

Nama : Fitri Cahya Kusumawati
NIM : L200170110
Kelas : D
Modul : 1 – 12

Laporan Praktikum 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!

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.

Contoh :

Perbankan yang memiliki cabang di setiap kotanya. Perusahaan bank tersebut hanya memiliki 1 database yang disimpan di server pusat, cabang – cabangnya terhubung melalui jaringan komputer untuk mengakses database yang terletak di server pusat tersebut.

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.
 - 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.
- Sistem File Manual : sistem yang digunakan untuk mengatasi semua permasalahan bisnis, menggunakan pengelolaan data secara tradisional dengan cara menyimpan record – record pada file yang terpisah yang di sebut juga sistem pemrosesan file.
 - Sistem basis data : sistem pemrosesan file ditinggalkan karena masih bersifat manual yang kemudian dikembangkan sistem pemrosesan dengan pendekatan basis data.
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 diisikan data berupa teks huruf, angka, dan simbol-simbol lain. Ukuran tipe data pada field NIM diatur sepanjang 10, artinya digit maksimum yang dapat dimasukkan ke dalam record adalah 10 digit. Jika data yang dimasukkan lebih dari 10 digit, maka digit ke-11 dan berikutnya tidak akan terbaca. Untuk pengaturan maksimum defaultnya sendiri adalah sepanjang 255 digit. Kemudian pada field NAMA, sama-sama diatur tipe data text, namun panjang ukurannya berbeda, yaitu 30, artinya kita dapat memasukkan digit maksimum ke dalam record yaitu sepanjang 30 digit, melebihi 30 data ke-31 dan seterusnya tidak akan terbaca. Pada field TANGGAL_LAHIR diatur tipe date/time yang berarti pada record dapat diisikan data berupa angka dan simbol.

Modul 2.

Tugas

Entities

- Mahasiswa : menyimpan semua data pribadi semua mahasiswa
- Dosen : menyimpan semua data pribadi semua dosen
- Mata_Kuliah : menyimpan informasi mata kuliah
- Ruang_Kelas : menyimpan informasi transaksi mahasiswa

Attributes

- Mahasiswa:

- NIM_mahasiswa : nomor identitas mahasiswa (varchar(10))PK
- nama_mahasiswa : nama lengkap mahasiswa (varchar(45))
- alamat_mahasiswa : alamat lengkap mahasiswa (varchar(100))

- Dosen :

- nip_dosen : nomor induk pegawai (integer)PK
- nama_dosen : nama lengkap dosen (varchar(45))
- alamat_dosen : alamat lengkap dosen (varchar(100))

- Mata_kuliah :

- Kode_matkul : kode identitas mata kuliah (varchar(10)) PK
- nama_matkul : nama mata kuliah (varchar(45))
- jumlah_sks : jumlah sks per 1 mata kuliah (integer)

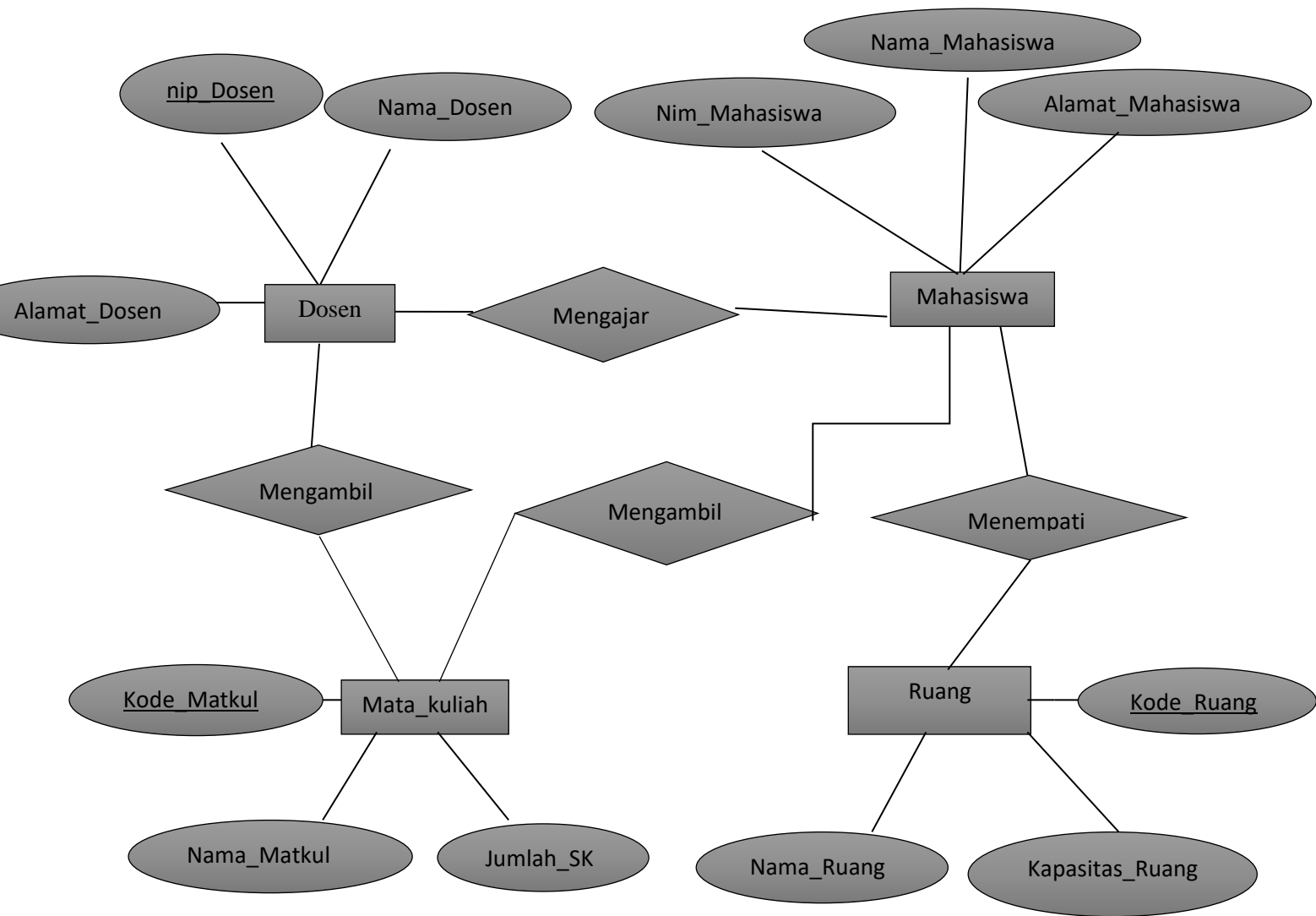
- Ruang_kelas :

- Kode_ruang : kode identitas ruangan (integer) PK
- nama_ruang : nama ruangan (varchar(50))
- kapasitas_ruang : jumlah maksimal mahasiswa dalam 1 ruang (integer)

	Mahasiswa	Dosen	Matkul	Kelas
Mahasiswa	-	n:n	n:n	1:1
Dosen		-	n:n	-
Matkul			-	-
Kelas				-

Hubungan

- **Dosen mengajar mahasiswa**
 - Tabel utama : **mahasiswa, dosen**
 - Tabel kedua : **mahasiswa_has_dosen**
 - Relationship : Many-to-many (n:n)
 - Attribute penghubung : **NIM_mahasiswa, nip_dosen** (FK **NIM_mahasiswa, nip_dosen** di **mahasiswa_has_dosen**)
- **Mahasiswa mengambil mata_kuliah**
 - Tabel utama : **mahasiswa, mata kuliah**
 - Tabel kedua : **mengambil**
 - Relationship : Many-to-many (n:n)
 - Attribute penghubung : **NIM_mahasiswa, kode_matkul** (FK **NIM_mahasiswa, Kode_matkul** di **mengambil**)
- **Mahasiswa menempati ruang kelas**
 - Tabel utama : **mahasiswa**
 - Tabel kedua : **ruang kelas**
 - Relationship : One-to-one(1:1)
 - Attribute penghubung : **NIM_mahasiswa**(FK **NIM_mahasiswa** di **ruang kelas**)
- **Dosen mengambil mata kuliah**
 - Tabel utama : **dosen, mata_kuliah**
 - Tabel kedua : **dosen_has_matakuliah**
 - Relationship : Many-to-Many(n:n)
 - Attribute penghubung : **nip_dosen, kode_matkul**(FK **nip_dosen, kode_matkul** di **dosen_has_mata_kuliah**)



Penjualan di apotek

Entities

- Pelanggan : menyimpan semua data pelanggan
- pegawai : menyimpan semua data pegawai
- obat : menyimpan informasi obat
- transaksi : menyimpan data transaksi

Attributes

- Pelanggan:

- id_pelanggan : nomor identitas pelanggan (integer)PK
- nama_pelanggan : nama lengkap pelanggan (varchar(45))
- alamat_pelanggan : alamat lengkap pelanggan (varchar(100))

- Pegawai :

- id_pegawai : nomor identitas pegawai (integer)PK
- nama_pegawai : nama lengkap pegawai (varchar(45))
- alamat_pegawai : alamat lengkap pegawai (varchar(100))

- obat :

- Kode_obat : kode obat (varchar(10)) PK
- nama_obat : nama obat (varchar(45))
- harga_obat : harga setiap obat (integer)

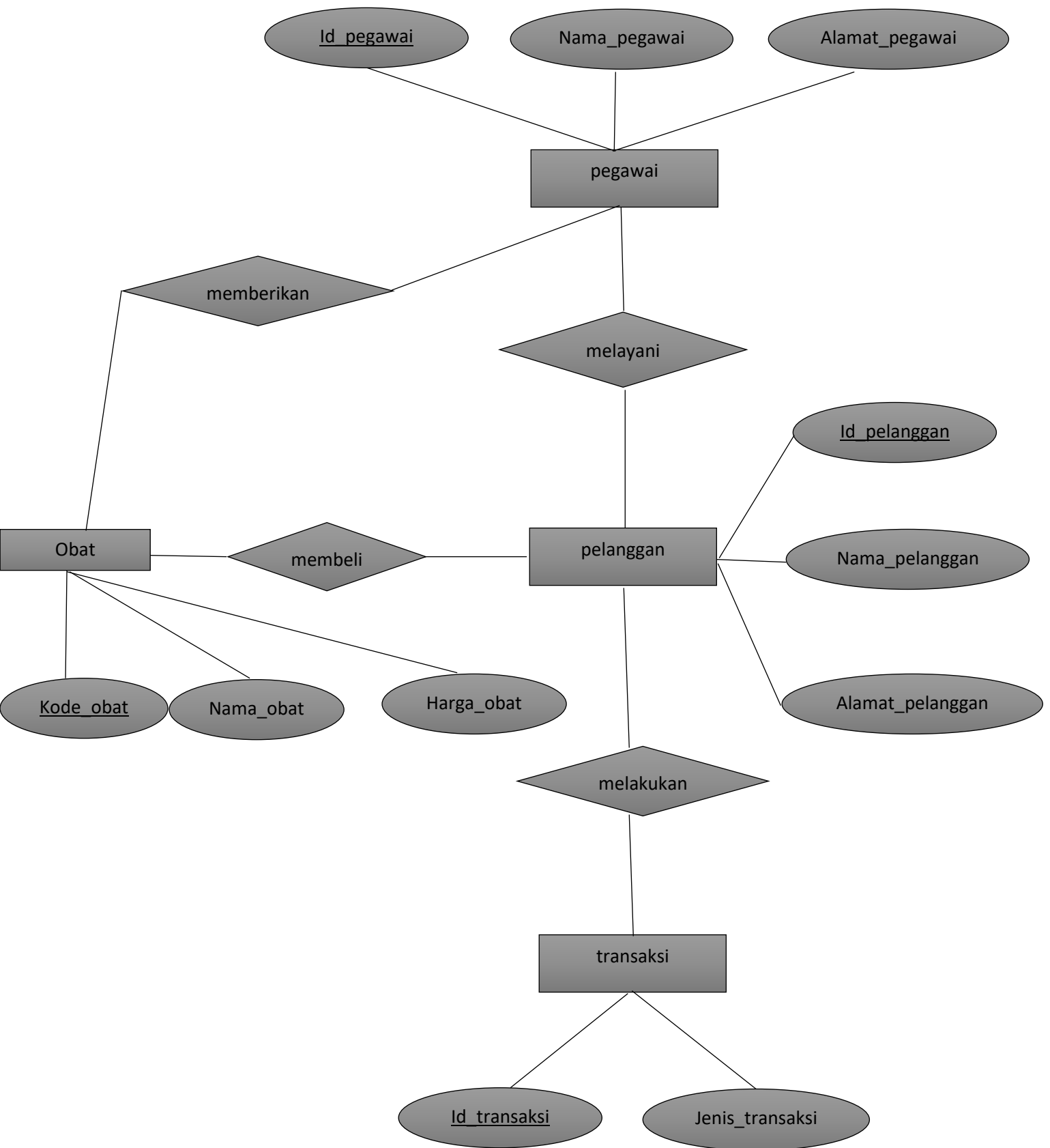
- transaksi :

- id_transaksi : id transaksi (integer) PK
- jenis_transaksi : jenis transaksi (varchar(45))

	Pelanggan	Pegawai	obat	Transaksi
Pelanggan	-	n:n	n:n	1:n
Pegawai		-	1:n	-
Obat			-	-
Transaksi				-

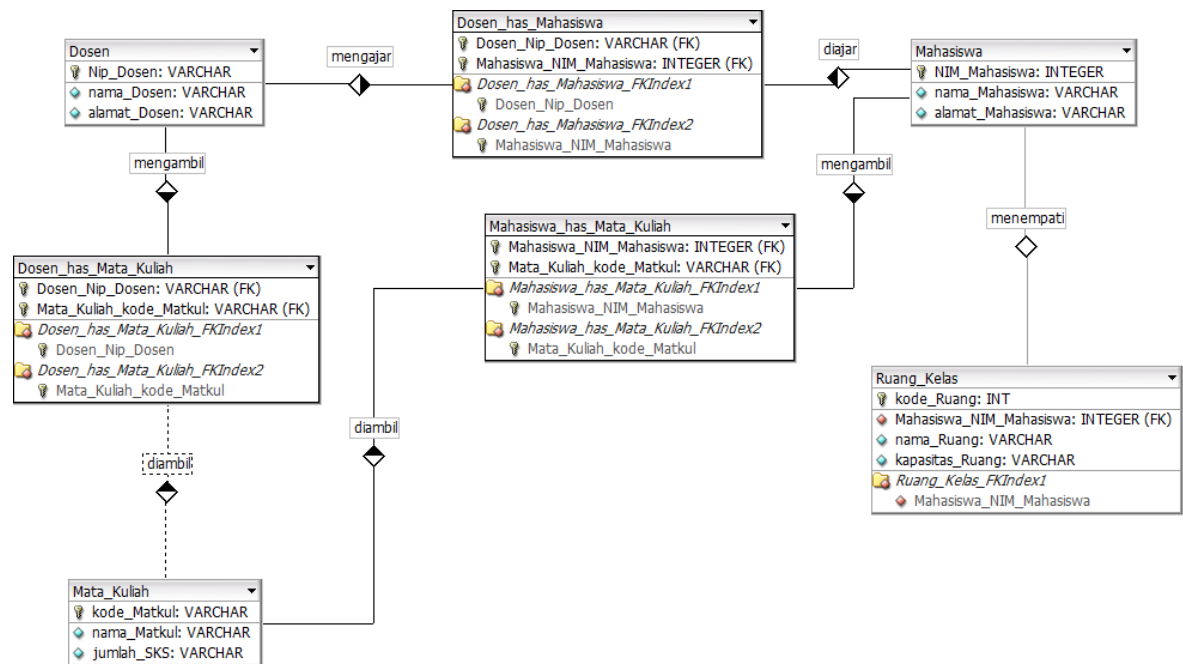
Hubungan

- **Pegawai melayani pelanggan**
 - Tabel utama : **pegawai, pelanggan**
 - Tabel kedua : **pelanggan_has_pegawai**
 - Relationship : Many-to-many (n:n)
 - Attribute penghubung : **id_pegawai, id_pelanggan** (FK id_pegawai, id_pelanggan di pelanggan_has_pegawai)
- **Pelanggan membeli obat**
 - Tabel utama : **pelanggan, obat**
 - Tabel kedua : **membeli**
 - Relationship : Many-to-many(n:n)
 - Attribute penghubung : **id_pelanggan, kode_obat**(FK id_pelanggan, kode_obat di membeli)
- **Pegawai memberikan Obat**
 - Tabel utama : **pegawai, obat**
 - Tabel kedua : **memberikan**
 - Relationship : One-to-many(1:n)
 - Attribute penghubung : **id_pegawai, kode_obat**(FK id_pegawai, kode_obat di memberikan)
- **Pelanggan melakukan Transaksi**
 - Tabel utama : **pelanggan, transaksi**
 - Tabel kedua : **melakukan**
 - Relationship : One-to-many(1:n)
 - Attribute penghubung : **id_pelanggan, id_transaksi**(FK id_pelanggan, id_transaksi di melakukan)

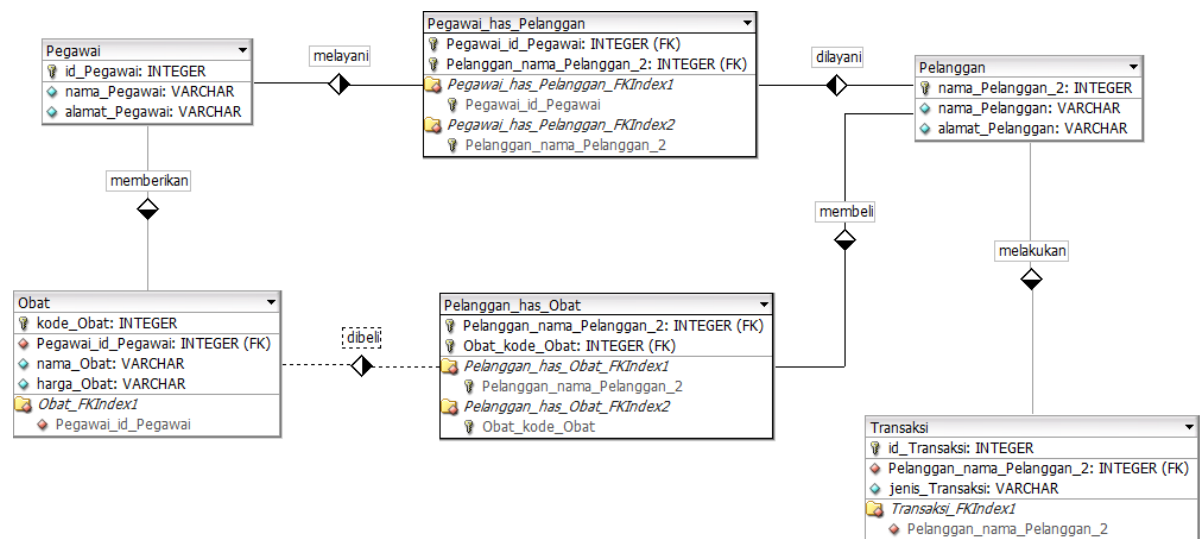


Modul 3.

1. Menggunakan program DB Designe untuk membuat ER Diagram pada modul 2 nomor 1.



2. Menggunakan program DB Designe untuk membuat ER Diagram pada modul 2 nomor 2.



Modul 4.

Tugas 1.

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

C:\Users\USER>cd..
C:\Users>cd..
C:\>cd C:/xampp/mysql/bin

C:\xampp\mysql\bin>login
'login' is not recognized as an internal or external command,
operable program or batch file.

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 28
Server version: 10.1.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

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

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

MariaDB [(none)]> use kuliah;
Database changed
MariaDB [kuliah]> CREATE TABLE dosen(
  -> nip_dosen INTEGER PRIMARY KEY,
  -> nama_dosen VARCHAR(45) NOT NULL,
  -> alamat_dosen VARCHAR(45) NOT NULL);
Query OK, 0 rows affected (0.19 sec)

MariaDB [kuliah]> CREATE TABLE matkul(
  -> kode_matkul INTEGER PRIMARY KEY,
  -> nama_matkul VARCHAR(45) NOT NULL,
  -> jumlah_matkul INTEGER DEFAULT 0 NOT NULL);
Query OK, 0 rows affected (0.18 sec)

MariaDB [kuliah]> CREATE TABLE mahasiswa(
  -> NIM_mahasiswa VARCHAR(20) PRIMARY KEY,
  -> nama_mahasiswa VARCHAR(45) NOT NULL,
  -> alamat_mahasiswa VARCHAR(45) NOT NULL,
  -> kode_ruangFK INTEGER REFERENCES ruang(kode_ruang) ON DELETE SET NULL ON UPDATE CASCADE);
Query OK, 0 rows affected (0.20 sec)
```

```
Command Prompt - mysql -u root -p
MariaDB [kuliah]> CREATE TABLE mahasiswa(
  -> NIM_mahasiswa VARCHAR(20) PRIMARY KEY,
  -> nama_mahasiswa VARCHAR(45) NOT NULL,
  -> alamat_mahasiswa VARCHAR(45) NOT NULL,
  -> kode_ruangFK INTEGER REFERENCES ruang(kode_ruang) ON DELETE SET NULL ON UPDATE CASCADE);
Query OK, 0 rows affected (0.20 sec)

MariaDB [kuliah]> CREATE TABLE ruang(
  -> kode_ruang INTEGER PRIMARY KEY,
  -> nama_ruang VARCHAR(45) NOT NULL,
  -> kapasitas_ruang INTEGER DEFAULT 0 NOT NULL);
Query OK, 0 rows affected (0.21 sec)

MariaDB [kuliah]> CREATE TABLE dosen_has_mahasiswa(
  -> nip_dosenFK INTEGER REFERENCES dosen(nip_dosen) ON DELETE SET NULL ON UPDATE CASCADE,
  -> NIM_mahasiswaFK VARCHAR(20) REFERENCES mahasiswa(NIM_mahasiswa) ON DELETE SET NULL ON UPDATE CASCADE);
Query OK, 0 rows affected (0.22 sec)

MariaDB [kuliah]> CREATE TABLE dosen_has_matkul(
  -> nip_dosenFK INTEGER REFERENCES dosen(nip_dosen) ON DELETE SET NULL ON UPDATE CASCADE,
  -> kode_matkulFK INTEGER REFERENCES matkul(kode_matkul) ON DELETE SET NULL ON UPDATE CASCADE);
Query OK, 0 rows affected (0.21 sec)

MariaDB [kuliah]> CREATE TABLE mahasiswa_has_matkul(
  -> NIM_mahasiswaFK VARCHAR(20) REFERENCES mahasiswa(NIM_mahasiswa) ON DELETE SET NULL ON UPDATE CASCADE,
  -> kode_matkulFK INTEGER REFERENCES matkul(kode_matkul) ON DELETE SET NULL ON UPDATE CASCADE);
Query OK, 0 rows affected (0.16 sec)

MariaDB [kuliah]> show tables;
+-----+
| Tables_in_kuliah |
+-----+
| dosen             |
| dosen_has_mahasiswa |
| dosen_has_matkul  |
| mahasiswa         |
| mahasiswa_has_matkul |
| matkul            |
| ruang             |
+-----+
7 rows in set (0.00 sec)
```

```
Command Prompt - mysql -u root -p

MariaDB [kuliah]> describe mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| NIM_mahasiswa | varchar(20) | NO | PRI | NULL | |
| nama_mahasiswa | varchar(45) | NO | | NULL | |
| alamat_mahasiswa | varchar(45) | NO | | NULL | |
| kode_ruangFK | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

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

MariaDB [kuliah]> describe ruang;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_ruang | int(11) | NO | PRI | NULL | |
| nama_ruang | varchar(45) | NO | | NULL | |
| kapasitas_ruang | int(11) | NO | | 0 | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [kuliah]> describe matkul;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode_matkul | int(11) | NO | PRI | NULL | |
| nama_matkul | varchar(45) | NO | | NULL | |
| jumlah_matkul | int(11) | NO | | 0 | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [kuliah]>
```

localhost / 127.0.0.1 / kuliah | php: X

localhost/phpmyadmin/db_structure.php?db=&db=kuliah

Server: 127.0.0.1 > Database: kuliah

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
dosen	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
dosen_has_mahasiswa	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
dosen_has_matkul	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
mahasiswa	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
mahasiswa_has_matkul	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
matkul	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
ruang	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K1B	-
7 tables	Sum	0	InnoDB	latin1_swedish_ci	112 K1B	0 B

☐ Check all With selected:

Print Data dictionary

Create table

Name: Number of columns:

Go

Console

Tugas 2.

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

C:\Users\USER>cd..
C:\Users>cd..
C:\>cd C:/xampp/mysql/bin
C:\xampp\mysql\bin>login
'login' is not recognized as an internal or external command,
operable program or batch file.
C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.28-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database Apotek;
ERROR 1007 (HY000): Can't create database 'apotek'; database exists
MariaDB [(none)]> create database Apotek1;
Query OK, 1 row affected (0.04 sec)

MariaDB [(none)]> use Apotek1;
Database changed
MariaDB [Apotek1]> CREATE TABLE pegawai(
  -> id_pegawai INTEGER PRIMARY KEY,
  -> nama_pegawai VARCHAR(45) NOT NULL,
  -> alamat_pegawai VARCHAR(45) NOT NULL);
Query OK, 0 rows affected (0.19 sec)

MariaDB [Apotek1]> CREATE TABLE pelanggan(
  -> id_pelanggan INTEGER PRIMARY KEY,
  -> nama_pelanggan VARCHAR(45) NOT NULL,
  -> alamat_pelanggan VARCHAR(45) NOT NULL);
Query OK, 0 rows affected (0.17 sec)

MariaDB [Apotek1]> CREATE TABLE obat(
  -> kode_obat INTEGER PRIMARY KEY,
  -> nama_obat VARCHAR(45) NOT NULL,
  -> harga_obat INTEGER DEFAULT 0 NOT NULL);
```

```
Command Prompt - mysql -u root -p
MariaDB [Apotek1]> CREATE TABLE pegawai_has_pelanggan(
  -> id_pegawaiFK INTEGER REFERENCES pegawai(id_pegawai) ON DELETE CASCADE ON UPDATE CASCADE,
  -> id_pelangganFK INTEGER REFERENCES pelanggan(id_pelanggan) ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.45 sec)

MariaDB [Apotek1]> CREATE TABLE pelanggan_has_obat(
  -> id_pelangganFK INTEGER REFERENCES pelanggan(id_pelanggan) ON DELETE CASCADE ON UPDATE CASCADE,
  -> kode_obatFK INTEGER REFERENCES obat(kode_obat) ON DELETE CASCADE ON UPDATE CASCADE);
Query OK, 0 rows affected (0.18 sec)

MariaDB [Apotek1]> show tables;
+-----+
| Tables_in_apotek1 |
+-----+
| obat              |
| pegawai            |
| pegawai_has_pelanggan |
| pelanggan          |
| pelanggan_has_obat |
+-----+
5 rows in set (0.00 sec)

MariaDB [Apotek1]> describe pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pegawai | int(11) | NO | PRI | NULL | |
| nama_pegawai | varchar(45) | NO | | NULL | |
| alamat_pegawai | varchar(45) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [Apotek1]> describe pelanggan;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_pelanggan | int(11) | NO | PRI | NULL | |
| nama_pelanggan | varchar(45) | NO | | NULL | |
| alamat_pelanggan | varchar(45) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

MariaDB [Apotek1]> describe obat
-> ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
```

```
Command Prompt - mysql -u root -p

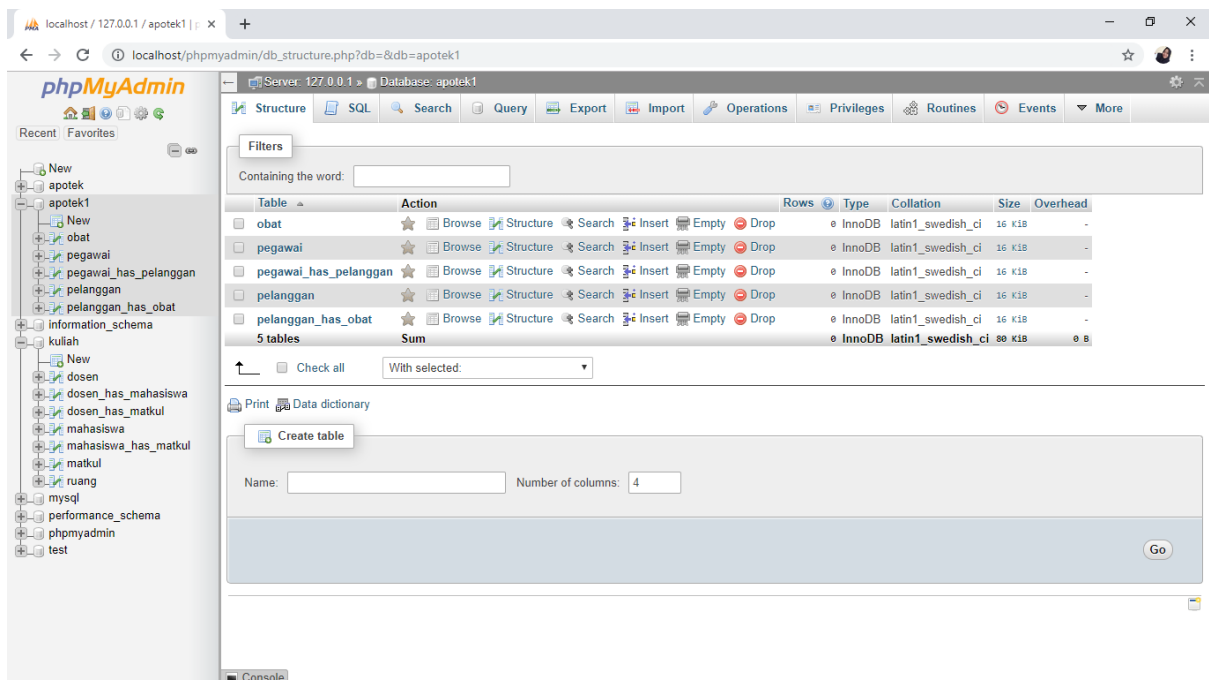
Tables_in_apotek1
+-----+
| obat |
| pegawai |
| pegawai_has_pelanggan |
| pelanggan |
| pelanggan_has_obat |
+-----+
5 rows in set (0.00 sec)

MariaDB [Apotek1]> describe pegawai;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id_pegawai | int(11) | NO | PRI | NULL | |
| nama_pegawai | varchar(45) | NO | | NULL | |
| alamat_pegawai | varchar(45) | NO | | NULL | |
+-----+
3 rows in set (0.01 sec)

MariaDB [Apotek1]> describe pelanggan;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id_pelanggan | int(11) | NO | PRI | NULL | |
| nama_pelanggan | varchar(45) | NO | | NULL | |
| alamat_pelanggan | varchar(45) | NO | | NULL | |
+-----+
3 rows in set (0.01 sec)

MariaDB [Apotek1]> describe obat
> ;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| kode_obat | int(11) | NO | PRI | NULL | |
| nama_obat | varchar(45) | NO | | NULL | |
| harga_obat | int(11) | NO | | 0 | |
+-----+
3 rows in set (0.01 sec)

MariaDB [Apotek1]>
```



Modul 5.

Tugas.

1. Nasabah

```
Command Prompt - mysql -u root -p
MariaDB [perbankan]> insert into nasabah(id_nasabah,nama_nasabah,alamat_nasabah)value
-> (11,'Slamet Budiyanto','Jl. Asia Afrika 112'),
-> (12,'Mursyid Efendi','Jl. Anggrek 3'),
-> (13,'Ajat Sudrajat','Jl. Peyum 21'),
-> (14,'Maulana Vikry','Jl. Nikisari 12'),
-> (15,'Sigit Prawiro','Jl. Inaja 13'),
-> (16,'Hendro Kartiko','Jl. Anggur 41'),
-> (17,'Nabila','Jl. Simaung 53'),
-> (18,'Nurhadi','Jl. Parangtritis 132'),
-> (19,'Kirana Murti','Jl. Tilang 19'),
-> (20,'Budi Sudarsono','Jl. Mundur 45');
Query OK, 10 rows affected (0.12 sec)
Records: 10 Duplicates: 0 Warnings: 0

MariaDB [perbankan]> 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 Eko 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
11	Slamet Budiyanto	Jl. Asia Afrika 112
12	Mursyid Efendi	Jl. Anggrek 3
13	Ajat Sudrajat	Jl. Peyum 21
14	Maulana Vikry	Jl. Nikisari 12
15	Sigit Prawiro	Jl. Inaja 13
16	Hendro Kartiko	Jl. Anggur 41
17	Nabila	Jl. Simaung 53
18	Nurhadi	Jl. Parangtritis 132
19	Kirana Murti	Jl. Tilang 19
20	Budi Sudarsono	Jl. Mundur 45

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

2. Cabang_bank

```
Command Prompt - mysql -u root -p
MariaDB [perbankan]> insert into cabang_bank(kode_cabang,nama_cabang,alamat_cabang)value
-> ('BRUA','Bank Rut Unit Anyer','Jl. H. Mukhlis 118'),
-> ('BRUN','Bank Rut Unit Ngawi','Jl. Pattimura 43'),
-> ('BRUG','Bank Rut Unit Gorontalo','Jl. Besar 25'),
-> ('BRUI','Bank Rut Unit Inggris','Jl. Supermin 23'),
-> ('BRUH','Bank Rut Unit Hungaria','Jl. Muh Yamin 41'),
-> ('BRUP','Bank Rut Unit Purwodadi','Jl. Sidan 12'),
-> ('BRUJ','Bank Rut Unit Jayapura','Jl. Bonjol 225'),
-> ('BRUR','Bank Rut Unit Rusia','Jl. Sutopo 33'),
-> ('BRUC','Bank Rut Unit Cilacap','Jl. H. Bagas 48'),
-> ('BRUL','Bank Rut Unit Lamongan','Jl. Pierre 4');
Query OK, 10 rows affected (0.09 sec)
Records: 10 Duplicates: 0 Warnings: 0

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

kode_cabang	nama_cabang	alamat_cabang
BRUA	Bank Rut Unit Anyer	Jl. H. Mukhlis 118
BRUB	Bank Rut Unit Boyolali	Jl. Ahmad Yani 45
BRUC	Bank Rut Unit Cilacap	Jl. H. Bagas 48
BRUG	Bank Rut Unit Gorontalo	Jl. Besar 25
BRUH	Bank Rut Unit Hungaria	Jl. Muh Yamin 41
BRUI	Bank Rut Unit Inggris	Jl. Supermin 23
BRUJ	Bank Rut Unit Jayapura	Jl. Bonjol 225
BRUK	Bank Rut Unit Klaten	Jl. Suparman 23
BRUL	Bank Rut Unit Lamongan	Jl. Pierre 4
BRUM	Bank Rut Unit Magelang	Jl. P.Tondan 63
BRUN	Bank Rut Unit Ngawi	Jl. Pattimura 43
BRUP	Bank Rut Unit Purwodadi	Jl. Sidan 12
BRUR	Bank Rut Unit Rusia	Jl. Sutopo 33
BRUS	Bank Rut Unit Surakarta	Jl. Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jl. Untung Surapati 12
BRUY	Bank Rut Unit Yogyakarta	Jl. Anggrek 21

```
16 rows in set (0.00 sec)

MariaDB [perbankan]>
```

3. Rekening

```
Command Prompt - mysql -u root -p

MariaDB [perbankan]> insert into rekening(no_rekening,kode_cabangFK,pin,saldo)value
-> (114,'BRUS','1123','310000'),
-> (115,'BRUJ','2123','320000'),
-> (116,'BRUH','3123','200000'),
-> (117,'BRUC','4123','330000'),
-> (118,'BRUD','5123','304000'),
-> (119,'BRUM','6123','380000'),
-> (120,'BRUA','7123','340000'),
-> (121,'BRUP','8123','350000'),
-> (122,'BRUL','9123','360000'),
-> (123,'BRUL','1023','370000');
Query OK, 10 rows affected (0.08 sec)
Records: 10 Duplicates: 0 Warnings: 0

MariaDB [perbankan]> 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 | BRIIS | 6666 | 3000000 |
| 107 | BRUS | 7777 | 1000000 |
| 108 | BRUB | 0000 | 5000000 |
| 109 | BRUB | 9999 | 0 |
| 110 | BRUY | 1234 | 550000 |
| 111 | BRIK | 4321 | 300000 |
| 112 | BRIK | 0123 | 300000 |
| 113 | BRUY | 8888 | 255000 |
| 114 | BRUS | 1123 | 310000 |
| 115 | BRUJ | 2123 | 320000 |
| 116 | BRUH | 3123 | 200000 |
| 117 | BRUC | 4123 | 330000 |
| 118 | BRUJ | 5123 | 304000 |
| 119 | BRUM | 6123 | 380000 |
| 120 | BRUA | 7123 | 340000 |
| 121 | BRUP | 8123 | 350000 |
| 122 | BRUL | 9123 | 360000 |
| 123 | BRUL | 1023 | 370000 |
+-----+-----+-----+-----+
23 rows in set (0.00 sec)

MariaDB [perbankan]>
```

4. Nasabah_has_rekening

```
Command Prompt - mysql -u root -p

MariaDB [perbankan]> insert into nasabah_has_rekening (id_nasabahFK,no_rekeningFK)value
-> (11,104),
-> (12,103),
-> (13,105),
-> (13,106),
-> (14,101),
-> (14,107),
-> (15,102),
-> (15,107),
-> (16,109),
-> (17,109),
-> (18,111),
-> (19,110),
-> (20,113),
-> (18,112),
-> (20,108);
Query OK, 15 rows affected (0.10 sec)
Records: 15 Duplicates: 0 Warnings: 0

MariaDB [perbankan]> 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 |
| 13 | 105 |
| 13 | 106 |
| 14 | 101 |
| 14 | 107 |
| 15 | 102 |
| 15 | 107 |
| 16 | 109 |
| 17 | 109 |
| 18 | 111 |
| 18 | 112 |
| 19 | 110 |
| 20 | 108 |
+-----+-----+
```

5. Transaksi

```
Command Prompt - mysql -u root -p
```

no_transaksi	id_nasabahFK	no_rekeningFK	jenis_transaksi	tanggal	jumlah
1	3	105	debit	2009-11-10 00:00:00	50000
2	2	103	debit	2009-11-10 00:00:00	40000
3	4	101	kredit	2009-11-12 00:00:00	20000
4	3	106	debit	2009-11-13 00:00:00	50000
5	5	107	kredit	2009-11-13 00:00:00	30000
6	1	104	kredit	2009-11-15 00:00:00	200000
7	9	110	kredit	2009-11-15 00:00:00	150000
8	5	102	debit	2009-11-16 00:00:00	20000
9	3	105	kredit	2009-11-18 00:00:00	50000
10	4	107	debit	2009-11-19 00:00:00	100000
11	2	103	debit	2009-11-19 00:00:00	100000
12	1	104	debit	2009-11-19 00:00:00	50000
13	4	107	kredit	2009-11-20 00:00:00	200000
14	3	105	debit	2009-11-21 00:00:00	40000
15	1	104	kredit	2009-11-22 00:00:00	100000
16	4	101	kredit	2009-11-22 00:00:00	20000
17	2	103	debit	2009-11-22 00:00:00	50000
18	5	102	debit	2009-11-25 00:00:00	50000
19	10	108	debit	2009-11-26 00:00:00	100000
20	3	106	kredit	2009-11-27 00:00:00	50000
21	2	103	kredit	2009-11-28 00:00:00	200000
22	3	105	kredit	2009-11-28 00:00:00	100000
23	5	102	debit	2009-11-30 00:00:00	20000
24	1	104	debit	2009-12-01 00:00:00	50000
25	2	103	debit	2009-12-02 00:00:00	40000
26	4	101	debit	2009-12-04 00:00:00	50000
27	2	103	kredit	2009-12-05 00:00:00	100000
28	5	102	kredit	2009-12-05 00:00:00	200000
29	7	109	debit	2009-12-05 00:00:00	100000
30	9	110	debit	2009-12-06 00:00:00	20000
31	3	105	debit	2009-12-07 00:00:00	40000
32	1	104	kredit	2009-12-07 00:00:00	100000
33	4	101	kredit	2009-12-08 00:00:00	20000
34	2	103	debit	2009-12-08 00:00:00	50000
35	5	102	debit	2009-12-09 00:00:00	50000
36	10	108	debit	2009-12-10 00:00:00	100000
37	3	106	kredit	2009-12-12 00:00:00	50000
38	2	103	kredit	2009-12-12 00:00:00	200000
39	3	105	kredit	2009-12-13 00:00:00	100000
40	5	102	debit	2009-12-14 00:00:00	20000
41	1	104	debit	2009-12-15 00:00:00	50000
42	1	104	debit	2009-12-16 00:00:00	50000
43	2	103	debit	2009-12-17 00:00:00	40000
44	4	101	debit	2009-12-18 00:00:00	50000
45	2	103	kredit	2009-12-19 00:00:00	100000
46	5	102	kredit	2009-12-20 00:00:00	200000
47	7	109	debit	2009-12-20 00:00:00	100000
48	9	110	debit	2009-12-21 00:00:00	20000
49	7	109	debit	2009-12-22 00:00:00	100000
50	9	110	debit	2009-12-23 00:00:00	20000

Modul 6.

Tugas.

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

```
MariaDB [perbankan]> SELECT nama_nasabah, alamat_nasabah FROM nasabah ORDER BY nama_nasabah ;
```

nama_nasabah	alamat_nasabah
Anggit	Solo
Budi Eko Prayogo	Jl. Kantil 30
Budi Murtono	Jl. Merak 22
Canka Lokananta	Jl. Tidar 86
Indri Hapsari	Jl. Sutoyo 5
Joko Mdo Kondo	Jl. Bareng jadian kagak
Jon Koplo	Jl. Angin Besar 12
Kartika Padmasari	Jl. Manggis 15
Maryati	Jl. MT. Haryono 31
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
Sutopo	Jl. Jendral Sudirman 12

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

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

```
MariaDB [perbankan]> SELECT jenis_transaksi, jumlah FROM transaksi WHERE jenis_transaksi = 'kredit';
```

jenis_transaksi	jumlah
kredit	20000
kredit	30000
kredit	200000
kredit	150000
kredit	50000
kredit	200000
kredit	100000
kredit	20000
kredit	50000
kredit	200000
kredit	100000
kredit	100000
kredit	200000
kredit	50000
kredit	250000
kredit	130000
kredit	357000
kredit	50000
kredit	50000
kredit	520000
kredit	50000
kredit	50000
kredit	50000

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

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

```
MariaDB [perbankan]> SELECT jenis_transaksi, jumlah FROM transaksi WHERE tanggal = '2009-11-21';
```

jenis_transaksi	jumlah
debit	40000

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

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

```
MariaDB [perbankan]> SELECT nama_nasabah, jenis_transaksi, jumlah FROM nasabah, transaksi WHERE nasabah.id_nasabah = transaksi.id_nasabahFK and jumlah = 20000;
```

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

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

```
MariaDB [perbankan]>
```

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

```
MariaDB [perbankan]> SELECT nama_nasabah, alamat_nasabah FROM nasabah WHERE nama_nasabah = 'Su%';
```

```
Empty set (0.00 sec)
```

```
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
Sutopo	Jl. Jendral Sudirman

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

Modul 7.

Tugas.

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

```
MariaDB [perbankan]> SELECT nama_nasabah, alamat_nasabah, jenis_transaksi, jumlah
FROM nasabah, transaksi WHERE nasabah.id_nasabah=transaksi.id_nasabah AND tran
saksi.jenis_transaksi = 'kredit' ORDER BY nasabah.nama_nasabah;
```

nama_nasabah	alamat_nasabah	jenis_transaksi	jumlah
Budi Eko Prayogo	Jl. Kanti 30	kredit	30000
Canka Locananta	Jl. Tidar 86	kredit	150000
Kartika Padmasari	Jl. Manggis 15	kredit	20000
Suparman	Jl. Hasanudin 81	kredit	50000
Sutopo	Jl. Jendral Sudirman 12	kredit	200000

5 rows in set (0.00 sec)

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

```
MariaDB [perbankan]> SELECT no_rekening, nama_nasabah, jenis_transaksi, jumlah FR
OM nasabah, transaksi WHERE nasabah.id_nasabah = transaksi.id_nasabah AND transa
ksi.tanggal = '2009-11-15' ORDER BY nasabah.nama_nasabah;
```

no_rekening	nama_nasabah	jenis_transaksi	jumlah
110	Canka Locananta	kredit	150000
104	Sutopo	kredit	200000

2 rows in set (0.00 sec)

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 F
ROM nasabah, transaksi WHERE nasabah.id_nasabah = transaksi.id_nasabah AND trans
aksi.jumlah= '20000';
```

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

2 rows in set (0.00 sec)

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

```
MariaDB [perbankan]> SELECT no_rekening, nama_nasabah, alamat_nasabah FROM nasab
ah, transaksi WHERE nasabah.id_nasabah = transaksi.id_nasabah AND nama_nasabah l
ike 'Su%';
```

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

4 rows in set (0.00 sec)

5. Tampilkan nomor rekening dengan alias "Nomor Rekening", nama nasabah dengan alias "Nama Nasabah", jumlah transaksi dengan alias "Jumlah Transaksi" dimana jenis transaksinya adalah debit! Urutkan berdasarkan nama nasabah!

```

MariaDB [perbankan]> SELECT nasabah_has_rekening.no_rekening AS 'NOMOR REKENING'
, nasabah.nama_nasabah AS 'NAMA NASABAH', transaksi.jumlah AS 'JUMLAH TRANSAKSI'
FROM nasabah_has_rekening, transaksi, nasabah WHERE nasabah.id_nasabah = nasaba
h_has_rekening.id_nasabah AND nasabah_has_rekening.id_nasabah = transaksi.id_nas
abah AND transaksi.jenis_transaksi = 'debit' ORDER BY nasabah.nama_nasabah;
+-----+-----+-----+
| NOMOR REKENING | NAMA NASABAH | JUMLAH TRANSAKSI |
+-----+-----+-----+
| 102 | Budi Eko Prayogo | 20000 |
| 107 | Budi Eko Prayogo | 20000 |
| 101 | Kartika Padmasari | 100000 |
| 106 | Kartika Padmasari | 100000 |
| 103 | Maryati | 40000 |
| 103 | Maryati | 100000 |
| 105 | Superman | 50000 |
| 105 | Superman | 50000 |
+-----+-----+-----+
8 rows in set (0.00 sec)

```

Modul 8.

Tugas.

1. Tampilkan jenis transaksi, jumlah transaksi dalam Rp dan total transaksi untuk nasabah yang bernama akhiran 'Kartika Padmasari' untuk masing – masing jenis transaksi!

```
MariaDB [perbankan]> select jenis_transaksi, jumlah as 'Rp', count(jumlah) from transaksi, nasabah where transaksi.id_nasabahFK = nasabah.id_nasabah and nama_nasabah = 'Kartika Padmasari' group by jenis_transaksi;
ERROR 1146 (42S02): Table 'perbankan.transaksi' doesn't exist
MariaDB [perbankan]> select jenis_transaksi, jumlah as 'Rp', count(jumlah) from transaksi, nasabah where transaksi.id_nasabahFK = nasabah.id_nasabah and 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 [perbankan]> select sum(saldo) 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 nasabah.nama_nasabah = 'Maryati';
```

sum(saldo)
750000

1 row in set (0.00 sec)

MariaDB [perbankan]>

3. Tampilkan Jumlah transaksi yang ditangani oleh masing masing cabang bank!

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

count(jumlah)	nama_cabang
2	Bank Rut Unit Boyolali
3	Bank Rut Unit Balikpapan
2	Bank Rut Unit Berau
1	Bank Rut Unit Kartasura
3	Bank Rut Unit Karanganyar
8	Bank Rut Unit Magelang
1	Bank Rut Unit Pontianak
18	Bank Rut Unit Surakarta
2	Bank Rut Unit Samarinda
2	Bank Rut Unit Sukoharjo
4	Bank Rut Unit Tarakan
1	Bank Rut Unit Tanjung Selor
3	Bank Rut Unit Yogyakarta

13 rows in set (0.00 sec)

MariaDB [perbankan]>

- Tampilkan nama nasabah dan jumlah saldo yang memiliki saldo antar Rp. 500.000 sampai Rp. 2.000.000!

```
MariaDB [perbankan]> select nasabah.nama_nasabah as "nama nasabah" , rekening.saldo as "jumlah saldo" from nasabah , rekening, transaksi where transaksi.no_rekeningfk=rekening.no_rekening and
-> transaksi.id_nasabahfk=nasabah.id_nasabah and rekening.saldo between '500000' and '2000000';
```

nama nasabah	jumlah saldo
Maryati	750000
Kartika Padmasari	500000
Sutopo	900000
Canka Lokananta	550000
Maryati	750000
Sutopo	900000
Sutopo	900000
Kartika Padmasari	500000
Maryati	750000
Budi Murtono	500000
Maryati	750000
Sutopo	900000
Maryati	750000
Kartika Padmasari	500000
Maryati	750000
Canka Lokananta	550000
Indri Hapsari	750000
Sari Murti	500000
Maryati	550000
Budi Eko Prayogo	550000
Jon Koplo	500000
Jon Koplo	500000
Budi Eko Prayogo	750000

23 rows in set (0.00 sec)

- Tampilkan nama nasabah, tanggal transaksi dan jumlah transaksi dalam Rp dimana jumlah transaksi di atas Rp 100.000 dan urutkan jumlah transaksi dari yang besar ke yang kecil!

```
MariaDB [perbankan]> select nasabah.nama_nasabah as "Nama Nasabah", transaksi.tanggal as "Tanggal Transaksi", sum(transaksi.jumlah) as "Jumlah <Rp>" from nasabah, transaksi where nasabah.id_nasabah=transaksi.id_nasabahfk AND transaksi.jumlah >100000 GROUP BY nasabah.nama_nasabah, transaksi.tanggal, transaksi.jumlah ORDER BY transaksi.jumlah DESC;
```

Nama Nasabah	Tanggal Transaksi	Jumlah <Rp>
Jon Koplo	2017-05-26 00:00:00	9700000
Jon Koplo	2017-05-26 00:00:00	8500000
Anggit	2017-05-26 00:00:00	700000
Anggit	2017-06-08 00:00:00	2100000
Maryati	2017-05-24 00:00:00	547000
Sutopo	2017-05-26 00:00:00	520000
Sari Murti	2017-05-24 00:00:00	357000
Budi Eko Prayogo	2017-06-14 00:00:00	263000
Budi Murtono	2017-05-24 00:00:00	250000
Kartika Padmasari	2009-11-20 00:00:00	200000
Maryati	2009-11-28 00:00:00	200000
Sutopo	2009-11-15 00:00:00	200000
Budi Eko Prayogo	2009-12-05 00:00:00	200000
Canka Lokananta	2009-11-15 00:00:00	150000
Indri Hapsari	2017-05-24 00:00:00	130000
Satria Eka Jaya	2017-05-24 00:00:00	125000

16 rows in set (0.00 sec)

Modul 9.

Tugas.

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

```
C:\xampp\mysql\bin>mysql -u Fitri -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.1.37-MariaDB mariadb.org binary distribution
```

2. Beri Privilege untuk user tersebut dengan ketentuan sebagai berikut
 - a. Dapat melakukan INSERT, UPDATE, dan DELETE pada tabel nasabah.

```
'Fitri'@'localhost' at line 1
MariaDB [mysql]> GRANT INSERT, UPDATE, DELETE on perbankan.nasabah to Fitri@localhost;
Query OK, 0 rows affected (0.00 sec)
```

- b. Hanya dapat melakukan SELECT pada tabel cabang_bank.

```
MariaDB [perbankan]> select * from cabang_bank;
ERROR 1142 (42000): SELECT command denied to user 'Fitri'@'localhost' for table 'cabang_bank'
```

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

- a. Lakukan perintah SELECT pada tabel nasabah.

```
MariaDB [perbankan]> select * from nasabah;
ERROR 1142 (42000): SELECT command denied to user 'Fitri'@'localhost' for table 'nasabah'
```

- b. Lakukan perintah INSERT pada tabel nasabah.

```
MariaDB [perbankan]> insert into nasabah(id_nasabah, nama_nasabah, alamat_nasabah) values (110, 'fitri', 'ngawi');
Query OK, 1 row affected (0.12 sec)
```

- c. Lakukan perintah INSERT pada tabel nasabah.

```
MariaDB [perbankan]> insert into cabang_bank(kode_cabang, nama_cabang, alamat_cabang) values ('BRU', 'Bank Rut Unit Boyolali', 'Jl. Ahmad Yani 45');
```

Modul 10.

Tugas 1.

```
MariaDB [perbankan]> select AVG(jumlah) from transaksi;
+-----+
| AVG(jumlah) |
+-----+
| 89600.0000  |
+-----+
1 row in set (0.00 sec)

MariaDB [perbankan]> select jumlah from transaksi where jumlah < (select AVG(jumlah) from transaksi);
+-----+
| jumlah |
+-----+
| 50000  |
| 40000  |
| 20000  |
| 50000  |
| 30000  |
| 20000  |
| 50000  |
| 50000  |
| 40000  |
| 20000  |
| 50000  |
| 50000  |
| 50000  |
| 20000  |
| 50000  |
| 40000  |
| 50000  |
| 20000  |
| 20000  |
| 25000  |
| 50000  |
| 50000  |
| 50000  |
| 50000  |
| 50000  |
| 80000  |
+-----+
26 rows in set (0.00 sec)
```

```
MariaDB [perbankan]> select jumlah from transaksi where jumlah > (select AVG(jumlah) from transaksi);
+-----+
| jumlah |
+-----+
| 200000 |
| 150000 |
| 100000 |
| 100000 |
| 200000 |
| 100000 |
| 100000 |
| 200000 |
| 100000 |
| 100000 |
| 200000 |
| 100000 |
| 100000 |
| 200000 |
| 100000 |
| 150000 |
| 100000 |
| 125000 |
| 125000 |
| 200000 |
| 175000 |
| 130000 |
| 200000 |
| 150000 |
+-----+
24 rows in set (0.00 sec)
```

```
MariaDB [perbankan]> select DISTINCT (nama_nasabah), jumlah from nasabah, transaksi where nasabah.id_nasabah = transaksi.id_nasabahFK AND jumlah > (select AVG(jumlah) from transaksi);
+-----+-----+
| nama_nasabah | jumlah |
+-----+-----+
| Sutopo       | 200000 |
| Canka Lokananta | 150000 |
| Kartika Padmasari | 100000 |
| Maryati      | 100000 |
| Kartika Padmasari | 200000 |
| Sutopo       | 100000 |
| Budi Martono  | 100000 |
| Maryati      | 200000 |
| Supanman     | 100000 |
| Budi Eko Prayogo | 200000 |
| Triandhya    | 100000 |
| Bening       | 100000 |
| Denindya     | 200000 |
| Khatulistiyawara | 100000 |
| Khatulistiyawara | 150000 |
| Puput        | 125000 |
| Khatulistiyawara | 200000 |
| Putri        | 175000 |
| Budi Martono | 130000 |
| Puput        | 150000 |
+-----+-----+
20 rows in set (0.00 sec)
```



```
MariaDB [perbankan]> select no_rekening from rekening where saldo > (select AVG(saldo) from rekening);
```

no_rekening
105
106
108
115
117
122

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

Tugas 2.

```
MariaDB [perkuliahan]> select mahasiswa.nim from mahasiswa, matkul, link_mahasiswa_matkul where mahasiswa.nim=link_mahasiswa_matkul.nim and matkul.kode_matkul=link_mahasiswa_matkul.kode_matkul and mahasiswa.nim not in (select link_mahasiswa_matkul.nim from link_mahasiswa_matkul where kode_matkul='TIF001') group by mahasiswa.nim;
```

nim
L200150125

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

```
MariaDB [perkuliahan]> select * from link_mahasiswa_matkul where kode_matkul='TIF001';
```

nim	kode_matkul	nilai
L200150003	TIF001	70
L200150008	TIF001	80
L200150099	TIF001	70
L200150112	TIF001	90
L200150113	TIF001	70
L200150117	TIF001	80
L200150118	TIF001	80
L200150123	TIF001	85
L200150127	TIF001	90
L200150128	TIF001	80
L200150129	TIF001	80
L200150144	TIF001	80
L200150146	TIF001	90
L200154001	TIF001	70

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

```
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_matkul in (select kode_matkul from matkul where kode_dosen='D004') group by mahasiswa.nim, mahasiswa.nama having count(matkul.kode_matkul)=(select count(kode_matkul) from matkul where kode_dosen='D004');
```

nim	nama
L200150008	Khofa Prayoga
L200150099	Purwantinah
L200150117	Udiyarti Endang Saputri
L200150123	Danindya Puput Muliana Putri
L200150128	Sulthana Dzakira Drajat
L200150129	Fendy
L200150144	Sam'an Alghozy
L200150146	Fakhrur Razi
L200154001	Khilyatin Ulin Fitri

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

```
MariaDB [perkuliahan]> update link_mahasiswa_matkul as link set link.kode_matkul='TIF010' where link.nim in (select distinct mahasiswa.nim from mahasiswa where link.kode_matkul='TIF009');
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Rows matched: 0 Changed: 0 Warnings: 0
```

Modul 11.

Tugas.

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

- INSERT

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

cnx=mysql.connector.connect(user='root', database='perbankan')
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=('5','102','debit', tanggal, '20000')
cursor.execute(tambah_transaksi, data_transaksi)

cnx.commit()

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

- UPDATE

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

cnx=mysql.connector.connect(user='root', database='perbankan')
cursor=cnx.cursor()
tanggal=datetime.now().date()
update_transaksi=('update transaksi set jumlah=%s where no_rekeningFK=%s')
data_update=('70000', '110')
cursor.execute(update_transaksi, data_update)

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

===== RESTART: C:/Python34/ekstensi1.py =====
nasabah dengan ID 9 melakukan transaksi kredit pada 23 May 2019 sejumlah 70000
nasabah dengan ID 9 melakukan transaksi kredit pada 27 May 2019 sejumlah 70000
nasabah dengan ID 34 melakukan transaksi debit pada 27 May 2019 sejumlah 150000
nasabah dengan ID 9 melakukan transaksi kredit pada 27 May 2019 sejumlah 70000
nasabah dengan ID 5 melakukan transaksi debit pada 27 May 2019 sejumlah 20000
>>>
```

- DELETE

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

cnx=mysql.connector.connect(user='root', database='perbankan')
cursor=cnx.cursor()
tanggal=datetime.now().date()
delete_transaksi=('delete from transaksi where id_nasabahFK=5')
cursor.execute(delete_transaksi)

cnx.commit()

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

```
===== RESTART: C:/Python34/ekstensi1.py =====
nasabah dengan ID 9 melakukan transaksi kredit pada 23 May 2019 sejumlah 70000
nasabah dengan ID 9 melakukan transaksi kredit pada 27 May 2019 sejumlah 70000
nasabah dengan ID 34 melakukan transaksi debit pada 27 May 2019 sejumlah 150000
nasabah dengan ID 9 melakukan transaksi kredit pada 27 May 2019 sejumlah 70000
```

2. Buat kode program python untuk mendapatkan :

a. Data nasabah

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

cnx=mysql.connector.connect(user='root', database='perbankan')
cursor=cnx.cursor()
query=('select id_nasabah, nama_nasabah, alamat_nasabah from nasabah')

cursor.execute(query)
for(id_nasabah, nama_nasabah, alamat_nasabah)in cursor:
    print('Berikut ini data-data nasabah dengan ID {}|atas nama {} yang beralamat di {}'.format(id_nasabah, nama_nasabah, alamat_nasabah))
cursor.close()
cnx.close()
///

===== RESTART: C:/Python34/t2a.py =====
Berikut ini data-data nasabah dengan ID 2 atas nama Maryati yang beralamat di Jl. Jendral Sudirman 12
Berikut ini data-data nasabah dengan ID 3 atas nama Suparman yang beralamat di Jl. Hasanudin 81
Berikut ini data-data nasabah dengan ID 4 atas nama Kartika Padmasari yang beralamat di Jl. Manggis 15
Berikut ini data-data nasabah dengan ID 5 atas nama Budi Eko Prayogo yang beralamat di Jl. Kantil 30
Berikut ini data-data nasabah dengan ID 6 atas nama Satria Eka Jaya yang beralamat di Jl. Slamet Riyadi 45
Berikut ini data-data nasabah dengan ID 7 atas nama Trianandya yang beralamat di Jl. Sutoyo 5
Berikut ini data-data nasabah dengan ID 8 atas nama Sari Murti yang beralamat di Jl. Pangandaran 11
Berikut ini data-data nasabah dengan ID 9 atas nama Canka Lokananta yang beralamat di Jl. Tidar 86
Berikut ini data-data nasabah dengan ID 10 atas nama Budi Martono yang beralamat di Jl. Merak 22
Berikut ini data-data nasabah dengan ID 11 atas nama Danindya yang beralamat di Jl. Aki Balak 22
Berikut ini data-data nasabah dengan ID 12 atas nama Puput yang beralamat di Jl. AMD 10
Berikut ini data-data nasabah dengan ID 13 atas nama Muliana yang beralamat di Jl. Jend. Sudirman 97
Berikut ini data-data nasabah dengan ID 14 atas nama Putri yang beralamat di Jl. Diponegoro 19
Berikut ini data-data nasabah dengan ID 15 atas nama Bening yang beralamat di Jl. Agatis 10
Berikut ini data-data nasabah dengan ID 16 atas nama Khatulistiwa yang beralamat di Jl. Ir. Soekarno 12
Berikut ini data-data nasabah dengan ID 17 atas nama Pandan Wangi yang beralamat di Jl. Imam Bonjol 20
Berikut ini data-data nasabah dengan ID 18 atas nama Andara yang beralamat di Jl. Agatis 10
Berikut ini data-data nasabah dengan ID 19 atas nama Rangga yang beralamat di Jl. Kedua 80
Berikut ini data-data nasabah dengan ID 20 atas nama Cinta yang beralamat di Jl. Ketiga 24
Berikut ini data-data nasabah dengan ID 21 atas nama Susilo yang beralamat di Sukoharjo
Berikut ini data-data nasabah dengan ID 25 atas nama Roro yang beralamat di Jl. Jendral Sudirman 15
Berikut ini data-data nasabah dengan ID 31 atas nama Susilo yang beralamat di Sukoharjo
Berikut ini data-data nasabah dengan ID 32 atas nama Ai yang beralamat di Mgt
Berikut ini data-data nasabah dengan ID 33 atas nama Dicky yang beralamat di Delanggu
Berikut ini data-data nasabah dengan ID 34 atas nama dimas yang beralamat di Jl.RRI
Berikut ini data-data nasabah dengan ID 40 atas nama Oki yang beralamat di Boyolali
Berikut ini data-data nasabah dengan ID 41 atas nama OkiM yang beralamat di Boyolali
Berikut ini data-data nasabah dengan ID 999 atas nama Fredi yang beralamat di Wonogiri
...
```

b. Data nasabah yang melakukan transaksi antara bulan oktober sampai desember

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

cnx=mysql.connector.connect(user='root', database='perbankan')
cursor=cnx.cursor()
query=('select `id_nasabahFK`, `jenis_transaksi`, `tanggal`, `jumlah` from transaksi where month(tanggal)between 10 and 12')

cursor.execute(query)
for(id_nasabahFK, jenis_transaksi, tanggal, jumlah)in cursor:
    print('Berikut data nasabah dengan ID {} melakukan transaksi {} pada {:%d %b %Y} sejumlah {}'.format(id_nasabahFK, jenis_transaksi, tanggal, jumlah))
cursor.close()
cnx.close()
```

Berikut data nasabah dengan ID 3 melakukan transaksi debit pada 10 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 2 melakukan transaksi debit pada 10 Nov 2009 sejumlah 40000
Berikut data nasabah dengan ID 4 melakukan transaksi kredit pada 12 Nov 2009 sejumlah 20000
Berikut data nasabah dengan ID 3 melakukan transaksi debit pada 13 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 1 melakukan transaksi kredit pada 15 Nov 2009 sejumlah 200000
Berikut data nasabah dengan ID 9 melakukan transaksi kredit pada 15 Nov 2009 sejumlah 70000
Berikut data nasabah dengan ID 3 melakukan transaksi kredit pada 18 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 4 melakukan transaksi debit pada 19 Nov 2009 sejumlah 100000
Berikut data nasabah dengan ID 2 melakukan transaksi debit pada 19 Nov 2009 sejumlah 100000
Berikut data nasabah dengan ID 1 melakukan transaksi debit pada 19 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 4 melakukan transaksi kredit pada 20 Nov 2009 sejumlah 200000
Berikut data nasabah dengan ID 3 melakukan transaksi debit pada 21 Nov 2009 sejumlah 40000
Berikut data nasabah dengan ID 1 melakukan transaksi kredit pada 22 Nov 2009 sejumlah 100000
Berikut data nasabah dengan ID 4 melakukan transaksi kredit pada 22 Nov 2009 sejumlah 20000
Berikut data nasabah dengan ID 2 melakukan transaksi debit pada 22 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 10 melakukan transaksi debit pada 26 Nov 2009 sejumlah 100000
Berikut data nasabah dengan ID 3 melakukan transaksi kredit pada 27 Nov 2009 sejumlah 50000
Berikut data nasabah dengan ID 2 melakukan transaksi kredit pada 28 Nov 2009 sejumlah 200000
Berikut data nasabah dengan ID 3 melakukan transaksi kredit pada 28 Nov 2009 sejumlah 100000
Berikut data nasabah dengan ID 1 melakukan transaksi debit pada 01 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 2 melakukan transaksi debit pada 02 Dec 2009 sejumlah 40000
Berikut data nasabah dengan ID 4 melakukan transaksi debit pada 04 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 2 melakukan transaksi kredit pada 05 Dec 2009 sejumlah 100000
Berikut data nasabah dengan ID 7 melakukan transaksi debit pada 05 Dec 2009 sejumlah 100000
Berikut data nasabah dengan ID 9 melakukan transaksi debit pada 06 Dec 2009 sejumlah 70000
Berikut data nasabah dengan ID 12 melakukan transaksi kredit pada 07 Dec 2009 sejumlah 20000
Berikut data nasabah dengan ID 11 melakukan transaksi kredit pada 07 Dec 2009 sejumlah 25000
Berikut data nasabah dengan ID 13 melakukan transaksi debit pada 08 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 15 melakukan transaksi debit pada 08 Dec 2009 sejumlah 100000
Berikut data nasabah dengan ID 11 melakukan transaksi debit pada 08 Dec 2009 sejumlah 200000
Berikut data nasabah dengan ID 11 melakukan transaksi kredit pada 10 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 16 melakukan transaksi debit pada 11 Dec 2009 sejumlah 100000
Berikut data nasabah dengan ID 14 melakukan transaksi kredit pada 11 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 17 melakukan transaksi kredit pada 12 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 16 melakukan transaksi debit pada 13 Dec 2009 sejumlah 150000
Berikut data nasabah dengan ID 10 melakukan transaksi debit pada 14 Dec 2009 sejumlah 100000
Berikut data nasabah dengan ID 12 melakukan transaksi kredit pada 14 Dec 2009 sejumlah 125000
Berikut data nasabah dengan ID 12 melakukan transaksi kredit pada 14 Dec 2009 sejumlah 125000
Berikut data nasabah dengan ID 16 melakukan transaksi debit pada 15 Dec 2009 sejumlah 200000
Berikut data nasabah dengan ID 13 melakukan transaksi debit pada 15 Dec 2009 sejumlah 50000
Berikut data nasabah dengan ID 14 melakukan transaksi kredit pada 16 Dec 2009 sejumlah 175000

Berikut data nasabah dengan ID 17 melakukan transaksi kredit pada 17 Dec 2009 sejumlah 80000
Berikut data nasabah dengan ID 10 melakukan transaksi kredit pada 18 Dec 2009 sejumlah 130000
Berikut data nasabah dengan ID 11 melakukan transaksi debit pada 19 Dec 2009 sejumlah 200000
Berikut data nasabah dengan ID 12 melakukan transaksi kredit pada 20 Dec 2009 sejumlah 150000

Modul 12.

Tugas.

Aplikasi berbasis python pada studi kasus Apotek, yang entitasnya :

- Obat
- Pasien
- Pemberian Obat

1. Obat

```
obat.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\obat.py (3.6.2)
File Edit Format Run Options Window Help

import mysql.connector
import os

db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="apotek"
)

def insert_data(db):
    id_obat = input("Masukan id obat: ")
    nama_obat = input("Masukan nama obat: ")
    harga_obat = input("Masukkan harga obat: ")

    val = (id_obat, nama_obat, harga_obat)
    cursor = db.cursor()
    sql = "INSERT INTO obat (id_obat, nama_obat, harga_obat) 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 obat"
    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)
    print("")
    id_obat = input("Masukan id obat> ")
    nama_obat = input("Masukan nama obat: ")
    harga_obat = input("Masukkan harga obat: ")

    sql = "UPDATE obat SET nama_obat=%s, harga_obat=%s WHERE id_obat=%s"
    val = (nama_obat, harga_obat, id_obat)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil diubah".format(cursor.rowcount))

def search_data(db):
    cursor = db.cursor()
    kata = input("Kata kunci: ")
    sql = "SELECT * FROM obat WHERE nama_obat LIKE %s OR harga_obat LIKE %s"
    val = ("%{}%".format(kata), "%{}%".format(kata))
    cursor.execute(sql, val)
    results = cursor.fetchall()

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

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    print("")
    id_obat = input("pilih id obat> ")
    sql = "DELETE FROM obat WHERE id_obat=%s"
    val = (id_obat,)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
    print("-----")
    menu = input("Pilih menu> ")

    #clear screen
    os.system("clear")

    if menu == "1":
        insert_data(db)
    elif menu == "2":
        show_data(db)
    elif menu == "3":
        update_data(db)
    elif menu == "4":
        delete_data(db)
    elif menu == "5":
        search_data(db)
    elif menu == "0":
        exit()
    else:
        print("Menu salah!")

if __name__ == "__main__":
    while True:
        show_menu(db)
```

```
obat.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\obat.py (3.6.2)
File Edit Format Run Options Window Help

id_obat = input("pilih id obat> ")
sql = "DELETE FROM obat WHERE id_obat=%s"
val = (id_obat,)
cursor.execute(sql, val)
db.commit()
print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
    print("-----")
    menu = input("Pilih menu> ")

    #clear screen
    os.system("clear")

    if menu == "1":
        insert_data(db)
    elif menu == "2":
        show_data(db)
    elif menu == "3":
        update_data(db)
    elif menu == "4":
        delete_data(db)
    elif menu == "5":
        search_data(db)
    elif menu == "0":
        exit()
    else:
        print("Menu salah!")

if __name__ == "__main__":
    while True:
        show_menu(db)
```

2. Pasien

pasien.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\pasien.py (3.6.2)

File Edit Format Run Options Window Help

```
import mysql.connector
import os

db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="apotek"
)

def insert_data(db):
    id_pasien = input("Masukan id pasien: ")
    nama_pasien = input("Masukan nama pasien: ")
    alamat_pasien = input("Masukkan alamat pasien: ")

    val = (id_pasien, nama_pasien, alamat_pasien)
    cursor = db.cursor()
    sql = "INSERT INTO pasien (id_pasien, nama_pasien, alamat_pasien) 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 pasien"
    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)
    print("")
    id_pasien = input("Masukan id pasien> ")
    nama_pasien = input("Masukan nama pasien: ")
    alamat_pasien = input("Masukkan alamat pasien: ")
```

pasien.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\pasien.py (3.6.2)

File Edit Format Run Options Window Help

```
sql = "UPDATE pasien SET nama_pasien=%s, alamat_pasien=%s WHERE id_pasien=%s"
val = (nama_pasien, alamat_pasien, id_pasien)
cursor.execute(sql, val)
db.commit()
print("{} data berhasil diubah".format(cursor.rowcount))

def search_data(db):
    cursor = db.cursor()
    kata = input("Kata Kunci: ")
    sql = "SELECT * FROM pasien WHERE nama_pasien LIKE %s OR alamat_pasien LIKE %s"
    val = ("%s" % kata, "%s" % kata)
    cursor.execute(sql, val)
    results = cursor.fetchall()

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

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    id_pasien = input("pilih id pasien> ")
    sql = "DELETE FROM pasien WHERE id_pasien=%s"
    val = (id_pasien,)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
```

pasien.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\pasien.py (3.6.2)

File Edit Format Run Options Window Help

```
id_pasien = input("pilih id pasien> ")
sql = "DELETE FROM pasien WHERE id_pasien=%s"
val = (id_pasien,)
cursor.execute(sql, val)
db.commit()
print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
    print("-----")
    menu = input("Pilih menu> ")

    #clear screen
    os.system("clear")

    if menu == "1":
        insert_data(db)
    elif menu == "2":
        show_data(db)
    elif menu == "3":
        update_data(db)
    elif menu == "4":
        delete_data(db)
    elif menu == "5":
        search_data(db)
    elif menu == "0":
        exit()
    else:
        print("Menu salah!")

if __name__ == "__main__":
    while True:
        show_menu(db)
```

Ln:1 Col:0

3. Pemberian Obat

```
pemberianobat.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\pemberianobat.py (3.6.2)
File Edit Format Run Options Window Help

import mysql.connector
import os

db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="apotek"
)

def insert_data(db):
    id_pemberian = input("Masukan id pemberian: ")
    jenis_pemberian = input("Masukan jenis pemberian: ")

    val = (id_pemberian, jenis_pemberian)
    cursor = db.cursor()
    sql = "INSERT INTO pemberianobat (id_pemberian, jenis_pemberian) 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 pemberianobat"
    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)
    print("")
    id_pemberian = input("Masukan id pemberian: ")
    jenis_pemberian = input("Masukan jenis pemberian: ")

    sql = "UPDATE pemberianobat SET jenis_pemberian=%s WHERE id_pemberian=%s"
    val = (jenis_pemberian, id_pemberian)

    sql = "UPDATE pemberianobat SET jenis_pemberian=%s WHERE id_pemberian=%s"
    val = (jenis_pemberian, id_pemberian)

    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil diubah".format(cursor.rowcount))

def search_data(db):
    cursor = db.cursor()
    kata = input("Kata kunci: ")
    sql = "SELECT * FROM pemberianobat WHERE id_pemberian LIKE %s OR jenis_pemberian LIKE %s"
    val = ("%{}%".format(kata), "%{}%".format(kata))
    cursor.execute(sql, val)
    results = cursor.fetchall()

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

def delete_data(db):
    cursor = db.cursor()
    show_data(db)
    print("")
    id_pemberian = input("Pilih id pemberian: ")
    sql = "DELETE FROM pemberianobat WHERE id_pemberian=%s"
    val = (id_pemberian,)
    cursor.execute(sql, val)
    db.commit()
    print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
    print("-----")
    menu = input("Pilih menu: ")

    #clear screen
    os.system("clear")

    if menu == "1":
        insert_data(db)
    elif menu == "2":
        show_data(db)
    elif menu == "3":
        update_data(db)
    elif menu == "4":
        delete_data(db)
    elif menu == "5":
        search_data(db)
    elif menu == "0":
        exit()
    else:
        print("Menu salah!")

if __name__ == "__main__":
    while True:
        show_menu(db)
```

```
pemberianobat.py - D:\MATKUL\SEM 4\prak basis data\Project Fitri\pemberianobat.py (3.6.2)
File Edit Format Run Options Window Help

id_pemberian = input("Pilih id pemberian: ")
sql = "DELETE FROM pemberianobat WHERE id_pemberian=%s"
val = (id_pemberian,)
cursor.execute(sql, val)
db.commit()
print("{} data berhasil dihapus".format(cursor.rowcount))

def show_menu(db):
    print("***** APLIKASI DATABASE PYTHON *****")
    print("***** APOTEK CAHYA *****")
    print("1. Insert Data")
    print("2. Tampilkan Data")
    print("3. Update Data")
    print("4. Hapus Data")
    print("5. Search Data")
    print("0. Keluar")
    print("-----")
    menu = input("Pilih menu: ")

    #clear screen
    os.system("clear")

    if menu == "1":
        insert_data(db)
    elif menu == "2":
        show_data(db)
    elif menu == "3":
        update_data(db)
    elif menu == "4":
        delete_data(db)
    elif menu == "5":
        search_data(db)
    elif menu == "0":
        exit()
    else:
        print("Menu salah!")

if __name__ == "__main__":
    while True:
        show_menu(db)
```

ER DIAGRAM

