

REKAP TUGAS
PRAKTIKUM SISTEM BASIS DATA



Disusun Oleh :

Nama : Angieta Putri W

NIM : L200170096

Kelas : D

PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

MODUL 1

1. Jelaskan mengapa dibutuhkan data!

Data diperlukan dalam segala hal, baik berupa pengukuran, pencatatan, pengumpulan informasi, maupun pengambilan keputusan semuanya memerlukan data. Dengan kata lain data sangat dibutuhkan karena informasi yang ada akan memberikan arti yang sangat penting baik untuk saat ini maupun untuk akan datang. Sehingga definisi dari data adalah informasi yang mengandung arti.

2. Jelaskan manfaat database dan contohnya!

- ✓ 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.

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

- Pendeskripsian 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)

- Database: Sekumpulan data yang saling berhubungan untuk mencapai suatu tujuan.
- Tabel: Merupakan hal yang paling mendasar dalam hal penyimpanan data yang 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.

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

pengolahan data secara manual segala bentuk pengolahan juga dilakukan secara manual (disusun, dihitung atau dibuat laporannya secara manual). Cara ini tentu saja membutuhkan ekstra tenaga dan waktu. Dan lebih sering lagi, diperlukan pengumpulan data-data yang sejenis secara berkali-kali dan dilakukan juga pengolahan dan pembuatan laporan secara berkali-kali pula. 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?

Tujuan utama DBMS adalah untuk menyediakan tinjauan abstrak dari data bagi user. Jadi sistem menyembunyikan informasi mengenai bagaimana data disimpan dan dirawat, tetapi data tetap dapat diambil dengan efisien. Pertimbangan efisien yang digunakan adalah bagaimana merancang struktur data yang kompleks, tetapi tetap dapat digunakan oleh pengguna yang masih awam, tanpa mengetahui kompleksitas struktur data. Basis data menjadi penting karena munculnya beberapa masalah bila tidak menggunakan data yang terpusat, seperti adanya duplikasi data, hubungan antar data tidak jelas, organisasi data dan update menjadi rumit.

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

Pada tabel Mhs, ada 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 diisi 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 diisi data berupa angka dan simbol.

MODUL 2

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 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!

Jawaban

Langkah-langkah perancangan database perkuliahan:

- a) Menentukan entitas(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 mata kuliah
 - Data ruang kelas : menyimpan informasi tentang data ruang kelas.
- b) Menentukan atribut (sifat sifat) masing masing entity sesuai kebutuhan database:
 - ❖ Mahasiswa
 - NIM_Mahasiswa : Nomor id untuk mahasiswa (integer)PK
 - Nama_Mahasiswa : Nama lengkap Mahasiswa (varchar(45))
 - Alamat_Mahasiswa : Alamat lengkap Mahasiswa (varchar(225))
 - ❖ Dosen
 - NIK_Dosen : Nomer identitas untuk Dosen (Integer)PK
 - Nama_Dosen : Nama lengkap Dosen (varchar(45))
 - Alamat_Dosen : Alamat lengkap Dosen (varchar(225))
 - ❖ Mata_Kuliah
 - Kode_MK : kode untuk matakuliah (varchar(10))PK
 - Nama_MK : nama matakuliah (varchar(20))PK
 - ❖ Ruang_Kelas
 - Kode_RK : Kode ruang kelas (varchar(10))PK
 - Nama_RK : Nama ruang kelas (varchar(20))PK

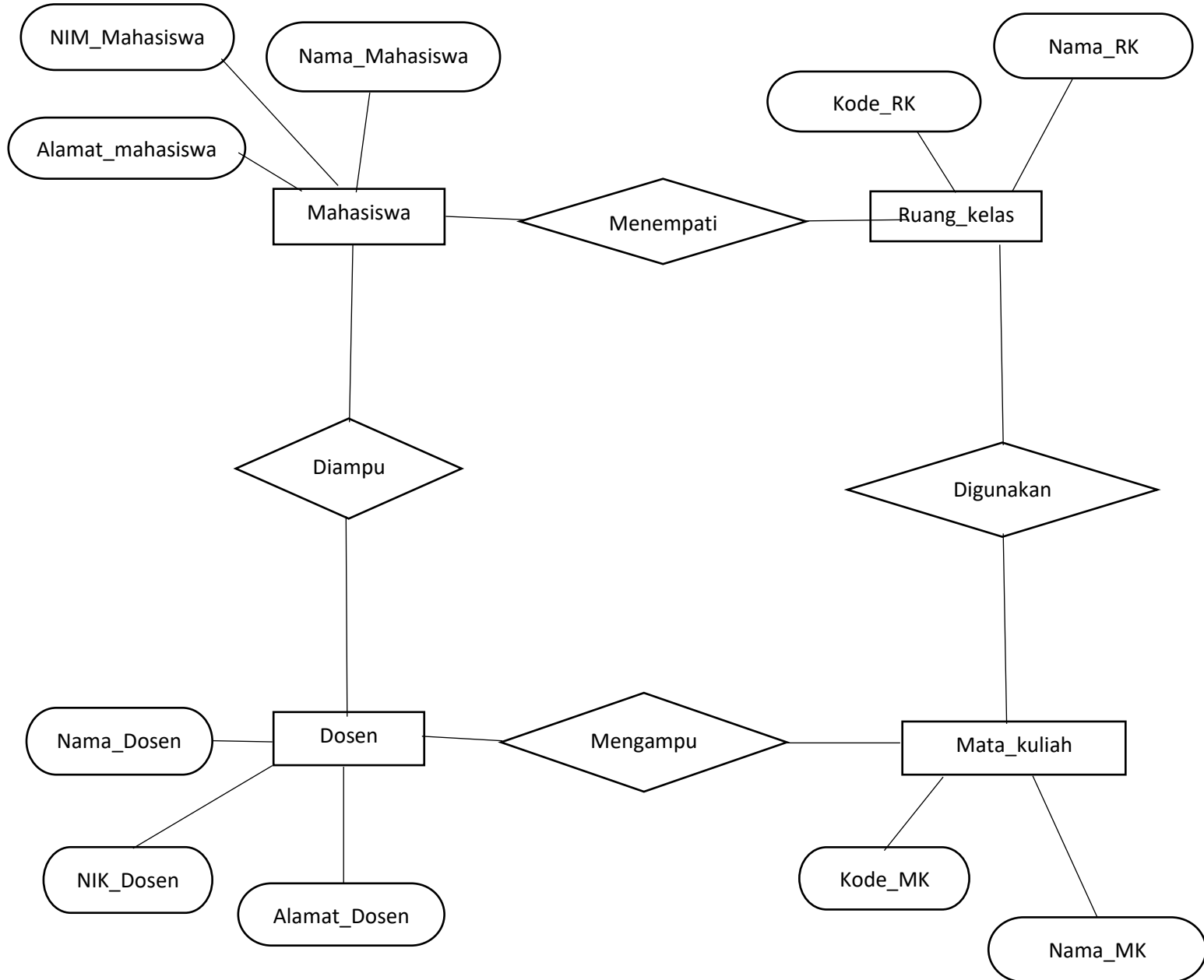
c) Menentukan relationship (hubungan) antar entitas

	Mahasiswa	Dosen	Mata_Kuliah	Ruang_Kelas
Mahasiswa	-	m:n	-	m:n
Dosen		-	m:n	-
Mata_Kuliah			-	m:n
Ruang_Kelas				-

Hubungan

- ❖ Mahasiswa diampu dosen.
 - Tabel utama : Mahasiswa, Dosen
 - Tabel kedua : Mahasiswa_has_Dosen
 - Relationship : many-to-many (m:n)
 - Atribut penghubung : NIM_Mahasiswa, NIK_Dosen
- ❖ Dosen Mengampu Mata_Kuliah
 - Tabel utama : dosen, mata_kuliah
 - Tabel kedua : dosen_MK
 - Relationship : many-to-many (m:n)
 - Atribut penghubung : NIK_Dosen, Kode_MK
- ❖ Mahasiswa menempati ruang_kelas
 - Tabel utama : mahasiswa, ruang_kelas
 - Tabel kedua : mahasiswa_has_RK
 - Relationship : many-to-many (m:n)
 - Atribut penghubung : NIM_Mahasiswa, Kode_RK
- ❖ Ruang_kelas digunakan Mata_kuliah
 - Tabel utama : ruang_kelas , mata_kuliah
 - Tabel kedua : RK_has_MK
 - Relationship : many-to-many (m:n)
 - Atribut penghubung : kode_RK, Kode_MK

d) 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 mengandung 4 buah entitas.

Jawaban

Langkah-langkah perancangan database :

- 1) Menentukan entities
 - penumpang : menyimpan semua data pribadi penumpang
 - kasir : menyimpan semua data pribadi kasir
 - pesawat : menyimpan informasi mengenai pesawat
 - tiket_pesawat : menyimpan informasi mengenai tiket pesawat
- 2) Menentukan attributes
 - a. penumpang
 - id_penumpang : nomor id penumpang (integer) PK
 - nama_penumpang : nama lengkap penumpang (varchar(45))
 - alamat_penumpang : alamat lengkap penumpang (varchar(255))
 - b. kasir
 - id_kasir : nomor id kasir (integer) PK
 - nama_kasir : nama lengkap kasir (varchar(45))
 - alamat_kasir : alamat lengkap kasir (varchar(255))
 - c. pesawat
 - kode_pesawat : kode pesawat (integer) PK
 - nama_pesawat : nama pesawat (varchar(45))
 - d. tiket_pesawat
 - kode_tiket : kode tiket pesawat (varchar(20)) PK
 - jumlah_tiket : jumlah tiket pesawat (varchar(20))
 - nomor_duduk : kapasitas setiap ruang kelas (varchar(20))
- 3) Menentukan relationship (hubungan) antar entitas

	penumpang	kasir	pesawat	tiket_pesawat
penumpang	-	m:n	m:n	m:n
kasir		-	-	-
pesawat			-	m:n
Tiket_pesawat				-

Hubungan

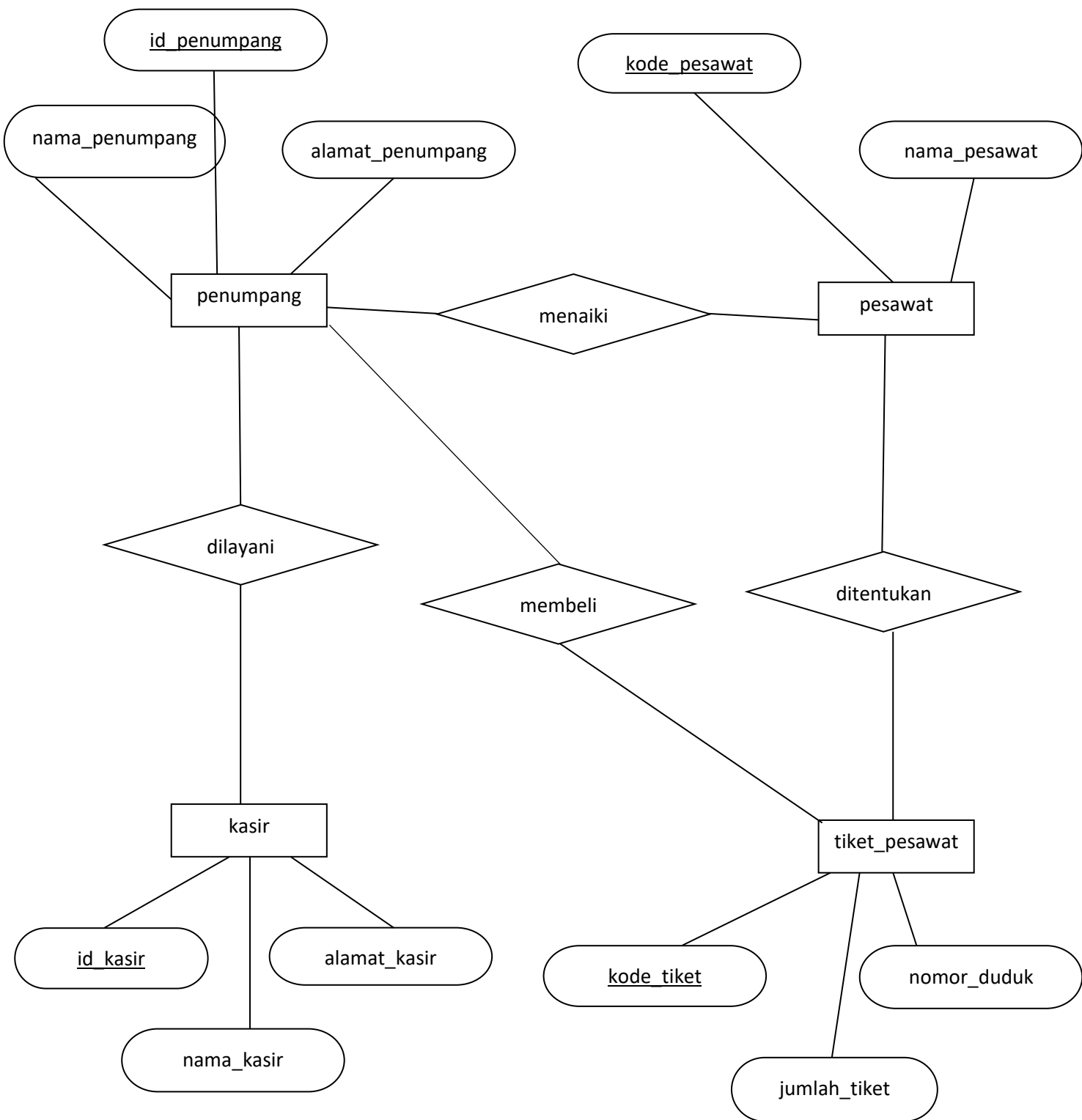
- ❖ kasir melayani penumpang:
 - Tabel utama: kasir, penumpang
 - Tabel kedua: kasir_has_penumpang
 - Relationship: Many-to-many(m:n)
 - Attribute penghubung: id_penumpang, id_kasir

- ❖ penumpang menaiki pesawat:
 - Tabel utama: penumpang, pesawat
 - Tabel kedua: penumpang_has_pesawat
 - Relationship: Many-to-many(m:n)
 - Attribute penghubung: id_penumpang, kode_pesawat

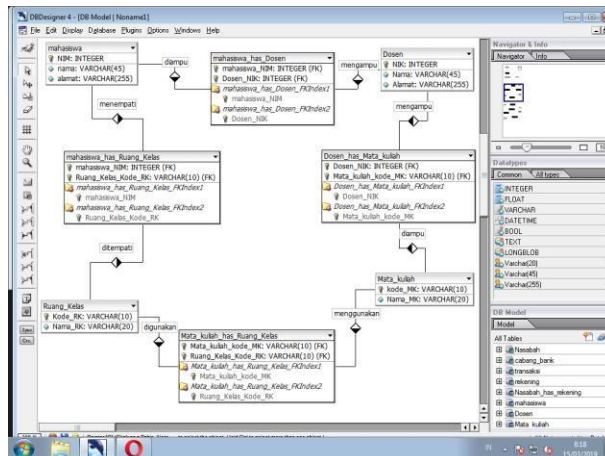
- ❖ tiket_pesawat dibeli penumpang:
 - Tabel utama: tiket_pesawat, penumpang
 - Tabel kedua: penumpang_has_tiket
 - Relationship: Many-to-many(m:n)
 - Attribute penghubung: kode_tiket,

- ❖ pesawat ditentukan tiket_pesawat :
 - Tabel utama: pesawat, tiket_pesawat
 - Tabel kedua: pesawat_has_tiket
 - Relationship: Many-to-many(m:n)
 - Attribute penghubung: kode_pesawat, kode_tiket

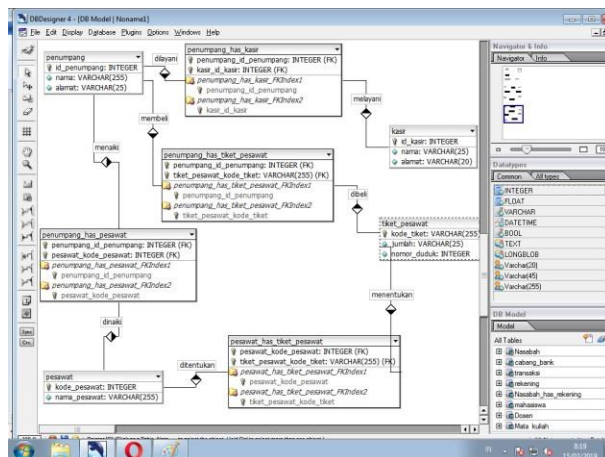
1. Menggambar ER-Diagram



❖ Database Universitas



❖ Database Pesawat



MODUL 4

DATABASE UNIVERSITAS

1. Membuat database baru dengan nama Universitas.

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

2. Menghubungkan ke dalam database yang telah dibuat.

```
MariaDB [(none)]> use universitas;  
Database changed
```

3. Membuat tabel Mahasiswa.

```
MariaDB [universitas]> CREATE TABLE Mahasiswa(  
  -> NIM_Mahasiswa INTEGER PRIMARY KEY,  
  -> Nama_Mahasiswa VARCHAR(45)NOT NULL,  
  -> Alamat_Mahasiswa VARCHAR(255)NOT NULL  
  -> );  
Query OK, 0 rows affected (0.47 sec)
```

4. Membuat tabel Dosen.

```
MariaDB [universitas]> CREATE TABLE Dosen(  
  -> NIK_Dosen INTEGER PRIMARY KEY,  
  -> Nama_Dosen VARCHAR(45)NOT NULL,  
  -> Alamat_Dosen VARCHAR(255)NOT NULL  
  -> );  
Query OK, 0 rows affected (0.45 sec)
```

5. Membuat tabel Mata_Kuliah.

```
MariaDB [universitas]> CREATE TABLE Mata_Kuliah(  
  -> Kode_MK VARCHAR(10) PRIMARY KEY,  
  -> Nama_MK VARCHAR(20) NOT NULL  
  -> );  
Query OK, 0 rows affected (0.21 sec)
```

6. Membuat tabel Ruang_Kelas.

```
MariaDB [universitas]> CREATE TABLE Ruang_Kelas(  
  -> Kode_RK VARCHAR(10) PRIMARY KEY,  
  -> Nama_RK VARCHAR(10) NOT NULL  
  -> );  
Query OK, 0 rows affected (0.85 sec)
```

7. Membuat tabel Mahasiswa_has_Dosen.

```
MariaDB [universitas]> CREATE TABLE Mahasiswa_has_Dosen(  
  -> NIM_MahasiswaFK INTEGER REFERENCES Mahasiswa(NIM_Mahasiswa)  
  -> ON DELETE CASCADE ON UPDATE CASCADE,  
  -> NIK_DosenFK INTEGER REFERENCES Dosen(NIK_Dosen)  
  -> ON DELETE CASCADE ON UPDATE CASCADE,  
  -> PRIMARY KEY(NIM_MahasiswaFK, NIK_DosenFK)  
  -> );  
Query OK, 0 rows affected (0.29 sec)
```

8. Membuat tabel Dosen_has_MK.

```
MariaDB [universitas]> CREATE TABLE Dosen_has_MK(
  -> NIK_DosenFK INTEGER REFERENCES Dosen(NIK_Dosen)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> Kode_MKFK INTEGER REFERENCES MK(Kode_MK)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> PRIMARY KEY(NIK_DosenFK, Kode_MKFK)
  -> );
Query OK, 0 rows affected (0.23 sec)
```

9. Membuat tabel Mahasiswa_has_RK.

```
MariaDB [universitas]> CREATE TABLE Mahasiswa_has_RK(
  -> NIM_MahasiswaFK INTEGER REFERENCES Mahasiswa(NIM_Mahasiswa)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> Kode_RKFK INTEGER REFERENCES RK(Kode_RK)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> PRIMARY KEY(NIM_MahasiswaFK, Kode_RKFK)
  -> );
Query OK, 0 rows affected (0.19 sec)
```

10. Membuat tabel RK_has_MK.

```
MariaDB [universitas]> CREATE TABLE RK_has_MK(
  -> Kode_RKFK INTEGER REFERENCES RK(Kode_RK)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> Kode_MKFK INTEGER REFERENCES MK(Kode_MK)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> PRIMARY KEY(Kode_RKFK, Kode_MKFK)
  -> );
Query OK, 0 rows affected (0.19 sec)
```

11. Mengecek hasil Pembuatan Database.

```
MariaDB [universitas]> show tables;
+-----+
| Tables_in_universitas |
+-----+
| dosen                  |
| dosen_has_mk           |
| mahasiswa              |
| mahasiswa_has_dosen    |
| mahasiswa_has_rk       |
| mata_kuliah            |
| rk_has_mk              |
| ruang_kelas            |
+-----+
8 rows in set (0.00 sec)
```

12. Melihat Struktur tabel Mahasiswa.

```
MariaDB [universitas]> describe Mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NIM_Mahasiswa  | int(11)       | NO   | PRI | NULL    |       |
| Nama_Mahasiswa | varchar(45)   | NO   |     | NULL    |       |
| Alamat_Mahasiswa | varchar(255) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

13. Melihat Struktur tabel Dosen.

```
MariaDB [universitas]> describe Dosen;
```

Field	Type	Null	Key	Default	Extra
NIK_Dosen	int(11)	NO	PRI	NULL	
Nama_Dosen	varchar(45)	NO		NULL	
Alamat_Dosen	varchar(255)	NO		NULL	

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

14. Melihat Struktur data Mata_Kuliah.

```
MariaDB [universitas]> describe Mata_Kuliah;
```

Field	Type	Null	Key	Default	Extra
Kode_MK	varchar(10)	NO	PRI	NULL	
Nama_MK	varchar(20)	NO		NULL	

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

15. Melihat Struktur data Ruang_Kelas.

```
MariaDB [universitas]> describe Ruang_Kelas;
```

Field	Type	Null	Key	Default	Extra
Kode_RK	varchar(10)	NO	PRI	NULL	
Nama_RK	varchar(10)	NO		NULL	

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

16. Melihat Struktur data Mahasiswa_has_Dosen.

```
MariaDB [universitas]> describe Mahasiswa_has_Dosen;
```

Field	Type	Null	Key	Default	Extra
NIM_MahasiswaFK	int(11)	NO	PRI	NULL	
NIK_DosenFK	int(11)	NO	PRI	NULL	

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

17. Melihat Struktur data Dosen_has_MK.

```
MariaDB [universitas]> describe Dosen_has_MK;
```

Field	Type	Null	Key	Default	Extra
NIK_DosenFK	int(11)	NO	PRI	NULL	
Kode_MKFK	int(11)	NO	PRI	NULL	

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

18. Melihat Struktur data Mahasiswa_has_RK.

```
MariaDB [universitas]> describe Mahasiswa_has_RK;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NIM_MahasiswaFK | int(11) | NO    | PRI  | NULL    |       |
| Kode_RKFK       | int(11) | NO    | PRI  | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

19. Melihat Struktur data RK_has_MK.

```
MariaDB [universitas]> describe RK_has_MK;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Kode_RKFK       | int(11) | NO    | PRI  | NULL    |       |
| Kode_MKFK       | int(11) | NO    | PRI  | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

DATABASE BANDARA

1. Membuat database baru dengan nama Bandara.

```
MariaDB [(none)]> create database Bandara;
Query OK, 1 row affected (0.05 sec)
```

2. Menghubungkan ke dalam database yang telah dibuat.

```
MariaDB [(none)]> use Bandara;
Database changed
```

3. Membuat tabel Penumpang.

```
MariaDB [Bandara]> CREATE TABLE Penumpang(
-> id_penumpang INTEGER PRIMARY KEY,
-> nama_penumpang VARCHAR(45) NOT NULL,
-> alamat_penumpang VARCHAR(255) NOT NULL
-> );
Query OK, 0 rows affected (0.27 sec)
```

4. Membuat tabel Kasir.

```
MariaDB [Bandara]> CREATE TABLE Kasir(
-> id_kasir INTEGER PRIMARY KEY,
-> nama_kasir VARCHAR(45) NOT NULL,
-> alamat_kasir VARCHAR(255) NOT NULL
-> );
Query OK, 0 rows affected (0.23 sec)
```

5. Membuat tabel Pesawat.

```
MariaDB [Bandara]> CREATE TABLE Pesawat(
-> kode_pesawat INTEGER PRIMARY KEY,
-> nama_pesawat VARCHAR(45) NOT NULL
-> );
Query OK, 0 rows affected (0.22 sec)
```

6. Membuat tabel Tiket_Pesawat.

```
MariaDB [Bandara]> CREATE TABLE Tiket_Pesawat(  
-> kode_tiket VARCHAR(20) PRIMARY KEY,  
-> jumlah_tiket VARCHAR(20) NOT NULL,  
-> nomor_duduk VARCHAR(20) NOT NULL  
-> );  
Query OK, 0 rows affected (0.24 sec)
```

7. Membuat tabel Kasir_has_Penumpang.

```
MariaDB [Bandara]> CREATE TABLE Kasir_has_Penumpang(  
-> id_kasirFK INTEGER REFERENCES kasir(id_kasir)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> id_penumpangFK INTEGER REFERENCES penumpang(id_penumpang)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> PRIMARY KEY(id_kasirFK, id_penumpangFK)  
-> );  
Query OK, 0 rows affected (0.22 sec)
```

8. Membuat tabel Penumpang_has_Pesawat.

```
MariaDB [Bandara]> CREATE TABLE Penumpang_has_Pesawat(  
-> id_penumpangFK INTEGER REFERENCES penumpang(id_penumpang)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> kode_pesawatFK INTEGER REFERENCES pesawat(kode_pesawat)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> PRIMARY KEY(id_penumpangFK, kode_pesawatFK)  
-> );  
Query OK, 0 rows affected (0.22 sec)
```

9. Membuat tabel Tiket_Pesawat_has_Penumpang.

```
MariaDB [Bandara]> CREATE TABLE Tiket_Pesawat_has_Penumpang(  
-> kode_tiketFK INTEGER REFERENCES tiket_pesawat(kode_tiket)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> id_penumpangFK INTEGER REFERENCES penumpang(id_penumpang)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> PRIMARY KEY(kode_tiketFK, id_penumpangFK)  
-> );  
Query OK, 0 rows affected (0.23 sec)
```

10. Membuat tabel Pesawat_has_Tiket_Pesawat.

```
MariaDB [Bandara]> CREATE TABLE Pesawat_has_Tiket_Pesawat(  
-> kode_pesawatFK INTEGER REFERENCES pesawat(kode_pesawat)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> kode_tiketFK INTEGER REFERENCES tiket_pesawat(kode_tiket)  
-> ON DELETE CASCADE ON UPDATE CASCADE,  
-> PRIMARY KEY(kode_pesawatFK, kode_tiketFK)  
-> );  
Query OK, 0 rows affected (0.28 sec)
```

11. Mengecek hasil Pembuatan Database.

```
MariaDB [Bandara]> show tables;
+-----+
| Tables_in_bandara |
+-----+
| kasir               |
| kasir_has_penumpang |
| penumpang          |
| penumpang_has_pesawat |
| pesawat            |
| pesawat_has_tiket_pesawat |
| tiket_pesawat       |
| tiket_pesawat_has_penumpang |
+-----+
8 rows in set (0.07 sec)
```

12. Melihat Struktur tabel Penumpang.

```
MariaDB [Bandara]> describe Penumpang;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_penumpang   | int(11)       | NO   | PRI | NULL    |       |
| nama_penumpang | varchar(45)   | NO   |     | NULL    |       |
| alamat_penumpang | varchar(255) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.18 sec)
```

13. Melihat Struktur tabel Kasir.

```
MariaDB [Bandara]> describe Kasir;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_kasir       | int(11)       | NO   | PRI | NULL    |       |
| nama_kasir     | varchar(45)   | NO   |     | NULL    |       |
| alamat_kasir   | varchar(255) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

14. Melihat Struktur data Pesawat.

```
MariaDB [Bandara]> describe Pesawat;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_pesawat   | int(11)       | NO   | PRI | NULL    |       |
| nama_pesawat   | varchar(45)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

15. Melihat Struktur data Tiket_Pesawat.

```
MariaDB [Bandara]> describe Tiket_Pesawat;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_tiket     | varchar(20)   | NO   | PRI | NULL    |       |
| jumlah_tiket   | varchar(20)   | NO   |     | NULL    |       |
| nomor_duduk    | varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```


16. Melihat Struktur data Kasir_has_Penumpang.

```
MariaDB [Bandara]> describe Kasir_has_Penumpang;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_kasirFK     | int(11)| NO   | PRI | NULL    |       |
| id_penumpangFK | int(11)| NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

17. Melihat Struktur data Penumpang_has_Pesawat .

```
MariaDB [Bandara]> describe Penumpang_has_Pesawat;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_penumpangFK | int(11)| NO   | PRI | NULL    |       |
| kode_pesawatFK | int(11)| NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

18. Melihat Struktur data Tiket_Pesawat_has_Penumpang.

```
MariaDB [Bandara]> describe Tiket_Pesawat_has_Penumpang;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_tiketFK   | int(11)| NO   | PRI | NULL    |       |
| id_penumpangFK | int(11)| NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

19. Melihat Struktur data Pesawat_has_Tiket_Pesawat.

```
MariaDB [Bandara]> describe Pesawat_has_Tiket_Pesawat;
+-----+-----+-----+-----+-----+-----+
| Field          | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_pesawatFK | int(11)| NO   | PRI | NULL    |       |
| kode_tiketFK   | int(11)| NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

MODUL 5

LATIHAN

1. Tabel nasabah

```
mysql> insert into nasabah(id_nasabah, nama_nasabah, alamat_nasabah) values (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'), (7, 'Indri Hapsari', 'Jl.Sutoyo 5')
-> , (8, 'Sari Murti', 'Jl.Pangandaran 11'), (9, 'Canka Lokananta', 'Jl.Tidar 86'), (10, 'Budi Murtono', 'Jl.Merak 22');
Query OK, 10 rows affected (0.06 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> 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
7	Indri Hapsari	Jl.Sutoyo 5
8	Sari Murti	Jl.Pangandaran 11
9	Canka Lokananta	Jl.Tidar 86
10	Budi Murtono	Jl.Merak 22

10 rows in set (0.00 sec)

2. Tabel cabang_bank

```
mysql> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values ('BRUS', 'Bank Rut Unit Surakarta', 'Jl.Slamet Riyadi 18'), ('BRUM', 'Bank Rut Unit Magelang', 'Jl.P.Tendean 63')
-> , ('BRUB', 'Bank Rut Unit Boyolali', 'Jl.Ahmad Yani 45'), ('BRUK', 'Bank Rut Unit Klaten', 'Jl.Suparman 23'), ('BRUY', 'Bank Rut Unit Yogyakarta', 'Jl.Anggrek 21')
-> , ('BRUW', 'Bank Rut Unit Wonogiri', 'Jl.Untung Suropati 12');
Query OK, 6 rows affected (0.07 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUB	Bank Rut Unit Boyolali	Jl.Ahmad Yani 45
BRUK	Bank Rut Unit Klaten	Jl.Suparman 23
BRUM	Bank Rut Unit Magelang	Jl.P.Tendean 63
BRUS	Bank Rut Unit Surakarta	Jl.Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jl.Untung Suropati 12
BRUY	Bank Rut Unit Yogyakarta	Jl.Anggrek 21

6 rows in set (0.00 sec)

3. Tabel rekening

```
mysql> insert into rekening(no_rekening, kode_cabangFK, pin, saldo) values (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);
Query OK, 13 rows affected (0.09 sec)
Records: 13 Duplicates: 0 Warnings: 0
```

```
mysql> 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

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

4. Tabel nasabah_has_rekening

```
mysql> insert into nasabah_has_rekening(id_nasabahFK, no_rekeningFK) values (1, 104), (2, 103), (3, 105), (3, 106),
-> (4, 101), (4, 107), (5, 102), (5, 107), (6, 109), (7, 109), (8, 111), (9, 110), (10, 113), (8, 112), (10, 108);
Query OK, 15 rows affected (0.07 sec)
Records: 15 Duplicates: 0 Warnings: 0
```

```
mysql> 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

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

5. Tabel transaksi

```
mysql> insert into transaksi(no_transaksi, no_rekeningFK, id_nasabahFK, jenis_transaksi, tanggal, jumlah) values
-> (NULL, 105, 3, 'debit', DEFAULT, 50000), (NULL, 103, 2, 'debit', DEFAULT, 40000), (NULL, 101, 4, 'kredit', DEFAULT, 20000),
-> (NULL, 106, 3, 'debit', DEFAULT, 50000), (NULL, 107, 5, 'kredit', DEFAULT, 30000), (NULL, 104, 1, 'kredit', DEFAULT, 200000),
-> (NULL, 110, 9, 'kredit', DEFAULT, 150000), (NULL, 102, 5, 'debit', DEFAULT, 20000), (NULL, 105, 3, 'kredit', DEFAULT, 50000), (NULL, 107, 4, 'debit', DEFAULT, 100000),
-> (NULL, 103, 2, 'debit', DEFAULT, 100000), (NULL, 104, 1, 'debit', DEFAULT, 50000),
-> (NULL, 107, 4, 'kredit', DEFAULT, 200000), (NULL, 105, 3, 'debit', DEFAULT, 40000), (NULL, 104, 1, 'kredit', DEFAULT, 100000),
-> (NULL, 101, 4, 'kredit', DEFAULT, 20000), (NULL, 103, 2, 'debit', DEFAULT, 50000), (NULL, 102, 5, 'debit', DEFAULT, 50000),
-> (NULL, 108, 10, 'debit', DEFAULT, 100000), (NULL, 106, 3, 'kredit', DEFAULT, 50000), (NULL, 103, 2, 'kredit', DEFAULT, 200000),
-> (NULL, 105, 3, 'kredit', DEFAULT, 100000), (NULL, 102, 5, 'debit', DEFAULT, 20000), (NULL, 104, 1, 'debit', DEFAULT, 50000),
-> (NULL, 103, 2, 'debit', DEFAULT, 40000), (NULL, 101, 4, 'debit', DEFAULT, 50000), (NULL, 103, 2, 'kredit', DEFAULT, 100000),
-> (NULL, 102, 5, 'kredit', DEFAULT, 200000), (NULL, 109, 7, 'debit', DEFAULT, 100000), (NULL, 110, 9, 'debit', DEFAULT, 20000);
Query OK, 30 rows affected (0.08 sec)
Records: 30 Duplicates: 0 Warnings: 0
```

```
mysql> select * from transaksi;
```

no_transaksi	id_nasabahFK	no_rekeningFK	jenis_transaksi	tanggal	jumlah
1	3	105	debit	2019-04-02 22:30:39	50000
2	2	103	debit	2019-04-02 22:30:39	40000
3	4	101	kredit	2019-04-02 22:30:39	20000
4	3	106	debit	2019-04-02 22:30:39	50000
5	5	107	kredit	2019-04-02 22:30:39	30000
6	1	104	kredit	2019-04-02 22:30:39	200000
7	9	110	kredit	2019-04-02 22:30:39	150000
8	5	102	debit	2019-04-02 22:30:39	20000
9	3	105	kredit	2019-04-02 22:30:39	50000
10	4	107	debit	2019-04-02 22:30:39	100000
11	2	103	debit	2019-04-02 22:30:39	100000
12	1	104	debit	2019-04-02 22:30:39	50000
13	4	107	kredit	2019-04-02 22:30:39	200000
14	3	105	debit	2019-04-02 22:30:39	40000
15	1	104	kredit	2019-04-02 22:30:39	100000
16	4	101	kredit	2019-04-02 22:30:39	20000
17	2	103	debit	2019-04-02 22:30:39	50000
18	5	102	debit	2019-04-02 22:30:39	50000
19	10	108	debit	2019-04-02 22:30:39	100000
20	3	106	kredit	2019-04-02 22:30:39	50000
21	2	103	kredit	2019-04-02 22:30:39	200000
22	3	105	kredit	2019-04-02 22:30:39	100000
23	5	102	debit	2019-04-02 22:30:39	20000
24	1	104	debit	2019-04-02 22:30:39	50000
25	2	103	debit	2019-04-02 22:30:39	40000
26	4	101	debit	2019-04-02 22:30:39	50000
27	2	103	kredit	2019-04-02 22:30:39	100000
28	5	102	kredit	2019-04-02 22:30:39	200000
29	7	109	debit	2019-04-02 22:30:39	100000
30	9	110	debit	2019-04-02 22:30:39	20000

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

6. Update

- Nasabah dengan nama "Indri Hapsari" pindah alamat ke "Jalan Slamet Riyadi No.34"

```
mysql> 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	Satrio Eka Jaya	Jl.Slamet Riyadi 45
7	Indri Hapsari	Jl.Sutoyo 5
8	Sari Murti	Jl.Pangandaran 11
9	Canka Lokananta	Jl.Tidar 86
10	Budi Murtono	Jl.Merak 22

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

```
mysql> update nasabah set alamat_nasabah = 'Jalan Slamet Riyadi No.34'
-> where nama_nasabah = 'Indri Hapsari';
Query OK, 1 row affected (0.11 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> 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
7	Indri Hapsari	Jalan Slamet Riyadi No.34
8	Budi Murti	Jl.Pengaderan 11
9	Canka Lokananta	Jl.Tidar 86
10	Budi Murtono	Jl.Merak 22

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

- b. Cabang dengan kode "BRUW" pindah ke alamat "Jalan A.Yani No.23"

```
mysql> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUB	Bank Rut Unit Boyolali	Jl.Ahmad Yani 45
BRUK	Bank Rut Unit Klaten	Jl.Suparman 23
BRUM	Bank Rut Unit Magelang	Jl.P.Tendean 63
BRUS	Bank Rut Unit Surakarta	Jl.Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jl.Untung Suropati 12
BRUY	Bank Rut Unit Yogyakarta	Jl.Anggrek 21

```
6 rows in set (0.00 sec)

mysql>
mysql> update cabang_bank set alamat_cabang = 'Jalan A.Yani No.23'
-> where kode_cabang = 'BRUW';
Query OK, 1 row affected (0.13 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUB	Bank Rut Unit Boyolali	Jl.Ahmad Yani 45
BRUK	Bank Rut Unit Klaten	Jl.Suparman 23
BRUM	Bank Rut Unit Magelang	Jl.P.Tendean 63
BRUS	Bank Rut Unit Surakarta	Jl.Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jalan A.Yani No.23
BRUY	Bank Rut Unit Yogyakarta	Jl.Anggrek 21

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

7. Delete

- Nasabah dengan id “7” menutup rekeningnya

```
mysql> delete from nasabah  
-> where id_nasabah = 7;  
Query OK, 1 row affected (0.08 sec)
```

```
mysql> 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

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

- Cabang dengan nama_cabang “Bank Rut Unit Magelang” menutup kantornya

```
mysql> delete from cabang_bank  
-> where nama_cabang = 'Bank Rut Unit Magelang';  
Query OK, 1 row affected (0.12 sec)
```

```
mysql> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUB	Bank Rut Unit Boyolali	Jl.Ahmad Yani 45
BRUK	Bank Rut Unit Klaten	Jl.Suparman 23
BRUS	Bank Rut Unit Surakarta	Jl.Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jalan A.Yani No.23
BRUY	Bank Rut Unit Yogyakarta	Jl.Anggrek 21

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

TUGAS

1) Tabel nasabah

```
mysql> insert into nasabah(id_nasabah, nama_nasabah, alamat_nasabah) values
-> (11, 'Aini Nur', 'Jl.Mawar 13'),
-> (12, 'Salsa Sasmita', 'Jl.Melati 56'),
-> (13, 'Angelita Putri', 'Jl.Pattimura 6'),
-> (14, 'Tyas Melani', 'Jl.Manggis 15'),
-> (15, 'Cinde Prawito', 'Jl.Anggrek 67'),
-> (16, 'Muhammad Rizky', 'Jl.Slamet Riyadi 5'),
-> (17, 'Ahmad Jibril', 'Jl.Melodi 8'),
-> (18, 'Sari Putri', 'Jl.Mangga 3'),
-> (19, 'Ardhia Dhiyani', 'Jl.Kelengkeng 7'),
-> (20, 'Ayu Cahyaningtyas', 'Jl.Budi 64');
Query OK, 10 rows affected (0.07 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> 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.Merek 22
11	Aini Nur	Jl.Mawar 13
12	Salsa Sasmita	Jl.Melati 56
13	Angelita Putri	Jl.Pattimura 6
14	Tyas Melani	Jl.Manggis 15
15	Cinde Prawito	Jl.Anggrek 67
16	Muhammad Rizky	Jl.Slamet Riyadi 5
17	Ahmad Jibril	Jl.Melodi 8
18	Sari Putri	Jl.Mangga 3
19	Ardhia Dhiyani	Jl.Kelengkeng 7
20	Ayu Cahyaningtyas	Jl.Budi 64

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

2) Tabel cabang_bank

```
mysql> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values
-> ('BRUJ', 'Bank Rut Unit Jakarta', 'Jl.Mawar 7'),
-> ('BRUA', 'Bank Rut Unit Aceh', 'Jl.Kelengkeng 5'),
-> ('BRUPA', 'Bank Rut Unit Papua', 'Jl.Manokwari 9'),
-> ('BRUG', 'Bank Rut Unit Gemolong', 'Jl.Sudirman 67'),
-> ('BRUT', 'Bank Rut Unit Ternate', 'Jl.Manggis 54'),
-> ('BRUP', 'Bank Rut Unit Pasuruan', 'Jl.Budi 14'),
-> ('BRUD', 'Bank Rut Unit Dumai', 'Jl.Pattimura 13'),
-> ('BRUBU', 'Bank Rut Unit Bnukittinggi', 'Jl.Semangka 6'),
-> ('BRUPP', 'Bank Rut Unit Padang Panjang', 'Jl.Minang 17'),
-> ('BRUBI', 'Bank Rut Unit Binjai', 'Jl.Budi 8');
Query OK, 10 rows affected (0.10 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> select * from cabang_bank;
```

kode_cabang	nama_cabang	alamat_cabang
BRUA	Bank Rut Unit Aceh	Jl.Kelengkeng 5
BRUB	Bank Rut Unit Boyolali	Jl.Ahmad Yani 45
BRUBI	Bank Rut Unit Binjai	Jl.Budi 8
BRUBU	Bank Rut Unit Bnukittinggi	Jl.Semangka 6
BRUD	Bank Rut Unit Dumai	Jl.Pattimura 13
BRUG	Bank Rut Unit Gemolong	Jl.Sudirman 67
BRUJ	Bank Rut Unit Jakarta	Jl.Mawar 7
BRUK	Bank Rut Unit Klaten	Jl.Suparman 23
BRUP	Bank Rut Unit Pasuruan	Jl.Budi 14
BRUPA	Bank Rut Unit Papua	Jl.Manokwari 9
BRUPP	Bank Rut Unit Padang Panjang	Jl.Minang 17
BRUS	Bank Rut Unit Surakarta	Jl.Slamet Riyadi 10
BRUT	Bank Rut Unit Ternate	Jl.Manggis 54
BRUW	Bank Rut Unit Wonogiri	Jalan A.Yani No.23
BRUY	Bank Rut Unit Yogyakarta	Jl.Anggrek 21

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

3) Tabel rekening

```
mysql> insert into rekening(no_rekening, kode_cabangFK, pin, saldo) values
-> (114, 'BRUPA', '4567', 200000),
-> (115, 'BRUPP', '5678', 450000),
-> (116, 'BRUA', '2345', 700000),
-> (117, 'BRUJ', '3456', 50000),
-> (118, 'BRUG', '6789', 200000),
-> (119, 'BRUT', '6543', 600000),
-> (120, 'BRUP', '7654', 1000000),
-> (121, 'BRUD', '8765', 500000),
-> (122, 'BRUBU', '9876', 200000),
-> (123, 'BRUI', '1580', 550000);
Query OK, 10 rows affected (0.18 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> 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	BRUPA	4567	200000
115	BRUPP	5678	450000
116	BRUA	2345	700000
117	BRUJ	3456	50000
118	BRUG	6789	200000
119	BRUT	6543	600000
120	BRUP	7654	1000000
121	BRUD	8765	500000
122	BRUBU	9876	200000
123	BRUI	1580	550000

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

4) Tabel nasabah_has_rekening

```
mysql> insert into nasabah_has_rekening(id_nasabahFK, no_rekeningFK) values
-> (11, 104), (12, 103),
-> (12, 105), (13, 106),
-> (14, 101), (14, 107),
-> (15, 102), (15, 107),
-> (16, 109), (17, 109);
Query OK, 10 rows affected (0.00 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

```
mysql> 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	104
12	103
12	105
13	106
14	101
14	107
15	102
15	107
16	109
17	109

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


5) Tabel Transaksi

```
mysql> insert into transaksi(no_transaksi, no_rekeningFK, id_nasabahFK, jenis_transaksi, tanggal, jumlah) values
mysql>
-> (NULL, 105, 11, 'debit', DEFAULT, 40000),
-> (NULL, 104, 11, 'kredit', DEFAULT, 100000),
-> (NULL, 101, 14, 'kredit', DEFAULT, 20000),
-> (NULL, 103, 12, 'debit', DEFAULT, 50000),
-> (NULL, 102, 15, 'debit', DEFAULT, 50000),
-> (NULL, 108, 20, 'debit', DEFAULT, 100000),
-> (NULL, 106, 13, 'kredit', DEFAULT, 50000),
-> (NULL, 103, 12, 'kredit', DEFAULT, 200000),
-> (NULL, 105, 13, 'kredit', DEFAULT, 100000),
-> (NULL, 102, 15, 'debit', DEFAULT, 20000),
-> (NULL, 104, 11, 'debit', DEFAULT, 50000),
-> (NULL, 103, 12, 'debit', DEFAULT, 40000),
-> (NULL, 101, 14, 'kredit', DEFAULT, 50000),
-> (NULL, 103, 12, 'kredit', DEFAULT, 100000),
-> (NULL, 102, 15, 'kredit', DEFAULT, 200000),
-> (NULL, 104, 11, 'debit', DEFAULT, 50000),
-> (NULL, 103, 12, 'debit', DEFAULT, 40000),
-> (NULL, 101, 14, 'debit', DEFAULT, 50000),
-> (NULL, 109, 17, 'debit', DEFAULT, 100000),
-> (NULL, 110, 19, 'debit', DEFAULT, 20000);
Query OK, 20 rows affected (0.08 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

```
mysql> select * from transaksi;
```

no_transaksi	id_nasabahFK	no_rekeningFK	jenis_transaksi	tanggal	jumlah
1	3	105	debit	2019-04-02 22:30:39	50000
2	2	103	debit	2019-04-02 22:30:39	40000
3	4	101	kredit	2019-04-02 22:30:39	20000
4	3	106	debit	2019-04-02 22:30:39	50000
5	5	107	kredit	2019-04-02 22:30:39	30000
6	1	104	kredit	2019-04-02 22:30:39	200000
7	9	110	kredit	2019-04-02 22:30:39	150000
8	5	102	debit	2019-04-02 22:30:39	20000
9	3	105	kredit	2019-04-02 22:30:39	50000
10	4	107	debit	2019-04-02 22:30:39	100000
11	2	103	debit	2019-04-02 22:30:39	100000
12	1	104	debit	2019-04-02 22:30:39	50000
13	4	107	kredit	2019-04-02 22:30:39	200000
14	3	105	debit	2019-04-02 22:30:39	40000
15	1	104	kredit	2019-04-02 22:30:39	100000

9	3	105	kredit	2019-04-02 22:30:39	50000
10	4	107	debit	2019-04-02 22:30:39	100000
11	2	103	debit	2019-04-02 22:30:39	100000
12	1	104	debit	2019-04-02 22:30:39	50000
13	4	107	kredit	2019-04-02 22:30:39	200000
14	3	105	debit	2019-04-02 22:30:39	40000
15	1	104	kredit	2019-04-02 22:30:39	100000
16	4	101	kredit	2019-04-02 22:30:39	20000
17	2	103	debit	2019-04-02 22:30:39	50000
18	5	102	debit	2019-04-02 22:30:39	50000
19	10	108	debit	2019-04-02 22:30:39	100000
20	3	106	kredit	2019-04-02 22:30:39	50000
21	2	103	kredit	2019-04-02 22:30:39	200000
22	3	105	kredit	2019-04-02 22:30:39	100000
23	5	102	debit	2019-04-02 22:30:39	20000
24	1	104	debit	2019-04-02 22:30:39	50000
25	2	103	debit	2019-04-02 22:30:39	40000
26	4	101	debit	2019-04-02 22:30:39	50000
27	2	103	kredit	2019-04-02 22:30:39	100000
28	5	102	kredit	2019-04-02 22:30:39	200000
29	7	109	debit	2019-04-02 22:30:39	100000
30	9	110	debit	2019-04-02 22:30:39	20000
31	11	105	debit	2019-04-02 23:48:16	40000
32	11	104	kredit	2019-04-02 23:48:16	100000
33	14	101	kredit	2019-04-02 23:48:16	20000
34	12	103	debit	2019-04-02 23:48:16	50000
35	15	102	debit	2019-04-02 23:48:16	50000
36	20	108	debit	2019-04-02 23:48:16	100000
37	13	106	kredit	2019-04-02 23:48:16	50000
38	12	103	kredit	2019-04-02 23:48:16	200000
39	13	105	kredit	2019-04-02 23:48:16	100000
40	15	102	debit	2019-04-02 23:48:16	20000
41	11	104	debit	2019-04-02 23:48:16	50000
42	12	103	debit	2019-04-02 23:48:16	40000
43	14	101	kredit	2019-04-02 23:48:16	50000
44	12	103	kredit	2019-04-02 23:48:16	100000
45	15	102	kredit	2019-04-02 23:48:16	200000
46	11	104	debit	2019-04-02 23:48:16	50000
47	12	103	debit	2019-04-02 23:48:16	40000
48	14	101	debit	2019-04-02 23:48:16	50000
49	17	109	debit	2019-04-02 23:48:16	100000
50	19	110	debit	2019-04-02 23:48:16	20000

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

MODUL 6

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

```
Command Prompt - mysql -u root -p
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)]> use perbankan;
Database changed
MariaDB [perbankan]> show tables;
+-----+
| Tables_in_perbankan |
+-----+
| cabang_bank          |
| nasabah              |
| nasabah_has_rekening |
| rekening             |
| transaksi            |
+-----+
5 rows in set (0.04 sec)

MariaDB [perbankan]> describe nasabah;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_nasabah     | int(11)       | NO   | PRI | NULL    |       |
| nama_nasabah   | varchar(45)   | NO   |     | NULL    |       |
| alamat_nasabah | varchar(255)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.09 sec)

MariaDB [perbankan]> select nama_nasabah, alamat_nasabah from nasabah order by nama_nasabah;
+-----+-----+
| nama_nasabah | alamat_nasabah |
+-----+-----+
| Budi Eko Prayogo | Jl. Kantil 30 |
| Budi Murtono    | Jl. Merak 22  |
| Canka Lokananta | Jl. Tidar 86  |
| Indri Hapsari   | Jl. Sutoyo 5  |
| Kartika Padmasari | Jl. Manggis 15 |
| Maryati         | Jl. MT. Haryono 31 |
| Sari Murti      | Jl. Pangandaran |
| Satria Eka Jaya | Jl. Slamet Riyadi |
| Suparman        | Jl. Hasanudin 81 |
| Sutopo          | Jl. Jendral Sudirman 12 |
+-----+-----+
```

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

```
MariaDB [perbankan]> describe transaksi;
```

Field	Type	Null	Key	Default	Extra
no_transaksi	bigint(20) unsigned	NO	PRI	NULL	auto_increment
id_nasabahFK	int(11)	YES		NULL	
no_rekeningFK	int(11)	YES		NULL	
jenis_transaksi	varchar(20)	NO		debit	
tanggal	datetime	NO		CURRENT_TIMESTAMP	
jumlah	int(11)	NO		NULL	

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

```
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

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

3. Tampilkan jenis transaksi dan jumlah transaksi yang melakukan transaksi pada tanggal 10 November 2009 dan diurutkan berdasarkan jumlah transaksi!

```
MariaDB [perbankan]> select jenis_transaksi, jumlah from transaksi where tanggal='2009-11-10' order by jumlah desc;
```

jenis_transaksi	jumlah
debit	50000
debit	40000

```
2 rows 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

```
3 rows in set (0.05 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)
```

```
MariaDB [perbankan]>
```

MODUL 7

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

```
Command Prompt - mysql -u root -p
MariaDB [perbankan]> describe nasabah;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_nasabah | int(11)       | NO   | PRI | NULL    |       |
| nama_nasabah | varchar(45)   | NO   |     | NULL    |       |
| alamat_nasabah | varchar(255) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.10 sec)

MariaDB [perbankan]> describe transaksi;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default      | Extra |
+-----+-----+-----+-----+-----+-----+
| no_transaksi | bigint(20) unsigned | NO   | PRI | NULL        | auto_increment |
| id_nasabahFK | int(11)       | YES  |     | NULL        |       |
| no_rekeningFK | int(11)       | YES  |     | NULL        |       |
| jenis_transaksi | varchar(20)   | NO   |     | debit       |       |
| tanggal      | datetime      | NO   |     | CURRENT_TIMESTAMP |       |
| jumlah        | int(11)       | NO   |     | NULL        |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.05 sec)

MariaDB [perbankan]> select nama_nasabah, alamat_nasabah, jenis_transaksi, jumlah from nasabah, transaksi where nasabah.
id_nasabah = transaksi.id_nasabahFK and transaksi = 'kredit' order by nama_nasabah;
ERROR 1054 (42S22): Unknown column 'transaksi' in 'where clause'
MariaDB [perbankan]> select nama_nasabah, alamat_nasabah, jenis_transaksi, jumlah from nasabah, transaksi where nasabah.
id_nasabah = transaksi.id_nasabahFK and jenis_transaksi = 'kredit' order by nama_nasabah;
+-----+-----+-----+-----+
| nama_nasabah | alamat_nasabah | jenis_transaksi | jumlah |
+-----+-----+-----+-----+
| Budi Eko Prayogo | Jl. Kantil 30 | kredit | 30000 |
| Canka Lokananta | Jl. Tidar 86 | kredit | 150000 |
| Kartika Padmasari | Jl. Manggis 15 | kredit | 200000 |
| Kartika Padmasari | Jl. Manggis 15 | kredit | 20000 |
| Kartika Padmasari | Jl. Manggis 15 | kredit | 20000 |
| 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 | 100000 |
+-----+-----+-----+-----+
```

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

```
MariaDB [perbankan]> describe rekening;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no_rekening | int(11) | NO | PRI | NULL | |
| kode_cabangFK | varchar(20) | YES | | NULL | |
| pin | varchar(20) | NO | | 1234 | |
| saldo | int(11) | NO | | 0 | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.05 sec)

MariaDB [perbankan]> select no_rekening, nama_nasabah, jenis_transaksi, jumlah from rekening, nasabah, transaksi where n
asabah.id_nasabah = transaksi.id_nasabahFK and rekening.no_rekening = transaksi.no_rekeningFK and tanggal = '2009-11-21'
order by nama_nasabah;
+-----+-----+-----+-----+
| no_rekening | nama_nasabah | jenis_transaksi | jumlah |
+-----+-----+-----+-----+
| 105 | Suparman | debit | 40000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [perbankan]>
```

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

```
MariaDB [perbankan]> select no_rekening, nama_nasabah, jenis_transaksi, jumlah from rekening, nasabah, transaksi where n
asabah.id_nasabah = transaksi.id_nasabahFK and rekening.no_rekening = transaksi.no_rekeningFK and jumlah = 20000;
+-----+-----+-----+-----+
| no_rekening | nama_nasabah | jenis_transaksi | jumlah |
+-----+-----+-----+-----+
| 101 | Kartika Padmasari | kredit | 20000 |
| 102 | Budi Eko Prayogo | debit | 20000 |
| 101 | Kartika Padmasari | kredit | 20000 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [perbankan]>
```

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

```
MariaDB [perbankan]> describe nasabah_has_rekening;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_nasabahFK | int(11) | NO | PRI | NULL | |
| no_rekeningFK | int(11) | NO | PRI | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)

MariaDB [perbankan]> select no_rekening, nama_nasabah, alamat_nasabah from rekening, nasabah, nasabah_has_rekening where
nasabah.id_nasabah = nasabah_has_rekening.id_nasabahFK and rekening.no_rekening = nasabah_has_rekening.no_rekeningFK and
nama_nasabah like 'Su%';
+-----+-----+-----+
| no_rekening | nama_nasabah | alamat_nasabah |
+-----+-----+-----+
| 104 | Sutopo | Jl. Jendral Sudirman 12 |
| 105 | Suparman | Jl. Hasanudin 81 |
+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [perbankan]>
```

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

Nomor Rekening	Nama Nasabah	Jumlah Transaksi
102	Budi Eko Prayogo	50000
102	Budi Eko Prayogo	20000
108	Budi Murtono	100000
107	Kartika Padmasari	100000
103	Maryati	100000
103	Maryati	40000
103	Maryati	50000
105	Superman	50000
106	Superman	50000
105	Superman	40000
104	Sutopo	50000

11 rows in set (0.00 sec)

```
MariaDB [perbankan]>
```

MODUL 8

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 jenis_transaksi, jumlah as 'Rp', count(jumlah) from transaksi, nasabah where nasabah.id_nasabahFK = transaksi.id_nasabahFK and nasabah.nama_nasabah = 'Kartika Padmasari' group by jenis_transaksi;
```

jenis_transaksi	Rp	count(jumlah)
debit	100000	1
kredit	20000	3

```
2 rows in set (0.03 sec)
```

MariaDB [perbankan]>

2. Berapa jumlah total saldo yang dimiliki oleh Maryati?

```
MariaDB [perbankan]> select sum(saldo) from nasabah, rekening, nasabah_has_rekening where nasabah_has_rekening.id_nasabahFK = nasabah.id_nasabahFK and nasabah_has_rekening.no_rekeningFK = rekening.no_rekening and nama_nasabah = 'Maryati';
```

sum(saldo)
750000

```
1 row in set (0.05 sec)
```

MariaDB [perbankan]>

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

```
MariaDB [perbankan]> select count(jumlah) from transaksi, cabang_bank, rekening where transaksi.no_rekeningFK = rekening.no_rekeningFK and rekening.kode_cabangFK = cabang_bank.kode_cabang group by cabang_bank.nama_cabang;
```

count(jumlah)
1
6
12
1

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

MariaDB [perbankan]>

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

```
MariaDB [perbankan]> select nama_nasabah, sum(saldo) from nasabah, nasabah_has_rekening, rekening where nasabah_has_rekening.id_nasabahFK = nasabah.id_nasabah and nasabah_has_rekening.no_rekeningFK = rekening.no_rekening and rekening.saldo between 500000 and 2000000;
+-----+-----+
| nama_nasabah | sum(saldo) |
+-----+-----+
| Sutopo       | 6200000    |
+-----+-----+
1 row in set (0.00 sec)

MariaDB [perbankan]>
```

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!

```
MariaDB [perbankan]> select nama_nasabah, tanggal, jumlah as 'Rp' from nasabah, transaksi where nasabah.id_nasabah = transaksi.id_nasabahFK and transaksi.jumlah > 100000 order by transaksi.jumlah desc;
+-----+-----+-----+
| nama_nasabah | tanggal           | Rp    |
+-----+-----+-----+
| Sutopo       | 2009-11-15 00:00:00 | 200000 |
| Kartika Padmasari | 2009-11-20 00:00:00 | 200000 |
| Canka Lokananta | 2009-11-15 00:00:00 | 150000 |
+-----+-----+-----+
3 rows in set (0.00 sec)

MariaDB [perbankan]>
```


MODUL 9

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

```
MariaDB [perbankan]> create user 'angietapw'@'localhost' identified by '1234';
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.

```
MariaDB [perbankan]> grant insert, update, delete on perbankan.nasabah to angietapw@localhost;
Query OK, 0 rows affected (0.00 sec)
```

- b. Hanya dapat melakukan SELECT pada tabel cabang_bank.

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

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

- a. Lakukan perintah SELECT pada tabel nasabah.

```
C:\xampp\mysql\bin>mysql -u angietapw -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)]> use perbankan
Database changed
MariaDB [perbankan]> select * from nasabah;
ERROR 1142 (42000): SELECT command denied to user 'angietapw'@'localhost' for table 'nasabah'
```

- b. Lakukan perintah INSERT pada tabel nasabah.

```
MariaDB [perbankan]> insert into nasabah(id_nasabah, nama_nasabah, alamat_nasabah) values
-> (96, 'Angieta', 'Sragen');
Query OK, 1 row affected (0.05 sec)
```

- c. Lakukan perintah INSERT pada tabel cabang_bank.

```
MariaDB [perbankan]> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values
-> ('BRUS', 'Bank Rut Unit Sragen', 'jl.sukowati 06');
ERROR 1142 (42000): INSERT command denied to user 'angietapw'@'localhost' for table 'cabang_bank'
MariaDB [perbankan]>
```

4. Analisa hasil percobaan.

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

MODUL 10

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' 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 [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)

MODUL 11

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
>>>
```


MODUL 12

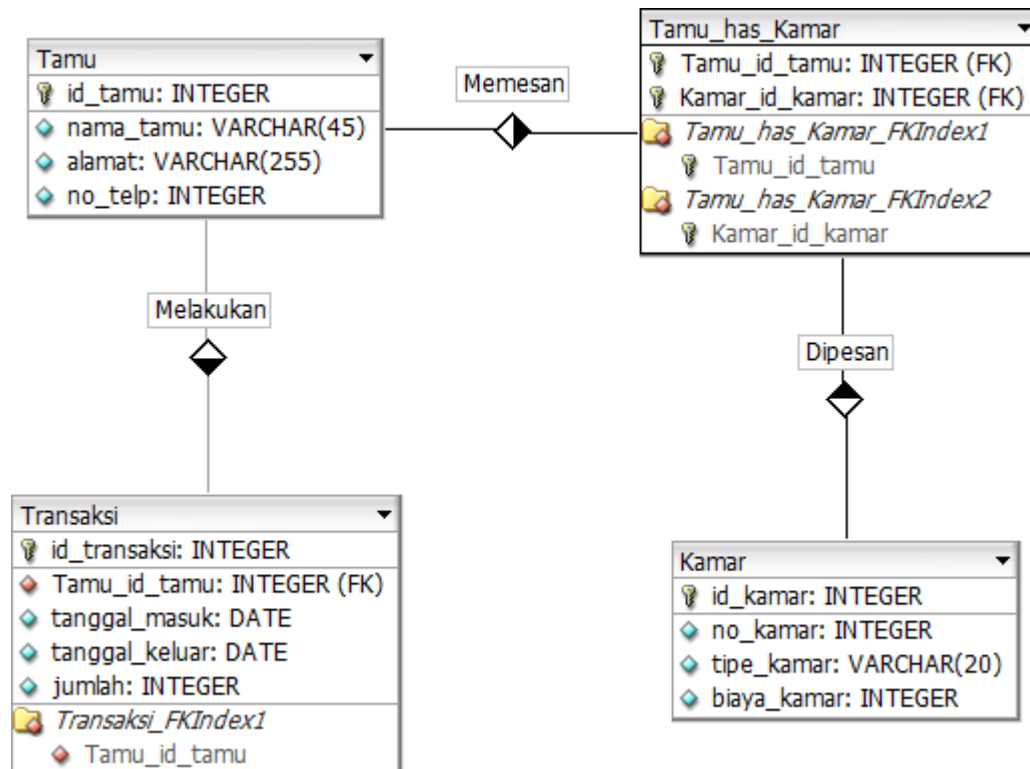
1. Aplikasi Perhotelan Studi Kasus :

Membuat rancangan sebuah database untuk menangani data-data perhotelan.

Data yang akan ditangani meliputi:

- Data Tamu
- Data Kamar
- Data Transaksi

2. Desain ER Diagram



3. Membuat Database Perhotelan

```
MariaDB [(none)]> create database Perhotelan;  
Query OK, 1 row affected (0.13 sec)  
  
MariaDB [(none)]> use perhotelan;  
Database changed
```

- Table Tamu

```
MariaDB [perhotelan]> create table tamu(
  -> id_tamu integer primary key,
  -> nama_tamu varchar(45) not null,
  -> alamat varchar(255));
Query OK, 0 rows affected (0.21 sec)

MariaDB [perhotelan]> insert into tamu values
  -> ('1', 'Angieta', 'Sragen'),
  -> ('2', 'Putri', 'Solo'),
  -> ('3', 'keyla', 'Ngawi');
Query OK, 3 rows affected (0.09 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [perhotelan]> select * from tamu;
+-----+-----+-----+
| id_tamu | nama_tamu | alamat |
+-----+-----+-----+
|      1 | Angieta   | Sragen |
|      2 | Putri     | Solo   |
|      3 | keyla     | Ngawi  |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Table Kamar

```
MariaDB [perhotelan]> create table kamar (
  -> id_kamar integer primary key,
  -> no_kamar integer not null,
  -> tipe_kamar varchar(20) not null,
  -> biaya_kamar integer );
Query OK, 0 rows affected (0.18 sec)

MariaDB [perhotelan]> insert into kamar values
  -> (1101, 101, 'Standar', 150000),
  -> (1102, 102, 'Standar', 150000),
  -> (2201, 201, 'Suite', 300000),
  -> (2202, 202, 'Suite', 300000);
Query OK, 4 rows affected (0.06 sec)
Records: 4 Duplicates: 0 Warnings: 0

MariaDB [perhotelan]> select * from kamar;
+-----+-----+-----+-----+
| id_kamar | no_kamar | tipe_kamar | biaya_kamar |
+-----+-----+-----+-----+
|      1101 |      101 | Standar    |      150000 |
|      1102 |      102 | Standar    |      150000 |
|      2201 |      201 | Suite      |      300000 |
|      2202 |      202 | Suite      |      300000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- Table Transaksi

```

MariaDB [perhotelan]> create table transaksi(
  -> id_transaksi integer primary key,
  -> tamu_id_tamu integer references tamu(id_tamu) on delete set null on update cascade,
  -> tanggal_masuk date,
  -> tanggal_keluar date,
  -> jumlah integer);
Query OK, 0 rows affected (0.22 sec)

MariaDB [perhotelan]> insert into transaksi values
  -> (1111, 1, '2019-06-17', '2019-06-18', 150000),
  -> (1112, 2, '2019-06-20', '2019-06-22', 300000),
  -> (1113, 3, '2019-06-25', '2019-06-26', 300000);
Query OK, 3 rows affected (0.10 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [perhotelan]> select * from transaksi;
+-----+-----+-----+-----+-----+
| id_transaksi | tamu_id_tamu | tanggal_masuk | tanggal_keluar | jumlah |
+-----+-----+-----+-----+-----+
|          1111 |             1 | 2019-06-17    | 2019-06-18    | 150000 |
|          1112 |             2 | 2019-06-20    | 2019-06-22    | 300000 |
|          1113 |             3 | 2019-06-25    | 2019-06-26    | 300000 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

- Table tamu_has_kamar

```

MariaDB [perhotelan]> create table tamu_has_kamar (
  -> tamu_id_tamu integer references tamu(id_tamu) on delete cascade on update cascade,
  -> kamar_id_kamar integer references kamar(id_kamar) on delete cascade on update cascade,
  -> primary key(tamu_id_tamu, kamar_id_kamar));
Query OK, 0 rows affected (0.17 sec)

MariaDB [perhotelan]> insert into tamu_has_kamar values
  -> (1, 1101),
  -> (2, 1102),
  -> (3, 2201);
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

MariaDB [perhotelan]> select * from tamu_has_kamar;
+-----+-----+
| tamu_id_tamu | kamar_id_kamar |
+-----+-----+
|             1 |             1101 |
|             2 |             1102 |
|             3 |             2201 |
+-----+-----+
3 rows in set (0.00 sec)

```

4. Aplikasi Perhotelan dengan Python

❖ Tamu

```
import mysql.connector
import os

db = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='',
    database='perhotelan'
)

def insert_data(db):
    id_tamu = input('Masukkan ID Tamu: ')
    nama_tamu = input('Masukkan Nama Tamu: ')
    alamat = input('Masukkan Alamat Tamu: ')
    val = (id_tamu, nama_tamu, alamat)
    cursor = db.cursor()
    sql = 'insert into tamu (id_tamu, nama_tamu, alamat) values (%s, %s, %s)'
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil disimpan".format(cursor.rowcount))

def show_data(db):
    cursor = db.cursor()
    sql = 'select * from tamu'
    cursor.execute(sql)
    results = cursor.fetchall()

    if cursor.rowcount < 0:
        print('Tidak ada data')
    else:
        for data in results:
            print(data)

def update_data(db):
    cursor = db.cursor()
    show_data(db)
    id_tamu = input('Pilih ID Tamu> ')
    nama_tamu = input('Nama baru: ')
    alamat = input('Alamat baru: ')
    sql = 'update tamu set nama_tamu=%s, alamat=%s where id_tamu=%s'
    val = (nama_tamu, alamat, id_tamu)
    cursor.execute(sql, val)
    db.commit()
    print('{} data berhasil diubah'.format(cursor.rowcount))
```

```

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    id_tamu = input('Pilih ID Tamu> ')
    sql = 'delete from tamu where id_tamu=%s'
    val = (id_tamu,)
    cursor.execute(sql, val)
    db.commit()
    print('{} data berhasil dihapus'.format(cursor.rowcount))

def show_menu(db):
    print('==== Aplikasi Database Perhotelan | ====')
    print('1. Tampilkan Data')
    print('2. Insert Data')
    print('3. Update Data')
    print('4. Hapus Data')
    print('5. Keluar')
    print('=====')
    menu = input('Pilih Menu: ')

    #clear screen
    os.system('clear')

    if menu == '1':
        show_data(db)
    elif menu == '2':
        insert_data(db)
    elif menu == '3':
        update_data(db)
    elif menu == '4':
        delete_data(db)
    elif menu == '5':
        exit()
    else:
        print('Menu Salah')

if __name__ == '__main__':
    while(True):
        show_menu(db)

```

- Menampilkan data (Select)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit

=====
Pilih Menu: 1
(1, 'Angieta', 'Sragen')
(2, 'Putri', 'Solo')
(3, 'keyla', 'Ngawi')
```

- Memasukkan data (Insert)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit

=====
Pilih Menu: 2
Masukkan ID Tamu: 4
Masukkan Nama Tamu: selly
Masukkan Alamat Tamu: jogja
1 Data berhasil disimpan
```

- Mengubah data (Update)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit

=====
Pilih Menu: 3
(1, 'Angieta', 'Sragen')
(2, 'Putri', 'Solo')
(3, 'keyla', 'Ngawi')
(4, 'selly', 'jogja')
Pilih ID Tamu> 4
Nama baru: roni
Alamat baru: bandung
1 Data Berhasil Diubah
```

- Menghapus data (delete)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit
=====
Pilih Menu: 4
(1, 'Angieta', 'Sragen')
(2, 'Putri', 'Solo')
(3, 'keyla', 'Ngawi')
(4, 'roni', 'bandung')
Pilih ID Tamu> 4
1 Data Berhasil Dihapus
```

❖ Kamar

```
import mysql.connector
import os
db = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='',
    database='perhotelan'
)
def insert_data(db):
    id_kamar = input('Masukkan ID Kamar: ')
    no_kamar = input('Masukkan No. Kamar: ')
    tipe_kamar = input('Masukkan Tipe Kamar: ')
    biaya_kamar = input('Harga Kamar: ')
    val = (id_kamar, no_kamar, tipe_kamar, biaya_kamar)
    cursor = db.cursor()
    sql = 'insert into kamar (id_kamar, no_kamar, tipe_kamar, biaya_kamar) values (%s, %s, %s, %s)'
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil disimpan".format(cursor.rowcount))

def show_data(db):
    cursor = db.cursor()
    sql = 'select * from kamar'
    cursor.execute(sql)
    results = cursor.fetchall()

    if cursor.rowcount < 0:
        print('Tidak ada data')
    else:
        for data in results:
            print(data)

def update_data(db):
    cursor = db.cursor()
    show_data(db)
    id_kamar = input('Masukkan ID Kamar> ')
    no_kamar = input('Masukkan No. Kamar: ')
    tipe_kamar = input('Masukkan Tipe Kamar: ')
    biaya_kamar = input('Harga Kamar: ')
    sql = 'update kamar set no_kamar=%s, tipe_kamar=%s, biaya_kamar=%s where id_kamar=%s'
    val = (no_kamar, tipe_kamar, biaya_kamar, id_kamar)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil diubah".format(cursor.rowcount))
```



```

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    id_kamar = input('Pilih ID Kamar> ')
    sql = 'delete from kamar where id_kamar=%s'
    val = (id_kamar,)
    cursor.execute(sql, val)
    db.commit()
    print('{} data berhasil dihapus'.format(cursor.rowcount))

def show_menu(db):
    print('==== Aplikasi Database Perhotelan ====')
    print('1. Tampilkan Data')
    print('2. Insert Data')
    print('3. Update Data')
    print('4. Hapus Data')
    print('5. Keluar')
    print('=====')
    menu = input('Pilih Menu: ')

    #clear screen
    os.system('clear')

    if menu == '1':
        show_data(db)
    elif menu == '2':
        insert_data(db)
    elif menu == '3':
        update_data(db)
    elif menu == '4':
        delete_data(db)
    elif menu == '5':
        exit()
    else:
        print('Menu Salah')

if __name__ == '__main__':
    while(True):
        show_menu(db)

```

- Menampilkan data (Select)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit
=====
Pilih Menu: 1
(1101, 101, 'Standar', 150000)
(1102, 102, 'Standar', 150000)
(2201, 201, 'Suite', 300000)
(2202, 202, 'Suite', 300000)
```

- Memasukkan data (Insert)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit
=====
Pilih Menu: 2
Masukkan ID Kamar: 2203
Masukkan No. Kamar: 203
Masukkan Tipe Kamar: Suite
Harga Kamar: 300000
1 Data berhasil disimpan
```

- Mengubah data (Update)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit
=====
Pilih Menu: 3
(1101, 101, 'Standar', 150000)
(1102, 102, 'Standar', 150000)
(2201, 201, 'Suite', 300000)
(2202, 202, 'Suite', 300000)
(2203, 203, 'Suite', 300000)
Masukkan ID Kamar> 2203
Masukkan No. Kamar: 103
Masukkan Tipe Kamar: Standar
Harga Kamar: 150000
1 Data Berhasil Diubah
```

- Menghapus data (delete)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Delete Data
5. Exit
=====
Pilih Menu: 4
(1101, 101, 'Standar', 150000)
(1102, 102, 'Standar', 150000)
(2201, 201, 'Suite', 300000)
(2202, 202, 'Suite', 300000)
(2203, 103, 'Standar', 150000)
Pilih ID Kamar> 2203
1 Data Berhasil Dihapus
```

❖ Transaksi

```
import mysql.connector
import os
db = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='',
    database='perhotelan'
)

def insert_data(db):
    id_transaksi = input('Masukkan ID Transaksi: ')
    tamu_id_tamu = input('Masukkan ID Tamu: ')
    tanggal_masuk = input('Tanggal Masuk: ')
    tanggal_keluar = input('Tanggal Keluar: ')
    jumlah = input('Total Transaksi: ')
    val = (id_transaksi, tamu_id_tamu, tanggal_masuk, tanggal_keluar, jumlah)
    cursor = db.cursor()
    sql = 'insert into transaksi (id_transaksi, tamu_id_tamu, tanggal_masuk, tanggal_keluar, jumlah) values (%s, %s, %s, %s, %s)'
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil disimpan".format(cursor.rowcount))

def show_data(db):
    cursor = db.cursor()
    sql = 'select * from transaksi'
    cursor.execute(sql)
    results = cursor.fetchall()

    if cursor.rowcount<0:
        print ('Tidak ada data')
    else:
        for data in results:
            print(data)

def update_data(db):
    cursor = db.cursor()
    show_data(db)
    id_transaksi = input('Masukkan ID Transaksi> ')
    tamu_id_tamu = input('Masukkan ID Tamu: ')
    tanggal_masuk = input('Tanggal Masuk: ')
    tanggal_keluar = input('Tanggal Keluar: ')
    jumlah = input('Total Transaksi: ')
    sql = 'update transaksi set tamu_id_tamu=%s, tanggal_masuk=%s, tanggal_keluar=%s, jumlah=%s where id_transaksi=%s'
    val = (tamu_id_tamu, tanggal_masuk, tanggal_keluar, jumlah, id_transaksi)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil diubah".format(cursor.rowcount))

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    id_transaksi = input('Pilih ID Transaksi> ')
    sql = 'delete from transaksi where id_transaksi=%s'
    val = (id_transaksi,)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil dihapus".format(cursor.rowcount))
```

```

def show_menu(db):
    print('==== Aplikasi Database Perhotelan ====')
    print('1. Tampilkan Data')
    print('2. Insert Data')
    print('3. Update Data')
    print('4. Hapus Data')
    print('5. Keluar')
    print('=====')
    menu = input('Pilih Menu: ')

    #clear screen
    os.system('clear')

    if menu == '1':
        show_data(db)
    elif menu == '2':
        insert_data(db)
    elif menu == '3':
        update_data(db)
    elif menu == '4':
        delete_data(db)
    elif menu == '5':
        exit()
    else:
        print('Menu Salah')

if __name__ == '__main__':
    while(True):
        show_menu(db)

```

- Menampilkan data (Select)

```

==== Aplikasi Database Perhotelan ====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 1
(1111, 1, datetime.date(2019, 6, 17), datetime.date(2019, 6, 18), 150000)
(1112, 2, datetime.date(2019, 6, 20), datetime.date(2019, 6, 22), 300000)
(1113, 3, datetime.date(2019, 6, 25), datetime.date(2019, 6, 26), 300000)

```

- Memasukkan data (Insert)

```

==== Aplikasi Database Perhotelan ====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 2
Masukkan ID Transaksi: 1114
Masukkan ID Tamu: 4
Tanggal Masuk: 2019-6-27
Tanggal Keluar: 2019-6-28
Total Transaksi: 150000
1 data berhasil disimpan

```

- Mengubah data (Update)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 3
(1111, 1, datetime.date(2019, 6, 17), datetime.date(2019, 6, 18), 150000)
(1112, 2, datetime.date(2019, 6, 20), datetime.date(2019, 6, 22), 300000)
(1113, 3, datetime.date(2019, 6, 25), datetime.date(2019, 6, 26), 300000)
(1114, 4, datetime.date(2019, 6, 27), datetime.date(2019, 6, 28), 150000)
Masukkan ID Transaksi> 1114
Masukkan ID Tamu: 4
Tanggal Masuk: 2019-6-27
Tanggal Keluar: 2019-6-30
Total Transaksi: 450000
1 data berhasil diubah
```

- Menghapus data (delete)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 4
(1111, 1, datetime.date(2019, 6, 17), datetime.date(2019, 6, 18), 150000)
(1112, 2, datetime.date(2019, 6, 20), datetime.date(2019, 6, 22), 300000)
(1113, 3, datetime.date(2019, 6, 25), datetime.date(2019, 6, 26), 300000)
(1114, 4, datetime.date(2019, 6, 27), datetime.date(2019, 6, 30), 450000)
pilih ID Transaksi> 1114
1 data berhasil dihapus
```

❖ Tamu_has_kamar

```
import mysql.connector
import os

db = mysql.connector.connect(
    host='localhost',
    user='root',
    passwd='',
    database='perhotelan'
)

def insert_data(db):
    tamu_id_tamu = input('Masukkan ID Tamu: ')
    kamar_id_kamar = input('Masukkan ID Kamar: ')
    val = (tamu_id_tamu, kamar_id_kamar)
    cursor = db.cursor()
    sql = 'insert into tamu_has_kamar (tamu_id_tamu, kamar_id_kamar) values (%s, %s)'
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil disimpan".format(cursor.rowcount))

def show_data(db):
    cursor = db.cursor()
    sql = 'select * from tamu_has_kamar'
    cursor.execute(sql)
    results = cursor.fetchall()

    if cursor.rowcount < 0:
        print('Tidak ada data')
    else:
        for data in results:
            print(data)

def update_data(db):
    cursor = db.cursor()
    show_data(db)
    tamu_id_tamu = input('Masukkan ID Tamu: ')
    kamar_id_kamar = input('Masukkan ID Kamar: ')
    sql = 'update tamu_has_kamar set kamar_id_kamar=%s where tamu_id_tamu=%s'
    val = (kamar_id_kamar, tamu_id_tamu)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil diubah".format(cursor.rowcount))
```

```

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    tamu_id_tamu = input('Pilih ID Tamu> ')
    sql = 'delete from tamu_has_kamar where tamu_id_tamu=%s'
    val = (tamu_id_tamu,)
    cursor.execute(sql, val)
    db.commit()
    print('{} data berhasil dihapus'.format(cursor.rowcount))

def show_menu(db):
    print('===== Aplikasi Database Perhotelan| =====')
    print('1. Tampilkan Data')
    print('2. Insert Data')
    print('3. Update Data')
    print('4. Hapus Data')
    print('5. Keluar')
    print('=====')
    menu = input('Pilih Menu: ')

    #clear screen
    os.system('clear')

    if menu == '1':
        show_data(db)
    elif menu == '2':
        insert_data(db)
    elif menu == '3':
        update_data(db)
    elif menu == '4':
        delete_data(db)
    elif menu == '5':
        exit()
    else:
        print('Menu Salah')

if __name__ == '__main__':
    while(True):
        show_menu(db)

```


- Menampilkan data (Select)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 1
(1, 1101)
(2, 1102)
(3, 2201)
(4, 2202)
```

- Memasukkan data (Insert)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 2
Masukkan ID Tamu: 5
Masukkan ID Kamar: 2203
1 data berhasil disimpan
```

- Mengubah data (Update)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 3
(1, 1101)
(2, 1102)
(3, 2201)
(4, 2202)
(5, 2203)
Masukkan ID Tamu: 5
Masukkan ID Kamar: 1103
1 data berhasil diubah
```

- Menghapus data (delete)

```
===== Aplikasi Database Perhotelan =====
1. Tampilkan Data
2. Insert Data
3. Update Data
4. Hapus Data
5. Keluar
=====
Pilih Menu: 4
(1, 1101)
(2, 1102)
(3, 2201)
(4, 2202)
(5, 1103)
Pilih ID Tamu> 5
1 data berhasil dihapus
```

