



## 1. Tujuan

Mempelajari bagaimana penggunaan perintah dasar SQL untuk mengelola data dengan Database MySQL.

## 2. Alat

- MySQL Server.
- MySQL Workbench
- Database dbtransaksi.sql (ERD ada di halaman terakhir materi ini) pada link berikut ini [https://drive.google.com/file/d/1wVWKbvznq5B\\_N\\_i-pbQgyvj4Unf6m2Dr/view?usp=sharing](https://drive.google.com/file/d/1wVWKbvznq5B_N_i-pbQgyvj4Unf6m2Dr/view?usp=sharing)

## 3. Persiapan

1. Pastikan sistem kalian sudah terinstall MySQL. Lakukan perintah berikut untuk mengecek versi MySQL.

```
Command Prompt
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Imam_Bhoim>mysql -V
mysql Ver 8.0.18 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\Imam_Bhoim>
```

2. Import file sql dbtransaksi.sql ke dalam database MySQL.  
Masuk ke mode MySQL

```
C:\Users\Imam_Bhoim>cd..
C:\Users>cd..
C:\>cd "Program Files"
C:\Program Files>cd MySQL
C:\Program Files\MySQL>cd "MySQL Server 8.0"
C:\Program Files\MySQL\MySQL Server 8.0>cd bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p -P 3310
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 100
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, 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>
```

3. Buat Database dbtransaksi

```
mysql> create database dbtransaksi;
Query OK, 1 row affected (0.57 sec)

mysql>
```

4. Masuk database dbtransaksi & Import file sql ke dalam database dbtransaksi.

Lalu tekan Enter

```
mysql> use dbtransaksi;
Database changed
mysql> source G:\dbtransaksi.sql

mysql> source G:\dbtransaksi.sql
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.01 sec)
```

## 4. SQL Dasar

### 1. Querying data

- SELECT FROM – menunjukkan cara menggunakan pernyataan SELECT FROM sederhana untuk mengkueri data dari satu tabel.

```
SELECT select_list
FROM table_name;
```

```
SELECT lastName
FROM employees;
```

```
SELECT
    lastName,
    firstName,
    jobTitle
FROM
    employees;
```

```
SELECT *
FROM employees;
```

- SELECT – pelajari cara menggunakan pernyataan SELECT tanpa mereferensikan tabel.

```
SELECT 1 + 1;
```

```
SELECT NOW();
```

```
SELECT CONCAT('John', ' ', 'Doe') AS name;
```

## 2. Sorting data

```
SELECT
    select_list
FROM
    table_name
ORDER BY
    column1 [ASC|DESC],
    column2 [ASC|DESC],
    ...;
```

```
SELECT
    contactLastname,
    contactFirstname
FROM
    customers
ORDER BY
    contactLastname;
```

```
SELECT
    contactLastname,
    contactFirstname
FROM
    customers
ORDER BY
    contactLastname DESC;
```

```
SELECT
    contactLastname,
    contactFirstname
FROM
    customers
ORDER BY
    contactLastname DESC ,
    contactFirstname ASC;
```

```
SELECT
    orderNumber,
    orderlinenumber,
    quantityOrdered * priceEach
FROM
    orderdetails
ORDER BY
    quantityOrdered * priceEach DESC;
```

```
SELECT
    orderNumber,
    orderLineNumber,
    quantityOrdered * priceEach AS subtotal
FROM
    orderdetails
ORDER BY subtotal DESC;
```

### 3. Filtering Data

- WHERE – pelajari cara menggunakan klausa WHERE untuk memfilter baris berdasarkan kondisi yang ditentukan

```
SELECT
    select_list
FROM
    table_name
WHERE
    search_condition;
```

```
SELECT
    lastname,
    firstname,
    jobtitle
FROM
    employees
WHERE
    jobtitle = 'Sales Rep';
```

```
SELECT
    lastName,
    firstName,
    jobTitle,
    officeCode
FROM
    employees
WHERE
    jobtitle = 'Sales Rep' OR
    officeCode = 1
ORDER BY
    officeCode ,
    jobTitle;
```

```
SELECT
    firstName,
    lastName
FROM
    employees
WHERE
    lastName LIKE '%son'
ORDER BY firstName;
```

```
SELECT
    firstName,
    lastName,
    officeCode
FROM
    employees
WHERE
    officeCode IN (1 , 2, 3)
ORDER BY
    officeCode;
```

```
SELECT
    lastname,
    firstname,
    jobtitle
FROM
    employees
WHERE
    jobtitle <> 'Sales Rep';
```

- **SELECT DISTINCT** – menunjukkan cara menggunakan operator **DISTINCT** dalam pernyataan **SELECT** untuk menghilangkan baris duplikat dalam kumpulan hasil.

```
SELECT DISTINCT
    select_list
FROM
    table_name
WHERE
    search_condition
ORDER BY
    sort_expression;
```

```
SELECT
    lastname
FROM
    employees
ORDER BY
    lastname;
```

```
SELECT
    DISTINCT lastname
FROM
    employees
ORDER BY
    lastname;
```

```
SELECT DISTINCT
    state, city
FROM
    customers
WHERE
    state IS NOT NULL
ORDER BY
    state,
    city;
```

- AND – memperkenalkan Anda ke operator AND untuk menggabungkan ekspresi Boolean untuk membentuk kondisi kompleks untuk memfilter data.

```
SELECT 1 AND 0, 0 AND 1, 0 AND 0, 0 AND NULL;
```

```
SELECT
    customername,
    country,
    state
FROM
    customers
WHERE
    country = 'USA' AND
    state = 'CA';
```

```
SELECT
    customername,
    country,
    state,
    creditlimit
FROM
    customers
WHERE
    country = 'USA' AND
    state = 'CA' AND
    creditlimit > 100000;
```

- OR – memperkenalkan Anda ke operator OR dan menunjukkan cara menggabungkan operator OR dengan operator AND untuk memfilter data.

```
SELECT 1 OR 1, 1 OR 0, 0 OR 1;
```

```
SELECT
    customername,
    country
FROM
    customers
WHERE country = 'USA' OR
       country = 'France';
```

```
SELECT
    customername,
    country,
    creditLimit
FROM
    customers
WHERE (country = 'USA'
       OR country = 'France')
       AND creditlimit > 100000;
```

```
SELECT
    customername,
    country,
    creditLimit
FROM
    customers
WHERE
    country = 'USA'
    OR country = 'France'
    AND creditlimit > 100000;
```

- IN – menunjukkan cara menggunakan operator IN dalam klausa WHERE untuk menentukan apakah suatu nilai cocok dengan nilai apa pun dalam suatu himpunan.

```
SELECT 1 IN (1,2,3);
```

```
SELECT 4 IN (1,2,3);
```

```
SELECT NULL IN (1,2,3);
```

```
SELECT 0 IN (1 , 2, 3, NULL);
```

```
SELECT
    officeCode,
    city,
    phone,
    country
FROM
    offices
WHERE
    country IN ('USA' , 'France');
```

```
SELECT
    officeCode,
    city,
    phone
FROM
    offices
WHERE
    country = 'USA' OR country = 'France';
```

- NOT IN – meniadakan operator IN menggunakan operator NOT untuk memeriksa apakah suatu nilai tidak cocok dengan nilai apa pun dalam satu set.

```
SELECT 1 NOT IN (1,2,3);
```

```
SELECT 0 NOT IN (1,2,3);
```

```
SELECT NULL NOT IN (1,2,3);
```

```
SELECT
    officeCode,
    city,
    phone
FROM
    offices
WHERE
    country NOT IN ('USA' , 'France')
ORDER BY
    city;
```

- BETWEEN – menunjukkan cara mengkueri data berdasarkan rentang menggunakan BETWEEN operator.

```
SELECT 15 BETWEEN 10 AND 20;
```

```
SELECT 15 BETWEEN 20 AND 30;
```

```
SELECT 15 NOT BETWEEN 10 AND 20;
```



```
SELECT
    productCode,
    productName,
    buyPrice
FROM
    products
WHERE
    buyPrice BETWEEN 90 AND 100;
```

```
SELECT
    productCode,
    productName,
    buyPrice
FROM
    products
WHERE
    buyPrice >= 90 AND buyPrice <= 100;
```

```
SELECT
    productCode,
    productName,
    buyPrice
FROM
    products
WHERE
    buyPrice NOT BETWEEN 20 AND 100;
```

```
SELECT
    productCode,
    productName,
    buyPrice
FROM
    products
WHERE
    buyPrice < 20 OR buyPrice > 100;
```

```
SELECT
    orderNumber,
    requiredDate,
    status
FROM
    orders
WHERE
    requireddate BETWEEN
        CAST('2003-01-01' AS DATE) AND
        CAST('2003-01-31' AS DATE);
```

- LIKE – memberi Anda teknik untuk mengkueri data berdasarkan pola.

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    firstName LIKE 'a%';
```

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    lastName LIKE '%on';
```

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    lastname LIKE '%on%';
```

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    firstname LIKE 'T_m';
```

```
SELECT
    employeeNumber,
    lastName,
    firstName
FROM
    employees
WHERE
    lastName NOT LIKE 'B%';
```

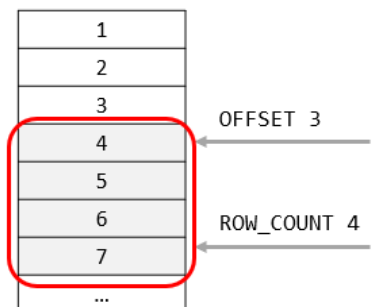
```
SELECT
    productCode,
    productName
FROM
    products
WHERE
    productCode LIKE '%\_20%';
```

```
SELECT
    productCode,
    productName
FROM
    products
WHERE
    productCode LIKE '%$_20%' ESCAPE '$';
```

- LIMIT – gunakan LIMIT untuk membatasi jumlah baris yang dikembalikan oleh pernyataan SELECT .

```
SELECT
    select_list
FROM
    table_name
LIMIT [offset,] row_count;
```

```
SELECT n FROM t
ORDER BY n
LIMIT 3, 4;
```



```
SELECT
    select_list
FROM
    table_name
ORDER BY
    sort_expression
LIMIT offset, row_count;
```

```
SELECT
    customerNumber,
    customerName,
    creditLimit
FROM
    customers
ORDER BY creditLimit DESC
LIMIT 5;
```

```
SELECT
    customerNumber,
    customerName,
    creditLimit
FROM
    customers
ORDER BY
    creditLimit,
    customerNumber
LIMIT 5;
```

```
SELECT
    customerNumber,
    customerName
FROM
    customers
ORDER BY customerName
LIMIT 10, 10;
```

```
SELECT DISTINCT
    state
FROM
    customers
WHERE
    state IS NOT NULL
LIMIT 5;
```

- IS NULL – menguji apakah suatu nilai NULL atau tidak dengan menggunakan operator IS NULL.

```
SELECT 1 IS NULL, -- 0
       0 IS NULL, -- 0
       NULL IS NULL; -- 1
```

```
SELECT 1 IS NOT NULL, -- 1
       0 IS NOT NULL, -- 1
       NULL IS NOT NULL; -- 0
```

```
SELECT
  customerName,
  country,
  salesrepemployeenumber
FROM
  customers
WHERE
  salesrepemployeenumber IS NULL
ORDER BY
  customerName;
```

```
SELECT
  customerName,
  country,
  salesrepemployeenumber
FROM
  customers
WHERE
  salesrepemployeenumber IS NOT NULL
ORDER BY
  customerName;
```

----- Sekian Terima Kasih -----

## Referensi

- <https://www.mysql.com/>
- <https://www.mysqltutorial.org/>

