

NAMA : PALISTYA NESTY HARIJANTI

KELAS : D

NIM : L200170083

REKAP PRAKTIKUM BASIS DATA MODUL 1-12

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!

Keuntungan Database

Adapun keuntungan database terhadap sistem pemrosesan adalah:

- 1) Kemubajiran data berkurang.
- 2) Integritas data dapat selalu terjaga.
- 3) Berbagai data dapat selalu dilakukan oleh setiap user.
- 4) Penggunaan data lebih mudah.
- 5) Konsistensi data dapat selalu terjaga.

Manfaat Database

Adapun manfaat database adalah:

- a) Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
- b) Menentukan kualitas informasi yaitu cepat, akurat, dan relevan, sehingga informasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- c) Mengatasi kerangkapan data (redundancy data).
- d) Menghindari terjadinya inkonsistensi data.
- e) Mengatasi kesulitan dalam mengakses data.
- f) Menyusun format yang standar dari sebuah data.
- g) Penggunaan oleh banyak pemakai (multiple user). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multiuser).
- h) Melakukan perlindungan dan pengamanan data. Setiap data hanya bisa diakses atau dimanipulasi oleh pihak yang diberi otoritas dengan memberikan login dan password terhadap masing-masing data.
- i) Agar pemakai mampu menyusun suatu pandangan (view) abstraksi dari data. Hal ini bertujuan menyederhanakan interaksi antara pengguna dengan sistemnya dan database dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer dan administrasinya.

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

yang menjadi acuan dalam pemilihan database adalah sebagai berikut.

- a) Pendeskripsian kebutuhan informasi dan data
- b) Spesifikasi data
- c) Pemrosesan yang diperlukan oleh data
- d) Pertimbangan keamanan
- e) Kecocokan dengan tipe aplikasi
- f) Bahasa query
- g) 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 (kolom) : Merupakan elemen dari tabel yang berisikan informasi tertentu yang spesifik tentang subjudul tabel pada sebuah item data.

Syarat-syarat pembentukan Field Name pada tabel:

- a) Harus Unik atau Spesifik
- b) Boleh disingkat
- c) Pemisah sebagai pengganti spasi dalam pembentuk field adalah tanda lambang "_"

Contoh: Kode Barang menjadi KdBarang, KodeBrg, Kd_Brg, Kd_Barang.

Record (baris): Sekumpulan data yang saling berkaitan tentang sebuah subjek tertentu, misalnya data seorang siswa akan disimpan dalam record yang terdiri dari beberapa kolom / field.

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

Tugas!

1. Rancangan database mengenai mahasiswa, dosen, ruang kelas, dan mata kuliah!

Langkah-langkah perancangan database:

1. Menentukan entities:

- a. Mahasiswa: menyimpan semua data pribadi semua mahasiswa
- b. Dosen : menyimpan semua data pribadi semua dosen
- c. Mata_kuliah : menyimpan semua data mata kuliah
- d. Ruang_kelas : menyimpan informasi semua ruang kelas

2. Menentukan attribute(sifat-sifat) masing-masing entiti sesuai kebutuhan database

a. Mahasiswa

- Nama_mahasiswa: nama lengkap mahasiswa (varchar(45))
- NIM_mahasiswa: nomor induk mahasiswa (varchar(25)) PK
- Alamat_mahasiswa: alamat lengkap mahasiswa (varchar(255))

b. Dosen

- Nama_dosen: nama lengkap dosen (varchar(45))
- NIP_dosen: nomor induk pegawai dosen (varchar(24)) PK
- Alamat: alamat lengkap dosen (varchar(255))

c. Mata_kuliah

- Nama_mata kuliah: nama mata kuliah yang diajarkan (varchar(25))
- Kode_MK: kode setiap mata kuliah (varchar(25)) PK

d. Ruang_kelas

- Nama_ruang: nama ruang kelas (varchar(25))
- Kode_ruang: kode ruang kelas (varchar(25)) PK

3. Menentukan relationship (hubungan) antar entitas

	mahasiswa	dosen	Mata_kuliah	Ruang_kelas
mahasiswa	-	m:n	m:n	n:1
dosen		-	m:n	n:1
Mata_kuliah			-	-
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, NIP_dosen, (FK NIM, NIP di Mahasiswa_has_Dosen)
- Mahasiswa mengambil Mata_kuliah
 - ✓ Tabel utama: mahasiswa, Mata_kuliah
 - ✓ Tabel kedua: mahasiswa_has_mata_kuliah
 - ✓ Relationship: many-to-many(m:n)
 - ✓ Atribut penghubung: nama_mahasiswa, kode_MK (FK nama_mahasiswa, kode_MK di mahasiswa_has_mata_kuliah)
- Mahasiswa menempati Ruang_kelas

- ✓ Tabel utama: mahasiswa, ruang_kelas
- ✓ Tabel kedua: ruang_kelas
- ✓ Relationship: one-to-many(1:n)
- ✓ Atribut penghubung: nim_mahasiswa (FK nim_mahasiswa di ruang_kelas)

➤ Dosen mengampu Mata_kuliah

- ✓ Tabel utama: dosen, mata_kuliah
- ✓ Tabel kedua: dosen_has_mata_kuliah
- ✓ Relationship: many-to-many(m:n)
- ✓ Atribut penghubung: NIP_dosen, kode_MK (FK NIP_dosen, kode_MK di dosen_has_mata_kuliah)

➤ Dosen menempati ruang_kelas

- ✓ Tabel utama: dosen, ruang_kelas
- ✓ Tabel kedua: ruang_kelas
- ✓ Relationship: one-to-many(1:n)
- ✓ Atribut penghubung: NIP_dosen(FK NIP_dosen di ruang_kelas)

2. Rancangan database mengenai Pembeli, Toko, Pemilik, dan Barang

Langkah-langkah perancangan database:

1. Menentukan entities:

- e. Pembeli: menyimpan semua data pribadi pembeli.
- f. Toko : menyimpan semua data tentang toko.
- g. Pemilik : menyimpan semua data tentang pemilik.
- h. Barang : menyimpan informasi semua tentang barang.

2. Menentukan atribut(sifat-sifat) masing-masing entiti sesuai kebutuhan database

a. Pembeli

- Id_pembeli: id yang dimiliki pembeli (varchar(25)) PK
- Nama_pembeli: nama lengkap pembeli (varchar(25))
- Alamat_pembeli: alamat lengkap pembeli (varchar(255))

b. Toko

- Nama_toko: nama toko (varchar(45))
- No_toko: nomor toko (varchar(24)) PK
- Alamat: alamat lengkap toko (varchar(255))

c. Pemilik

- Nama_pemilik: nama lengkap pemilik toko (varchar(25))
- Id_pemilik: id yang dimiliki pemilik (varchar(25)) PK
- Alamat: alamat lengkap pemilik (varchar(225))

d. Barang

- Nama_barang: nama barang (varchar(25))
- Kode_barang: kode barang (varchar(25)) PK
- Jenis: jenis barang yang di jual (varchar(25))

3. Menentukan relationship (hubungan) antar entitas

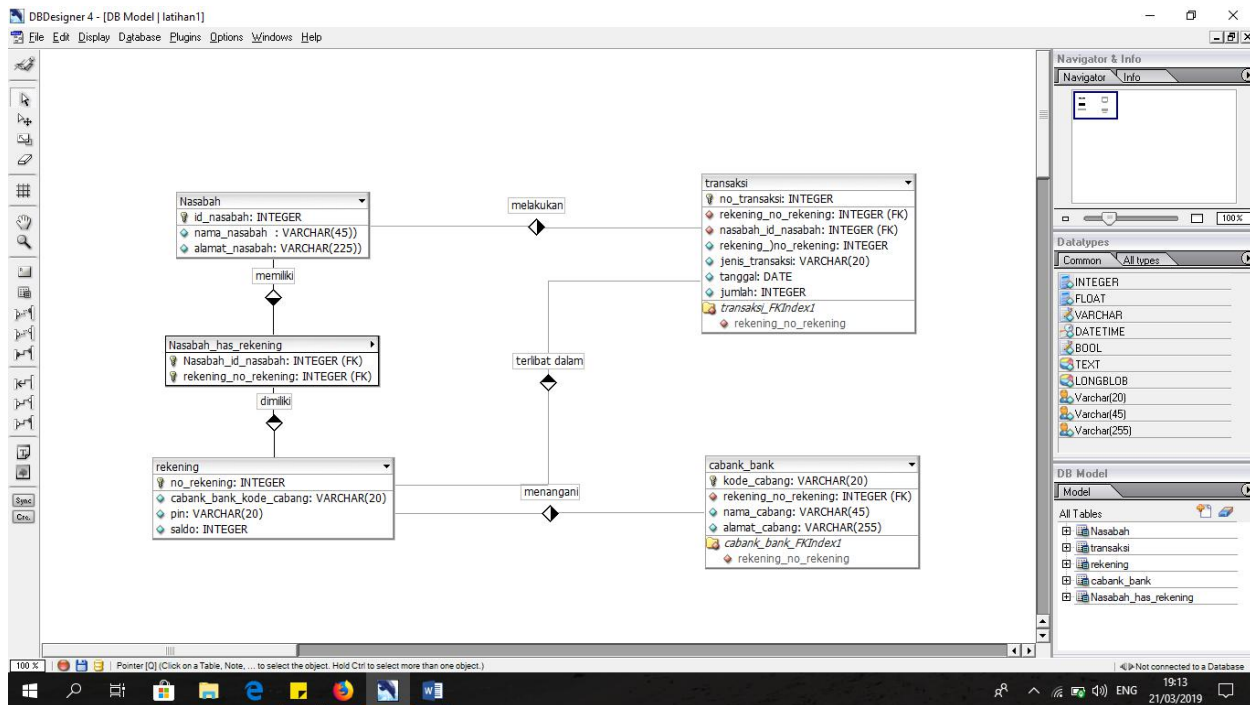
	pembeli	toko	pemilik	barang
pembeli	-	1:1	-	n:1
toko		-	m:n	m:n
pemilik			-	-
barang				-

Hubungan

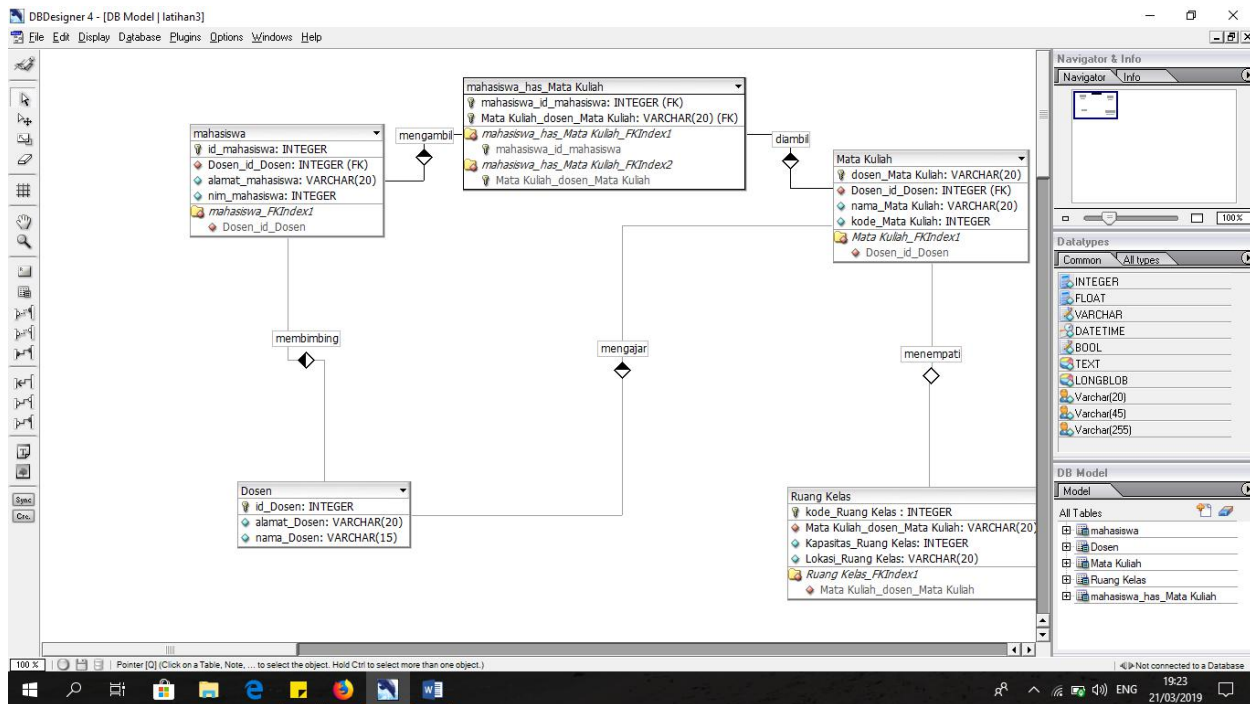
- Pembeli mengunjungi toko:
 - ✓ Tabel utama: pembeli, toko
 - ✓ Tabel kedua: toko
 - ✓ Relationship: one-to-one(1:1)
 - ✓ Atribut penghubung:
- Pembeli membeli barang:
 - ✓ Tabel utama: pembeli, barang
 - ✓ Tabel kedua: pembeli_has_barang
 - ✓ Relationship: one_to_many(1:n)
 - ✓ Atribut penghubung: id_pembeli(FK id_pembeli di barang)
- Toko dimiliki pemilik:
 - ✓ Tabel utama: toko, pemilik
 - ✓ Tabel kedua: toko_has_pemilik
 - ✓ Relationship: many_to_many(m:n)
 - ✓ Atribut penghubung: id_toko, id_pemilik(FK id_toko, id_pemilik di toko_has_pemilik)
- Toko penempatan barang
 - ✓ Tabel utama: toko, barang
 - ✓ Tabel kedua: toko_has_barang
 - ✓ Relationship: many-to-many(m:n)
 - ✓ Atribut penghubung: id_toko, id_barang (FK id_toko, id_barang di toko_has_barang)

MODUL 3

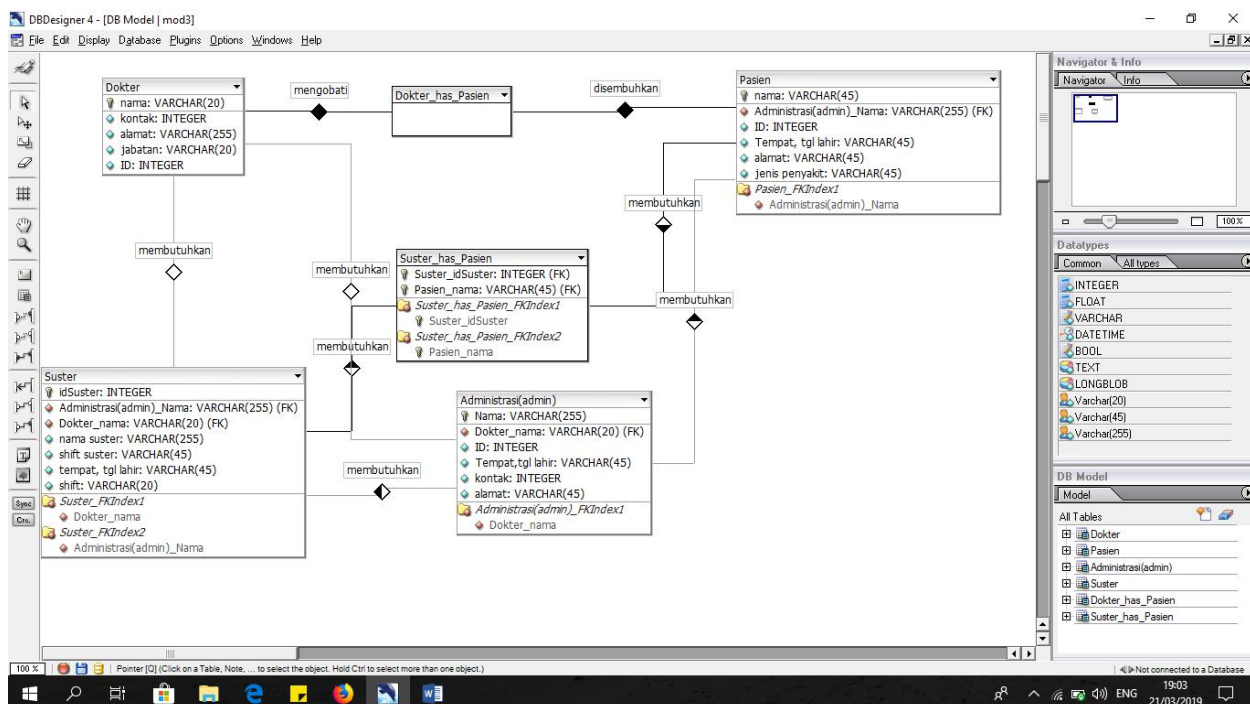
LATIHAN 1



LATIHAN 2



TUGAS



MODUL 4

PERBANKAN

```
Command Prompt - mysql -u root

C:\>cd xampp\mysql\bin

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

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

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

MariaDB [(none)]> create database Perbankan;
Query OK, 1 row affected (0.07 sec)

MariaDB [(none)]> use Perbankan
Database changed
MariaDB [Perbankan]> create table Nasabah (
  -> id_nasabah INTEGER PRIMARY KEY,
  -> nama_nasabah VARCHAR(45) NOT NULL,
  -> alamat_nasabah VARCHAR(225) NOT NULL
  -> );
Query OK, 0 rows affected (0.30 sec)

MariaDB [Perbankan]> create table cabang_bank (
  -> kode_cabang VARCHAR(20) PRIMARY KEY,
  -> nama_cabang VARCHAR(45) UNIQUE NOT NULL,
  -> alamat_cabang VARCHAR(225) NOT NULL
  -> );
Query OK, 0 rows affected (0.57 sec)

MariaDB [Perbankan]> create table Rekening (
  -> no_rekening INTEGER PRIMARY KEY,
  -> kode_cabangFK VARCHAR(20) REFERENCES cabang_bank(kode_cabang)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> pin VARCHAR(20) DEFAULT '1234' NOT NULL,
  -> saldo INTEGER DEFAULT 0 NOT NULL
  -> );
Query OK, 0 rows affected (0.28 sec)

MariaDB [Perbankan]> create table Transaksi (
  -> no_nasabahFK INTEGER REFERENCES nasabah(id_nasabah)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> no_rekeningFK INTEGER REFERENCES rekening(no_rekening)
```

```
Command Prompt - mysql -u root
Query OK, 0 rows affected (0.28 sec)

MariaDB [Perbankan]> create table Transaksi (
  -> no_nasabahFK INTEGER REFERENCES nasabah(id_nasabah)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> no_rekeningFK INTEGER REFERENCES rekening(no_rekening)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> jenis_transaksi VARCHAR(20) DEFAULT 'debit' NOT NULL,
  -> tanggal DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  -> jumlah INTEGER NOT NULL CHECK (jumlah>=20000)
  -> );
Query OK, 0 rows affected (0.29 sec)

MariaDB [Perbankan]> create table nasabah_has_rekening (
  -> id_nasabahFK INTEGER REFERENCES nasabah(id_nasabah)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> no_rekeningFK INTEGER REFERENCES rekening(id_rekening)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> PRIMARY KEY (id_nasabahFK, no_rekeningFK)
  -> );
Query OK, 0 rows affected (0.23 sec)

MariaDB [Perbankan]> show tables;
+-----+
| Tables_in_perbankan |
+-----+
| cabang_bank          |
| nasabah              |
| nasabah_has_rekening |
| rekening             |
| transaksi            |
+-----+
5 rows in set (0.07 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(225) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.12 sec)
```

DATA KULIAH

```
Command Prompt - mysql -u root

MariaDB [Perbankan]> use kuliah
Database changed
MariaDB [kuliah]> create table Mahasiswa;
ERROR 1113 (42000): A table must have at least 1 column
MariaDB [kuliah]> create table mahasiswa (
  -> nim_mahasiswa CHAR(20) PRIMARY KEY,
  -> nama_mahasiswa VARCHAR(40) NOT NULL,
  -> alamat_mahasiswa VARCHAR(225) NOT NULL
  -> );
Query OK, 0 rows affected (0.17 sec)

MariaDB [kuliah]> create table dosen (
  -> nip_dosen CHAR(20) PRIMARY KEY,
  -> nama_dosen VARCHAR(40) NOT NULL,
  -> alamat_dosen VARCHAR(225) NOT NULL
  -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corr
5
MariaDB [kuliah]> create table dosen (
  -> nip_dosen CHAR(20) PRIMARY KEY,
  -> nama_dosen VARCHAR(40) NOT NULL,
  -> alamat_dosen VARCHAR(225) NOT NULL
  -> );
Query OK, 0 rows affected (0.39 sec)

MariaDB [kuliah]> create table mata_kuliah (
  -> id_mk INTEGER(20) PRIMARY KEY,
  -> nama_mk VARCHAR(40) NOT NULL
  -> );
Query OK, 0 rows affected (0.19 sec)

MariaDB [kuliah]> create table ruang_kelas (
  -> id_ruangkelas INTEGER(20) PRIMARY KEY,
  -> nama_ruangkelas VARCHAR(40) NOT NULL
  -> );
Query OK, 0 rows affected (0.27 sec)

MariaDB [kuliah]> show tables;
+-----+
| Tables_in_kuliah |
+-----+
| dosen            |
| mahasiswa        |
+-----+
```



```
Command Prompt - mysql -u root

dosen
mahasiswa
mata_kuliah
ruang_kelas
4 rows in set (0.00 sec)

MariaDB [kuliah]> describe dosen;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip_dosen | char(20) | NO | PRI | NULL | |
| nama_dosen | varchar(40) | NO | | NULL | |
| alamat_dosen | varchar(225) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

MariaDB [kuliah]> describe mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim_mahasiswa | char(20) | NO | PRI | NULL | |
| nama_mahasiswa | varchar(40) | NO | | NULL | |
| alamat_mahasiswa | varchar(225) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.06 sec)

MariaDB [kuliah]> describe mata_kuliah;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_mk | int(20) | NO | PRI | NULL | |
| nama_mk | varchar(40) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

MariaDB [kuliah]> describe ruang_kelas;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_ruangkelas | int(20) | NO | PRI | NULL | |
| nama_ruangkelas | varchar(40) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

PERTOKOAN

```
Command Prompt - mysql -u root

MariaDB [kuliah]> use pertokoan;
ERROR 1049 (42000): Unknown database 'pertokoan'
MariaDB [kuliah]> use pertokoan
ERROR 1049 (42000): Unknown database 'pertokoan'
MariaDB [kuliah]> create database pertokoan;
Query OK, 1 row affected (0.00 sec)

MariaDB [kuliah]> use pertokoan
Database changed
MariaDB [pertokoan]>
MariaDB [pertokoan]> create table pemilik(
-> id_pemilik INTEGER PRIMARY KEY,
-> nama_pemilik VARCHAR(20) NOT NULL,
-> alamat_pemilik VARCHAR(225) NOT NULL
-> );
Query OK, 0 rows affected (0.23 sec)

MariaDB [pertokoan]> create table toko(
-> id_toko INTEGER PRIMARY KEY,
-> nama_toko VARCHAR(20) NOT NULL,
-> alamat_toko VARCHAR(225) NOT NULL
-> );
Query OK, 0 rows affected (0.61 sec)

MariaDB [pertokoan]> create table pembeli(
-> id_pembeli INTEGER PRIMARY KEY,
-> nama_pembeli VARCHAR(20) NOT NULL,
-> alamat_pembeli VARCHAR(225) NOT NULL
-> );
Query OK, 0 rows affected (0.18 sec)

MariaDB [pertokoan]> create table barang (
-> id_barang INTEGER PRIMARY KEY,
-> nama_barang VARCHAR(20) NOT NULL,
-> jenis_barang VARCHRA(40) NOT NULL,
-> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that correspond
NOT NULL,
)' at line 4
MariaDB [pertokoan]> create table barang (
-> id_barang INTEGER PRIMARY KEY,
-> nama_barang VARCHAR(20) NOT NULL,
-> jenis_barang VARCHRA(40) NOT NULL
-> );
```

```

) at line 4
MariaDB [pertokoan]> show tables;
+-----+
| Tables_in_pertokoan |
+-----+
| pembeli            |
| pemilik            |
| toko                |
+-----+
3 rows in set (0.00 sec)

MariaDB [pertokoan]> describe pemilik
+-----+
>
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| id_pemilik | int(11)       | NO   | PRI | NULL    |       |
| nama_pemilik | varchar(20)   | NO   |     | NULL    |       |
| alamat_pemilik | varchar(225) | NO   |     | NULL    |       |
+-----+
3 rows in set (0.02 sec)

MariaDB [pertokoan]> describe toko;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| id_toko    | int(11)       | NO   | PRI | NULL    |       |
| nama_toko  | varchar(20)   | NO   |     | NULL    |       |
| alamat_toko | varchar(225) | NO   |     | NULL    |       |
+-----+
3 rows in set (0.02 sec)

MariaDB [pertokoan]> describe pembeli;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| id_pembeli | int(11)       | NO   | PRI | NULL    |       |
| nama_pembeli | varchar(20)   | NO   |     | NULL    |       |
| alamat_pembeli | varchar(225) | NO   |     | NULL    |       |
+-----+
3 rows in set (0.02 sec)

MariaDB [pertokoan]>

```



MODUL 5

Database “Kuliah”

Entity Dosen

```
MariaDB [kuliah]> insert into dosen values
-> (1001, "Sutopo", "Jl. Jendral Sudorman 12"),
-> (1002, "Maryati", "Jl. MT. Haryono31"),
-> (1003, "Suparman", "Jl. Hasanudin 81"),
-> (1004, "Kartika Padmasari", "Jl. Manggis 12"),
-> (1005, "Budi Eko Prayogo", "Jl. Kantil 30");
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [kuliah]> select*from dosen;
+-----+-----+-----+
| nip_dosen | nama_dosen | alamat_dosen |
+-----+-----+-----+
| 1001      | Sutopo     | Jl. Jendral Sudorman 12 |
| 1002      | Maryati    | Jl. MT. Haryono31      |
| 1003      | Suparman   | Jl. Hasanudin 81       |
| 1004      | Kartika Padmasari | Jl. Manggis 12        |
| 1005      | Budi Eko Prayogo | Jl. Kantil 30         |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Entity Mahasiswa

```
MariaDB [kuliah]> insert into mahasiswa values
-> (170081, "Intan", "Surakarta"),
-> (170082, "Heni", "Kalimantan"),
-> (170083, "Dila", "Palembang"),
-> (170084, "Sari", "Sragen"),
-> (170085, "Lia", "Ngawi");
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [kuliah]> select*from mahasiswa;
+-----+-----+-----+
| nim_mahasiswa | nama_mahasiswa | alamat_mahasiswa |
+-----+-----+-----+
| 170081        | Intan          | Surakarta         |
| 170082        | Heni           | Kalimantan         |
| 170083        | Dila           | Palembang         |
| 170084        | Sari           | Sragen            |
| 170085        | Lia            | Ngawi             |
| L200170083    | Palistya       | Sragen            |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Entity Mata_Kuliah

```
MariaDB [kuliah]> insert into mata_kuliah values
-> (001, "Basis Data"),
-> (002, "Pemograman Visual"),
-> (003, "Logika dan Himpunan"),
-> (004, "Aljabar Linier"),
-> (005, "Matematika Diskrit");
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
MariaDB [kuliah]> select*from mata_kuliah;
+-----+-----+
| id_mk | nama_mk |
+-----+-----+
| 1     | Basis Data |
| 2     | Pemograman Visual |
| 3     | Logika dan Himpunan |
| 4     | Aljabar Linier |
| 5     | Matematika Diskrit |
+-----+-----+
5 rows in set (0.00 sec)
```

Entity Ruang_Kelas

```
MariaDB [kuliah]> insert into ruang_kelas values
-> (1, "Mawar"),
-> (2, "Melati"),
-> (3, "Dahlia"),
-> (4, "Anggrek"),
-> (5, "Tulip");
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
MariaDB [kuliah]> select*from ruang_kelas;
+-----+-----+
| id_ruangkelas | nama_ruangkelas |
+-----+-----+
| 1             | Mawar           |
| 2             | Melati          |
| 3             | Dahlia          |
| 4             | Anggrek         |
| 5             | Tulip           |
+-----+-----+
5 rows in set (0.00 sec)
```

```
MariaDB [kuliah]>
```


Database “Pertokoan”

Entity Pembeli

```
MariaDB [kuliaah]> use pertokoan;
Database changed
MariaDB [pertokoan]> insert into pembeli values
  -> (0011, "Andi", "Jl. Pahlawan 20"),
  -> (0012, "Mela", "Jl. Tulip 11"),
  -> (0013, "Doni", "Jl. Jambu 01"),
  -> (0014, "Anto", "Jl. Gatot Subroto 34"),
  -> (1125, "Ria", "Jl. Sukarno 28");
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [pertokoan]> select*from pembeli;
+-----+-----+-----+
| id_pembeli | nama_pembeli | alamat_pembeli |
+-----+-----+-----+
|          11 | Andi        | Jl. Pahlawan 20 |
|          12 | Mela        | Jl. Tulip 11    |
|          13 | Doni        | Jl. Jambu 01    |
|          14 | Anto        | Jl. Gatot Subroto 34 |
|         1125 | Ria         | Jl. Sukarno 28  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Entity Pemilik

```
MariaDB [pertokoan]> insert into pemilik values
  -> (0098761239876, "Palistya Nesty ", "Jl. Jendral Sudirman 55");
Query OK, 1 row affected, 1 warning (0.05 sec)

MariaDB [pertokoan]> select*from pemilik;
+-----+-----+-----+
| id_pemilik | nama_pemilik | alamat_pemilik |
+-----+-----+-----+
| 2147483647 | Palistya Nesty | Jl. Jendral Sudirman 55 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Entity Toko

```
MariaDB [pertokoan]> insert into toko values
  -> (12345, "Makmur Sejahtera ", "Jl. Jendral Sudirman 55");
Query OK, 1 row affected (0.04 sec)

MariaDB [pertokoan]> select*from toko;
+-----+-----+-----+
| id_toko | nama_toko | alamat_toko |
+-----+-----+-----+
|    12345 | Makmur Sejahtera | Jl. Jendral Sudirman 55 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

MODUL 6

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

```
MariaDB [perbankannew]> 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
Kartika Padmasari	Jl. Manggis 15
Maryati	Jl. MT. Haryono 31
Sari Murti	Jl. Pangandaran 11
Satria Eka Jaya	Jl. Slamet Riyadi 45
Suparman	Jl. Hasanudin 81
Sutopo	Jl. Jendral Sudirman 12

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

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

```
MariaDB [perbankannew]> 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	200000
kredit	100000
kredit	200000
kredit	50000
kredit	150000
kredit	100000
kredit	150000
kredit	100000
kredit	100000
kredit	200000

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

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

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

```
MariaDB [perbankannew]> 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  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

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

```
MariaDB [perbankannew]> select nama_nasabah, alamat_nasabah from nasabah where nama_nasabah like "Su%";
+-----+-----+
| nama_nasabah | alamat_nasabah |
+-----+-----+
| Sutopo        | Jl. Jendral Sudirman 12 |
| Superman      | Jl. Hasanudin 81       |
+-----+-----+
2 rows in set (0.00 sec)
```

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 | Superman    | 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 | Superman    | 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 jumlah transaksi untuk nasabah yang bernama akhiran 'Kartika Padmawati' untuk masing-masing jenis transaksi.

```
MariaDB [perbankanew]> select jenis_transaksi, jumlah as 'Rp', count(jumlah) from transaksi
-> , nasabah where nasabah.id_nasabah = transaksi.id_nasabahFK and nasabah.nama_nasabah =
-> 'Kartika Padmasari' group by jenis_transaksi;
```

jenis_transaksi	Rp	count(jumlah)
debit	100000	2
kredit	20000	3

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

2. Berapa total saldo yang dimiliki Maryati

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

sum(saldo)
750000

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

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

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

count(jumlah)
6
6
12
22
4

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

4. Tampilkan nama nasabah dan jumlah saldo yang memiliki saldo antara rp. 500.000 sampai rp. 200.000

```
MariaDB [perbankanew]> 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	6700000

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

5. Tampilkan nama nasabha, tanggal transaksi dan jumlah transaksi dalam rupiah dimana jumlah transaksi diatas Rp. 100.000 dan urutkan jumlah transaksi dari besar ke kecil.

```
MariaDB [perbankannew]> 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
Budi Eko Prayogo	2019-01-03 00:00:00	200000
Budi Eko Prayogo	2019-01-11 00:00:00	200000
Budi Eko Prayogo	2009-12-05 00:00:00	200000
Maryati	2009-11-28 00:00:00	200000
Kartika Padmasari	2009-11-20 00:00:00	200000
Budi Murtono	2019-01-06 00:00:00	150000
Canka Lokananta	2009-11-15 00:00:00	150000
Budi Eko Prayogo	2019-01-03 00:00:00	150000

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


MODUL 9

1. Buatlah user baru sesuai pada nama Anda masing-masing.

```
Command Prompt - mysql -u paw -p
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ASUS-USER>cd\

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

C:\xampp\mysql\bin>mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 29
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 'paw'@'localhost' identified by '123';
Query OK, 0 rows affected (0.00 sec)

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

2. Beri privilege untuk user tersebut dengan ketentuan sebagai berikut ini:
 - a. Dapat melakukan INSERT, UPDATE, dan DELETE pada tabel nasabah.
 - b. Hanya dapat melakukan SELECT pada tabel cabang_bank

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

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

MariaDB [mysql]> exit;
Bye

C:\xampp\mysql\bin>mysql -u paw -p
Enter password: ***
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
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.
```

3. Coba lakukan kasus berikut ini dengan login sebagai user yang dibuat
 - a. Lakukan perintah SELECT pada tabel nasabah.
 - b. Lakukan perintah INSERT pada tabel nasabah.
 - c. Lakukan perintah INSERT pada tabel cabang_bank.

```
MariaDB [(none)]> use perbankan;
Database changed
MariaDB [perbankan]> select * from nasabah;
ERROR 1142 (42000): SELECT command denied to user 'paw'@'localhost' for table 'nasabah'
MariaDB [perbankan]> insert into nasabah(id_nasabah, nama_nasabah, alamat_nasabah) values (22, 'Paw', 'Wonogiri');
Query OK, 1 row affected (0.09 sec)

MariaDB [perbankan]> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values ('BRUA', 'Bank Rut Unit Aceh', 'Wonogiri');
ERROR 1142 (42000): INSERT command denied to user 'paw'@'localhost' for table 'cabang_bank'
MariaDB [perbankan]>
```

MODUL 10

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 link_mahasiswa_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

MySQL dan Python

Tugas

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

a. Perintah insert

```
*ekstensi.py - C:/Users/dessyna/AppData/Local/Programs/Python/Python37-32/ekstensi.py (3.7.3)*
File Edit Format Run Options Window Help
from datetime import date,datetime,timedelta
import mysql.connector

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

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

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

51 rows in set (0.00 sec)

b. Perintah update

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

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

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

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

51 rows in set (0.00 sec)

c. Perintah delete

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

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

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

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

50 rows in set (0.00 sec)

2. Buatlah kode program pyhton untuk mendapatkan :

a. Data nasabah

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

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

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

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

b. Data nasabah yang melakukan transaksi antara oktober sampai desember

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

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

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

```
ID: 3 Nama: Superman 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: Superman 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: Superman 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: Superman 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: Superman Alamat: Jl. Hasanudin 81
ID: 2 Nama: Maryati Alamat: Jl. MT. Haryono 31
ID: 3 Nama: Superman 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
>>>
```

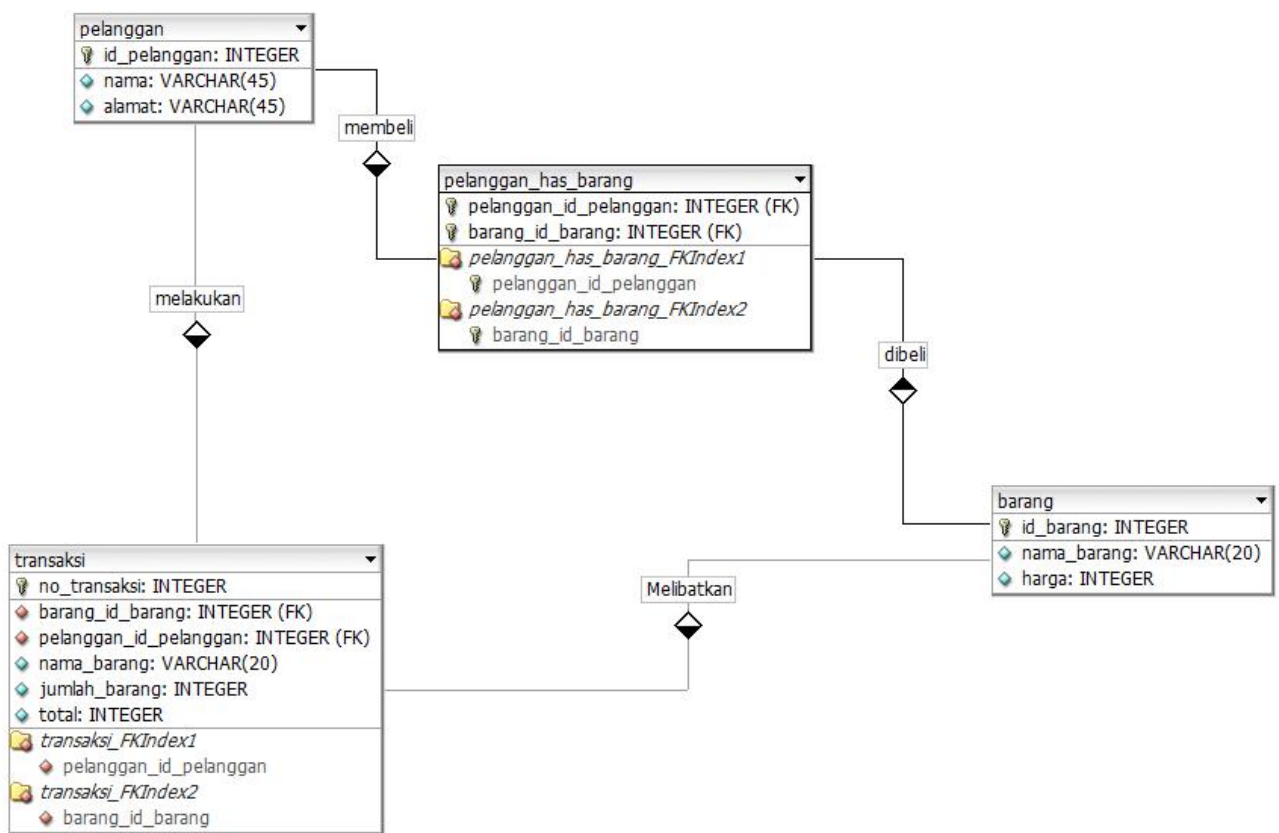

MODUL 12

1. Buatlah sebuah aplikasi (berbasis Python atau berbasis Java) pada studi kasus berikut ini :


Tiket Kereta Api, dengan minimal entitas yang ada, antara lain :'

- Barang
- Pelanggan
- Transaksi

2. Pada aplikasi tersebut, buatlah ER Diagram-nya



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



ID	Nama	Harga
1	pensil	2000000
2	ytfvj	50000
3	tas	20000
5	gfffy	20000
2789	hunga	25000

a. Proses View (Select)

```
try{
    String sql = "SELECT * FROM barang";
    java.sql.Connection conn = (Connection)fig.configDB();
    java.sql.Statement stm = conn.createStatement();
    java.sql.ResultSet res = stm.executeQuery(sql);

    while(res.next()){
        model.addRow(new Object[]{res.getString(1),res.getString(2),res.getString(3)});
    }
    tabelbarang.setModel(model);
}catch (SQLException e){
    System.out.println("Error : " + e.getMessage());
}
```

b. Insert (memasukkan data)

```
] private void insertActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        String sql = "INSERT INTO barang VALUES ('"+id2.getText()+"', '"+nama2.getText()+"', '"+harga2.getText()+"')";  
        java.sql.Connection conn=(Connection)fig.configDB();  
        java.sql.PreparedStatement pst=conn.prepareStatement(sql);  
        pst.execute();  
        JOptionPane.showMessageDialog(null, "Penyimpanan Data Berhasil");  
    } catch (SQLException | HeadlessException e) {  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
}
```

c. Update (mengubah data)

```
] private void updateActionPerformed(java.awt.event.ActionEvent evt) {  
    String sql = "UPDATE barang SET id='"+ id2.getText()+"',nama='"+nama2.getText()+"',harga='"+harga2.getText()+"'WHERE  
    java.sql.Connection conn = (Connection)fig.configDB();  
    java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
    pstmt.execute();  
    JOptionPane.showMessageDialog(null, "Edit Data Berhasil...");  
} catch (HeadlessException | SQLException e){  
    JOptionPane.showMessageDialog(this, e.getMessage());  
}  
tampilkan_data();  
kosongkan_form();
```

d. Delete (menghapus data)

```
] private void deleteActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        String sql = "DELETE FROM barang WHERE id='"+id2.getText()+"'";  
        java.sql.Connection conn = (Connection)fig.configDB();  
        java.sql.PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.execute();  
        JOptionPane.showMessageDialog(null, "Hapus Data Berhasil...");  
    } catch (HeadlessException | SQLException e){  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    }  
    tampilkan_data();  
    kosongkan_form(); // TODO add your handling code here:  
}
```

