = "update transaksi set kode_transaksi ='" + kodetrans + "', tanggal_transaksi ='" + tanggal
            + "', lama_sewa ='" + lamainap + "', total_transaksi ='" + total + "', dp ='" + dp
            + "', sisa_bayar ='" + sisabayar + "', id_tamu ='" + idtamu + "', no_kamar ='" + nomerkamar
            + "' where kode_transaksi ='" + txtkode_trans.getText()+"'";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Diubah");
        kosongkan_form();
    } catch (HeadlessException | SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

d. Delete (menghapus data)

```

private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    kodetrans=String.valueOf(txtkode_trans.getText());
    tanggal=String.valueOf(datetanggal.getText());
    idtamu=cmbid_tamu.getItemAt(cmbid_tamu.getSelectedIndex()).toString();
    nomerkamar=cmbno_kamar.getItemAt(cmbno_kamar.getSelectedIndex()).toString();
    lamainap=Integer.parseInt(txtlama_inap.getText());
    total=Integer.parseInt(txttotal.getText());
    dp=Integer.parseInt(txtdp.getText());
    sisabayar=Integer.parseInt(txtsisabayar.getText());
    try {
        sql = "delete from transaksi where kode_transaksi = '"+kodetrans+"'";
        st=con.createStatement();
        st.execute(sql);
        tampildata();
        JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data Gagal Dihapus \n" + e.getMessage());
    }
}

```