

**SURAT TUGAS**

388.1/F.Bishum-UTY/D/VIII/2022

Yang bertanda tangan di bawah ini :

Nama : Dr. R. Yohanes Radjaban, M.Hum.  
Jabatan : Dekan Bisnis & Humaniora  
Perguruan Tinggi : Universitas Teknologi Yogyakarta

Dengan ini memberikan tugas kepada :

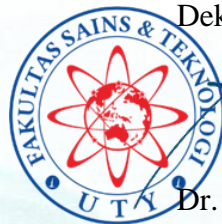
| No | Nama Penulis             | NIDN       | Dosen Pengampu Prodi             |
|----|--------------------------|------------|----------------------------------|
| 1  | Vivianti, S.Pd., M.Pd.   | 0520109001 | Pendidikan Teknologi & Informasi |
| 2  | Tri Widodo, S.T., M.Kom. | 0523128601 | Pendidikan Teknologi & Informasi |

untuk berperan aktif sebagai Penulis Modul Praltikum “Sistem Basis Data Praktik” yang akan digunakan oleh mahasiswa Prodi Pendidikan Teknologi & Informasi Program Sarjana pada semester semester ganjil T.A 2022/2023. Demi kelancaran tugas-tugas tersebut, diharapkan koordinasi, kerjasama dan bertanggungjawab kepada Kaprodi.

Demikian untuk dapat dilaksanakan sebagaimana mestinya dan mohon memberikan laporan sesudahnya. Atas perhatian dan kerjasamanya kami ucapkan terimakasih.

Yogyakarta, 1 Agustus 2022

Dekan



*[Handwritten Signature]*  
Dr. R. Yohanes Radjaban, M.Hum.  
NIK. 11 0994 016



# MODUL PRAKTIKUM SISTEM BASIS DATA



Nama Mahasiswa:

NPM:

# **SISTEM BASIS DATA**

**Oleh:**

**Vivianti**

**Tri Widodo**

**Universitas Teknologi Yogyakarta**

**2022**

# Sistem Basis Data

**Penulis:**

Vivianti, S.Pd., M.Pd.

Tri Widodo, S.T., M.Kom.

**ISBN:****Penerbit:**

Universitas Teknologi Yogyakarta

**Redaksi:**

Jl. Siliwangi, Jombor, Sleman, Yogyakarta

Email: [publikasi@uty.ac.id](mailto:publikasi@uty.ac.id)

Website: [www.uty.ac.id](http://www.uty.ac.id)

Cetakan pertama, September 2022

Hak Cipta @2022 pada penulis

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun tanpa ijin dari penulis

## KATA PENGANTAR

Dengan mengucapkan syukur kepada Allah SWT yang senantiasa melimpahkan rahmat, hidayah, inayah serta karunia-NYA kepada penyusun, sehingga penyusun dapat menyelesaikan modul Sistem Basis Data ini.

Modul ini disusun untuk membantu mahasiswa dalam kegiatan praktikum sistem basis data di program studi Pendidikan Teknologi Informasi Universitas Teknologi Yogyakarta. Modul praktikum ini berisi serangkaian aktivitas yang mengharuskan mahasiswa mampu menggunakan *tools* DBMS dan memahami perintah-perintah Structure Query Language (SQL). Dengan memiliki kemampuan tersebut diharapkan akan menjadi modal dasar bagi mahasiswa dalam mengembangkan suatu aplikasi basisdata pada sebuah sistem informasi untuk menyelesaikan suatu permasalahan yang ada di masyarakat.

Penyusun berharap semoga modul ini bermanfaat bagi dosen maupun mahasiswa. Terima kasih penyusun sampaikan kepada semua pihak yang telah membantu dan memberikan kritik maupun saran yang membangun demi terwujudnya modul pembelajaran ini.

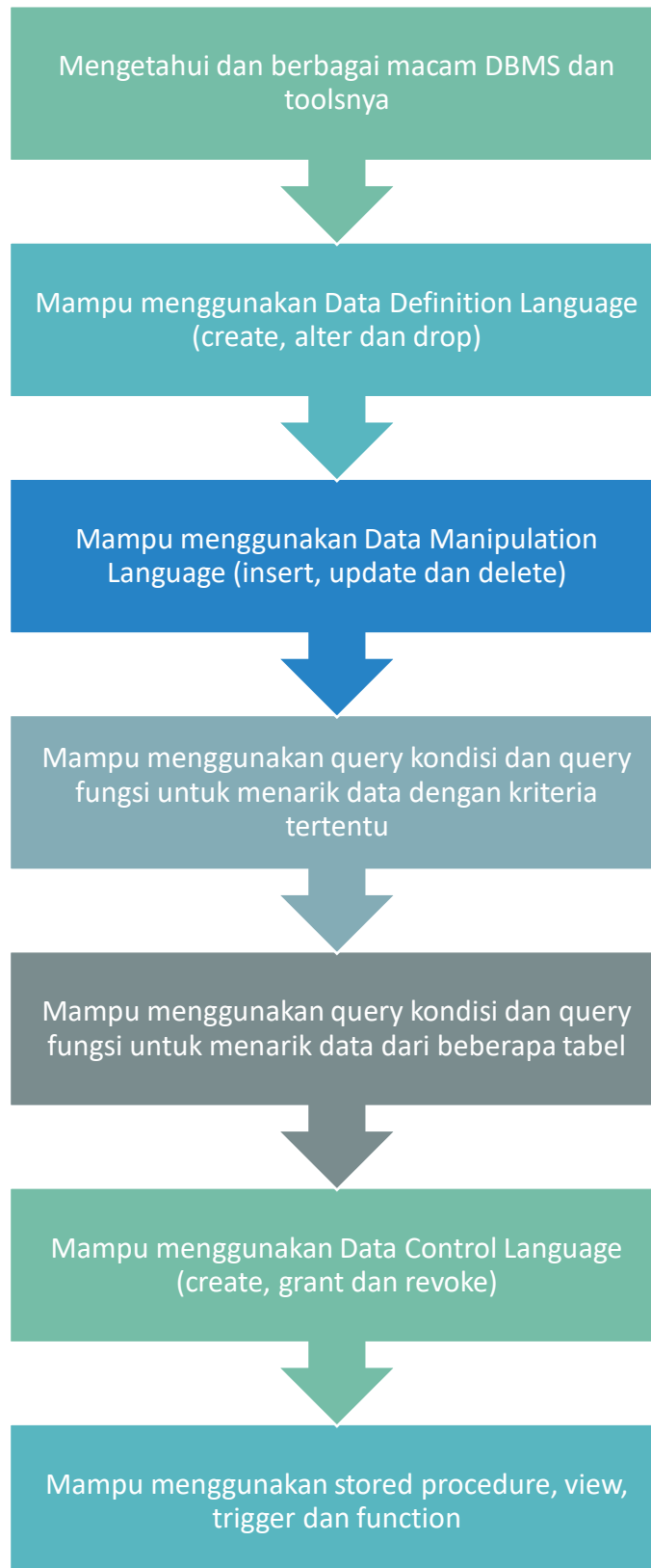
Yogyakarta, September 2022

Penyusun

## DAFTAR ISI

|   |     |
|---|-----|
| KATA PENGANTAR .....  | iii |
| BAB 1_PENDAHULUAN.....                                      | 1   |
| BAB 2_MODEL DATA ENTITY RELATIONSHIP .....                  | 4   |
| BAB 3_DATABASE MANAGEMENT SYSTEM (DBMS).....                | 8   |
| BAB 4_DATA DEFINITION LANGUAGE (DDL).....                   | 15  |
| BAB 5_DATA MANIPULATION LANGUAGE (DML).....                 | 20  |
| BAB 6_QUERY KONDISI DAN QUERY FUNGSI .....                  | 24  |
| BAB 7_QUERY RELASI (ACCESSING DATA IN MULTIPLE TABLES)..... | 30  |
| BAB 8_PENGAYAAN.....  | 38  |
| BAB 9_DATA CONTROL LANGUAGE (DCL).....                      | 41  |
| BAB 10_STORED PROCEDURE .....                               | 47  |
| BAB 11_VIEW.....  | 50  |
| BAB 12_TRIGGER .....  | 53  |
| BAB 13_FUNCTION .....                                       | 57  |
| DAFTAR PUSTAKA .....  | 60  |

## PETA KOMPETENSI



# BAB 1

## PENDAHULUAN



### 1.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Memahami pengertian sistem basis data
- Memahami penggunaan sistem basis data



### 1.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

- Mampu menjelaskan tentang sistem basis data
- Mampu menjelaskan penggunaan sistem basis data



### 1.3. URAIAN MATERI

#### 1.3.1. PENGANTAR

Sistem Basis data muncul untuk menanggapi permasalahan manajemen komputerisasi data. Sebagai contoh dalam kasus komputerisasi data dalam organisasi universitas dapat menyimpan informasi tentang data dosen, data mahasiswa, data fakultas, data matakuliah dan lain sebagainya.

#### 1.3.2. KRITERIA BASIS DATA

Kriteria yang ada pada basis data :

- Berorientasi pada data dan bukan pada program yang akan menggunakannya.
- Data dalam basis data dapat berkembang dengan mudah. Maksud berkembang di sini adalah berkembang dalam hal volume maupun strukturnya.
- Data yang ada dapat memenuhi kebutuhan sistem baru secara mudah.
- Data dapat digunakan dengan cara berbeda-beda.
- Kerangkapan data (*data redundancy*) minimal

#### 1.3.3. ASPEK PENTING BASIS DATA

Dalam penyusunan basis data ada beberapa batasan/aturan yang harus ditaati. Batasan tersebut berhubungan dengan 5 aspek penting yang harus diperhatikan, yaitu:

- Kerangkapan data = data muncul secara berulang pada file basis data
- Inkonsistensi data = munculnya data yang tidak konsisten pada kolom yang sama dalam satu atau beberapa file data yang dihubungkan



- c. Data terisolasi = program aplikasi tidak dapat mengakses data tertentu pada basis data
- d. Keamanan data = perlindungan data dalam basis data
- e. Integritas data = meyakinkan bahwa data-data yang tersimpan dalam basis data selalu berada dalam kondisi yang benar, up to date, konsisten dan selalu tersedia.



#### 1.4. TUGAS

Jawab pertanyaan di bawah ini

- a. Jelaskan pengertian sistem basis data! (sertakan sumber jika mengutip dari pernyataan orang lain).
- b. Menurut anda, seberapa pentingkah penggunaan basis data saat ini? Berikan contoh pentingnya penggunaan basis data.
- c. Jelaskan perkembangan basis data pada saat ini!

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 2

### MODEL DATA ENTITY RELATIONSHIP



#### 2.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- a. Memahami pengertian dan fungsi ERM
- b. Memahami perancangan basis data menggunakan ERD
- c. Memahami penentuan relasi tabel dan penentuan atribut dari masing-masing entitas
- d. Memahami penentuan primary key (PK) dan foreign key (FK)



#### 2.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

- a. Mampu menjelaskan tentang pengertian dan fungsi ERD
- b. Mampu merancang basis data menggunakan ERD
- c. Mampu membuat relationship antar tabel
- d. Mampu menentukan primary key (PK) dan foreign key (FK)



#### 2.3. URAIAN MATERI

##### 2.3.1. MODEL DATA ENTITY RELATIONSHIP

Entity Relationship Model merupakan model data yang dikembangkan berbasis obyek. Entity Relationship Model digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logik. Entity Relationship Model digambarkan dalam bentuk diagram yang disebut Entity Relationship Diagram. Untuk menggambarkan Entity Relationship Diagram digunakan simbol-simbol grafis tertentu.

##### 2.3.2. KOMPONEN ENTITY RELATIONSHIP DIAGRAM

Sebuah Entity Relationship Diagram tersusun atas 3 komponen, yaitu entitas, atribut, dan kerelasian antar entitas.

###### a. Entitas

Entitas merupakan obyek-obyek dasar yang terkait di dalam sistem. Obyek dasar dapat berupa benda, orang atau hal apapun yang keterangannya perlu disimpan dalam basis data. Entitas digambarkan dengan simbol persegi panjang. Dalam pembuatan basis data entitas ini nantinya akan menjadi sebuah tabel.

## b. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada suatu entitas yang perlu disimpan dalam basis data. Atribut dinyatakan dalam simbol elips.

Dalam basis data, atribut ini nantinya akan menjadi *field*/kolom dari setiap tabel yang didefinisikan dengan entitas. Kolom yang memiliki nilai paling unik nantinya bisa dinyatakan sebagai Primary Key (Kunci Utama). Field yang menjadi Primary Key pada satu tabel bisa menjadi kunci pada tabel lain, yang disebut Foreign Key (Kunci Tamu).

## c. Kerelasian Antar Entitas

Kerelasian antar entitas mendefinisikan hubungan antara dua buah entitas. Kerelasian dinyatakan dalam bentuk belah ketupat. Dalam basis data, hubungan relasi ini nantinya yang menentukan relasi antar tabel.

### 2.3.3. KARDINALITAS

Pemetaan kardinalitas mengungkapkan jumlah hubungan relasi antara satu entitas dengan entitas lainnya. Pemetaan kardinalitas berfungsi untuk menentukan relasi antar tabel di database. Pemetaan kardinalitas terdiri dari:

#### a. One to one (relasi satu ke satu)

Dalam relasi dari satu ke satu merupakan pernyataan bahwa setiap record dalam tabel A hanya dapat memiliki satu *record* yang bersesuaian dalam tabel B, dan sebaliknya.

#### b. One to many (relasi satu ke banyak)

Relasi satu ke banyak adalah jenis relasi yang paling umum atau paling banyak digunakan. Dalam relasi satu ke banyak menyatakan sebuah record dalam tabel A dapat memiliki banyak record bersesuaian dalam tabel B, tetapi sebuah record dalam tabel B hanya memiliki sebuah record yang bersesuaian dalam tabel A.

#### c. Many to one (relasi banyak ke satu)

Relasi banyak ke satu menyatakan beberapa record dalam tabel A memiliki sebuah record yang bersesuaian dalam tabel B. Sebaliknya, record dalam tabel B dapat memiliki banyak record bersesuaian dalam tabel A.

#### d. Many to many (relasi banyak ke banyak)

Relasi many to many berarti sebuah record dalam tabel A dapat memiliki banyak record yang bersesuaian dalam tabel B dan sebuah record dalam tabel B dapat memiliki banyak record yang bersesuaian dalam tabel A.

#### 2.3.4. KEY

Key merupakan satu atau gabungan atribut bersifat unik yang digunakan untuk mengidentifikasi setiap record dalam relasi. Ada beberapa macam key dalam basis data.

- a. Simple Key: Kunci relasi yang tersusun atas satu atribut. Simple key terjadi bila sifat unik telah dapat terpenuhi dengan menggunakan sebuah atribut saja.
- b. Composite Key: Kunci yang tersusun atas gabungan atribut, terjadi bila tidak dipenuhi oleh sebuah atribut tetapi harus menggabungkan lebih dari satu/beberapa atribut.
- c. Candidate Key: Satu atau gabungan atribut yang bersifat unik yang dapat digunakan untuk membedakan setiap record dalam relasi.
- d. Primary Key: Salah satu dari Candidate Key yang dipilih sebagai kunci utama untuk membedakan setiap record dalam relasi.
- e. Foreign Key: Satu/gabungan sembarang atribut yang menjadi Primary Key pada suatu tabel dan menjadi key di tabel lainnya.
- f. Alternate Key: Bagian dari Candidate Key yang tidak dipilih sebagai Primary Key.



#### 2.4. TUGAS

Buatlah rancangan diagram ER untuk sebuah sistem informasi akademik sebuah universitas.

Rancangan meliputi:

- a. Penentuan entitas yang terkait
- b. Hubungan antar entitas beserta kardinalitasnya
- c. Relasi tabel

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 3

### DATABASE MANAGEMENT SYSTEM (DBMS)



#### 3.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Memahami pengertian dan fungsi DBMS
- Mengetahui jenis DBMS



#### 3.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

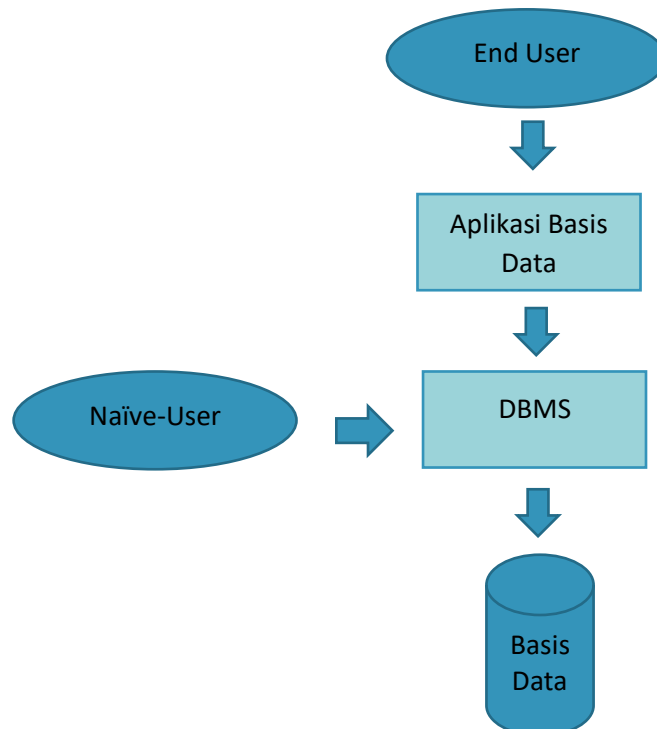
- Mampu memahami pengertian dan fungsi DBMS
- Mampu mengetahui jenis-jenis DBMS



#### 3.3. URAIAN MATERI

##### 3.3.1. DATABASE DATABASE MANAGEMENT SYSTEM (DBMS)

Database Management System (DBMS) Merupakan perangkat lunak yang didesain untuk melakukan penyimpanan dan pengaturan basis data. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan data, dll. Secara garis besar penggunaan DBMS seperti terlihat pada bagan berikut:



DBMS juga digunakan untuk melakukan proses manipulasi (input, update, delete) terhadap data dalam tabel. DBMS juga menjembatani antara basis data dengan pengguna. Beberapa software atau perangkat lunak DBMS yang sering digunakan antara lain:

- a. Microsoft SQL Server - <http://www.microsoft.com/sql/>
- b. Oracle - <http://www.oracle.com>
- c. MySQL – <http://www.mysql.com>
- d. Microsoft Acces - <https://www.microsoft.com/>

### 3.3.2. MySQL

My SQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang multithread dan multi-user. My SQL adalah implementasi dari sistem manajemen basis data relasional (RDBMS).

Beberapa Fitur yang ada di Database MySQL:

- a. RDBMS
- b. Mendukung Views
- c. Arsitektur Client-Server
- d. Free Download
- e. Stabil
- f. Keamanan yang baik

Secara Umum terdapat 3 jenis perintah SQL yaitu :

- a. DDL (Data Definition Language)

DDL merupakan perintah SQL yang berhubungan dengan pendefinisian suatu struktur database, yaitu database dan table. Beberapa perintah dasar yang termasuk DDL ini antara lain: CREATE, ALTER, RENAME dan DROP.

- b. DML (Data Manipulation Language)

DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data. Perintah SQL yang termasuk dalam DML antara lain: SELECT, INSERT, UPDATE dan DELETE.

- c. DCL (Data Control Language)

DCL merupakan perintah SQL yang berhubungan dengan manipulasi user dan hak akses (priviledges). Perintah SQL yang termasuk dalam DCL adalah GRANT dan REVOKE.



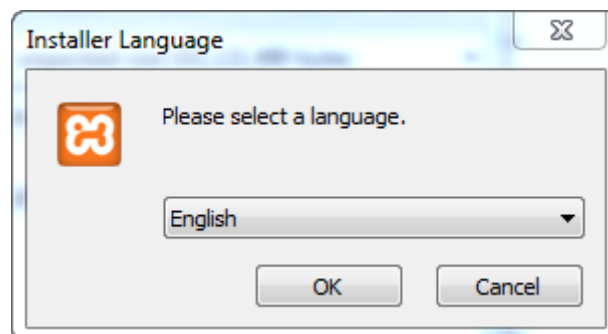
### 3.3.3. XAMPP

XAMPP adalah sebuah paket perangkat lunak web server apache yang di dalamnya juga sudah menyediakan paket MySQL. XAMPP merupakan software yang mudah digunakan dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya dari penggunaan XAMPP adalah tersedianya paket yang cukup lengkap Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya hanya dengan satu kali instalasi. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.

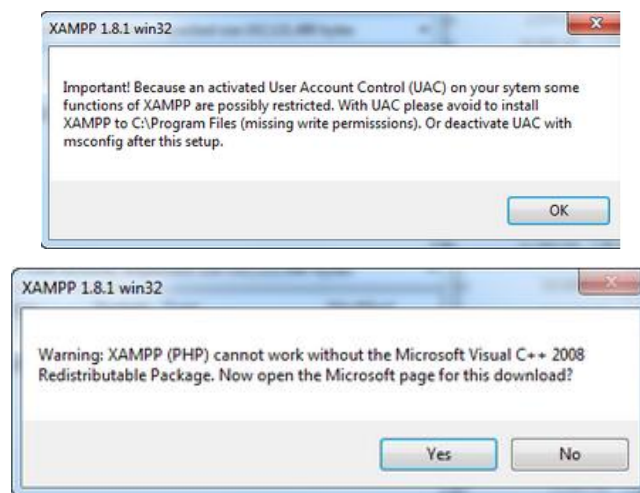
#### a. Menginstal XAMPP

Berikut ini langkah untuk menginstal XAMPP:

- 1) Buka file installer XAMP, selanjutnya akan muncul jendela “installer language” seperti pada gambar di bawah ini. Pilih bahasa, sebaiknya gunakan Bahasa Inggris (English). Klik OK.



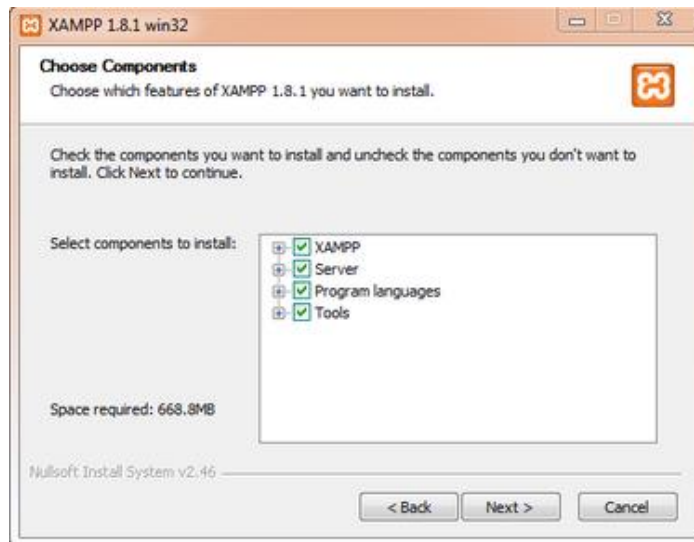
- 2) Pada proses ini terkadang muncul pesan *error*. Jika ada, abaikan saja dan lanjutkan dengan klik OK dan YES.



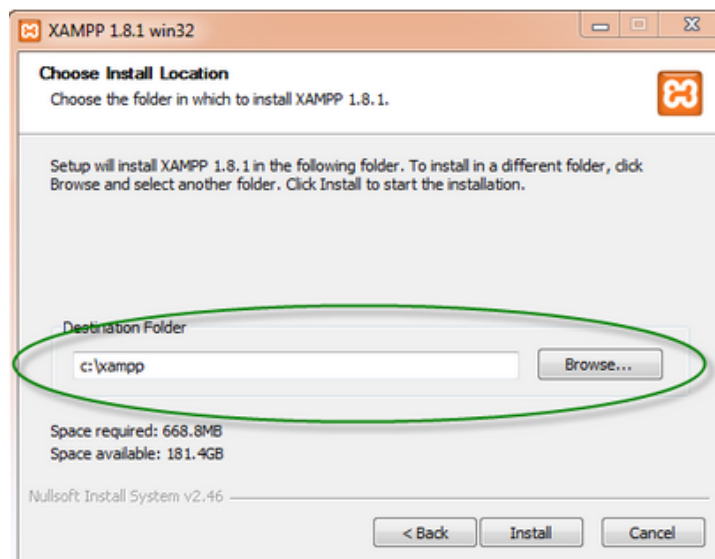
- 3) Selanjutnya akan muncul jendela yang isinya meminta untuk menutup semua aplikasi yang sedang berjalan. Tutup aplikasi lain yang sedang berjalan, lalu klik tombol **Next**.



4) Centang aplikasi yang akan diinstal dan klik tombol Next.



5) Tahap selanjutnya adalah menentukan lokasi penyimpanan. Secara default penyimpanan akan diarahkan ke lokasi **c:\xampp**. Klik tombol **Install**.



6) Tunggu hingga proses instalasi selesai lalu klik tombol Finish untuk menyelesaikannya.



### b. Menjalankan MySQL pada XAMPP

Berikut ini cara menjalankan MySQL pada XAMPP:

- 1) Bukalah aplikasi XAMPP melalui **Start Menu** atau **Desktop**.
- 2) Setelah terbuka, klik tombol **Start** pada kolom **Action** sehingga tombol tersebut berubah menjadi **Stop**. Untuk menjalankan MySQL cukup klik tombol **start** pada aplikasi Apache dan MySQL.



### c. Mengakses PHP MyAdmin

Berikut ini cara untuk mengakses PHP MyAdmin

- 1) Buka browser. Ketikkan **http://localhost/xampp** di address bar dan tekan enter.
- 2) Klik link PHP MyAdmin yang tersedia di bagian kiri laman
- 3) PHP MyAdmin siap untuk digunakan

### 3.3.4. Database Manager

Aplikasi database manager diperlukan untuk mempermudah proses manajemen database yang bisa digunakan untuk melihat, mengubah, menghapus, ataupun membuat data ke dalam database. Banyak pengembang yang membuat database manager untuk relational database atau basis data relasional, di antaranya adalah SQL Yog dan SQL Workbench.



## 3.4. TUGAS

- a. Instal MySQL pada komputer kalian, explore fungsi-fungsi yang ada pada PHP MyAdmin.
- b. Instal SQL Yog dan explore fungsi tools yang ada di dalamnya.
- c. Buat laporan, tempelkan screen shoot langkah nya

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 4

### DATA DEFINITION LANGUAGE (DDL)



#### 4.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Mengenal SQL
- Memahami penggunaan SQL
- Memahami perintah create, alter, drop
- Memahami cara pembuatan database dan tabel serta Mendefinisikan atribut database dan tabel menggunakan syntax SQL



#### 4.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa mampu menggunakan perintah-perintah SQL untuk membuat dan mendefinisikan atribut database dan tabel.



#### 4.3. URAIAN MATERI

##### 4.3.1. DATA DEFINITION LANGUAGE

DDL (Data Definition Language) merupakan perintah dalam SQL yang digunakan untuk menjelaskan objek dari database. DDL berfungsi untuk mendefinisikan database dan atribut (kolom) dalam tabel.

##### 4.3.2. STATEMENT DATA DEFINITION LANGUAGE

Syntax query yang termasuk dalam DDL adalah CREATE, ALTER, DROP, dan CONSTRAINT.

###### a. CREATE

Perintah yang digunakan untuk membuat database, tabel.

Contoh penggunaan:

- Membuat database:

```
CREATE DATABASE <nama_database>;
```

- Membuat tabel:

```
CREATE TABLE <nama_tabel> (  
<field_1> <tipe data>(size),  
<field_2> <tipe data>(size),  
<field_3> <tipe data>(size)  
);
```

- Membuat primary key dan foreign key pada saat membuat tabel

```
CREATE TABLE <nama_tabel> (
<field_1> <tipe data>(size) PRIMARY KEY,
<field_2> <tipe data>(size),
<field_3> <tipe data>(size),
FOREIGN KEY (<field_yang_akan_dijadikan_FK>) REFERENCES
tabel_referensi(<PK_tabel_referensi>)
);
```

Atau bisa menggunakan constraint

```
CREATE TABLE <nama_tabel> (
<field_1> <tipe data>(size) PRIMARY KEY,
<field_2> <tipe data>(size),
<field_3> <tipe data>(size),
CONSTRAINT <field_yang_akan_dijadikan_FK> FOREIGN KEY (<PK_tabel_referensi>)
REFERENCES <tabel_referensi>(<PK_tabel_referensi>)
);
```

4) Menambahkan primary key pada tabel yang sudah ada

```
ALTER TABLE nama_tabel ADD PRIMARY KEY(nama_field);
```

5) Menghapus primary key

```
ALTER TABLE nama_tabel DROP PRIMARY KEY;
```

## b. ALTER

Perintah yang digunakan untuk merubah skema kolom pada tabel.

Contoh penggunaan:

1) Menambah field/kolom:

```
ALTER TABLE <nama tabel> ADD <nama field><tipe data>;
```

2) Merubah tipe data:

```
ALTER TABLE <nama tabel> ALTER COLUMN <nama field> <tipe data baru>
```

3) Mengubah nama kolom

```
ALTER TABLE <nama_tabel> CHANGE <nama_field_lama> <nama_field_baru>
<tipe_data_baru>(<size_tipe_data> NOT NULL;
```

## c. DROP

Perintah yang digunakan untuk menghapus database atau tabel.

Contoh penggunaan:

1) Menghapus tabel

```
DROP TABLE <nama_tabel>;
```

2) Menghapus database

```
DROP DATABASE <nama_database>;
```



#### 4.4. TUGAS

Kerjakan latihan berikut ini dan buatlah laporan.

a. Buatlah database dengan nama SWALAYAN

```
create database SWALAYAN
```

b. Buatlah tabel PELANGGAN

|    | Column Name   | Data Type   | Allow Nulls              |
|----|---------------|-------------|--------------------------|
| PK | id_pelanggan  | varchar(6)  | <input type="checkbox"/> |
|    | nama          | varchar(30) | <input type="checkbox"/> |
|    | jenis_kelamin | varchar(1)  | <input type="checkbox"/> |
|    | alamat        | varchar(30) | <input type="checkbox"/> |
|    | telepon       | varchar(13) | <input type="checkbox"/> |
|    | tgl_lahir     | date        | <input type="checkbox"/> |

```
CREATE TABLE PELANGGAN (  
id_pelanggan varchar(6) PRIMARY KEY,  
nama varchar(30) NOT NULL,  
jenis_kelamin varchar(1) NOT NULL,  
alamat varchar(30) NOT NULL,  
telepon varchar(13) NOT NULL,  
tgl_lahir date NOT NULL  
)
```

c. Buatlah tabel KARYAWAN

|    | Column Name | Data Type   | Allow Nulls              |
|----|-------------|-------------|--------------------------|
| PK | id_karyawan | varchar(6)  | <input type="checkbox"/> |
|    | nama        | varchar(30) | <input type="checkbox"/> |
|    | alamat      | varchar(30) | <input type="checkbox"/> |
|    | telepon     | varchar(13) | <input type="checkbox"/> |
|    | tgl_lahir   | date        | <input type="checkbox"/> |

```
CREATE TABLE KARYAWAN (  
id_karyawan varchar(6) PRIMARY KEY,  
nama varchar(30) NOT NULL,  
alamat varchar(30) NOT NULL,  
telepon varchar(13) NULL,  
tgl_lahir date NOT NULL  
)
```



- d. Tambahkan kolom jenis\_pelanggan ke tabel PELANGGAN

```
ALTER TABLE PELANGGAN ADD jenis_pelanggan varchar(6);
```

- e. Tambahkan kolom gaji ke tabel KARYAWAN

```
ALTER TABLE KARYAWAN ADD gaji float;
```

- f. Mengubah tipe data gaji pada tabel KARYAWAN ke integer

```
ALTER TABLE KARYAWAN ALTER COLUMN gaji int;
```

- g. Menghapus tabel

- 1) buat tabel

```
CREATE TABLE beli(  
id_beli char(6) PRIMARY KEY,  
barang_dibeli varchar(20) NOT NULL,  
)
```

- 2) hapus tabel

```
DROP TABLE beli;
```

- h. Membuat tabel transaksi

```
CREATE TABLE penjualan(  
kode_penjualan VARCHAR(6) PRIMARY KEY,  
tanggal_transaksi DATE,  
id_pelanggan VARCHAR(6),  
id_karyawan VARCHAR(6),  
FOREIGN KEY (id_pelanggan) REFERENCES pelanggan(id_pelanggan),  
FOREIGN KEY (id_karyawan) REFERENCES karyawan(id_karyawan)  
);
```

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 5

### DATA MANIPULATION LANGUAGE (DML)



#### 5.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Mengenal SQL
- Memahami penggunaan SQL
- Memahami perintah insert, update, delete, select
- Memahami cara memasukkan, mengubah, dan menghapus data ke dalam tabel menggunakan syntax SQL



#### 5.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa mampu menggunakan perintah-perintah SQL untuk memasukkan, mengubah, dan menghapus data dalam tabel



#### 5.3. URAIAN MATERI

##### 5.3.1. DATA MANIPULATION LANGUAGE

DML (Data Manipulation Language) merupakan perintah dalam SQL yang memungkinkan pengguna untuk memanipulasi data dalam tabel. Yang dimaksud memanipulasi data adalah:

- Menambah data baru pada suatu tabel
- Merubah data yang sudah ada pada suatu tabel
- Menghapus data pada suatu tabel
- Pengambilan informasi yang disimpan dalam basisdata

##### 5.3.2. STATEMENT DATA MANIPULATION LANGUAGE

Syntax query yang termasuk dalam DDL adalah INSERT, UPDATE, DELETE, dan SELECT.

###### a. INSERT

Perintah INSERT digunakan untuk memasukkan/menyimpan data baru pada suatu tabel.

Contoh penggunaan:

```
INSERT INTO <nama tabel> VALUES ('<isi field_1>', '<isi field_2>', '<isi field_3>', '<isi field_4>', '<isi field_5>');
```

###### b. UPDATE

Perintah UPDATE digunakan untuk merubah data yang tersimpan pada suatu tabel.

Contoh Penggunaan:

```
UPDATE <nama tabel> SET <nama field_1=isi baru_1, nama field_2=isi baru_2,.....,
nama field_n=isi baru_n WHERE kriteria> ;
```

Pada perintah UPDATE, harus disertakan kriteria baris/data/row yang akan diubah karena jika tidak diberikan kriteria maka bisa merubah semua data yang ada pada tabel. Kriteria yang dimaksud adalah mengacu pada PRIMARY KEY di tabel yang akan diubah.

### c. DELETE

Perintah DELETE digunakan untuk menghapus data yang tersimpan pada suatu tabel.

Contoh Penggunaan :

```
DELETE FROM <nama tabel> WHERE <kriteria>;
```

Pada perintah DELETE, harus disertakan kriteria baris/data/row yang akan dihapus karena jika tidak diberikan kriteria maka bisa menghapus semua data yang ada pada tabel. Kriteria yang dimaksud adalah mengacu pada PRIMARY KEY di tabel yang akan dihapus.

### d. SELECT

Perintah SELECT hanya digunakan untuk menampilkan data yang ada di dalam suatu tabel.

Contoh Penggunaan:

```
SELECT * FROM nama_tabel;
```

Tanda bintang mengandung arti all/semua.

Jika ingin menampilkan kolom tertentu pada suatu tabel, bisa menggunakan perintah:

```
SELECT nama_kolom1, nama_kolom2, nama_kolom3
FROM nama_tabel;
```

Jika ingin menampilkan kolom tertentu pada suatu tabel dan dengan kriteria tertentu, bisa menggunakan perintah WHERE seperti berikut ini:

```
SELECT nama_kolom1, nama_kolom2, nama_kolom3
FROM nama_tabel
WHERE predikat;
```

Dimana predikat adalah suatu kondisi yang ditentukan oleh user. Predikat bisa melibatkan operasi logika AND, OR, NOT, atau operasi perbandingan =, <, >, <>.



## 5.4. TUGAS

Kerjakan latihan berikut ini dan buatlah laporannya.

- a. Buat tabel pelanggan dengan ketentuan sebagai berikut:

|   | Column Name     | Data Type   | Allow Nulls              |
|---|-----------------|-------------|--------------------------|
| 🔑 | id_pelanggan    | varchar(6)  | <input type="checkbox"/> |
|   | nama            | varchar(30) | <input type="checkbox"/> |
|   | jenis_kelamin   | varchar(1)  | <input type="checkbox"/> |
|   | alamat          | varchar(30) | <input type="checkbox"/> |
|   | telepon         | varchar(13) | <input type="checkbox"/> |
|   | tgl_lahir       | date        | <input type="checkbox"/> |
|   | jenis_pelanggan | varchar(6)  | <input type="checkbox"/> |

- b. Memasukkan data ke tabel pelanggan

```
INSERT INTO pelanggan VALUES ('PLG001', 'arlin', 'P', 'jl.sudirman
yogya', '0274777666', '1990-10-20', 'gold')
```

- c. Masukkan data pelanggan yang lain:

| id_pelanggan | nama   | jenis_kelamin | alamat                        | telepon    | tgl_lahir           | jenis-pelanggan |
|--------------|--------|---------------|-------------------------------|------------|---------------------|-----------------|
| PLG002       | burhan | L             | jl. bambu 222a                | 0274123456 | 1 Januari 1997      | gold            |
| PLG003       | cici   | P             | jl. merdeka<br>wonosari       | 0274555555 | 5 Mei 1989          | silver          |
| PLG004       | dino   | L             | gg. jati 44 bantul            | 0274111111 | 30 Desember<br>1966 | silver          |
| PLG005       | ena    | P             | jl. gedung baru<br>12b sleman | 0274191209 | 7 April 1978        | gold            |

- d. Merubah data tersimpan

```
UPDATE pelanggan SET alamat='jl.baru 222' WHERE id_pelanggan='PLG001'
```

- e. Menghapus data pelanggan dengan id\_pelanggan PLG002

```
DELETE FROM pelanggan WHERE id_pelanggan='PLG002'
```

- f. Melihat seluruh isi tabel pelanggan

```
SELECT * FROM pelanggan
```

- g. Menampilkan data nama dan jenis pelanggan dari tabel pelanggan

```
SELECT nama, jenis_pelanggan FROM pelanggan
```

- h. Menampilkan kolom nama, alamat dan telepon untuk pelanggan gold

```
SELECT nama, alamat, telepon FROM pelanggan WHERE
jenis_pelanggan='gold'
```

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 6

### QUERY KONDISI DAN QUERY FUNGSI



#### 6.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- a. Mengetahui SQL
- b. Memahami penggunaan SQL
- c. Mengetahui yang termasuk query kondisi
- d. Memahami penggunaan sintak query kondisi
- e. Memahami penggunaan sintak query fungsi



#### 6.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

- a. Mampu menjelaskan penggunaan query kondisi dan query fungsi
- b. Mampu menggunakan sintak query kondisi dan query fungsi



#### 6.3. URAIAN MATERI

##### 6.3.1. QUERY KONDISI

Query kondisi adalah perintah query yang digunakan untuk menampilkan sesuai dengan yang diinginkan. Kondisi yang dibentuk dalam perintah SQL memiliki sifat membandingkan dengan menggunakan tanda sama dengan (=). Beberapa perintah SQL yang digunakan untuk kondisi antara lain LIKE, ORDER BY, ASC/DESC, GROUP BY.

##### a. LIKE

Fungsi LIKE hampir sama seperti fungsi tanda sama dengan (=).

```
SELECT * FROM nama_tabel WHERE nama_field LIKE 'kondisi';
```

Terdapat 3 cara penempatan tanda tersebut dan masing-masing mempunyai makna yang berbeda.

##### 1) %like%

Tanda persen (%) yang diletakkan di depan dan dibelakang suatu kata atau huruf untuk kriteria tertentu, artinya menampilkan data (record) yang mengandung kata tersebut.

##### 2) %like

Tanda persen (%) yang diletakkan di depan suatu kata atau huruf untuk kriteria tertentu, artinya menampilkan data (record) yang dibelakangnya mengandung kata tersebut.

### 3) like%

Tanda persen (%) yang diletakkan dibelakang suatu kata atau huruf untuk kriteria tertentu, artinya menampilkan data (record) yang didepannya mengandung kata tersebut.

#### b. ORDER BY

Fungsi ORDER BY digunakan untuk mengurutkan sesuai yang diinginkan

Contoh penggunaan :

```
SELECT * FROM nama_tabel ORDER BY nama_field;
```

#### c. ASC / DESC

Ada 2 perintah dalam pengurutan yaitu ASC (Ascending) dan DESC (Descending). Contoh penggunaan :

```
SELECT * FROM nama_tabel ORDER BY nama_field DESC;
```

#### d. GROUP BY / HAVING

Fungsi GROUP BY digunakan untuk mengelompokkan berdasarkan persamaan isi datanya.

Contoh penggunaan:

```
SELECT nama_field FROM nama_tabel GROUP BY nama_field;
```

Jika akan menampilkan data persamaan dari fungsi agregat, maka harus ditambahkan HAVING, karena WHERE tidak dapat digunakan untuk menampilkan persamaan data hasil fungsi agregat.

Contoh penggunaan:

```
SELECT nama_field, aggregate_function(nama_kolom) FROM nama_tabel  
WHERE nama_field operator nilai GROUP BY nama_field HAVING  
aggregate_function (nama_field) nilai operator;
```



### 6.3.2. QUERY FUNGSI

Query fungsi adalah perintah query yang digunakan untuk memanipulasi suatu data yang akan ditampilkan, antara lain DATE, Aggregate, Aritmatika.

#### a. Date

Date menunjukkan tanggal. Format tanggal yang biasa digunakan yyyy-mm-dd HH:ii:ss.

dengan kata lain:

tahun:bulan:tanggal jam:menit:detik



Query untuk menunjukkan tanggal saat ini:

```
SELECT GETDATE()
```

b. Aggregate

Fungsi Aggregate digunakan untuk menghasilkan perhitungan secara cepat. Yang termasuk dalam fungsi aggregate antara lain SUM, MIN, MAX, AVG (Average), dan COUNT.

Fungsi-fungsi yang bisa digunakan dalam SQL adalah :

MAX: untuk mencari nilai maksimal dari suatu kolom

MIN: untuk mencari nilai minimal dari suatu kolom

AVG: untuk mencari nilai rata-rata

SUM: untuk mencari nilai jumlah

COUNT: untuk mencari nilai cacah

c. Aritmatika

Fungsi aritmatika digunakan untuk menghasilkan perhitungan yang diambil dari record atau perhitungan tersendiri. Yang termasuk dalam fungsi aritmatika antara lain penjumlahan (+), pengurangan (-), perkalian (\*), pembagian (/), pembagian sisa (%).

Contoh penggunaan:

```
SELECT 8+4 AS JUMLAH, 8-4 AS SELISIH, 8*4 AS PERKALIAN, 8/3 AS PEMBAGIAN,  
12%3 AS MODULUS;
```



## 6.4. LATIHAN

a. Restore database swalayan yang telah diunggah di elearning

b. Menampilkan data pelanggan yang memiliki nama belakang danang

```
SELECT * FROM pelanggan WHERE nama LIKE '%danang'
```

|   | id_pelanggan | nama           | jenis_kelamin | alamat         | telepon     | jenis_pelanggan |
|---|--------------|----------------|---------------|----------------|-------------|-----------------|
| 1 | PLG007       | samijan danang | L             | kasihan bantul | 08277777777 | silver          |

c. Menampilkan data pelanggan yang memiliki nama depan danang

```
SELECT * FROM pelanggan WHERE nama LIKE 'danang%'
```

|   | id_pelanggan | nama           | jenis_kelamin | alamat       | telepon     | jenis_pelanggan |
|---|--------------|----------------|---------------|--------------|-------------|-----------------|
| 1 | PLG006       | danang karomat | L             | sewon bantul | 08266666666 | gold            |

d. Menampilkan data pelanggan yang namanya mengandung kata 'ana'

```
SELECT * FROM pelanggan WHERE nama LIKE '%ana%'
```

|   | id_pelanggan | nama           | jenis_kelamin | alamat                    | telepon      | jenis_pelanggan |
|---|--------------|----------------|---------------|---------------------------|--------------|-----------------|
| 1 | PLG001       | pipit destiana | P             | jl. wonosari gunung kidul | 082111111111 | gold            |
| 2 | PLG002       | novita meliana | P             | wirosaban yogyakarta      | 082222222222 | gold            |
| 3 | PLG006       | danang karomat | L             | sewon bantul              | 082666666666 | gold            |
| 4 | PLG007       | samijan danang | L             | kasihan bantul            | 082777777777 | silver          |

e. Menampilkan data pelanggan urut berdasarkan nama (Ascending)

```
SELECT * FROM pelanggan ORDER BY nama
```

atau

```
SELECT * FROM pelanggan ORDER BY nama ASC
```

|   | id_pelanggan | nama           | jenis_kelamin | alamat                         | telepon      | jenis_pelanggan |
|---|--------------|----------------|---------------|--------------------------------|--------------|-----------------|
| 1 | PLG003       | buchori ahmat  | L             | jl. kalimantan yogyakarta      | 082333333333 | silver          |
| 2 | PLG006       | danang karomat | L             | sewon bantul                   | 082666666666 | gold            |
| 3 | PLG005       | eni sutanti    | P             | jl. kali progo 45 A yogyakarta | 082555555555 | silver          |
| 4 | PLG004       | muaz dahlan    | L             | pathuk gunung kidul            | 082444444444 | gold            |
| 5 | PLG002       | novita meliana | P             | wirosaban yogyakarta           | 082222222222 | gold            |
| 6 | PLG001       | pipit destiana | P             | jl. wonosari gunung kidul      | 082111111111 | gold            |
| 7 | PLG007       | samijan danang | L             | kasihan bantul                 | 082777777777 | silver          |

f. Menampilkan data pelanggan urut berdasarkan nama (Descending)

```
SELECT * FROM pelanggan ORDER BY nama DESC
```

|   | id_pelanggan | nama           | jenis_kelamin | alamat                         | telepon      | jenis_pelanggan |
|---|--------------|----------------|---------------|--------------------------------|--------------|-----------------|
| 1 | PLG007       | samijan danang | L             | kasihan bantul                 | 082777777777 | silver          |
| 2 | PLG001       | pipit destiana | P             | jl. wonosari gunung kidul      | 082111111111 | gold            |
| 3 | PLG002       | novita meliana | P             | wirosaban yogyakarta           | 082222222222 | gold            |
| 4 | PLG004       | muaz dahlan    | L             | pathuk gunung kidul            | 082444444444 | gold            |
| 5 | PLG005       | eni sutanti    | P             | jl. kali progo 45 A yogyakarta | 082555555555 | silver          |
| 6 | PLG006       | danang karomat | L             | sewon bantul                   | 082666666666 | gold            |
| 7 | PLG003       | buchori ahmat  | L             | jl. kalimantan yogyakarta      | 082333333333 | silver          |

g. Menampilkan kelompok jenis pelanggan apa saja yang ada pada data pelanggan

```
SELECT jenis_pelanggan FROM pelanggan GROUP BY jenis_pelanggan
```

|   | jenis_pelanggan |
|---|-----------------|
| 1 | gold            |
| 2 | silver          |

h. Menampilkan pembelian barang kode transaksi dan total harga pembelian dengan total harga pembelian diatas 20000

```
SELECT kode_transaksi, SUM(total) FROM detail_transaksi GROUP BY
kode_transaksi HAVING SUM(total)>20000
```

|   | kode_transaksi | (No column name) |
|---|----------------|------------------|
| 1 | TRS002         | 28500            |
| 2 | TRS006         | 51000            |

atau bisa ditambahkan AS untuk memberi nama tabel total harga

```
SELECT kode_transaksi, SUM(total) AS total_harga FROM detail_transaksi
GROUP BY kode_transaksi HAVING SUM(total)>20000
```

|   | kode_transaksi | total_harga |
|---|----------------|-------------|
| 1 | TRS002         | 28500       |
| 2 | TRS006         | 51000       |



## 6.5. TUGAS

Dengan menggunakan database yang telah digunakan pada latihan sebelumnya, kerjakan tugas berikut dan buatlah laporannya.

- a. Tampilkan rata-rata gaji karyawan
- b. Tampilkan gaji karyawan yang terendah
- c. Tampilkan gaji karyawan yang tertinggi
- d. Tampilkan jumlah karyawan yang lahir setelah tahun 1990
- e. Tampilkan daftar barang dan harganya, urutkan dari barang yang paling mahal
- f. Tampilkan data pelanggan yang beralamatkan di Bantul dan nama depannya adalah danang
- g. Tampilkan data karyawan yang lahir setelah tahun 1986, urut tanggal lahir dari yang paling muda
- h. Tampilkan id\_karyawan dari karyawan yang melayani pada tanggal 29 february 2022
- i. Tampilkan pembelian dengan jumlah item barang lebih dari 2 dalam satu transaksi
- j. Tampilkan data barang yang ada diskonnya, urut data yang diskonnya paling besar

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 7

### QUERY RELASI (ACCESSING DATA IN MULTIPLE TABLES)



#### 7.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Memahami perintah query relasi
- Memahami penggunaan sintak query relasi WHERE-AND dan JOIN



#### 7.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

- Mampu menjelaskan konsep query relasi
- Mampu menggunakan sintak query relasi WHERE-AND dan JOIN



#### 7.3. URAIAN MATERI

Query Relasi digunakan untuk mengakses satu atau lebih tabel dalam suatu database dalam waktu bersamaan. Perintah query relasi bisa dilakukan dengan 2 macam perintah, yaitu WHERE-AND dan JOIN.

##### 7.3.1. WHERE AND

Perintah query WHERE-AND biasa disebut juga Join Without Join Statement.

Sintak umum:

```
SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2 FROM nama_tabel1, nama_tabel2
WHERE nama_tabel1.key=nama_tabel2.key AND 'kondisi' ;
```

Contoh:

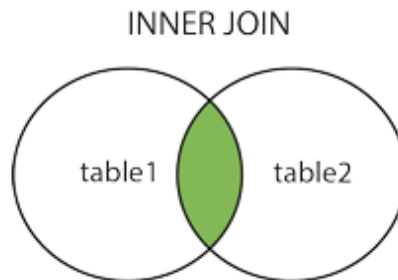
Menampilkan nama pelanggan yang melakukan transaksi pembelian pada tanggal 28 Februari 2022

```
SELECT transaksi.tgl_transaksi, pelanggan.nama FROM transaksi, pelanggan
WHERE transaksi.id_pelanggan = pelanggan.id_pelanggan AND
transaksi.tgl_transaksi='2022-02-28'
```

### 7.3.2. JOIN

#### a. INNER JOIN

INNER JOIN merupakan jenis join yang paling umum digunakan untuk menampilkan record dari beberapa tabel. Perbedaan perintah tersebut terletak pada relasinya yaitu ON (untuk JOIN) dan AND (untuk WHERE-AND).



Sintak Umum:

```
SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2 FROM nama_tabel1 INNER JOIN  
nama_tabel2 ON nama_tabel1.key=nama_tabel2.key
```

Contoh:

Menampilkan nama pelanggan yang melakukan transaksi pembelian pada tanggal 28 Februari 2022

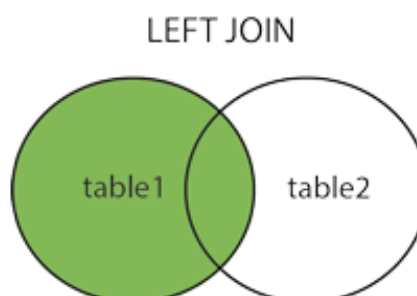
```
SELECT transaksi.tgl_transaksi, pelanggan.nama FROM transaksi INNER  
JOIN pelanggan ON transaksi.id_pelanggan = pelanggan.id_pelanggan AND  
transaksi.tgl_transaksi='2022-02-28'
```

#### b. OUTER JOIN

OUTER JOIN menghasilkan record yang ada maupun tidak ada pasangannya pada tabel lain.

##### 1) LEFT JOIN

LEFT JOIN digunakan untuk mengembalikan semua elemen data set A, terlepas dari apakah nilai kunci ada dalam data set B.



Sintak Umum:

```
SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2 FROM nama_tabel1 LEFT JOIN  
nama_tabel2 ON nama_tabel1.key=nama_tabel2.key
```

Contoh:

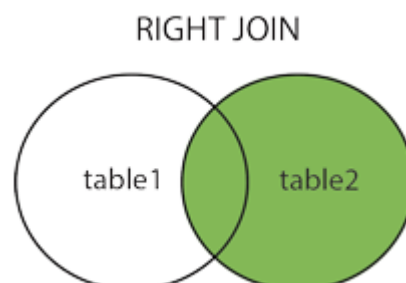
```
select barang.nama, supplier.nama from barang LEFT JOIN supplier ON  
barang.id_supplier=supplier.id_supplier
```

Hasil:

|    | nama          | nama              |
|----|---------------|-------------------|
| 1  | tensoplas     | cv. sehat sentosa |
| 2  | ciki citos    | cv. snack ring    |
| 3  | sari roti     | cv. snack ring    |
| 4  | keju asin     | cv. snack ring    |
| 5  | abon sapi     | cv. snack ring    |
| 6  | sarimi        | cv. snack ring    |
| 7  | akuah         | pt. banyu merapi  |
| 8  | oki jeli d... | pt. banyu merapi  |
| 9  | mijon         | pt. banyu merapi  |
| 10 | piring        | pt. aneka jaya    |
| 11 | tang          | pt. aneka jaya    |
| 12 | obeng         | pt. aneka jaya    |
| 13 | NULL          | pt. baru muncul   |

## 2) RIGHT JOIN

RIGHT JOIN digunakan untuk mengembalikan semua elemen data set B, terlepas dari apakah nilai kunci ada dalam data set A.



Sintak Umum:

```
SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2 FROM nama_tabel1 RIGHT JOIN  
nama_tabel2 ON nama_tabel1.key=nama_tabel2.key
```

CONTOH:

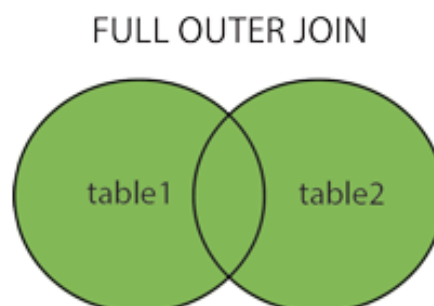
```
select barang.nama,supplier.nama from barang RIGHT JOIN supplier ON
barang.id_supplier=supplier.id_supplier
```

Hasil:

|    | nama           | nama             |
|----|----------------|------------------|
| 2  | ciki citos     | cv. snack ring   |
| 3  | sari roti      | cv. snack ring   |
| 4  | keju asin      | cv. snack ring   |
| 5  | abon sapi      | cv. snack ring   |
| 6  | sarimi         | cv. snack ring   |
| 7  | akuah          | pt. banyu merapi |
| 8  | oki jeli drink | pt. banyu merapi |
| 9  | mijon          | pt. banyu merapi |
| 10 | piring         | pt. aneka jaya   |
| 11 | tang           | pt. aneka jaya   |
| 12 | obeng          | pt. aneka jaya   |
| 13 | NULL           | pt. baru muncul  |

### c. FULL JOIN

FULL JOIN mengembalikan semua data dari set A maupun set B meskipun tidak ada pasangannya. My SQL tidak support fitur FULL JOIN. Untuk menggunakan FULL JOIN maka harus mengkombinasikan penggunaan RIGHT JOIN, UNION dan LEFT JOIN.



Sintak Umum:

```
(SELECT nama_tabel1.nama_field1, nama_tabel2.nama_field2 FROM nama_tabel1 LEFT JOIN
nama_tabel2 ON nama_tabel1.key=nama_tabel2.key UNION SELECT nama_tabel1.nama_field1,
nama_tabel2.nama_field2 FROM nama_tabel1 RIGHT JOIN nama_tabel2 ON
nama_tabel1.key=nama_tabel2.key)
```



CONTOH:

```
SELECT transaksi.tgl_transaksi, pelanggan.nama FROM
transaksi LEFT JOIN pelanggan ON transaksi.id_pelanggan =
pelanggan.id_pelanggan
UNION
SELECT transaksi.tgl_transaksi, pelanggan.nama FROM
transaksi RIGHT JOIN pelanggan ON transaksi.id_pelanggan =
pelanggan.id_pelanggan
```

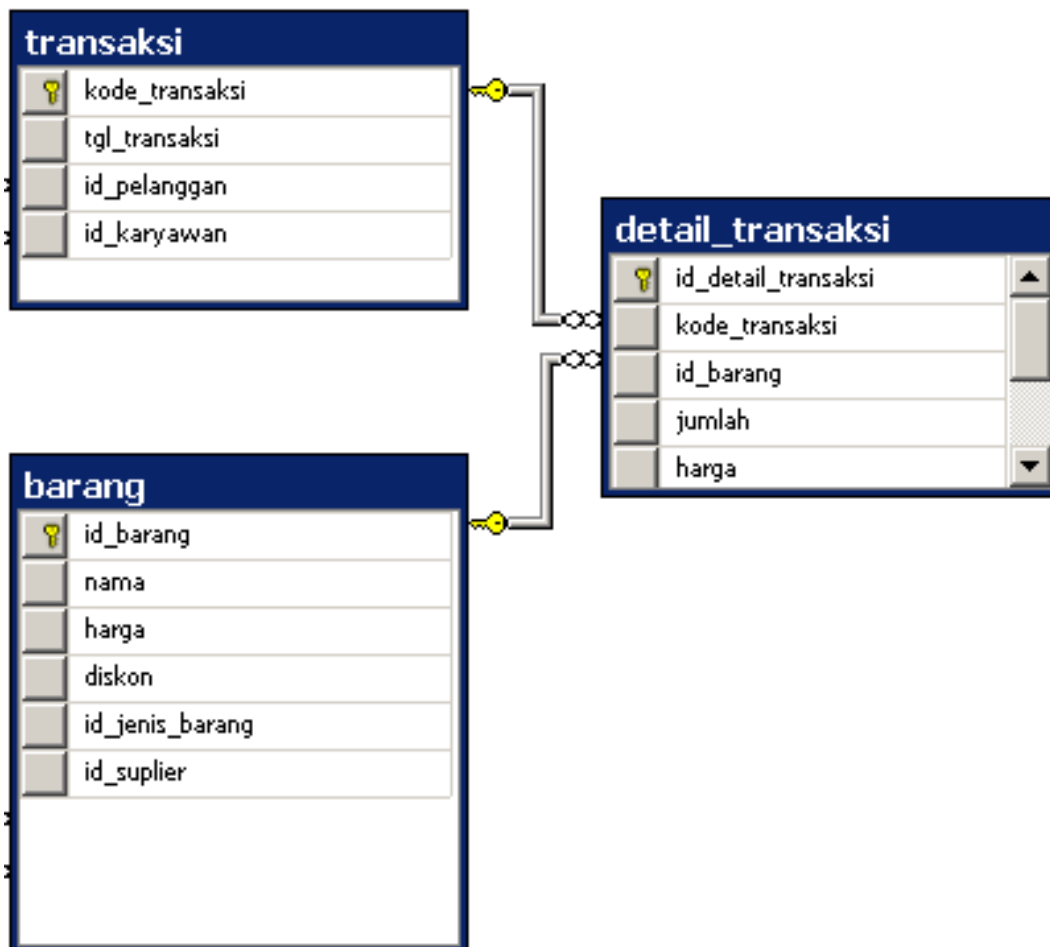
### 7.3.3. PENGGUNAAN JOIN UNTUK RELASI LEBIH DARI 2 TABEL

Berikut ini contoh penggunaan JOIN untuk relasi lebih dari 2 tabel:

Kasus:

Tampilkan kode transaksi dan nama barang yang terjual pada tanggal 28 Februari 2022.

Perhatikan tabel yang berkaitan dengan pertanyaan tersebut. Kode transaksi terdapat di tabel transaksi, nama barang di tabel barang, tanggal transaksi pada tabel transaksi. Sedangkan tabel transaksi dan tabel barang tidak berhubungan secara langsung, tapi melewati tabel detail transaksi. Untuk itu JOIN melibatkan tabel detail transaksi



Sintak:

```
select transaksi.kode_transaksi, barang.nama from barang INNER JOIN
detail_transaksi ON barang.id_barang = detail_transaksi.id_barang INNER
JOIN transaksi ON
detail_transaksi.kode_transaksi=transaksi.kode_transaksi WHERE
transaksi.tgl_transaksi='2022-03-01'
```

Hasil:

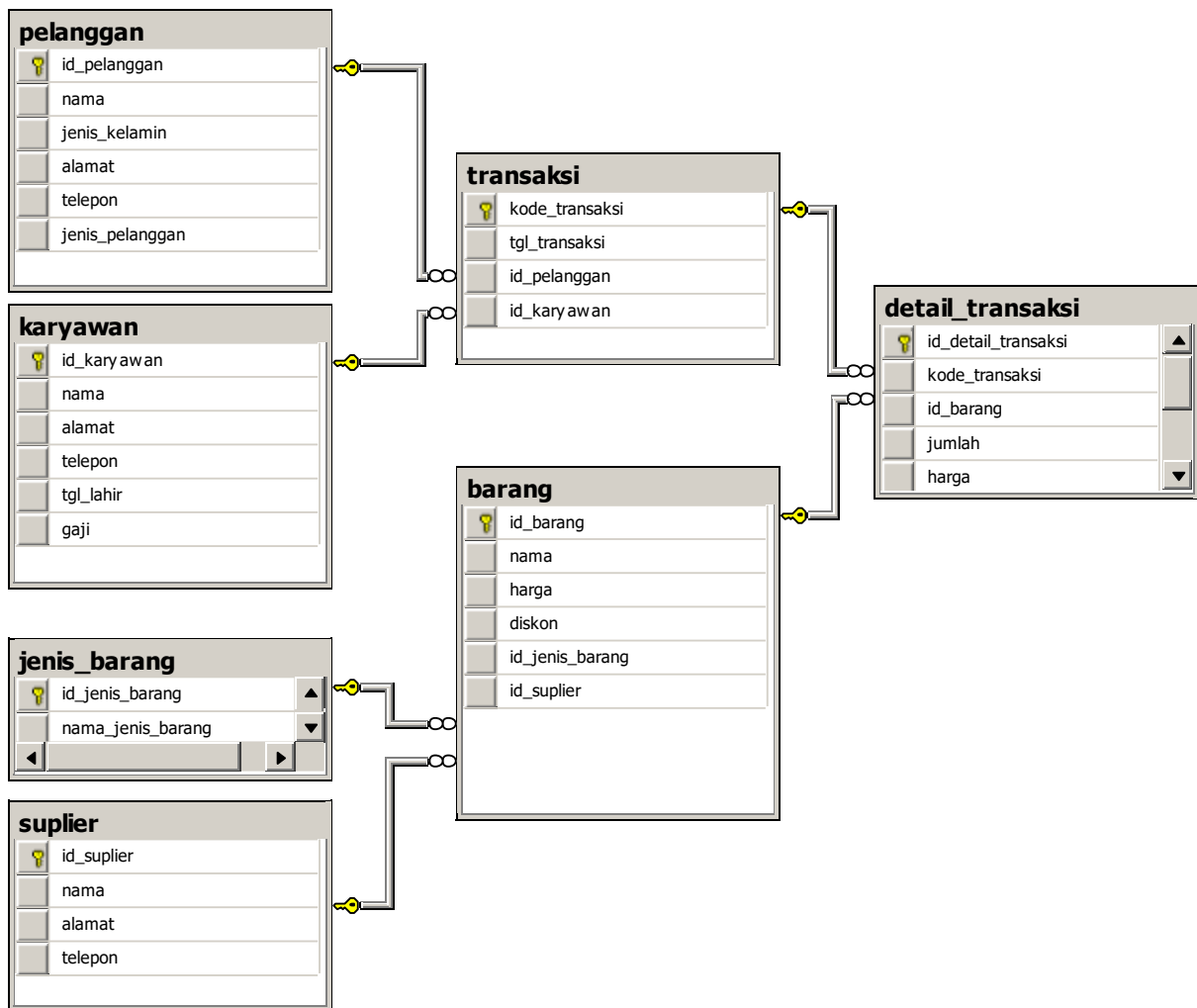
| <input type="checkbox"/> | kode_transaksi | nama       |
|--------------------------|----------------|------------|
| <input type="checkbox"/> | TRS005         | ciki citos |
| <input type="checkbox"/> | TRS005         | tensoplas  |
| <input type="checkbox"/> | TRS006         | ciki citos |
| <input type="checkbox"/> | TRS006         | keju asin  |
| <input type="checkbox"/> | TRS007         | sari roti  |



#### 7.4. TUGAS

Dengan menggunakan database yang telah digunakan pada latihan sebelumnya, kerjakan tugas berikut dan buatlah laporannya.

Perhatikan relasi pada database untuk mempermudah mengerjakan tugas.



- Tampilkan nama barang dan nama jenis barangnya
- Tampilkan nama barang dan jumlah pembelian masing-masing barang pada transaksi dengan kode TRS003
- Tampilkan nama supplier, nama barang, dan harganya untuk barang dengan harga diatas 20000
- Tampilkan nama barang, nama jenis barang, nama supplier, dan harga barang untuk barang yang berjenis makanan dan harganya lebih dari 5000
- Tampilkan nama barang, nama supplier, harga barang, persen diskon, diskon dalam rupiah dan harga barang setelah didiskon dari barang yang memiliki diskon
- Tampilkan nama pelanggan, tanggal transaksi, dan nama barang yang dibeli pada tanggal 28 Februari 2022

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 8

### PENGAYAAN



#### 8.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

Memahami penggunaan data definition language, data manipulation language, query kondisi, query fungsi dan query relasi



#### 8.2. INDIKATOR

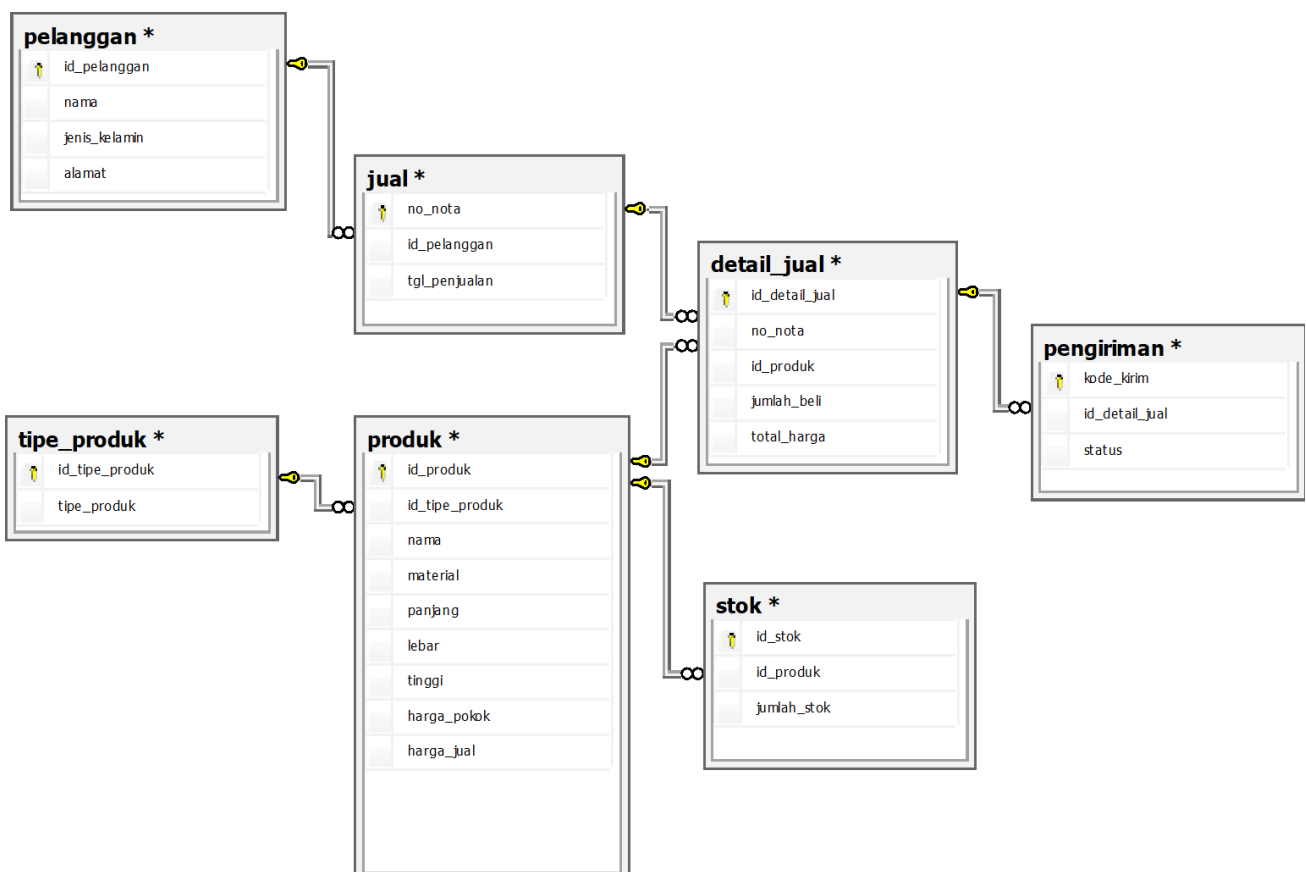
Setelah mempelajari Bab ini, mahasiswa:

Menguasai penggunaan data definition language, data manipulation language, query kondisi, query fungsi dan query relasi



#### 8.3. URAIAN KEGIATAN

Pada bab ini mahasiswa diminta untuk melakukan restore database yang telah diunggah di elearning. Perhatikan struktur database berikut, lalu kerjakan soal dan buat laporannya.





## 8.4. TUGAS

Dengan menggunakan database toko yang telah diunggah di elearning, kerjakan tugas berikut:

- a. Buatlah **tabel pemesanan** dengan field sebagai berikut:

| nama field       | tipe data  | keterangan                       |
|------------------|------------|----------------------------------|
| id_pemesanan     | varchar(6) | primary key                      |
| id_pelanggan     | varchar(6) | foreign key dari tabel pelanggan |
| id_produk        | varchar(6) | foreign key dari tabel produk    |
| jumlah_pemesanan | int        |                                  |
| tgl_pesan        | date       |                                  |
| total_bayar      | int        |                                  |

- b. Isilah tabel pemesanan dengan data berikut:

| id_pemesanan | Id_pelanggan | id_produk | jumlah_pemesanan | tgl_pesan     | total_bayar |
|--------------|--------------|-----------|------------------|---------------|-------------|
| PSN001       | PLG004       | PRD011    | 2                | 15 Maret 2022 | 7000000     |

- c. Berapakan jumlah produk yang terbuat dari bahan mahoni?
- d. Turunkan harga pokok produk yang berbahan kayu kelapa sebesar 15% dari harga semula
- e. Tampilkan nama produk dan harga jualnya yang harga jualnya kurang dari 500000
- f. Berapa jumlah seluruh barang yang belum dikirim?
- g. Ada berapa produk yang berjenis hiasan dan terjual pada tanggal 15 maret 2022?
- h. Tipe produk apa saja yang terjual pada tanggal 15 maret 2022? (perhatikan jangan ada data dobel)
- i. Jika seorang pembeli ingin membeli lemari dengan budget di bawah 2.800.000 dan panjangnya tidak lebih dari 130, maka produk apayang direkomendasikan?
- j. Jika seorang pembeli ingin membeli dipan, tapi tidak mau yang berbahan jati jepara, maka produk apa yang direkomendasikan?
- k. Seseorang memiliki kamar yang sangat sempit, ia hanya memiliki bagian kecil untuk tempat menaruh dipan. panjang bagian itu hanya 220 cm dan lebar nya 155 cm. maka dipan yang mana yang memungkinkan untuk ditaruh diruang itu?
- l. Seorang pembeli ingin membeli dua lusin kursi taman single, berapa kekurangan stoknya?
- m. barang apa yang memiliki keuntungan paling kecil? dan berapa besar keuntungannya?
- n. Ke alamat mana saja lemari harus dikirim untuk yang belum terkirim?
- o. Berapa jumlah barang yang dibeli oleh masing-masing pelanggan? tampilkan dengan urutan dari yang paling sedikit

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 9

### DATA CONTROL LANGUAGE (DCL)



#### 9.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

- Memahami perintah pembuatan user, privilege, grant, dan revoke
- Memahami cara pengaturan hak akses ke dalam database



#### 9.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

- Mampu menggunakan perintah pembuatan user, privilege, grant, dan revoke
- Mampu melakukan pengaturan hak akses ke dalam database



#### 9.3. URAIAN MATERI

Data Control Language (DCL) merupakan perintah yang dapat digunakan untuk menentukan hak akses kepada pengguna. Ada berbagai tipe hak akses yang dapat diberikan pada user, berikut diantaranya:

| Hak Akses        | Keterangan   |
|------------------|--|
| ALL [PRIVILEGES] | Memberikan seluruh hak akses, kecuali GRANT OPTION   |
| ALTER            | Hak akses untuk merubah tabel (ALTER TABLE)  |
| CREATE           | Hak akses untuk membuat tabel dan database   |
| CREATE USER      | Hak akses untuk membuat, menghapus, dan mengubah user (CREATE USER, DROP USER, RENAME USER, dan REVOKE ALL PRIVILEGES) |
| DELETE           | Hak akses untuk menghapus data   |
| DROP             | Hak akses untuk menghapus database, tabel dan view   |
| INSERT           | Hak akses untuk menambahkan data   |
| SELECT           | Hak akses untuk melihat data   |
| SHOW DATABASES   | Hak akses untuk melihat seluruh database   |
| SHUTDOWN         | Hak akses untuk mysqladmin shutdown  |
| UPDATE           | Hak akses untuk memperbaharui data   |

##### 9.3.1. MEMBUAT USER BARU

Untuk membuat user baru digunakan perintah Create User.

Sintak:

```
CREATE USER 'nama_user@'localhost' IDENTIFIED BY 'password';
```



Contoh:

```
CREATE USER 'vivi'@'localhost' IDENTIFIED BY '123';
```

Perintah di atas digunakan untuk membuat user dengan nama 'vivi dengan nama host 'localhost' dan password '123'.

### 9.3.2. GRANT

Grant digunakan untuk memberikan hak akses kepada user.

a. Sintak umum:

```
GRANT hak_akses ON nama_database.nama_tabel TO user@nama_host;
```

b. Perintah pengaturan untuk hak akses penuh

```
GRANT ALL PRIVELEGES ON *.* TO 'nama_user'@'password';
```

Contoh:

```
GRANT ALL PRIVELEGES ON *.* TO 'vivi'@'123';
```

c. Perintah pengaturan untuk hak akses tertentu kepada user

```
GRANT hak_akses1, hak_akses2, hak_akses3, ... hak_aksesn ON *.* TO  
'nama_user'@'nama_host' identified by 'password';
```

Contoh:

```
GRANT select, insert ON *.* TO 'vivi'@'localhost' identified by '123';
```

### 9.3.3. REVOKE

Revoke digunakan untuk memberikan hak akses kepada user.

a. Sintak umum :

```
REVOKE hak_akses1, hak_akses2, hak_akses3, ... hak_aksesn ON  
nama_database.nama_tabel FROM user@nama_host;
```

Contoh:

```
REVOKE select, insert ON nama_database.nama_tabel FROM  
user@nama_host;
```

- b. Perintah menghapus seluruh hak akses untuk suatu user

```
REVOKE ALL PRIVILEGES ON *.* FROM 'nama_user'@'nama_host' identified by 'password';
```

Contoh:

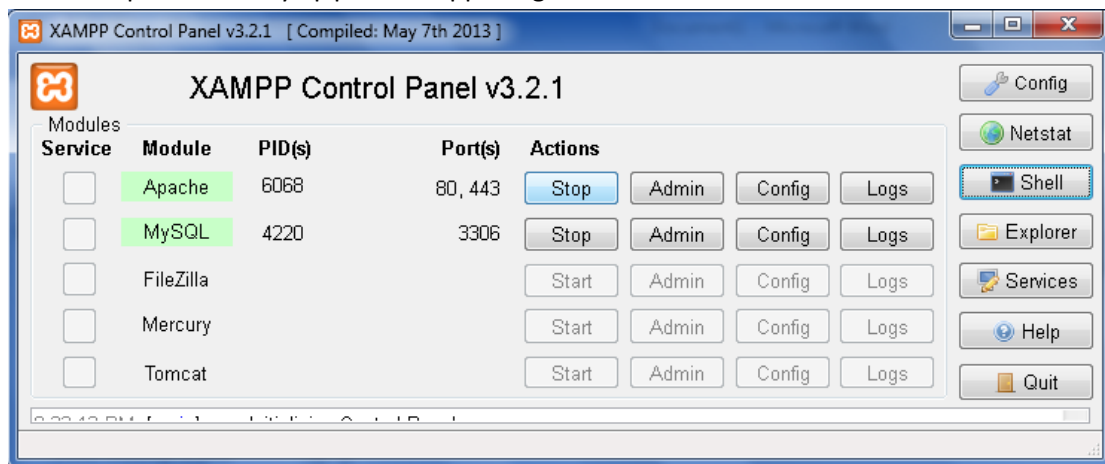
```
REVOKE ALL PRIVILEGES ON *.* FROM 'vivi'@'localhost' identified by '123';
```



## 9.4. TUGAS

Kerjakan latihan berikut dan buat laporannya.

- a. Aktifkan apache dan mysql pada xampp dengan klik tombol start



- b. Buka commad prompt  
c. masuk ke direktori c:\xamppmysql\bin

```
D:\>cd /d c:\xampp\mysql\bin
```

- d. masuk ke mysql dengan user root

```
c:\xampp\mysql\bin>mysql -u root -p
```

- e. setelah itu akan muncul permintaan password, kosongkan saja

```
Enter password:
```

- f. Tahap selanjutnya akan muncul notifikasi seperti berikut:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.6.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.
```

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

mysql>
```

- g. Buatlah sebuah database dengan nama 'perpustakaan' dengan cara mengetikkan query langsung. Jangan lupa harus diakhiri dengan ; atau \g
- h. buat database perpustakaan
- i. Buatlah sebuah tabel dengan nama tabel buku, dengan ketentuan:

| Nama kolom   | Tipe data   | Keterangan  |
|--------------|-------------|-------------|
| id_buku      | varchar(6)  | primary key |
| judul_buku   | varchar(50) |             |
| pengarang    | varchar(50) |             |
| penerbit     | varchar(30) |             |
| tahun_terbit | Int         |             |

- j. Isi dengan data seperti berikut:

| Id_buku | Judul_buku                     | pengarang | penerbit  | Tahun_terbit |
|---------|--------------------------------|-----------|-----------|--------------|
| B001    | Belajar database               | Sutopo    | Gramedia  | 2015         |
| B002    | Membangun Website untuk pemula | Yanto     | Ilmu kita | 2014         |

- k. Buat user atas nama masing-masing dan berikan password
- l. Beri hak akses **select** , **insert**, dan **update**
- m. Exit dari root

```
mysql> exit
```

akan muncul notifikasi

```
bye
```

- n. Cobalah login dengan user kalian masing-masing
- o. Gunakan database perpustakaan yang telah kalian buat tadi.
- p. Cobalah untuk menampilkan data dengan perintah **select**, apakah bisa?
- q. Cobalah untuk menginput data dengan perintah **insert**, apakah bisa?
- r. Cobalah untuk **update**/merubah salah satu data, apakah bisa?
- s. **Buatlah sebuah tabel baru** dengan nama tabel karyawan, apakah bisa?

**berikut data tabel karyawan:**

| Nama Kolom | Tipe data   | Keterangan  |
|------------|-------------|-------------|
| id_petugas | varchar(6)  | primary key |
| nama       | varchar(20) |             |

- t. Keluar dari user kalian, masuk ke root
- u. Tambahkan grant create
- v. Keluar dari root, masuk ke user kalian
- w. Jalankan lagi query untuk membuat tabel karyawan, apakah bisa?
- x. Setelah berhasil memberi hak akses, cobalah untuk menghapus hak akses dengan perintah revoke.
- y. Keluar dari user kalian, masuk ke root, hapus hak akses select untuk user kalian
- z. kembali masuk ke user kalian
- aa. Cobalah untuk menggunakan query select, apakah bisa?
- bb. Melihat hak akses juga bisa dilakukan dengan cara

```
Show grants for 'vivi'@'localhost'
```

- cc. Jelaskan mengapa user kalian tidak bisa memberi hak akses/menggunakan query grant??!

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 10

### STORED PROCEDURE



#### 10.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:

Memahami fungsi dan cara menggunakan stored procedure



#### 10.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:

Mampu menggunakan stored procedure



#### 10.3. URAIAN MATERI

##### 10.3.1. STORED PROCEDURE

Stored procedure memungkinkan sebuah aplikasi berbasis database (SQL) mendefinisikan dan kemudian memanggil prosedur melalui pernyataan SQL. Salah satu keuntungan menggunakan Stored procedure dalam aplikasi terdistribusi adalah dapat digunakannya satu perintah CALL pada aplikasi untuk mengerjakan perintah yang sama pada modul program yang berbeda.

Sintak umum:

```
DELIMITER //  
CREATE PROCEDURE <nama_procedure> (IN <nama_variabel1> tipe_data, IN  
<nama_variabel2> tipe_data, IN <nama_variabel3> tipe_data)  
BEGIN  
[perintahnya]  
END//
```

Keterangan :

<nama\_variabel1> : PARAMETER

Merupakan nama variabel yang digunakan sebagai parameter masukannya.

[perintahnya] : Dapat berupa perintah INSERT, UPDATE, DELETE, SELECT, DLL.

Dalam perintah tersebut juga dapat berisi lebih dari beberapa perintah SQL.

Cara akses:

```
CALL <nama_procedure> (data1, data2, ..., dataN)
```

CONTOH:

berikut ini contoh prosedur untuk mengisi tabel petugas dan menampilkannya

```
DELIMITER //  
CREATE PROCEDURE petugasbaru (IN kd VARCHAR(5), IN nama VARCHAR(100), IN lahir DATE)  
BEGIN  
INSERT INTO petugas VALUES (kd, nama, lahir);  
SELECT * FROM PETUGAS;  
END//
```

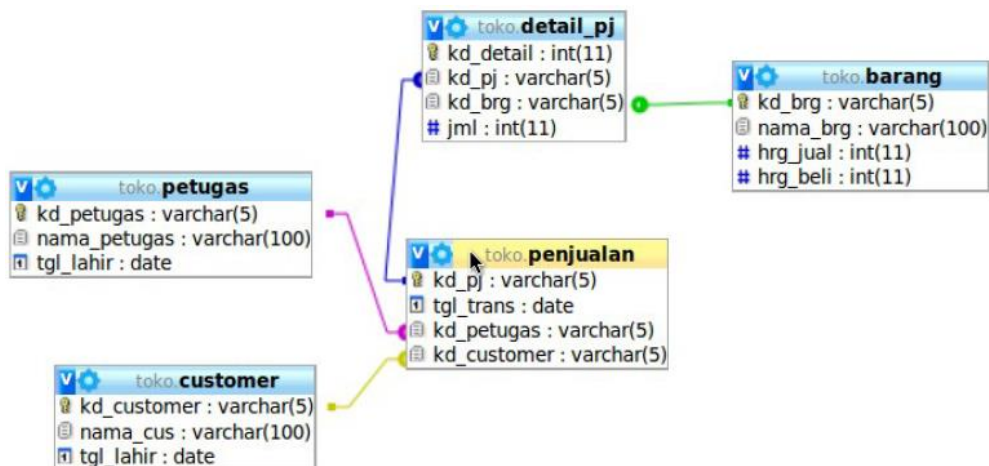
Cara akses:

```
CALL petugasbaru ('P07','Didi','1987-06-05')
```



## 10.4. TUGAS

Restore database toko yang telah diunggah di elearning. Berikut diagramnya:



Kerjakan latihan berikut dan buat laporannya di lembar yang telah disediakan.

a. Buat Store Procedure untuk :

- 1) Update tabel barang (update stok) dan menampilkan tabel data
- 2) Input tabel penjualan dan detail\_pj sekaligus,serta jumlah stok barang pada tabel barang otomatis berkurang sejumlah barang yang terjual pada transaksi tersebut.

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |



## BAB 11

### VIEW



#### 11.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:  
Memahami fungsi dan cara menggunakan view



#### 11.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:  
Mampu menggunakan view



#### 11.3. URAIAN MATERI

##### 11.3.1. VIEW

Cara membuat view:

```
CREATE VIEW <nama_view>  
AS  
[Perintahnya]
```

Cara akses:

```
Select * from <nama_view>
```

Contoh:

```
CREATE VIEW tampilharga  
AS  
Select nama_brg,hrg_jual from barang;  
Cara memanggilnya:  
Select * from tampil_harga;
```

---

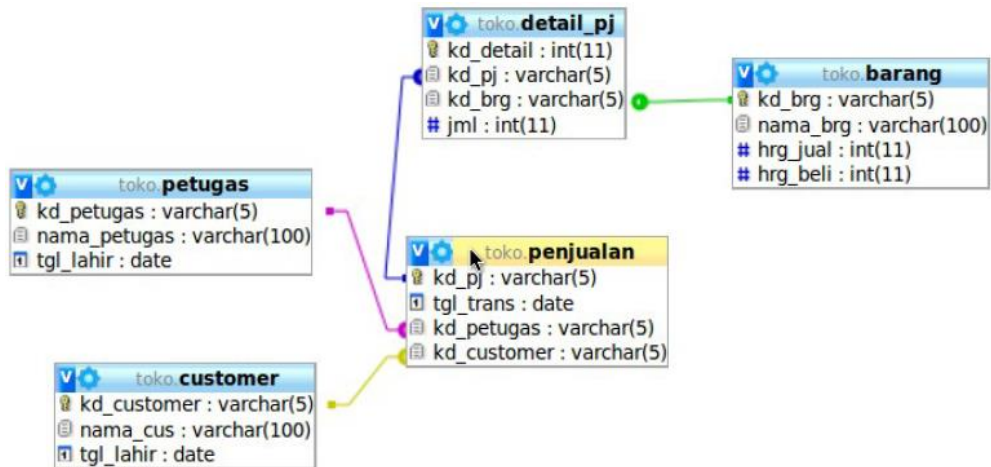
**Catatan:** contoh praktik di bab ini masih menggunakan database toko seperti bab 10.

---



## 11.4. TUGAS

Restore database toko yang telah diunggah di elearning. Berikut diagramnya:



Kerjakan latihan berikut dan buat laporannya di lembar yang telah disediakan.

b. Buatlah view untuk menampilkan:

- 1) Petugas, Customer dan penjualan

| Kd_pj | Tgl_trans | Nama_customer | Nama_petugas |
|-------|-----------|---------------|--------------|
|-------|-----------|---------------|--------------|

- 2) Tabel penjualan, detail\_pj, barang

| Kd_detail | Tgl_trans | Nama_brg | jml | keuntungan |
|-----------|-----------|----------|-----|------------|
|-----------|-----------|----------|-----|------------|

- 3) Total harga (SUM(hrg\*jml)) per transaksi penjualan :

| Kd_pj | Tanggal | Total_harga |
|-------|---------|-------------|
|-------|---------|-------------|

- 4) Keuntungan per tanggal transaksi

| Tgl_trans | keuntungan |
|-----------|------------|
|-----------|------------|

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 12

### TRIGGER



#### 12.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:  
Memahami fungsi dan cara menggunakan trigger



#### 12.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:  
Mampu menggunakan trigger



#### 12.3. URAIAN MATERI

##### 12.3.1. TRIGGER

Trigger merupakan sekumpulan perintah atau sintaks yang akan secara otomatis dijalankan jika terjadi operasi tertentu dalam tabel atau view. Trigger digunakan untuk memanggil satu atau beberapa perintah SQL secara otomatis sebelum atau sesudah terjadi proses INSERT, UPDATE atau DELETE dari suatu tabel.

Berikut ini bentuk umum perintah untuk membuat trigger:

```
CREATE TRIGGER name  
[BEFORE | AFTER] [INSERT | UPDATE | DELETE]  
ON tablename  
FOR EACH ROW statement
```

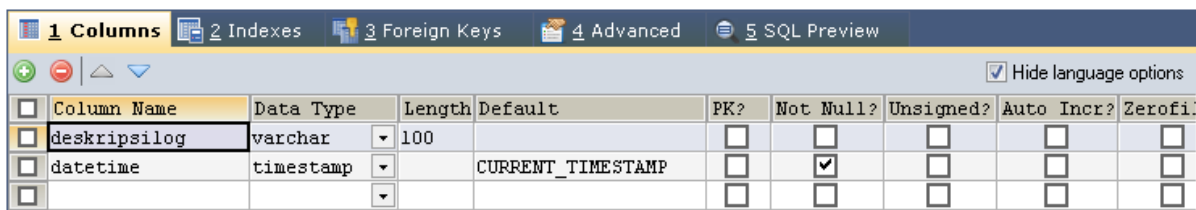
Keterangan dari bentuk umum perintah membuat trigger:

- Name** : Nama trigger mengikuti peraturan penamaan variabel / identifier dalam MySQL
- [BEFORE | AFTER]** : digunakan untuk menentukan kapan proses secara otomatis akan dieksekusi, sebelum atau sesudah proses.
- [INSERT | UPDATE | DELETE]** : digunakan untuk menentukan event (proses) yang dijadikan trigger (pemicu) untuk menjalankan perintah-perintah di dalam triggers.
- Tablename** : Nama tabel dimana trigger berada.
- Statement** : merupakan sekumpulan perintah atau query yang akan secara otomatis dijalankan jika event / proses yang didefinisikan sebelumnya aktif.

Statement atau perintah dalam trigger dapat berupa satu perintah saja, dan dapat juga beberapa perintah sekaligus. Jika terdapat beberapa perintah dalam trigger, maka gunakan perintah **BEGIN** dan **END** untuk mengawali dan mengakhiri perintah. Di dalam statement trigger, kita dapat mengakses record tabel sebelum atau sesudah proses dengan menggunakan **NEW** dan **OLD**. **NEW** digunakan untuk mengambil record yang akan diproses (insert atau update), sedangkan **OLD** digunakan untuk mengakses record yang sudah diproses (update atau delete).

contoh:

tambahkan tabel baru yaitu log yang fungsinya untuk menyimpan aktifitas yang terjadi.



| Column Name  | Data Type | Length | Default           | PK?                      | Not Null?                           | Unsigned?                | Auto                     | Incr?                    | Zerofill?                |
|--------------|-----------|--------|-------------------|--------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| deskripsilog | varchar   | 100    |                   | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| datetime     | timestamp |        | CURRENT_TIMESTAMP | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|              |           |        |                   | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Buatlah sebuah trigger yang fungsinya untuk mengisi tabel log apabila terjadi transaksi input data di tabel penjualan

```
DELIMITER //
CREATE TRIGGER isi_tabel_log
AFTER INSERT ON penjualan
FOR EACH ROW BEGIN
INSERT INTO LOG (deskripsi)
VALUES ('transaksi penjualan');
END//
```

---

*Catatan: contoh praktik di bab ini masih menggunakan database toko seperti bab 10.*

---

Untuk menghapus trigger, dapat menggunakan perintah **DROP TRIGGER** dengan diikuti dengan nama tabel dan nama trigger-nya. Berikut ini bentuk umum dan contoh perintah untuk menghapus trigger.

Bentuk umum dan contoh menghapus trigger:

**DROP TRIGGER tablename.triggername;**

Contoh menghapus trigger bernama „before\_insert“ yang ada di tabel pelanggan.

**DROP TRIGGER penjualan. isi\_tabel\_log;**



#### 12.4. TUGAS

Restore database swalayan yang pernah digunakan untuk praktikum pada bab 6 dan 7. Buatlah contoh trigger (kasus bebas). Buat laporan.

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## BAB 13

### FUNCTION



#### 13.1. KOMPETENSI DASAR

Setelah mempelajari Bab ini, mahasiswa:  
Memahami fungsi dan cara menggunakan function



#### 13.2. INDIKATOR

Setelah mempelajari Bab ini, mahasiswa:  
Mampu menggunakan function



#### 13.3. URAIAN MATERI

##### 13.3.1. FUNCTION

Function merupakan suatu kumpulan perintah atau statement yang disimpan dan dieksekusi di server database MySQL. Kita dapat menyusun program sederhana berbasis sintaks SQL untuk menjalankan fungsi tertentu. Hal ini menjadikan aplikasi yang kita buat lebih efektif dan efisien.

Dalam function dikenal istilah variabel. Variabel digunakan untuk menyimpan suatu nilai secara temporer (sementara) di memori. Variabel akan hilang saat sudah tidak digunakan lagi. Variabel dalam MySQL sebelum dapat digunakan, pertama kali harus dideklarasikan terlebih dahulu. Berikut ini bentuk umum pendeklarasian suatu variabel di MySQL:

```
DECLARE variable_name DATATYPE [DEFAULT value];
```

##### Contoh 1:

```
DECLARE jumlah INT;  
DECLARE jumlah INT DEFAULT '1';
```

Berikut ini contoh penggunaan function:

misalnya kita akan membuat function untuk menghitung jumlah barang yang terjual

```
DELIMITER //  
CREATE FUNCTION jumlahbarangdibeli(kodeproduk VARCHAR(5))  
RETURNS INT  
BEGIN  
DECLARE jumlah INT;  
SELECT SUM(jml) INTO jumlah FROM detail_pj  
WHERE kd_brg=kodeproduk;
```



```
RETURN jumlah;  
END//
```

cara memanggilnya adalah:

```
SELECT jumlahbarangdibeli('B08');
```

### contoh 2:

penggunaan function dengan seleksi kondisi

```
DELIMITER //  
CREATE FUNCTION cekpembelian (kd_pelanggan VARCHAR(5))  
RETURNS VARCHAR (100)  
BEGIN  
DECLARE jumlah INT;  
SELECT COUNT(kd_pj) INTO jumlah FROM penjualan  
WHERE kd_customer=kd_pelanggan;  
IF (jumlah > 0) THEN  
RETURN CONCAT("anda sudah bertransaksi sebanyak ",  
jumlah, " kali");  
ELSE  
RETURN "anda belum pernah melakukan transaksi";  
END IF;  
END//
```

Cara panggil:

```
SELECT cekpembelian('C01');
```



## 13.4. TUGAS

Kerjakan latihan berikut dan buat laporannya.

- a. buatlah sebuah database baru (bebas)  
dengan menggunakan database yang sudah kalian buat, buatlah:
  - 1) Procedure
  - 2) View
  - 3) Trigger
  - 4) Function(masing-masing satu contoh)
- b. Buatlah laporan yang berisi:
  - 1) Nama database
  - 2) Diagram
  - 3) Query
  - 4) Screenshoot hasil

|                            |              |         |
|----------------------------|--------------|---------|
| Lembar Jawab               | Nama         |         |
|                            | NPM          |         |
|                            | Tanda tangan |         |
|                            |              |         |
| Tanda tangan asisten/dosen |              | Tanggal |
|                            |              |         |

## DAFTAR PUSTAKA

Sutanta, E. 2011. *Basis Data dalam Tinjauan Konseptual*. Yogyakarta: Penerbit Andi.

Widodo, A. W., & Kurnianingtyas, D. 2017. *Sistem Basis Data*. Malang: Universitas Brawijaya Press.

**@2022**

Diterbitkan oleh: Universitas Teknologi Yogyakarta

Jl. Siliwangi, Jombor, Sleman, Yogyakarta

Email : [publikasi@uty.ac.id](mailto:publikasi@uty.ac.id)

Website : [uty.ac.id](http://uty.ac.id)