Pendahuluan MySQL

oleh Maulana Ifandika
22-2-2025

Daftar Isi:

|Pendahuluan

|Jenis MySQL
Berbagai jenis MySQL, ini diperuntukan untuk penggunaannya.
1. MySQL Standard Edition

2. MySQL Community Edition
Jenis MySQL yang gratis.

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL NDB Cluster
- MySQL Router
- MySQL Shell
- MySQL Operator
- MySQL NDB Operator
- MySQL Workbench
- MySQL Installer for Windows (umum)

3. MySQL Enterprise Edition
MySQL Enterprise Edition adalah jenis yang berbayar dan mencakup serangkaian fitur canggih, alat manajemen, dan dukungan teknis terlengkap untuk MySQL.

4. MySQL NDB Cluster CGE
MySQL NDB Cluster adalah database transaksional sumber terbuka waktu nyata yang dirancang untuk akses cepat dan selalu aktif ke data dalam kondisi throughput tinggi.

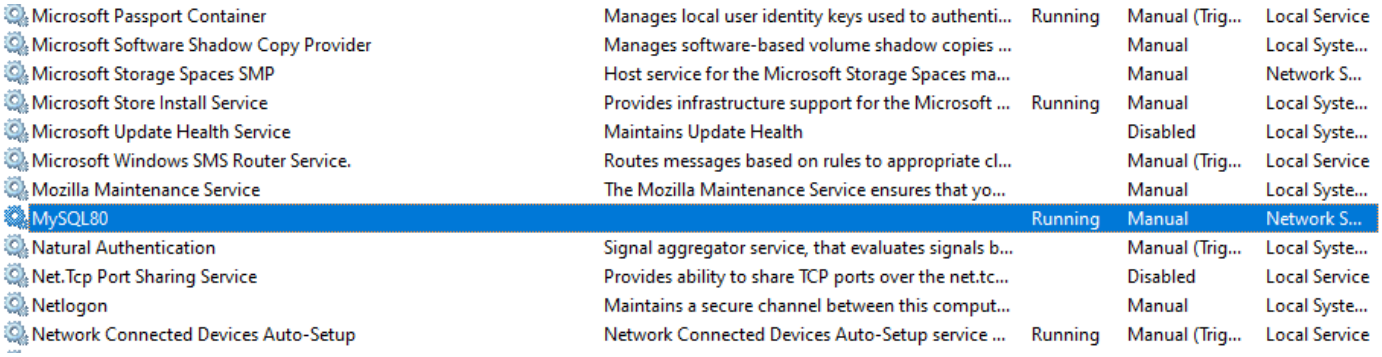
- MySQL NDB Cluster
- MySQL NDB Cluster Manager
- Ditambah lagi, semua yang ada di MySQL Enterprise Edition

5. MySQL Enterprise Edition Downloads
Akses rangkaian lengkap fitur MySQL Enterprise Edition secara gratis sambil belajar, mengembangkan, dan membuat prototipe.

- Unduhan meliputi MySQL Enterprise Server, Backup, Router, Shell, dan Konektor.

|Download, Instal, dan Konfigurasi
Konjungi website mysql dan ke bagian download untuk mengunduh MySQL. Pada contoh ini menggunakan MySQL Community Edition yang versi MySQL Installer karena hanya tinggal install (sudah mencakup banyak hal dan siap digunakan).

| MySQL Service pada Windows
Service > MySQL80



Jika ingin mematika maka set opsi ke stop, pada contoh service mysql berjalan. Jika set secara otomatis/manual/disable.

Lalu untuk mengakses MySQL lewat teriminal windows

```
$ mysql -u [username] -p
Enter Password: [password]
```

-u digunakan untuk username, dan -p untuk password, password tidak bisa ditulis langsung, jika hanya -u tanpa -p sedangkan username perlu password maka akan eror.

```
$ mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
```

Contoh disini dengan username root & password Mifandika

```
$ mysql -u root -p
Enter password: *********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.34 MySQL Community Server - GPL
```

```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

| Membuat Pengguna Baru

Bawaan dari MySQL adalah root, dimana root dapat mengakses seluruh komponen pada MySQL. Untuk membuat pengguna baru dengan akses yang terbatas dengan perintah ini. Sebelum membuat user baru, untuk menampilkan seluruh user yang ada.

Menampilkan daftar user
Untuk menapilkan daftar user.

```
mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| example          | localhost |
| ifandika         | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
| user             | localhost |
+------------------+-----------+
7 rows in set (0.00 sec)
```

Jika ingin melihat seluruh kolom pada mysql.user tinggal ganti * pada SELECT * ...;

Membuat user baru
Membuat user baru secara umum

```
$ CREATE USER '[username]'@'[host]' IDENTIFIED BY '[password]';
```

Membaut user baru dengan penambahan extra plugin

```
$ CREATE USER '[username]'@'[host]' IDENTIFIED WITH [authentication_plugin] BY '[password]';
```

Dengan ketentuan sebagai berikut.

- **username** : nama user/pengguna baru, contoh root
- **host** : adalah alamat yang akan digunakan, jika lokal menggunakan localhost
- **authentication_plugin** : semacam aturan tambahan untuk pengguna ini
- **password** : password/kata sandi yang akan digunakan untuk login pengguna ini

Untuk contoh membuat pengguna baru.

Membuat pengguna baru yang secara umum.

```
mysql> CREATE USER 'ifandika'@'localhost' IDENTIFIED BY 'ifandika';
Query OK, 0 rows affected (0.04 sec)
```

***Catatan**
Ada masalah yang diketahui pada beberapa versi PHP yang menyebabkan masalah dengan caching_sha2_password. Jika Anda berencana untuk menggunakan basis data ini dengan aplikasi PHP — phpMyAdmin, misalnya — Anda mungkin ingin membuat pengguna yang akan mengautentikasi dengan plugin mysql_native_password yang lama, meskipun masih aman:

```
mysql> CREATE USER 'example'@'localhost' IDENTIFIED WITH mysql_native_password BY 'example';
Query OK, 0 rows affected (0.01 sec)
```

Setelah membuat user, coba masuk dengan user baru.

```
PS C:\Users\Ifandika> mysql -u ifandika -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

Masuk dengan user baru berhasil.

Memberikan Izin (Granting a User Permissions)
Ini agar pengguna/user yang telah dibuat diberikan akses izin yang ditentukan.

Izin untuk spesifik

```
$ GRANT [privilege] ON [database].[table] TO '[username]'@'[host]';
```

Izin untuk banyak database/tabel

```
$ GRANT ALL PRIVILEGES ON [database/*].[tabel/*] TO '[username]'@'[host]';
```

Keterangan:

- **database** : database yang akan diakses oleh pengguna tersebut
- **table** : (Opsional) spesifik tabel yang akan diakses/digunakan
- **username** : nama user/pengguna baru
- **host** : alamat dari user/pengguna

Untuk daftar privilege:

Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.

UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
INDEX	Ability to create an index on an existing table.
CREATE	Ability to perform CREATE TABLE statements.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
DROP	Ability to perform DROP TABLE statements.
GRANT OPTION	Allows you to grant the privileges that you possess to other users.
ALL	Grants all permissions except GRANT OPTION.

Contoh user ifandika diberikan akses database hero dengan tabel fighter didalamnya.

```
mysql> GRANT PRIVILEGE ON hero.fighter TO 'ifandika'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Contoh user ifandika diberikan akses database hero dengan semua tabel didalamnya.

```
mysql> GRANT ALL PRIVILEGES ON hero.* TO 'ifandika'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

***Catatan**

Penambahan WITH GRANT OPTION dapat menyebabkan pengguna tersebut memberikan izin pengguna baru dengan semaunya.

```
$ GRANT ALL PRIVILEGES ON [db/*].[table/*] TO '[username]'@'[host]' WITH GRANT OPTION;
```

Contoh dengan user ifandika.

```
mysql> GRANT ALL PRIVILEGES ON hero.* TO 'ifandika'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'ifandika'@'localhost';
+-----------------------------------------------------------------------------+
| Grants for ifandika@localhost                                               |
+-----------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `ifandika`@`localhost`                                |
| GRANT ALL PRIVILEGES ON `hero`.* TO `ifandika`@`localhost` WITH GRANT OPTION |
+-----------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

Untuk pembuktian kita buat user baru.

```
mysql> CREATE USER 'user_baru'@'localhost' IDENTIFIED BY 'user_baru';
Query OK, 0 rows affected (0.03 sec)
```

Kemudian masuk dengan user ifandika dan coba memberikan izin ke pengguna baru.

```
PS C:\Users\Ifandika> mysql -u ifandika -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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>
```

Kemudian berikan izin ke user baru.

```
mysql> GRANT PRIVILEGE ON hero.fighter TO 'user_baru'@'localhost';
ERROR 3619 (HY000): Illegal privilege level specified for fighter

mysql> GRANT ALL PRIVILEGES ON hero.fighter TO 'user_baru'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

Hanya bisa memberikan GRANT ALL PRIVILEGES. Untuk lebih tentang REVOKE
https://www.techonthenet.com/mysql/grant_revoke.php

Terakhir yang penting adalah menyimpan perubaan yang terjadi setelah membuat pengguna baru.

```
$ FLUSH PRIVILEGES;
```

Contoh:

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

Melihat izin yang diberikan
Untuk melihat izin yang diberikan pada pengguna bisa dengan perintah.

```
$ SHOW GRANTS FOR '[username]'@'[host]';
```

Untuk contoh dengan user ifandika yang tadi telah dibuat.

```
mysql> SHOW GRANTS FOR 'ifandika'@'localhost';
+-----------------------------------------------------------+
| Grants for ifandika@localhost                             |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO `ifandika`@`localhost`              |
| GRANT ALL PRIVILEGES ON `hero`.* TO `ifandika`@`localhost` |
+-----------------------------------------------------------+
2 rows in set (0.00 sec)
```

Mengubah Izin
Jika ingin mengubah atau mengganti izin yang telah diberikan bisa dengan REVOKE

```
$ REVOKE [type_of_permission] ON [database].[table] FROM '[username]'@'[host]';
```

Keterangan:

- **type_of_permission** : jenis dari izin yang diberikan, contoh USAGE, ALL PRIVILEGES
- **database** : database yang diberikan izin
- **table** : tabel yang diberikan izin
- **username** : username tersebut
- **host** : host tersebut

Contoh menghapus izin ALL PRIVILEGES milik user ifandika.

```
mysql> SHOW GRANTS FOR 'ifandika'@'localhost';
+-----------------------------------------------------------+
| Grants for ifandika@localhost                             |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO `ifandika`@`localhost`              |
```

```
| GRANT ALL PRIVILEGES ON `hero`.* TO `ifandika`@`localhost` |
+------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> REVOKE ALL PRIVILEGES ON hero.* FROM 'ifandika'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'ifandika'@'localhost';
+--------------------------------------------+
| Grants for ifandika@localhost              |
+--------------------------------------------+
| GRANT USAGE ON *.* TO `ifandika`@`localhost` |
+--------------------------------------------+
1 row in set (0.00 sec)
```

Hapus user

Untuk menghapus pengguna/user yang telah dibuat dengan perintah drop, mirip menghapus database.

```
$ DROP USER '[username]'@'[localhost]';
```

Untuk contoh, menghapus user example

```
mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| example          | localhost |
| ifandika         | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
| user             | localhost |
+------------------+-----------+
7 rows in set (0.00 sec)

mysql> DROP USER 'example'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| ifandika         | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
| user             | localhost |
+------------------+-----------+
6 rows in set (0.00 sec)
```

Simpan setiap perubahan

Menyimpan setiap perubahan dengan perintah.

```
$ FLUSH PRIVILEGES;
```

|Perintah Dasar MySQL

Perintah dasar yang digunakan pada MySQL, setiap perintah diakhiri dengan tanda titik koma (;).

Menampilkan database

Kode:

```
$ SHOW DATABASES;
```

Contoh:

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| examples           |
| hero               |
| information_schema |
```

```
| mysql              |
| new                |
| performance_schema |
| person             |
| sys                |
+--------------------+
8 rows in set (0.01 sec)
```

Membuat database

Membuat database baru

Kode:

```
$ CREATE DATABASE <nama_database>
```

Contoh:

```
$ CREATE DATABASE animals_db;
```

Menghapus database

Kode:

```
$ DROP DATABASE [nama_database];
```

Contoh:

```
mysql> DROP DATABASE tempo;
Query OK, 0 rows affected (0.05 sec)
```

Memakai/menggunakan database

Kode:

```
$ mysql> USE [nama_database];
```

Contoh:

```
mysql> USE hero;
Database changed
```

Menampilkan tabel

Kode:

```
$ SHOW TABLES;
```

Contoh:

```
mysql> SHOW TABLES;
+----------------------+
| Tables_in_hero       |
+----------------------+
| fighter              |
+----------------------+
1 row in set (0.01 sec)
```

Membuat Tabel

Membuat tabel pada database, untuk jenis atau daftar dari ENGINE yang ada
https://dev.mysql.com/doc/refman/8.4/en/storage-engines.html
Salah satu yang umum digunakan adalah InnoDB.

Feature	MyISAM	Memory	InnoDB	Archive	NDB
B-tree indexes	Yes	Yes	Yes	No	No
Backup/ point-in-time	Yes	Yes	Yes	Yes	Yes

recovery (note 1)					
Cluster database support	No	No	No	No	Yes
Clustered indexes	No	No	Yes	No	No
Compressed data	Yes (note 2)	No	Yes	Yes	No
Data caches	No	N/A	Yes	No	Yes
Encrypted data	Yes (note 3)	Yes (note 3)	Yes (note 4)	Yes (note 3)	Yes (note 5)
Foreign key support	No	No	Yes	No	Yes
Full-text search indexes	Yes	No	Yes (note 6)	No	No
Geospatial data type support	Yes	No	Yes	Yes	Yes
Geospatial indexing support	Yes	No	Yes (note 7)	No	No
Hash indexes	No	Yes	No (note 8)	No	Yes
Index caches	Yes	N/A	Yes	No	Yes
Locking granularity	Table	Table	Row	Row	Row
MVCC	No	No	Yes	No	No
Replication support (note 1)	Yes	Limited (note 9)	Yes	Yes	Yes
Storage limits	256TB	RAM	64TB	None	384EB
T-tree indexes	No	No	No	No	Yes
Transactions	No	No	Yes	No	Yes
Update statistics for data dictionary	Yes	Yes	Yes	Yes	Yes

Kode:

```
$ CREATE TABLE [namaTabel] (
  [kategori1] [tipeData],
  [kategori1] [tipeData] [ekpresi],
  [kategori1] [tipeData] [ekpresi] ...,
  UNIQUE KEY ([data]),
  PRIMARY KEY ([data])
) ENGINE = <nama>;
```

Contoh:

```
mysql> CREATE TABLE contoh(
  id INT SIGNED NOT NULL,
  nama VARCHAR(10) NOT NULL,
  umur INT(3) NOT NULL ZEROFILL
  PRIMARY KEY (id)
) ENGINE = InnoDB;
```

Menghapus tabel

Kode:

$ DROP TABLE [nama_tabel];

Contoh:

```
mysql> DROP TABLE tempo;
Query OK, 0 rows affected (0.01 sec)
```

Menambahkan Data Ke Tabel

Ada 2 cara dalam menambahkan data, cara ke 1

Kode:

```
$ INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Contoh:

```
mysql> INSERT INTO fighter (id, name) VALUES (1, 'Argus');
Query OK, 1 row affected (0.01 sec)
```

Cara yang ke 2

Kode:

```
$ INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

Contoh:

```
mysql> INSERT INTO fighter VALUES (1, 'Argus');
Query OK, 1 row affected (0.01 sec)
```

Memperbarui Data Pada Tabel

Kode:

```
$ UPDATE [table_name]
SET [column1] = [value1], [column2] = [value2], ...
WHERE [condition];
```

Contoh:

```
mysql> UPDATE fighter SET name = 'Hayabusa' WHERE fighter.id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Menghapus Data Pada Tabel

Kode:

```
$ DELETE FROM [nama_tabel] WHERE [kondisi];
```

Contoh:

```
mysql> DELETE FROM fighter WHERE fighter.id = 2;
Query OK, 1 row affected (0.01 sec)
```

Mengambil informasi dari database dan tabel

Mengambil informasi seputar database yang digunakan, struktur dari tabel.
Mengambil informasi nama database yang digunakan.

Kode:

```
$ SELECT DATABASE();
```

Contoh:

```
mysql> SELECT DATABASE();
+------------+
| DATABASE() |
+------------+
| hero       |
+------------+
1 row in set (0.00 sec)
```

Mengambil informasi struktur dari tabel;

Kode:

```
$ DESCRIBE [nama_tabel];
```

Contoh:

```
mysql> DESCRIBE fighter;
+-------+-------------+------+-----+---------+-------+
```

```
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| id    | int         | NO   | PRI | NULL    |       |
| name  | varchar(20) | NO   |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

Mengambil Data Dari Tabel

Mengambil data dari dalam tabel secara spesifik atau keseluruhan.
Untuk mengambil data secara keseluruhan.

Kode:
```
$ SELECT * FROM [nama_tabel];
```

Contoh:
```
mysql> SELECT * FROM fighter;
+----+----------+
| id | name     |
+----+----------+
|  1 | Hayabusa |
|  2 | Saber    |
+----+----------+
2 rows in set (0.00 sec)
```

Untuk mengambil data secara spesifik per by kolom.

Kode:
```
$ mysql> SELECT [tabel].[kolom], [kolom], ... FROM [nama_tabel];
```

Contoh:
```
mysql> SELECT fighter.id, fighter.name FROM fighter;
+----+----------+
| id | name     |
+----+----------+
|  1 | Hayabusa |
|  2 | Saber    |
+----+----------+
2 rows in set (0.00 sec)
```

Fungsi MIN dan MAX

Fungsi MIN digunakan untuk mengambil nilai terkecil pada kolom tersebut.
Fungsi MAX digunakan untuk mengambil nilai terbesar pada kolom tersebut.

Menggunakan AS sebagai nama kolom untuk hasil yang ditampilkan.

Untuk kode MIN.
Kode:
```
$ SELECT MIN([nama_kolom])
FROM [nama_tabel]
WHERE [kondisi];
```

Contoh:
```
mysql> SELECT MIN(Price) AS SmallestPrice FROM Products;
+------------------+
| SmallestPrice    |
+------------------+
| 2.50             |
+------------------+
```

Untuk kode MAX.

Kode:

```
$ SELECT MAX([nama_kolom])
FROM [nama_tabel]
WHERE [kondisi];
```

Contoh:

```
mysql> SELECT MAX(Price) AS LargestPrice FROM Products;
+----------------+
| LargestPrice   |
+----------------+
| 263.50         |
+----------------+
```

Operator Pada MySQL

Operator Aritmatika

Operator	Nama	Contoh	Hasil
+	Penjumlahan	SELECT 30 + 20;	50
-	Pengurangan	SELECT 30 - 20;	10
*	Perkalian	SELECT 10 * 2;	20
/	Pembagian	SELECT 30 / 10;	3
%	Modulo	SELECT 17 % 5;	2

Operator BitWise

Operator	Nama
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR (exclusive or)

Operator Perbandingan

Operator	Nama	Contoh
<	Kurang dari	SELECT * FROM Products WHERE Price < 30;
>	Lebih dari	SELECT * FROM Products WHERE Price > 30;
<=	Kurang dari sama dengan	SELECT * FROM Products WHERE Price <= 30;
>=	Lebih dari sama dengan	SELECT * FROM Products WHERE Price >= 30;
=	Sama dengan	SELECT * FROM Products WHERE Price = 30;
<>	Tidak sama	SELECT * FROM Products WHERE Price <> 30;

Komentar

Menambahkan komentar sebagai informasi tambahan. Komentar ada 2 jenis Single-Line (Tunggal) & Multi-Line (Ganda).

Untuk Single-line

Kode:

```
-- [komentar]
```

Contoh:

```
-- Select all:
SELECT * FROM Customers;
...
SELECT * FROM Customers -- WHERE City='Berlin';
```

Untuk Multi-line

Kode:

```
/*
[komentar]
[komentar]
*/
```

Contoh:
```
/*Select all the columns
of all the records
in the Customers table:*/
SELECT * FROM Customers;
...
SELECT CustomerName, /*City,*/ Country FROM Customers;
...
SELECT * FROM Customers WHERE (CustomerName LIKE 'L%'
OR CustomerName LIKE 'R%' /*OR CustomerName LIKE 'S%'
OR CustomerName LIKE 'T%'*/ OR CustomerName LIKE 'W%')
AND Country='USA'
ORDER BY CustomerName;
```

Kondisi

Kondisi digunakan untuk operasi degan data, semisal ingin mengambil data tertentu, dsb. Menggunakan operator perbandingan dan beberapa penambahan operator.

Operator	Nama
=	Sama
>	Lebih besar
<	Lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
<>	Tidak sama, mungkin yang lain seperti ini !=
BETWEEN	Rentan nilai antara/range nilai
LIKE	Untuk pencarian
IN	Untuk multi nilai & spesifik

Kode:
```
$ WHERE [kondisi];
```

Contoh:
```
SELECT Name FROM Customers WHERE Country = 'Mexico';
```

Contoh untuk BETWEEN
Kode:
```
$ SELECT Name FROM Products WHERE Price BETWEEN 50 AND 60;
```

Contoh untuk LIKE, mengambil data berdasarkan City degan awalan ‘a’
Kode:
```
$ SELECT * FROM Customers WHERE City LIKE 'a%';
```

Contoh untuk IN
Kode:
```
$ SELECT * FROM Customers WHERE City IN ('Paris','London');
```

Sortir Dengan ORDER BY

Kata kunci ORDER BY digunakan untuk mengurutkan kumpulan hasil dalam urutan menaik atau menurun. Kata kunci ORDER BY mengurutkan catatan dalam urutan menaik secara default. Untuk mengurutkan catatan dalam urutan menurun, gunakan kata kunci DESC.

Kode:
```
SELECT [kolom1], [kolom2], ...
FROM [nama_tabel]
ORDER BY [kolom1], [kolom2], ... ASC|DESC;
```

Contoh:
```
SELECT * FROM Customers
ORDER BY Country;
...
SELECT * FROM Customers
ORDER BY Country DESC;
...
SELECT * FROM Customers
ORDER BY Country, CustomerName;
...
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```

Operator AND OR NOT

Klausa WHERE dapat dikombinasikan dengan operator AND, OR, dan NOT.
Operator AND dan OR digunakan untuk memfilter catatan berdasarkan lebih dari satu kondisi:

- Operator AND menampilkan catatan jika semua kondisi yang dipisahkan oleh AND adalah TRUE.
- Operator OR menampilkan catatan jika salah satu kondisi yang dipisahkan oleh OR adalah TRUE.

Operator NOT menampilkan catatan jika kondisinya NOT TRUE.

Untuk operator AND
Kode:
```
SELECT [kolom1], [kolom2], ...
FROM [nama_tabel]
WHERE [kondisi1] AND [kondisi2] AND [kondisi3] ...;
```

Contoh:
```
SELECT * FROM Customers WHERE Country = 'Germany' AND City = 'Berlin';
```

Untuk operator OR
Kode:
```
SELECT [kolom1], [kolom2], ...
FROM [nama_tabel]
WHERE [kondisi1] OR [kondisi2] OR [kondisi3] ...;
```

Contoh:
```
SELECT * FROM Customers WHERE City = 'Berlin' OR City = 'Stuttgart';
```

Untuk operator NOT
Kode:
```
SELECT [kolom1], [kolom2], ...
FROM [nama_tabel]
WHERE NOT [kondisi];
```

Contoh:
```
$ SELECT * FROM Customers WHERE NOT Country = 'Germany';
```

Mengkombinasikan ke-3 operator tersebut.

Kode:
```
$ SELECT * FROM Customers WHERE Country = 'Germany' AND (City = 'Berlin' OR City = 'Stuttgart');
```

Membersihkan layar atau screen

Perintah untuk membersihkan layar/screen pada MySQL

Kode:
$ \! cls

Join

Klausa JOIN digunakan untuk menggabungkan baris dari dua atau lebih tabel, berdasarkan kolom terkait di antara keduanya.

Jenis-jenis dari join

INNER JOIN : adalah menggabungkan data yang sama
LEFT JOIN : semua data tabel kiri
RIGHT JOIN : mengambil semua data tabel kanan dan yang sama di tabel kiri
CROSS JOIN : mengambil keseluruhan dari kedua tabel

INNER JOIN
employees:

id	name	department_id
1	John	1
2	Alice	2
3	Bob	1

departments:

id	department_name
1	HR
2	IT

Untuk SQL Querynya

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments
ON employees.departments_id = departments.id;
```

Untuk hasilnya

name	department_name
John	HR
Alice	IT

Bob HR

LEFT JOIN
employees:

id	name	department_id
1	John	1
2	Alice	2
3	Bob	NULL

departments:

id	department_name
1	HR
2	IT

Untuk SQL Querynya

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments
ON employees.departments_id = departments.id;
```

Untuk hasilnya

name	department_name
John	HR
Alice	IT
Bob	NULL

Terlihat semua data pada tabel kiri (employees) akan diambil tanpa terkecuali tidak sama dengan kondisi data kanan (departments)

RIGHT JOIN
employees:

id	name	department_id
1	John	1
2	Alice	2
3	Bob	NULL

departments:

id	department_name
1	HR
2	IT
3	Marketing

Untuk SQL Querynya

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments
ON employees.departments_id = departments.id;
```

Untuk hasilnya

name	department_name
John	HR

Alice IT
NULL Marketing

Terlihat ada data null pada kolom name karena tidak ada data yang sesuai dengan tabel department, sedangkan seluruh data pada tabel departments diambil.

CROSS JOIN
employees:

id	name
1	John
2	Alice
3	Bob

departments:

id	department_name
1	HR
2	IT

Untuk SQL Querynya

```
SELECT employees.name, departments.department_name
FROM employees
INNER JOIN departments;
```

Kemudian untuk hasilnya adalah jumlah baris tabel 1 x jumlah baris tabel 2, pada contoh 3 x 2 = 6

name	department_name
John	HR
John	IT
Alice	HR
Alice	IT
Bob	HR
Bob	IT

Hasil adalah setiap 1 data pada tabel kiri (employees) akan dipasangkan dengan kemungkinan seluruh data yang ada pada tabel kanan (departments)

COUNT, AVG, SUM

Beberapa fungsi yang tersedia untuk bisa digunakan,

COUNT()
Untuk mencari total baris pada tabel

```
sales:
id      product_name   amount
1       Widget         10
2       Gadget         15
3       Widget         12
4       Widget         20
```

Untuk SQL Querynya

```
SELECT COUNT(*) AS total_sales
FROM sales;
```

Untuk hasilnya

```
total_sales
4
```

AVG()
Untuk mencari nilai rata2 pada suatu kolom

```
sales:
id      product_name   amount
1       Widget         10
2       Gadget         15
3       Widget         12
4       Widget         20
```

Untuk SQL Querynya

```
SELECT AVG(amount) AS average_amount
FROM sales;
```

Untuk hasil

```
average_amount
14.25
```

Untuk sebagai pembuktian

```
PS C:\Users\Ifandika> node
Welcome to Node.js v20.14.0.
Type ".help" for more information.
> console.log( (10+15+12+20) / 4 );
14.25
undefined
```

SUM()
Untuk mencari nilai total dari suatu kolom

```
sales:
id      product_name   amount
1       Widget         10
2       Gadget         15
3       Widget         12
4       Widget         20
```

Untuk SQL Querynya

```
SELECT SUM(amount) AS total_amount
FROM sales;
```

Untuk hasilnya

```
total_amount
57
```

LIMIT

Membatasi jumlah data record yang akan diambil

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

Untuk contoh

```
sales:
id      product_name   amount
1       Widget         10
2       Gadget         15
3       Widget         12
4       Widget         20
5       Gadget         30
6       Widget         25
7       Gadget         18
```

Kemudian untuk SQL Querynya

```
SELECT * FROM sales
LIMIT 3;
```

Untuk hasilnya

```
id      product_name amount
1       Widget       10
2       Gadget       15
3       Widget       12
```

Jika ingin mengambil dengan posisi start tertentu, atau mulai dari data ke berapa

```
SELECT * FROM sales
LIMIT 3 OFFSET 3;
```

Untuk hasilnya

```
id      product_name   amount
4       Widget         20
5       Gadget         30
6       Widget         25
```

Operator UNION

UNION digunakan untuk mengkombinasikan hasil himpunan 2 atau lebih dari pernyataan SELECT.

- Dengan syarat jumlah kolom sama

Untuk sintaks UNION

```
SELECT column_name(s) FROM table1
UNION
SELECT column_name(s) FROM table2;
```

Untuk sintaks UNION ALL

```
SELECT column_name(s) FROM table1
UNION ALL
SELECT column_name(s) FROM table2;
```

Untuk contoh

```
sales table:
id     product_name   amount
1      Widget         10
2      Gadget         15
3      Widget         12

returns table:
id     product_name   amount
1      Widget         5
2      Gadget         7
3      Gadget         3
```

Untuk SQL Querynya

```
SELECT product_name
FROM sales
UNION
SELECT product_name
FROM returns;
```

Untuk hasilnya

```
product_name
Widget
Gadget
```

Jika ingin mengambil seluruh data (data duplikat), maka gunakan UNION ALL

```
SELECT product_name
FROM sales
UNION ALL
SELECT product_name
FROM returns;
```

Maka untuk hasilnya

```
product_name
Widget
Gadget
Widget
Gadget
Gadget
```

Operator GROUP BY

Operator/pernyataan untuk mengelompokkan sebuah data, GROUP BY biasanya berbarengan dengan fungsi lain, semisal COUNT, AVG, SUM

Sintaks:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Contoh dengan sederhana
table: sales

```
+----+--------------+-------+------------+
| id | product_name | amount| sales_date |
+----+--------------+-------+------------+
|  1 | Laptop       |     5 | 2025-03-01 |
|  2 | Laptop       |     3 | 2025-03-02 |
|  3 | Smartphone   |     7 | 2025-03-01 |
|  4 | Smartphone   |     2 | 2025-03-03 |
|  5 | Tablet       |     4 | 2025-03-02 |
+----+--------------+-------+------------+
```

Untuk SQL Querynya

```
SELECT product_name, SUM(amount) AS total_sales
FROM Sales
GROUP BY product_name;
```

Untuk hasilnya

```
+--------------+-------------+
| product_name | total_sales |
+--------------+-------------+
| Laptop       | 8           |
| Smartphone   | 9           |
| Tablet       | 4           |
+--------------+-------------+
```

Terlihat mendapat total hasil dari setiap prduk dimana dari fungsi SUM kemudian kita kelompokkan dengan GROUP BY name.

HAVING Clause

HAVING Caluse ditambahkan ke MySQL adalah sebagai pengganti untuk WHERE karena tidak bisa untuk agregate function, atau fungsi yang berlapis.

Sintaks:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Untuk contoh sederhana.
table: sales

```
+----+--------------+-------+------------+
| id | product_name | amount| sales_date |
+----+--------------+-------+------------+
|  1 | Laptop       |     5 | 2025-03-01 |
|  2 | Laptop       |     3 | 2025-03-02 |
|  3 | Smartphone   |     7 | 2025-03-01 |
|  4 | Smartphone   |     2 | 2025-03-03 |
|  5 | Tablet       |     4 | 2025-03-02 |
+----+--------------+-------+------------+
```

Untuk SQL Querynya

```
SELECT product_name, SUM(amount) AS total_amount
FROM Sales
GROUP BY product_name
HAVING SUM(amount) >= 6;
```

Untuk hasilnya

```
+--------------+-------------+
| product_name | total_amount|
+--------------+-------------+
| Laptop       | 8           |
| Smartphone   | 9           |
+--------------+-------------+
```

Aliases

Alias digunakan sebagai nama lain dari tabel, kolom yang digunakan, juga bisa utuk nama lain dari tabel/kolom untuk output yang akan ditampilkan.

Alias Column Syntax

```
SELECT column_name AS alias_name
FROM table_name;
```

Alias Table Syntax

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

Untuk contoh sederhananya
tabel: Sales

```
+----+--------------+-------+------------+
| id | product_name | amount| sales_date |
+----+--------------+-------+------------+
|  1 | Laptop       |     5 | 2025-03-01 |
|  2 | Smartphone   |     7 | 2025-03-01 |
|  3 | Tablet       |     4 | 2025-03-02 |
|  4 | Laptop       |     3 | 2025-03-02 |
|  5 | Smartphone   |     2 | 2025-03-03 |
+----+--------------+-------+------------+
```

Untuk SQL Quernya

```
SELECT s.product_name AS product, SUM(s.amount) AS total_sales
FROM Sales AS s
GROUP BY s.product_name;
```

Untuk hasilnya

```
+-----------+-------------+
| product   | total_sales |
+-----------+-------------+
| Laptop    | 8           |
| Smartphone| 9           |
| Tablet    | 4           |
+-----------+-------------+
```

Terlihat bahwa nama kolom s.product_name kita ganti menjadi product, kemudian s.amount menjadi total_sales, lalu untuk tabel dari Sales menjadi s

Operator EXISTS

Ini digunakan untuk kondisi mengecek data pada tabel apakah ada atau kosong

Sintaks:

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS (SELECT column_name FROM table_name WHERE condition);
```

Contoh dengan sederhana

table: sales

```
+----+--------------+-------+------------+
| id | product_name | amount| sales_date |
+----+--------------+-------+------------+
|  1 | Laptop       |     5 | 2025-03-01 |
|  2 | Laptop       |     3 | 2025-03-02 |
|  3 | Smartphone   |     7 | 2025-03-01 |
|  4 | Smartphone   |     2 | 2025-03-03 |
|  5 | Tablet       |     4 | 2025-03-02 |
+----+--------------+-------+------------+
```

Untuk SQL Querynya

```
SELECT DISTINCT product_name
FROM Sales S
WHERE EXISTS (
    SELECT 1 -- Sama seperti SELECT *
    FROM Sales
    WHERE product_name = S.product_name
    GROUP BY product_name
    HAVING SUM(amount) > 5
);
```

Untuk hasilnya

```
+--------------+
| product_name |
+--------------+
| Laptop       |
| Smartphone   |
+--------------+
```

Disini mengambil data yang jumlah (amount) > 5, maka Tablet tidak ikut

LENGTH

Mengambil panjang data sebuah string/var char

Sintaks:
LENGTH(x)

Untuk contoh sederhana

table: Users

```
+----+---------+
| id | name    |
+----+---------+
|  1 | Alice   |
|  2 | Bob     |
|  3 | Charlie |
|  4 | David   |
|  5 | Eve     |
+----+---------+
```

Untuk SQL Querynya

```
# MySQL
SELECT id, name, LENGTH(name) AS name_length
FROM users;
```

Untuk hasilnya

```
+----+---------+-------------+
| id | name    | name_length |
+----+---------+-------------+
|  1 | Alice   |           5 |
|  2 | Bob     |           3 |
|  3 | Charlie |           7 |
|  4 | David   |           5 |
|  5 | Eve     |           3 |
+----+---------+-------------+
```