

LAPORAN PRAKTIKUM SISTEM BASIS DATA



Disusun oleh :

Raihan Mazarul Hidayat

L200180162

PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA SURAKARTA

Modul 1

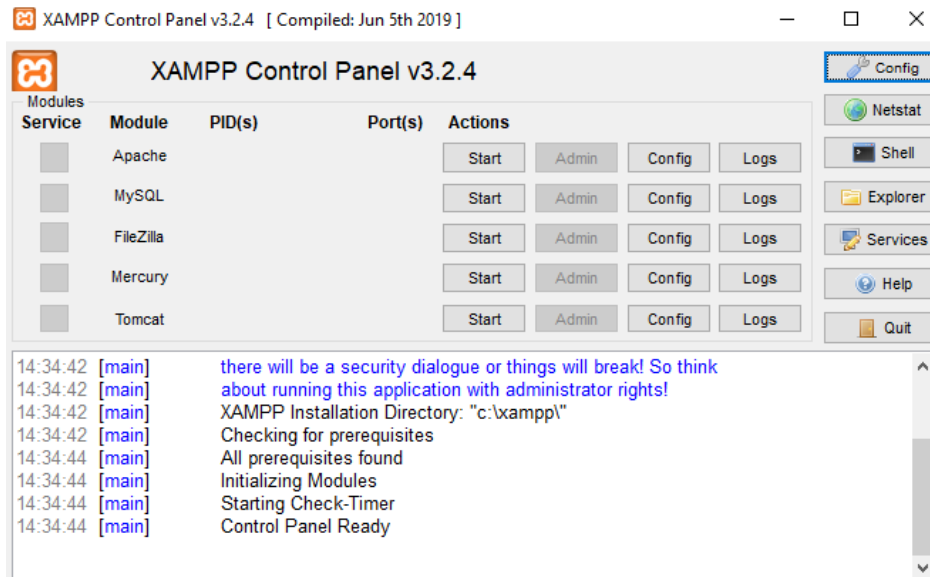
Pengenalan dan Instalasi MySQL

A. Praktikum

1. Mengakses PhpMyAdmin

Langkah 1.

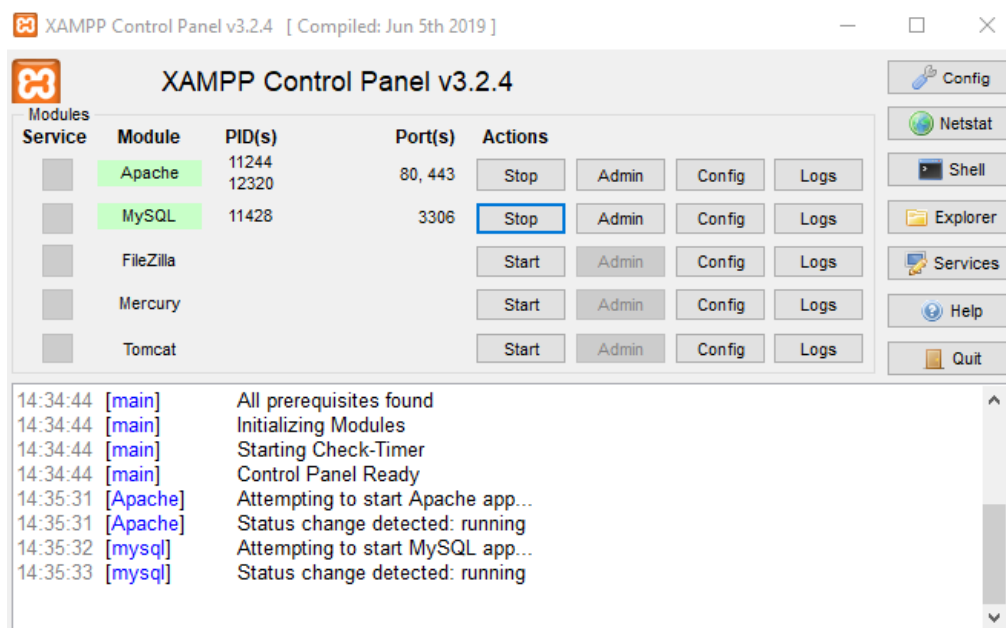
Membuka XAMPP Control Panel.



Langkah 2.

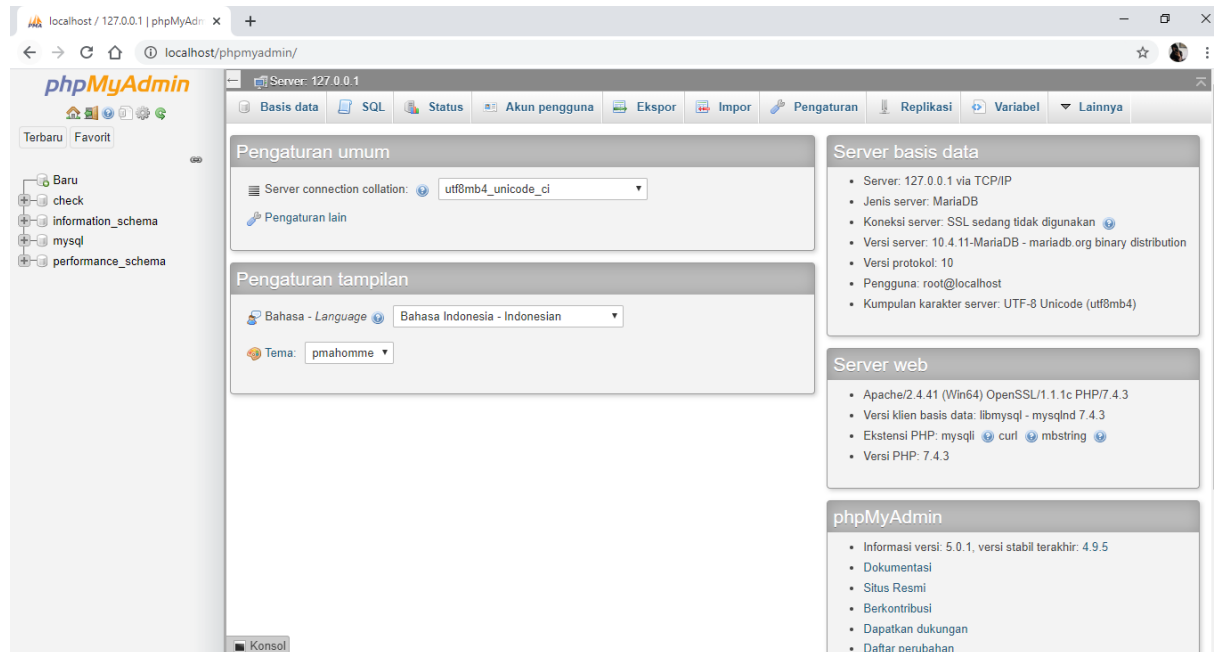
Menjalankan Apache Server dan MySQL Server dengan menekan tombol “Start”.

Menunggu hingga muncul warna hijau pada nama Module.



Langkah 3.

Membuka web browser dan mengetikkan <http://localhost/phpmyadmin/>



Langkah 4.

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

1. Mengakses MySQL lewat command prompt.

Langkah 1.

Memastikan bahwa server MySQL telah berjalan

Langkah 2.

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

Langkah 3.

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

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\HP-DESKTOP>cd C:\xampp\mysql\bin
C:\xampp\mysql\bin>
```

Langkah 4.

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

```
C:\Windows\system32\cmd.exe - mysql -u root -p
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\HP-DESKTOP>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 22
Server version: 10.4.11-MariaDB mariadb.org binary distribution

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

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

MariaDB [(none)]>
```

Langkah 5.

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

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| check    |
| information_schema |
| mysql    |
| performance_schema |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]>
```

B. Tugas

1. Jelaskan mengapa dibutuhkan data! Jawab:

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

2. Jelaskan manfaat database dan contohnya! Jawab:

- Sebagai komponen utama atau penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
- Menentukan kualitas informasi yaitu cepat, akurat, dan relevan, sehingga informasi yang disajikan tidak basi. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
- Mengatasi kerangkapan data (redundancy data)
- Menghindari terjadinya inkonsistensi data.
- Mengatasi kesulitan dalam mengakses data.
- Menyusun format yang standar dari sebuah data.
- Penggunaan oleh banyak pemakai (multiple user). Sebuah database bisa dimanfaatkan sekaligus secara bersama oleh banyak pengguna (multiuser).
- Melakukan perlindungan dan pengamanan data. Setiap data hanya bisa diakses atau dimanipulasi oleh pihak yang diberi otoritas dengan memberikan login dan password terhadap masing-masing data.

Contohnya: database rumah sakit, database akademik, database perusahaan, database bank, dll.

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

Jawab:

- Deskripsikan kebutuhan informasi dan data Spesifikasi data
- Pemrosesan yang diperlukan oleh data Pertimbangan keamanan
- Kecocokan dengan tipe aplikasi Bahasa query
- Biaya tak langsung terhadap pemrosesan

4. Jelaskan istilah atau terminology yang digunakan dalam Database (database, table, field, record)!

Jawab:

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

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

Jawab:

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

6. Mengapa dibutuhkan DBMS? Jawab:

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

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

Jawab:

Pada tabel Mhs, terdapat 6 field yaitu: NIM, NAMA, TEMPAT_LAHIR, TANGGAL_LAHIR, ALAMAT, TELEPON. Masing-masing field diatur sesuai tipe data yang dibutuhkan. Pada field NIM diatur tipe data text, berarti pada record dapat diisikan data berupa teks huruf, angka, dan simbol-simbol lain. Ukuran tipe data pada field NIM diatur sepanjang 10, artinya digit maksimum yang dapat dimasukkan ke dalam record adalah 10 digit. Jika data yang dimasukkan lebih dari 10 digit, maka digit ke-11 dan berikutnya tidak akan terbaca. Untuk pengaturan maksimum defaultnya sendiri adalah sepanjang 255 digit. Kemudian pada field NAMA, sama-sama diatur tipe data text, namun panjang ukurannya berbeda, yaitu 30, artinya kita dapat memasukkan digit maksimum ke dalam record yaitu sepanjang 30 digit, melebihi 30 data ke-31 dan seterusnya tidak akan terbaca. Pada field TANGGAL_LAHIR diatur tipe date/time yang berarti pada record dapat diisikan data

berupa angka dan symbol. Pada field TELEPHONE diatur tipe integer, berarti pada record dapat diisi angka saja.

Modul 2

Perancangan Basis Data

A. Praktikum

Merancang basis data untuk permasalahan berikut ini.

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

Langkah-langkah perncangan database perbankan:

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

nasabah : menyimpan semua data pribadi semua nasabah

rekening : menyimpan informasi semua rekening yang telah dibuka

cabang_bank : menyimpan informasi tentang semua cabang bank

transaksi : menyimpan informasi tentang semua transaksi yang telah terjadi

2. Menentukan attributes masing-masing *entity* sesuai kebutuhan database

nasabah :

- id_nasabah : nomor id untuk nasabah (integer) PK
- nama_nasabah : nama lengkap nasabah (varchar(45))
- alamat_nasabah : alamat lengkap nasabah (varchar(255))

rekening :

- no_rekening : nomor rekening (integer) PK
- pin : personal identification number (varchar(10))
- saldo : jumlah saldo rekening dalam Rp (integer)

cabang_bank :

- kode_cabang : kode untuk cabang bank (varchar(10)) PK
- nama_cabang : nama lengkap cabang bank (varchar(20))
- alamat_cabang : alamat lengkap cabang bank (varchar(255))

transaksi :

- no_transaksi : nomer transaksi (integer) PK
- jenis_transaksi : kredit atau debit (varchar(10))
- tanggal : tanggal terjadinya transaksi (date)
- jumlah : besarnya transaksi dalam Rp (integer)

3. Menentukan relationship (hubungan) antar entitas

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

Hubungan

nasabah memiliki rekening :

- tabel utama : nasabah, rekening
- tabel kedua : nasabah_has_rekening ○ relationship : many-to-many (m:n)
- attribute penghubung : id_nasabah, no_rekening (FK id_nasabah, no_rekening di nasabah_has_rekening)

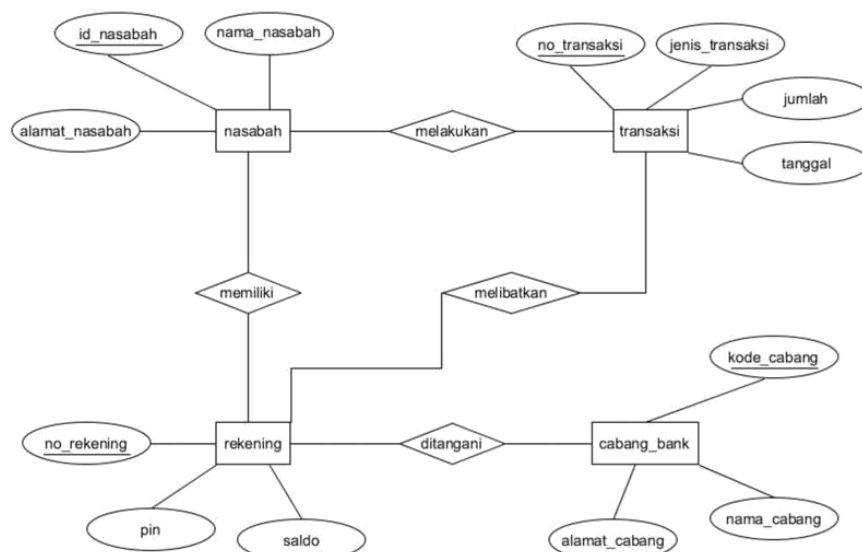
nasabah melakukan transaksi :

- tabel utama : nasabah
- tabel kedua : transaksi
- relationship : one-to-many(1:n)
- attribute penghubung : id_nasabah (FK id_nasabah di transaksi) cabang_bank menangani rekening :
- tabel utama : cabang_bank
- tabel kedua : rekening
- rekening : one-to-many (1:n)
- attribute penghubung : kode_cabang (FK kode_cabang di rekening)

rekening terlibat dalam transaksi :

- tabel utama : rekening
- tabel kedua : transaksi
- relationship : one-to-many (1:n)
- attribute penghubung : no_rekening (FK no_rekening di transaksi)

4. Menggambar ERD Diagram



B. Tugas

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

Jawab:

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

mahasiswa : menyimpan semua data pribadi semua mahasiswa

dosen : menyimpan semua data pribadi semua dosen

matakuliah : menyimpan informasi tentang semua mata kuliah

kelas : menyimpan informasi tentang data semua ruang kelas

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

mahasiswa :

- nim : nim untuk mahasiswa (varchar(20)) PK
- nama_mhs : nama lengkap mahasiswa (varchar(45))
- alamat_mhs : alamat lengkap mahasiswa (varchar(255))
- tgllahir_mhs : tanggal lahir mahasiswa (date)

dosen :

- nip_dosen : nip untuk dosen (varchar(20)) PK
- nama_dosen : nama lengkap dosen (varchar(45))
- alamat_dosen : alamat lengkap dosen (varchar(255))
- kontak_dosen : nomor telephone dosen (integer)

matakuliah :

- kode_mk : kode untuk mata kuliah (varchar(20)) PK
- nama_mk : nama mata kuliah (varchar(45))
- jml_sks : jumlah sks mata kuliah (integer)
- semester : semester mata kuliah (integer)

kelas :

- kode_ruang : kode ruang kelas (varchar(20)) PK
- kapasitas : kapasitas ruang kelas (integer)

- C. Menentukan relationship antar entitas

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

Hubungan

mahasiswa diampu dosen :

- tabel utama : dosen
- tabel kedua : mahasiswa
- relationship : many-to-one (n:1)
- attribute penghubung : nip_dosen (FK nip_dosen di dosen)

dosen mengampu matakuliah :

- tabel utama : matakuliah
- tabel kedua : dosen
- relationship : many-to-one(n:1)
- attribute penghubung : kode_mk (FK kode_mk di matakuliah)

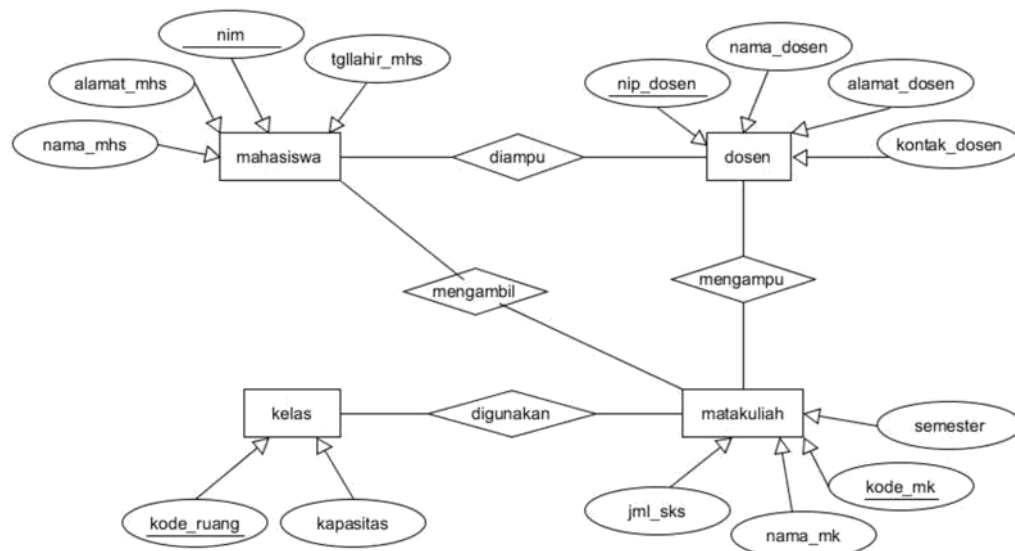
mahasiswa mengambil matakuliah :

- tabel utama : mahasiswa
- tabel kedua : mahasiswa_has_matakuliah
- rekening : many-to-many (m:n)
- attribute penghubung : nim, kode_mk (FK nim, kode_mk di mahasiswa_has_matakuliah)

kelas digunakan untuk matakuliah :

- tabel utama : kelas
- tabel kedua : matakuliah
- relationship : one-to-one (1:1)
- attribute penghubung : kode_ruang (FK kode_ruang di matakuliah)

D. Menggambar ERD Diagram



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

Database Rumah Sakit

- A. Menentukan entities (object-object dasar) yang perlu ada di database. pasien :
menyimpan semua data pribadi semua pasien
dokter : menyimpan semua data pribadi semua dokter
receptionist : menyimpan semua data pribadi semua receptionist
kamar : menyimpan informasi tentang data semua kamar
- B. Menentukan attributes (sifat-sifat) masing-masing *entity* sesuai kebutuhan database
- pasien* :
- id_pasien : id untuk pasien (varchar(15)) PK
 - nama_pasien : nama lengkap pasien (varchar(75))
 - alamat_pasien : alamat lengkap pasien (varchar(255))
 - umur_pasien : umur pasien (integer)
- dokter* :
- id_dokter : id untuk dokter (varchar(15)) PK
 - nama_dokter : nama lengkap dokter (varchar(75))
 - alamat_dokter : alamat lengkap dokter (varchar(255))
 - spesialis : spesialis dokter (varchar(20))
- receptionist* :
- id_receptionist : id untuk receptionist (varchar(10)) PK
 - nama_receptionist : nama receptionist (varchar(50))
 - alamat_receptionist : alamat lengkap receptionist (varchar(255))
 - jam_jaga : jam jaga receptionist (varchar(10))
- kamar* :
- no_kamar : nomer kamar (varchar(10)) PK
 - kelas_kamar : kelas kamar (integer)
- C. Menentukan relationship (hubungan) antar entitas

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

Hubungan

dokter memeriksa pasien :

tabel utama : dokter

tabel kedua : pasien

relationship : one-to-many (1:n)

attribute penghubung : id_dokter (FK id_dokter di pasien)

receptionist melayani pasien :

tabel utama : receptionist, pasien

tabel kedua : receptionist_has_pasien

relationship : many-to-many(m:n)

attribute penghubung : id_receptionist, id_pasien (FK id_recptionist, id_pasien di receptionist_has_pasien)

pasien menggunakan kamar :

tabel utama : pasien

tabel kedua : kamar

rekening : one-to-one (1:1)

attribute penghubung : id_pasien(FK id_pasien di kamar)

dokter visit kamar :

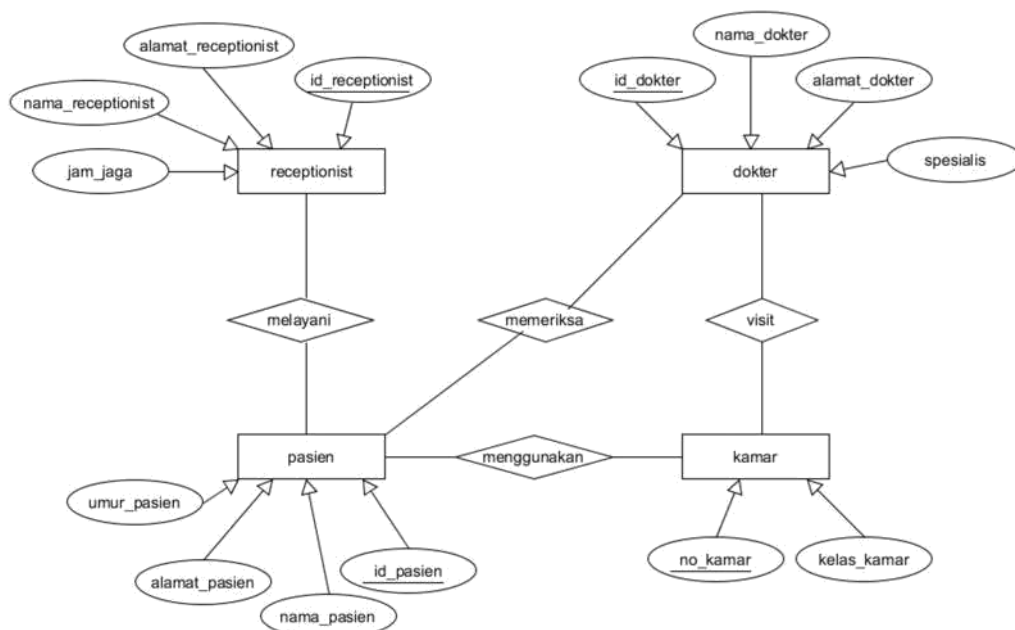
tabel utama : dokter

tabel kedua : kamar

relationship : one-to-many (1:n)

attribute penghubung : id_dokter (FK id_dokter di kamar)

D. Menggambar ERD Diagram



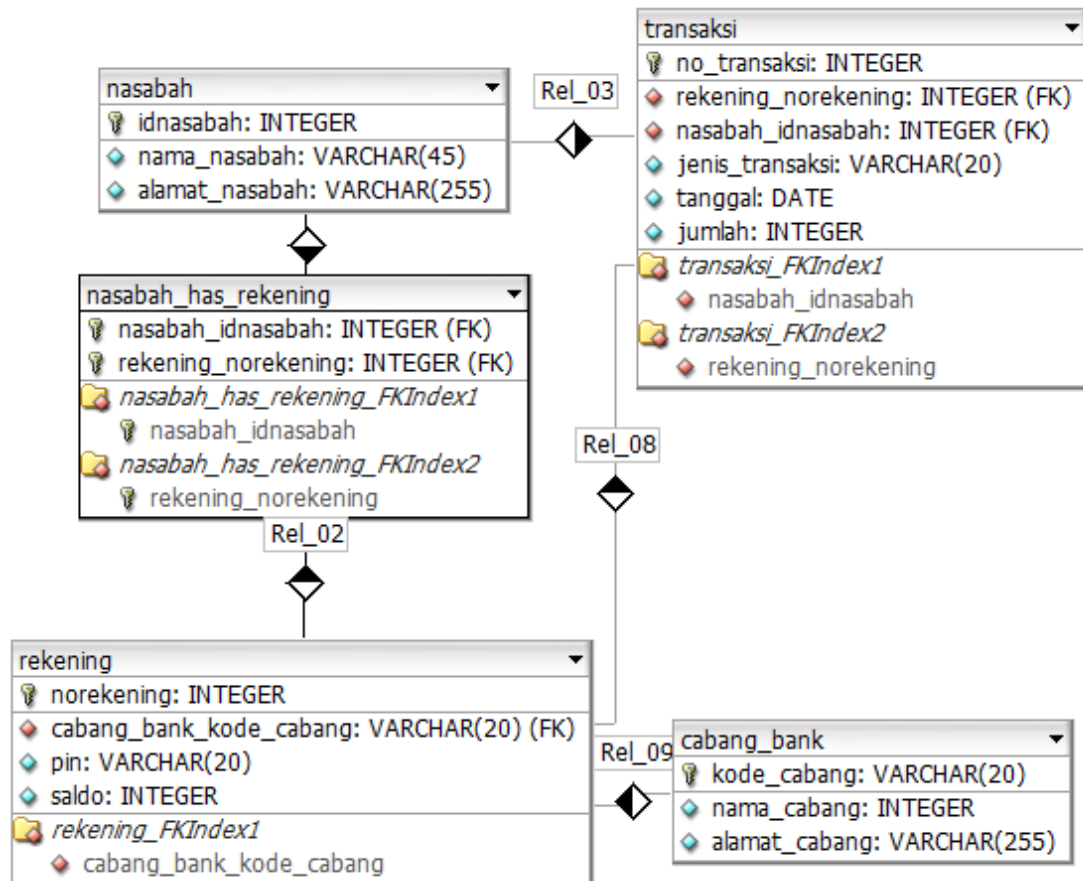
Modul 3

Perancangan E-R Diagram dengan DBDesigner

A. Praktikum

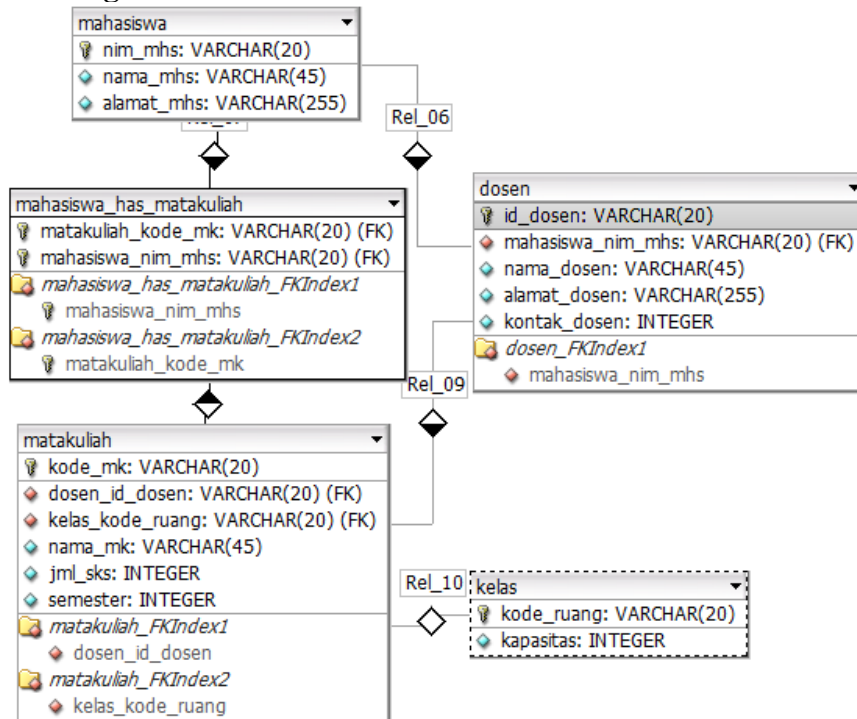
Menggambar ER Diagram dengan DB Designer

1. Jalankan program aplikasi DB Designer
2. Buat tabel nasabah, transaksi, cabang_bank, rekening

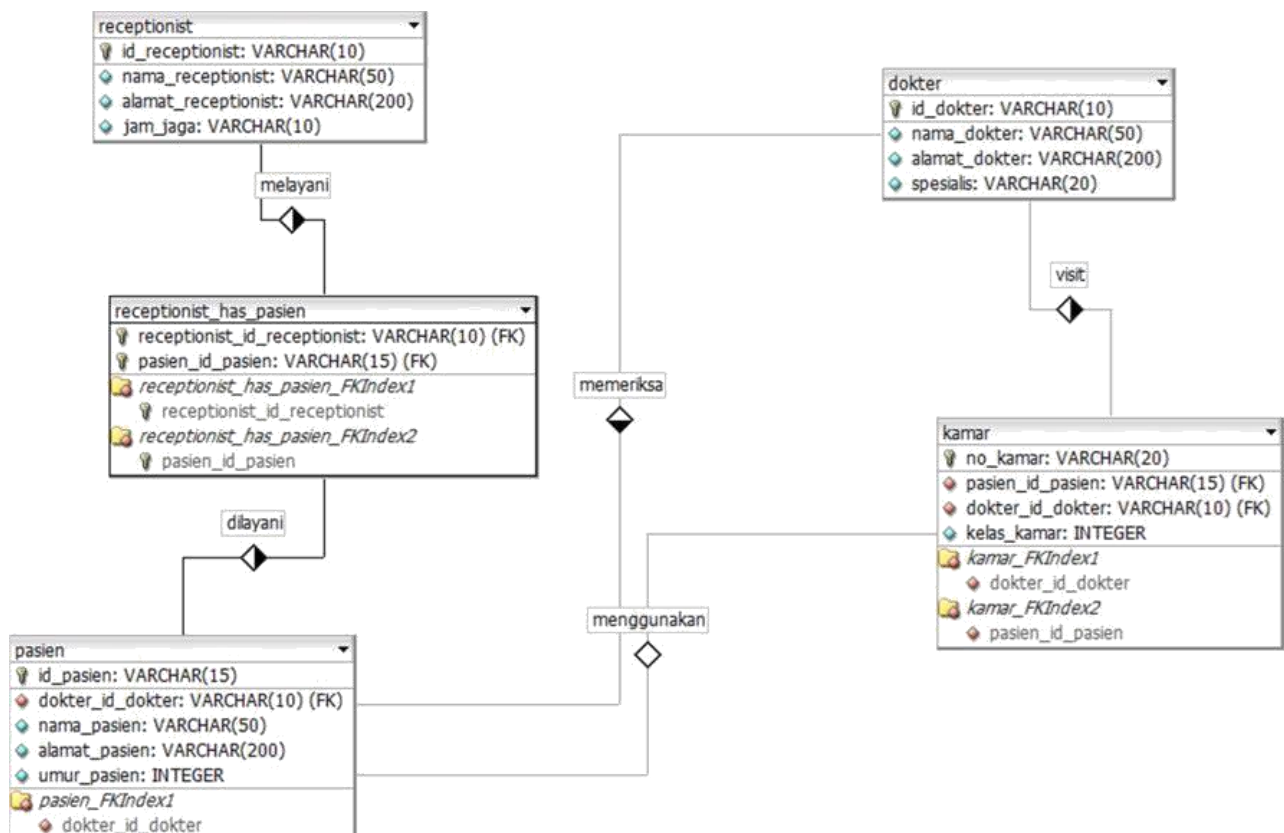


B. Tugas

ER Diagram datakuliah



ER Diagram Rumah Sakit



Modul 4

Perancangan E-R Diagram dengan DBDesigner

A. Praktikum

1. Jalankan Xampp Control Panel
2. Jalankan server Mysql dan Apace
3. Buka cmd dan login sebagai root ke Mysql “mysql -u root -p
4. Buat database baru “create database perbankan;”
5. Hubungkan kedalam database yang telah dibuat “use perbankan;”

```
C:\Users\HP-DESKTOP>cd C:\xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 23
Server version: 10.4.11-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 (0.127 sec)

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

6. Membuat tabel nasabah

```
CREATE TABLE nasabah(
id_nasabah INTEGER PRIMARY KEY,
nama_nasabah VARCHAR(45) NOT NULL,
alamat_nasabah VARCHAR(255) NOT NULL);
```

7. Membuat tabel cabang_bank

```
CREATE TABLE cabang_bank(
kode_cabang VARCHAR(20) PRIMARY KEY,
nama_cabang VARCHAR(45) UNIQUE NOT NULL,
alamat_cabang VARCHAR(255) NOT NULL);
```

8. Membuat tabel rekening

```
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);

```
MariaDB [perbankan]> CREATE TABLE nasabah(  
-> id_nasabah INTEGER PRIMARY KEY,  
-> nama_nasabah VARCHAR(45) NOT NULL,  
-> alamat_nasabah VARCHAR(255) NOT NULL);  
Query OK, 0 rows affected (0.779 sec)  
  
MariaDB [perbankan]> CREATE TABLE cabang_bank(  
-> kode_cabang VARCHAR(20) PRIMARY KEY,  
-> nama_cabang VARCHAR(45) UNIQUE NOT NULL,  
-> alamat_cabang VARCHAR(255) NOT NULL);  
Query OK, 0 rows affected (0.479 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 (0.526 sec)
```

9. Membuat tabel transaksi

```
CREATE TABLE transaksi(  
no_transaksi SERIAL PRIMARY KEY,  
id_nasabahFK INTEGER REFERENCES nasabah(id_nasabah)  
ON DELETE SET NULL ON UPDATE CASCADE,  
no_rekeningFK INTEGER REFERENCES rekening(no_rekening)  
ON DELETE SET NULL 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));
```

10. Membuat tabel nasabah_has_rekening

```
CREATE TABLE nasabah_has_rekening(  
Id_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,  
PRIMARY KEY(id_nasabahFK, no_rekeningFK));
```

```

MariaDB [perbankan]> CREATE TABLE transaksi(
  -> no_transaksi SERIAL PRIMARY KEY,
  -> id_nasabahFK INTEGER REFERENCES nasabah(id_nasabah)
  -> ON DELETE SET NULL ON UPDATE CASCADE,
  -> no_rekeningFK INTEGER REFERENCES rekening(no_rekening)
  -> ON DELETE SET NULL 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 (0.563 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(no_rekening)
  -> ON DELETE CASCADE ON UPDATE CASCADE,
  -> PRIMARY KEY(id_nasabahFK, no_rekeningFK));
Query OK, 0 rows affected (0.385 sec)

```

11. Untuk cek hasil pembuatan database gunakan perintah 'show tables'

```

MariaDB [perbankan]> show tables;
+-----+
| Tables_in_perbankan |
+-----+
| cabang_bank         |
| nasabah             |
| nasabah_has_rekening |
| rekening            |
| transaksi           |
+-----+
5 rows in set (0.009 sec)

```

12. Untuk melihat struktur tiap tabel gunakan perintah 'describe'

```

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 (0.226 sec)

```

B. Tugas

Implementasi database “data kuliah” pada tugas modul 2 ke program mysql.

1. Jalankan Xampp Control Panel
2. Jalankan server Mysql dan Apache
3. Buka cmd dan login sebagai root ke Mysql “mysql -u root -p
4. Buat database baru “create database datakuliah;”
5. Hubungkan kedalam database yang telah dibuat “use datakuliah;”

```
MariaDB [(none)]> create database datakuliah;  
Query OK, 1 row affected (0.002 sec)  
  
MariaDB [(none)]> use datakuliah;  
Database changed
```

6. Buat tabel mahasiswa, dosen, matakuliah, kelas dan mahasiswa_has_matakuliah

```
MariaDB [datakuliah]> CREATE TABLE mahasiswa (  
-> nim_mhs VARCHAR(20) NOT NULL,  
-> nama_mhs VARCHAR(45) NULL,  
-> alamat_mhs VARCHAR(255) NULL,  
-> PRIMARY KEY(nim_mhs)  
-> );  
Query OK, 0 rows affected (0.477 sec)  
  
MariaDB [datakuliah]> CREATE TABLE dosen (  
-> id_dosen VARCHAR(20) NOT NULL,  
-> mahasiswa_nim_mhs VARCHAR(20) NOT NULL,  
-> nama_dosen VARCHAR(45) NULL,  
-> alamat_dosen VARCHAR(255) NULL,  
-> kontak_dosen INTEGER UNSIGNED NULL,  
-> PRIMARY KEY(id_dosen),  
-> INDEX dosen_FKIndex1(mahasiswa_nim_mhs)  
-> );  
Query OK, 0 rows affected (0.389 sec)  
  
MariaDB [datakuliah]> CREATE TABLE matakuliah (  
-> kode_mk VARCHAR(20) NOT NULL,  
-> dosen_id_dosen VARCHAR(20) NOT NULL,  
-> kelas_kode_ruang VARCHAR(20) NOT NULL,  
-> nama_mk VARCHAR(45) NULL,  
-> jml_sks INTEGER UNSIGNED NULL,  
-> semester INTEGER UNSIGNED NULL,  
-> PRIMARY KEY(kode_mk),  
-> INDEX matakuliah_FKIndex1(dosen_id_dosen),  
-> INDEX matakuliah_FKIndex2(kelas_kode_ruang)  
-> );  
Query OK, 0 rows affected (0.658 sec)  
  
MariaDB [datakuliah]> CREATE TABLE kelas (  
-> kode_ruang VARCHAR(20) NOT NULL,  
-> kapasitas INTEGER UNSIGNED NULL,  
-> PRIMARY KEY(kode_ruang)  
-> );  
  
MariaDB [datakuliah]> CREATE TABLE mahasiswa_has_matakuliah (  
-> matakuliah_kode_mk VARCHAR(20) NOT NULL,  
-> mahasiswa_nim_mhs VARCHAR(20) NOT NULL,  
-> PRIMARY KEY(matakuliah_kode_mk, mahasiswa_nim_mhs),  
-> INDEX mahasiswa_has_matakuliah_FKIndex1(mahasiswa_nim_mhs),  
-> INDEX mahasiswa_has_matakuliah_FKIndex2(matakuliah_kode_mk)  
-> );  
Query OK, 0 rows affected (0.505 sec)
```

Modul 5

Data Manipulation Language (DML)

A. Praktikum

1. Jalankan Xampp control panel
2. Jalankan server Apache dan Mysql
3. Buka cmd, login sebagai root ke Mysql
4. Pilih database perbankan “use perbankan”
5. Masukkan data-data kedalam data perbankan

INSERT : nasabah, cabang_bank

```
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 31
6	Satria Eka Jaya	Jl. Slamet Riyadi 45
7	Indri Hapsari	Jl. Sutoyo 5
8	Sari Murti	Jl. Pangandaran 11
9	Canka Lokananta	Jl. Tidar 86
10	Budi Murtono	Jl. Merak 22

```
10 rows in set (0.001 sec)
```

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

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

```
6 rows in set (0.001 sec)
```

INSERT rekening, nasabah_has_rekening

```
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	BRUS	6666	3000000
107	BRUS	7777	1000000
108	BRUB	0000	5000000
109	BRUB	9999	0
110	BRUY	1234	550000
111	BRUK	4321	150000
112	BRUK	0123	3000000
113	BRUY	8888	255000

```
13 rows in set (0.000 sec)
```

```
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

```
15 rows in set (0.000 sec)
```

INSERT transaksi

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

no_transaksi	id_nasabahFK	no_rekeningFK	Jenis_transaksi	tanggal	jumlah
1	3	105	debit	2009-11-10 00:00:00	50000
2	2	103	debit	2009-11-10 00:00:00	40000
3	4	101	kredit	2009-11-12 00:00:00	20000
4	3	106	debit	2009-11-13 00:00:00	50000
5	5	107	kredit	2009-11-13 00:00:00	30000
6	1	104	kredit	2009-11-15 00:00:00	200000
7	9	110	kredit	2009-11-15 00:00:00	150000
8	5	102	debit	2009-11-16 00:00:00	20000
9	3	105	kredit	2009-11-18 00:00:00	50000
10	4	107	debit	2009-11-19 00:00:00	100000
11	2	103	debit	2009-11-19 00:00:00	100000
12	1	104	debit	2009-11-19 00:00:00	50000
13	4	107	kredit	2009-11-20 00:00:00	200000
14	3	105	debit	2009-11-21 00:00:00	40000
15	1	104	kredit	2009-11-22 00:00:00	100000
16	4	101	kredit	2009-11-22 00:00:00	20000
17	2	103	debit	2009-11-22 00:00:00	50000
18	5	102	debit	2009-11-25 00:00:00	50000
19	10	108	debit	2009-11-26 00:00:00	100000
20	3	106	kredit	2009-11-27 00:00:00	50000
21	2	103	kredit	2009-11-28 00:00:00	200000
22	3	105	kredit	2009-11-28 00:00:00	100000
23	5	102	debit	2009-11-30 00:00:00	20000
24	1	104	debit	2009-12-01 00:00:00	50000
25	2	103	debit	2009-12-02 00:00:00	40000
26	4	101	debit	2009-12-04 00:00:00	50000
27	2	103	kredit	2009-12-05 00:00:00	100000
28	5	102	kredit	2009-12-05 00:00:00	200000
29	7	109	debit	2009-12-05 00:00:00	100000
30	9	110	debit	2009-12-06 00:00:00	20000

```
30 rows in set (0.001 sec)
```

UPDATE

- Nasabah dengan nama "Indri Hapsari" pindah alamat ke Jl. Slamet Riyadi 34

```
MariaDB [perbankan]> UPDATE nasabah SET alamat_nasabah = 'Jl. Slamet Riyadi 34' WHERE nama_nasabah = 'Indri Hapsari';
Query OK, 1 row affected (0.155 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

- Cabang dengan kode "BRUW" pindah ke alamat "Jl. A. Yani 23"

```
MariaDB [perbankan]> UPDATE cabang_bank SET alamat_cabang = 'Jl. A. Yani 23' WHERE kode_cabang = 'BRUW';
Query OK, 1 row affected (0.111 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

DELETE

- Nasabah dengan id 7 menutup rekeningnya

```
MariaDB [perbankan]> DELETE FROM nasabah WHERE id_nasabah = '7';
Query OK, 1 row affected (0.152 sec)
```

- Cabang Bank Rut Unit Magerlang menutup kantornya

```
MariaDB [perbankan]> DELETE FROM cabang_bank WHERE nama_cabang = 'Bank Rut Unit Magelang';
Query OK, 1 row affected (0.097 sec)
```

B. Tugas

Masukkan beberapa *record* ke setiap tabel dalam database yang telah anda buat pada modul 4. Print out hasil implementasi rancangan tersebut dan analisa hasilnya. (Masing-masing 10 record untuk tabel master dan 20 record untuk tabel transaksi)

Tabel nasabah

```
MariaDB [perbankan]> INSERT INTO `nasabah` (`id_nasabah`, `nama_nasabah`, `alamat_nasabah`) VALUES
-> (11, 'Aulia Husnia', 'Jl. Jahe 12'),
-> (12, 'Anita Rohmawati', 'Jl. Laos 31'),
-> (13, 'Karina Melati', 'Jl. Kunyit 81'),
-> (14, 'Anton Pamungkas', 'Jl. Pala 15'),
-> (15, 'Muh Prayoga', 'Jl. Ketumbar 31'),
-> (16, 'Nur Safitri', 'Jl. Pandan 45'),
-> (17, 'Rahmadi', 'Jl. Lada 5'),
-> (18, 'Galih Aprianto', 'Jl. Bawang 11'),
-> (19, 'Yuki', 'Jl. Kemiri 86'),
-> (20, 'Snowie', 'Jl. Kencur 22');
Query OK, 10 rows affected (0.069 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 31
6	Satria Eka Jaya	Jl. Slamet Riyadi 45
8	Sari Murti	Jl. Pangandaran 11
9	Canka Lokananta	Jl. Tidar 86
10	Budi Murtono	Jl. Merak 22
11	Aulia Husnia	Jl. Jahe 12
12	Anita Rohmawati	Jl. Laos 31
13	Karina Melati	Jl. Kunyit 81
14	Anton Pamungkas	Jl. Pala 15
15	Muh Prayoga	Jl. Ketumbar 31
16	Nur Safitri	Jl. Pandan 45
17	Rahmadi	Jl. Lada 5
18	Galih Aprianto	Jl. Bawang 11
19	Yuki	Jl. Kemiri 86
20	Snowie	Jl. Kencur 22

19 rows in set (0.001 sec)

Tabel cabang_bank

```
MariaDB [perbankan]> INSERT INTO `cabang_bank` (`kode_cabang`, `nama_cabang`, `alamat_cabang`) VALUES
-> ('BRUA', 'Bank Rut Unit Aceh', 'Jl. Apel 12'),
-> ('BRUD', 'Bank Rut Unit Denpasar', 'Jl. Semangka 21'),
-> ('BRUP', 'Bank Rut Unit Pekalongan', 'Jl. Pepaya 33'),
-> ('BRUG', 'Bank Rut Unit Gorontalo ', 'Jl. Srikaya 41'),
-> ('BRUJ', 'Bank Rut Unit Jakarta', 'Jl. Durian 19'),
-> ('BRUN', 'Bank Rut Unit Ngawi', 'Jl. Brokoli 46'),
-> ('BRUBB', 'Bank Rut Unit Bangka Belitung', 'Jl. Selada 25'),
-> ('BRUL', 'Bank Rut Unit Lombok', 'Jl. Tomat 64'),
-> ('BRBW', 'Bank Rut Unit Banyuwangi', 'Jl. Asparagus 17'),
-> ('BRUC', 'Bank Rut Unit Cirebon', 'Jl. Jeruk 99');
```

Query OK, 10 rows affected (0.081 sec)

Records: 10 Duplicates: 0 Warnings: 0

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

kode_cabang	nama_cabang	alamat_cabang
BRUB	Bank Rut Unit Boyolali	Jl. Ahmad Yani 45
BRUK	Bank Rut Unit Klaten	Jl. Suparman 23
BRUS	Bank Rut Unit Surakarta	Jl. Slamet Riyadi 18
BRUW	Bank Rut Unit Wonogiri	Jl. A. Yani 23
BRUY	Bank Rut Unit Yogyakarta	Jl. Anggrek 21
BRUA	Bank Rut Unit Aceh	Jl. Apel 12
BRUD	Bank Rut Unit Denpasar	Jl. Semangka 21
BRUP	Bank Rut Unit Pekalongan	Jl. Pepaya 33
BRUG	Bank Rut Unit Gorontalo	Jl. Srikaya 41
BRUJ	Bank Rut Unit Jakarta	Jl. Durian 19
BRUN	Bank Rut Unit Ngawi	Jl. Brokoli 46
BRUBB	Bank Rut Unit Bangka Belitung	Jl. Selada 25
BRUL	Bank Rut Unit Lombok	Jl. Tomat 64
BRBW	Bank Rut Unit Banyuwangi	Jl. Asparagus 17
BRUC	Bank Rut Unit Cirebon	Jl. Jeruk 99

15 rows in set (0.000 sec)

Tabel rekening

```
MariaDB [perbankan]> INSERT INTO `rekening` (`no_rekening`, `kode_cabangFK`, `Pin`, `Saldo`) VALUES
-> (201, 'BRUA', '1001', 800000),
-> (202, 'BRUD', '2002', 950000),
-> (203, 'BRUP', '3003', 150000),
-> (204, 'BRUC', '4004', 100000),
-> (205, 'BRUG', '5005', 120000),
-> (206, 'BRUJ', '6006', 130000),
-> (207, 'BRUN', '7007', 140000),
-> (208, 'BRUBB', '8008', 150000),
-> (209, 'BRUL', '9009', 160000),
-> (210, 'BRUBW', '2100', 770000);
```

Query OK, 10 rows affected (0.158 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 | BRUS          | 6666 | 3000000 |
|          107 | BRUS          | 7777 | 1000000 |
|          108 | BRUB          | 0000 | 5000000 |
|          109 | BRUB          | 9999 | 0 |
|          110 | BRUY          | 1234 | 550000 |
|          111 | BRUK          | 4321 | 150000 |
|          112 | BRUK          | 0123 | 3000000 |
|          113 | BRUY          | 8888 | 255000 |
|          201 | BRUA          | 1001 | 800000 |
|          202 | BRUD          | 2002 | 950000 |
|          203 | BRUP          | 3003 | 150000 |
|          204 | BRUC          | 4004 | 100000 |
|          205 | BRUG          | 5005 | 120000 |
|          206 | BRUJ          | 6006 | 130000 |
|          207 | BRUN          | 7007 | 140000 |
|          208 | BRUBB         | 8008 | 150000 |
|          209 | BRUL          | 9009 | 160000 |
|          210 | BRUBW         | 2100 | 770000 |
+-----+-----+-----+-----+
23 rows in set (0.000 sec)

```

Tabel nasabah_has_rekening

```

MariaDB [perbankan]> INSERT INTO `nasabah_has_rekening` (`Id_nasabahFK`, `no_rekeningFK`) VALUES
-> (11, 201),
-> (12, 202),
-> (13, 203),
-> (14, 204),
-> (15, 205),
-> (16, 206),
-> (17, 207),
-> (18, 208),
-> (19, 209),
-> (20, 210);
Query OK, 10 rows affected (0.160 sec)
Records: 10  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	201
12	202
13	203
14	204
15	205
16	206
17	207
18	208
19	209
20	210

```
25 rows in set (0.000 sec)
```

Input record tabel transaksi

```

MariaDB [perbankan]> INSERT INTO `transaksi` (`no_transaksi`, `id_nasabahFK`, `no_rekeningFK`, `Jenis_transaksi`, `tanggal`, `jumlah`) VALUES
-> (31, 19, 209, 'debit', '2010-1-10 00:00:00', 50000),
-> (32, 18, 208, 'debit', '2010-1-10 00:00:00', 40000),
-> (33, 17, 207, 'kredit', '2010-1-12 00:00:00', 20000),
-> (34, 16, 206, 'debit', '2010-1-13 00:00:00', 50000),
-> (35, 15, 205, 'kredit', '2010-1-13 00:00:00', 30000),
-> (36, 14, 204, 'kredit', '2010-1-15 00:00:00', 200000),
-> (37, 13, 203, 'kredit', '2010-1-15 00:00:00', 150000),
-> (38, 12, 102, 'debit', '2010-1-16 00:00:00', 20000),
-> (39, 11, 101, 'kredit', '2010-1-18 00:00:00', 50000),
-> (40, 12, 202, 'debit', '2010-1-19 00:00:00', 100000),
-> (41, 13, 203, 'debit', '2010-1-19 00:00:00', 100000),
-> (42, 14, 204, 'debit', '2010-1-19 00:00:00', 50000),
-> (43, 15, 205, 'kredit', '2010-1-20 00:00:00', 200000),
-> (44, 16, 206, 'debit', '2010-1-21 00:00:00', 40000),
-> (45, 17, 207, 'kredit', '2010-1-22 00:00:00', 100000),
-> (46, 18, 208, 'kredit', '2010-1-22 00:00:00', 20000),
-> (47, 19, 209, 'debit', '2010-1-22 00:00:00', 50000),
-> (48, 20, 210, 'debit', '2010-1-25 00:00:00', 50000),
-> (49, 20, 210, 'debit', '2010-1-26 00:00:00', 100000),
-> (50, 1, 106, 'kredit', '2010-1-27 00:00:00', 50000);
Query OK, 20 rows affected (0.158 sec)
Records: 20 Duplicates: 0 Warnings: 0

```

```

MariaDB [perbankan]> select * from transaksi;

```

no_transaksi	id_nasabahFK	no_rekeningFK	Jenis_transaksi	tanggal	jumlah
1	3	105	debit	2009-11-10 00:00:00	50000
2	2	103	debit	2009-11-10 00:00:00	40000
3	4	101	kredit	2009-11-12 00:00:00	20000
4	3	106	debit	2009-11-13 00:00:00	50000
5	5	107	kredit	2009-11-13 00:00:00	30000
6	1	104	kredit	2009-11-15 00:00:00	200000
7	9	110	kredit	2009-11-15 00:00:00	150000
8	5	102	debit	2009-11-16 00:00:00	20000
9	3	105	kredit	2009-11-18 00:00:00	50000
10	4	107	debit	2009-11-19 00:00:00	100000
11	2	103	debit	2009-11-19 00:00:00	100000
12	1	104	debit	2009-11-19 00:00:00	50000
13	4	107	kredit	2009-11-20 00:00:00	200000
14	3	105	debit	2009-11-21 00:00:00	40000
15	1	104	kredit	2009-11-22 00:00:00	100000
16	4	101	kredit	2009-11-22 00:00:00	20000
17	2	103	debit	2009-11-22 00:00:00	50000
18	5	102	debit	2009-11-25 00:00:00	50000
19	10	108	debit	2009-11-26 00:00:00	100000
20	3	106	kredit	2009-11-27 00:00:00	50000
21	2	103	kredit	2009-11-28 00:00:00	200000
22	3	105	kredit	2009-11-28 00:00:00	100000
23	5	102	debit	2009-11-30 00:00:00	20000
24	1	104	debit	2009-12-01 00:00:00	50000
25	2	103	debit	2009-12-02 00:00:00	40000
26	4	101	debit	2009-12-04 00:00:00	50000
27	2	103	kredit	2009-12-05 00:00:00	100000
28	5	102	kredit	2009-12-05 00:00:00	200000
29	7	109	debit	2009-12-05 00:00:00	100000
30	9	110	debit	2009-12-06 00:00:00	20000
31	19	209	debit	2010-01-10 00:00:00	50000
32	18	208	debit	2010-01-10 00:00:00	40000
33	17	207	kredit	2010-01-12 00:00:00	20000
34	16	206	debit	2010-01-13 00:00:00	50000
35	15	205	kredit	2010-01-13 00:00:00	30000
36	14	204	kredit	2010-01-15 00:00:00	200000
37	13	203	kredit	2010-01-15 00:00:00	150000
38	12	102	debit	2010-01-16 00:00:00	20000
39	11	101	kredit	2010-01-18 00:00:00	50000
40	12	202	debit	2010-01-19 00:00:00	100000
41	13	203	debit	2010-01-19 00:00:00	100000
42	14	204	debit	2010-01-19 00:00:00	50000
43	15	205	kredit	2010-01-20 00:00:00	200000
44	16	206	debit	2010-01-21 00:00:00	40000
45	17	207	kredit	2010-01-22 00:00:00	100000
46	18	208	kredit	2010-01-22 00:00:00	20000
47	19	209	debit	2010-01-22 00:00:00	50000
48	20	210	debit	2010-01-25 00:00:00	50000
49	20	210	debit	2010-01-26 00:00:00	100000
50	1	106	kredit	2010-01-27 00:00:00	50000

```

50 rows in set (0.001 sec)

```

Modul 6

Query Standard dan Variasinya

A. Praktikum

1. Jalankan Xampp control panel
2. Jalankan server Apache dan Mysql
3. Buka cmd, login sebagai root ke Mysql
4. Pilih database perbankan “use perbankan”
5. Tampilkan nama bank dan alamat bank untuk semua cabang bank dan diurutkan berdasarkan nama bank

SELECT nama_cabang, alamat_cabang FROM cabang_bank ORDER BY nama_cabang;

```
MariaDB [perbankan]> SELECT nama_cabang, alamat_cabang FROM cabang_bank ORDER BY nama_cabang;
```

nama_cabang	alamat_cabang
Bank Rut Unit Aceh	Jl. Apel 12
Bank Rut Unit Bangka Belitung	Jl. Selada 25
Bank Rut Unit Banyuwangi	Jl. Asparagus 17
Bank Rut Unit Boyolali	Jl. Ahmad Yani 45
Bank Rut Unit Cirebon	Jl. Jeruk 99
Bank Rut Unit Denpasar	Jl. Semangka 21
Bank Rut Unit Gorontalo	Jl. Srikaya 41
Bank Rut Unit Jakarta	Jl. Durian 19
Bank Rut Unit Klaten	Jl. Superman 23
Bank Rut Unit Lombok	Jl. Tomat 64
Bank Rut Unit Ngawi	Jl. Brokoli 46
Bank Rut Unit Pekalongan	Jl. Pepaya 33
Bank Rut Unit Surakarta	Jl. Slamet Riyadi 18
Bank Rut Unit Wonogiri	Jl. A. Yani 23
Bank Rut Unit Yogyakarta	Jl. Anggrek 21

15 rows in set (0.107 sec)

6. Tampilkan nomor rekening, pin, dan jumlah saldo dari yang paling besar ke yang paling kecil

SELECT no_rekening, pin, saldo FROM rekening ORDER BY saldo DESC;

```

MariaDB [perbankan]> SELECT no_rekening, pin, saldo FROM rekening ORDER BY saldo DESC;
+-----+-----+-----+
| no_rekening | pin  | saldo |
+-----+-----+-----+
| 108         | 0000 | 5000000 |
| 112         | 0123 | 3000000 |
| 106         | 6666 | 3000000 |
| 105         | 5555 | 2000000 |
| 107         | 7777 | 1000000 |
| 202         | 2002 | 950000  |
| 104         | 4444 | 900000  |
| 201         | 1001 | 800000  |
| 210         | 2100 | 770000  |
| 103         | 3333 | 750000  |
| 110         | 1234 | 550000  |
| 101         | 1111 | 500000  |
| 102         | 2222 | 350000  |
| 113         | 8888 | 255000  |
| 209         | 9009 | 160000  |
| 203         | 3003 | 150000  |
| 111         | 4321 | 150000  |
| 208         | 8008 | 150000  |
| 207         | 7007 | 140000  |
| 206         | 6006 | 130000  |
| 205         | 5005 | 120000  |
| 204         | 4004 | 100000  |
| 109         | 9999 | 0        |
+-----+-----+-----+
23 rows in set (0.001 sec)

```

B. 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
Anita Rohmawati	Jl. Laos 31
Anton Pamungkas	Jl. Pala 15
Aulia Husnia	Jl. Jahe 12
Budi Eko Prayogo	Jl. Kantil 31
Budi Murtono	Jl. Merak 22
Canka Lokananta	Jl. Tidar 86
Galih Aprianto	Jl. Bawang 11
Karina Melati	Jl. Kunyit 81
Kartika Padmasari	Jl. Manggis 15
Maryati	Jl. MT. Haryono 31
Muh Prayoga	Jl. Ketumbar 31
Nur Safitri	Jl. Pandan 45
Rahmadi	Jl. Lada 5
Sari Murti	Jl. Pangandaran 11
Satria Eka Jaya	Jl. Slamet Riyadi 45
Snowie	Jl. Kencur 22
Suparman	Jl. Hasanudin 81
Sutopo	Jl. Jendral Sudirman 12
Yuki	Jl. Kemiri 86

```
19 rows in set (0.001 sec)
```

2. Tampilkan jenis transaksi dan jumlah transaksi dimana jenis transaksinya 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	20000
kredit	30000
kredit	200000
kredit	150000
kredit	50000
kredit	200000
kredit	100000
kredit	20000
kredit	50000

```
22 rows in set (0.002 sec)
```

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

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

jenis_transaksi	jumlah
debit	40000

1 row in set (0.129 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
Rahmadi	kredit	20000
Anita Rohmawati	debit	20000
Galih Aprianto	kredit	20000

8 rows in set (0.102 sec)

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

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

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

2 rows in set (0.001 sec)

6. Menampilkan table nasabah dimana nama nasabah “yuki” dan alamat nasabah “Jl. Kemiri 86”

```
MariaDB [perbankan]> SELECT * FROM nasabah WHERE nama_nasabah = 'Yuki' AND alamat_nasabah = 'Jl. Kemiri 86';
```

id_nasabah	nama_nasabah	alamat_nasabah
19	Yuki	Jl. Kemiri 86

1 row in set (0.220 sec)

7. Jika kita ingin menampilkan dimana kondisi jenis transaksi 'debit', tanggal '2009-11-19' dan jumlah '100000'

```
MariaDB [perbankan]> SELECT * FROM transaksi WHERE jenis_transaksi = 'debit' AND tanggal = '2009-11-19' AND jumlah = '100000';
```

no_transaksi	id_nasabahFK	no_rekeningFK	Jenis_transaksi	tanggal	jumlah
10	4	107	debit	2009-11-19 00:00:00	100000
11	2	103	debit	2009-11-19 00:00:00	100000

2 rows in set (0.197 sec)

8. Menampilkan table nasabah dengan 2 baris setelah melewati 2 baris pertama

```
MariaDB [perbankan]> SELECT * FROM nasabah ORDER BY id_nasabah LIMIT 2 OFFSET 2;
```

id_nasabah	nama_nasabah	alamat_nasabah
3	Suparman	Jl. Hasanudin 81
4	Kartika Padmasari	Jl. Manggis 15

```
2 rows in set (0.086 sec)
```

9. Menampilkan table cabang bank dengan kondisi STRING DAN NOT NULL

```
MariaDB [perbankan]> SELECT * FROM cabang_bank WHERE kode_cabang IS NOT NULL AND nama_cabang LIKE 'Bank Rut Unit 5%';
```

kode_cabang	nama_cabang	alamat_cabang
BRUS	Bank Rut Unit Surakarta	Jl. Slamet Riyadi 18

```
1 row in set (0.150 sec)
```

10. Menampilkan table nasabah dengan pencarian string dan null menggunakan klausa order, limit dan offset

```
MariaDB [perbankan]> SELECT * FROM nasabah WHERE nama_nasabah LIKE 'Anita%' OR alamat_nasabah IS NOT NULL ORDER BY nama_nasabah LIMIT 5 OFFSET 5;
```

id_nasabah	nama_nasabah	alamat_nasabah
9	Canka Lokananta	Jl. Tidar 86
18	Galih Aprianto	Jl. Bawang 11
13	Karina Melati	Jl. Kunyit 81
4	Kartika Padmasari	Jl. Manggis 15
2	Maryati	Jl. MT. Haryono 31

```
5 rows in set (0.001 sec)
```