



# Data Manipulation Language



DML (Data Manipulation Language) adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data, misalnya untuk pengambilan, penyisipan, pengubahan dan penghapusan data.

# DML Syntax

Insert

- menambahkan baris pada suatu tabel

Update

- mengubah isi data pada satu atau beberapa kolom pada suatu tabel

Delete

- menghapus satu baris, baris dengan kondisi tertentu atau seluruh baris

Select

- menampilkan isi dari suatu tabel yang dapat dihubungkan dengan tabel yang lainnya

# Insert

Menambah baris dengan mengisi data pada setiap kolom

```
INSERT INTO namabel VALUES (nilai1,nilai2,nilai-n);
```

The screenshot shows the MySQL Workbench interface. On the left, a tree view displays the database structure. Under 'Tables', the 'jenisfilm' table is selected. Its 'Columns' section lists 'kode' (char(5)), 'jenis' (char(6), Nullable), and 'harga' (int(11), Nullable). The 'Indexes' section shows a primary key named 'PRIMARY' on the 'kode' column. On the right, the 'Table Data' tab is active, showing the current data in the 'jenisfilm' table:

	kode	jenis	harga
*	E001	Action	90000
*	(NULL)	komedi	0

The screenshot shows the MySQL Workbench interface. On the left, a tree view displays the database structure. Under 'Tables', the 'jenisfilm' table is selected. Its 'Columns' section lists 'kode' (char(5)), 'jenis' (char(6), Nullable), and 'harga' (int(11), Nullable). The 'Indexes' section shows a primary key named 'PRIMARY' on the 'kode' column. On the right, the 'Table Data' tab is active, showing the current data in the 'jenisfilm' table:

	kode	jenis	harga
*	E001	Action	90000
*	(NULL)	komedi	0

- Jika data bertipe string, date atau time (contoh : action, horor, 2007-11-10) maka pemberian nilainya diapit dengan tanda petik tunggal ('horor') atau petik ganda ("horor").

# Insert

Menambah baris dengan hanya mengisi data pada kolom tertentu

```
INSERT INTO namabel (kolom1,kolom2,kolom-n) VALUES  
(nilai1,nilai2,nilai-n);
```

The screenshot shows a database schema browser on the left. It lists several databases: 'aku' and 'coba'. Under 'Tables', there is a table named 'jenisfilm'. The table has three columns: 'kode' (char(5)), 'jenis' (char(6), Nullable), and 'harga' (int(11), Nullable). A primary key constraint 'PRIMARY (kode)' is defined on the 'kode' column.

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a code editor with the following SQL query:

```
1 INSERT INTO jenisfilm (kode,harga) VALUES ('F002',50000);
```

In the bottom-right pane, there is a table viewer titled '2 Table Data'. The table has three columns: 'kode', 'jenis', and 'harga'. It contains three rows:

	kode	jenis	harga
	F001	Action	90000
	F002	komedi	50000
*	(NULL)	komedi	0

- Jika data bertipe numerik (2500, 400) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda

# Delete

**DELETE FROM namabel [WHERE kondisi];**

Autocomplete: [Tab]->Next Tag. [Ctrl+S]

```
1  DELETE FROM jenisfilm;
```

1 Messages 2 Table Data 3 Info

All Rows Rows

	kode	jenis	harga
*	(NULL)	komedi	0

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Ma

```
1  DELETE FROM jenisfilm WHERE kode='F001';
```

1 Messages 2 Table Data 3 Info

All Rows Rows in a Range First

	kode	jenis	harga
	F002	Drama	80000
*	(NULL)	komedi	0

Perintah dalam tanda [] bersifat opsional untuk menghapus suatu baris dengan suatu kondisi tertentu.

# Update

```
UPDATE namabel SET kolom1 = nilai1, kolom2 = nilai2  
[WHERE kondisi];
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->L

```
1 UPDATE jenisfilm  
2 SET jenis = 'Action', harga = 800000  
3
```

1 Messages 2 Table Data 3 Info

All Rows Rows in a Range

	kode	jenis	harga
	F001	Action	800000
	F002	Action	800000
	F003	Action	800000
	F004	Action	800000
*	(NULL)	komedi	0

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->L

```
1 UPDATE jenisfilm  
2 SET jenis = 'Drama', harga = 60000  
3 WHERE kode='F003'  
4
```

1 Messages 2 Table Data 3 Info

All Rows Rows in a Range

	kode	jenis	harga
	F001	Action	800000
	F002	Action	800000
	F003	Drama	60000
	F004	Action	800000
*	(NULL)	komedi	0

Perintah dalam tanda [] bersifat opsional untuk menghapus suatu baris dengan suatu kondisi tertentu.

# Select

Menampilkan data untuk semua kolom menggunakan asterisk (\*)

```
SELECT * FROM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Shift+Space]

```
1 SELECT * FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with icons for creating new connections, opening existing ones, and other database management functions. Below the toolbar, a tab bar indicates the current tab is '1 Result'. Underneath the tab bar, there are several icons: a plus sign for creating new tables, a magnifying glass for search, a refresh symbol, and a table icon. To the right of these icons, the text '(Read Only)' is displayed. The main area of the window is a data grid showing the results of the query. The table has four columns: 'kode', 'jenis', 'harga', and an unnamed column represented by a small square icon. There are five rows of data:

	kode	jenis	harga
	F001	Action	800000
	F002	Action	800000
	F003	Drama	60000
	F004	Action	800000

# Select

Menampilkan data untuk kolom tertentu

```
SELECT kolom1,kolom2,kolom-n FROM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]:

```
1  SELECT kode,jenis FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a code editor window with the following content:

```
1  SELECT kode,jenis FROM jenisfilm;
```

Below the code editor is a toolbar with several icons. The first icon, which looks like a grid with a plus sign, is highlighted. To its right are icons for a table, a search bar, and other database-related functions. A dropdown menu next to the icons shows the text "(Read Only)".

At the bottom of the interface is a results grid. The columns are labeled "kode" and "jenis". The data is as follows:

	kode	jenis
	F001	Action
	F002	Action
	F003	Drama
	F004	Action

# Select

Menampilkan data dengan kondisi data tertentu dengan klausa WHERE

```
SELECT * FROM namabel WHERE kondisi;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Matching

```
1 SELECT * FROM jenisfilm WHERE jenis='Action';
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a tab bar with four tabs: '1 Result' (selected), '2 Profiler', '3 Messages', and '4 Tasks'. Underneath the tabs, there are more icons for navigating between queries and managing sessions. The main area contains a SQL query editor with the code 'SELECT \* FROM jenisfilm WHERE jenis='Action';'. Below the editor is a results grid. The grid has three columns: 'kode', 'jenis', and 'harga'. It displays three rows of data: F001 (Action, 90000), F002 (Action, 70000), and F004 (Action, 800000). The first row (F001) is highlighted with a yellow background.

	kode	jenis	harga
	F001	Action	90000
	F002	Action	70000
	F004	Action	800000

Beberapa operator perbandingan yang dapat digunakan pada klausa WHERE

- = (sama dengan)
- > (lebih dari)
- < (kurang dari)
- <> (tidak sama dengan)
- $\geq$  (lebih dari atau sama dengan)
- $\leq$  (kurang dari atau sama dengan).

Adapun operator lain, yaitu : AND, OR, NOT, BETWEEN-AND, IN dan LIKE

# Select

Memberikan nama lain pada kolom

```
SELECT namakolomlama AS namakolombaru FROM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Ma

```
1  SELECT jenis AS jenisfilm FROM JENISFILM;
```

III

1 Result 2 Profiler 3 Messages

(Read Only)

	jenisfilm
Action	Action
Action	Drama
Action	Action

# Select

Menggunakan alias untuk nama tabel

```
SELECT namaalias.namakolom1, namaalias.namakolom-n  
FROM namatabel namaalias;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List Mat

```
1  SELECT J.jenis, J.harga FROM jenisfilm J;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the SQL code. To the right of the editor is a results pane. The results pane has tabs for 'Result', 'Profiler', and 'Messages'. The 'Result' tab is selected and displays a table with four rows of data. The table has columns named 'jenis' and 'harga'. The data is as follows:

	jenis	harga
1	Action	90000
2	Action	70000
3	Drama	60000
4	Action	800000

# Select

Klausa **ORDER BY** digunakan untuk mengurutkan data berdasarkan kolom tertentu sesuai dengan tipe data yang dimiliki.

**SELECT \* FROM namabel ORDER BY namakolom;**

The screenshot shows the MySQL Workbench interface. At the top, there is an autocomplete tooltip: "Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List". Below it is a code editor window containing the following SQL query:

```
1  SELECT * FROM jenisfilm ORDER BY jenis;
```

Below the code editor is a toolbar with three tabs: **1 Result**, **2 Profiler**, and **3 Messages**. The **1 Result** tab is selected. A status bar below the toolbar indicates "(Read Only)". The main area displays a table with the following data:

	kode	jenis	harga
	F001	Action	90000
	F002	Action	70000
	F004	Action	800000
	F003	Drama	60000

Tambahkan ASC untuk pengurutan secara ascending (menaik) atau atau tambahkan DESC untuk pengurutan secara descending (menurun), dibelakang namakolom.

## Select

Function **COUNT** digunakan untuk menghitung jumlah baris suatu kolom pada tabel

```
SELECT COUNT (namakolom) FORM namatabel;
```

Autocomplete: [Tab]-> Next Tag. [Ctrl+Space]->

```
1 SELECT COUNT(jenis) FROM jenisfilm;
```

1 Result    2 Profiler    3 Messages

Read-only result

COUNT (jenis)
4

## Select

Function **SUM** digunakan untuk menghitung jumlah nilai suatu kolom pada tabel

```
SELECT SUM (namakolom) FORM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]

```
1 SELECT SUM(harga) FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with icons for connection, schema browser, profiler, and messages. Below the toolbar, a tab bar shows '1 Result' (selected), '2 Profiler', and '3 Messages'. Under the 'Result' tab, there's a dropdown menu set to 'Read-only result'. The main area displays a single row of results from a query:

	sum(harga)
	310000

## Select

Function **AVG** digunakan untuk menghitung rata-rata dari nilai suatu kolom pada tabel

```
SELECT AVG (namakolom) FORM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]

```
1 SELECT AVG(harga) FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar is a tab bar with three tabs: '1 Result' (selected), '2 Profiler', and '3 Messages'. Underneath the tabs, there is a toolbar with icons for creating new queries, opening existing ones, and other database operations. A dropdown menu labeled 'Read-only result' is open. The main area displays the query 'SELECT AVG(harga) FROM jenisfilm;' and its result, which is '77500.0000'. The result is shown in a table with two columns: one for the column name ('AVG(harga)') and one for the value ('77500.0000').

## Select

Function **MIN** digunakan untuk menampilkan nilai terkecil dari suatu kolom pada tabel

```
SELECT MIN (namakolom) FORM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]

```
1 SELECT MIN(harga) FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a tab bar with three tabs: '1 Result' (selected), '2 Profiler', and '3 Messages'. Underneath the tabs, there are four small icons followed by the text 'Read-only result'. The main area displays a single row of data in a table format. The first column is empty (grey square). The second column is labeled 'MIN (harga)' and contains the value '60000'.

	MIN (harga)
	60000

## Select

Function **MAX** digunakan untuk menampilkan nilai terkecil dari suatu kolom pada tabel

```
SELECT MAX (namakolom) FORM namatabel;
```

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]

```
1 SELECT MAX(harga) FROM jenisfilm;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a tab bar with three tabs: '1 Result' (selected), '2 Profiler', and '3 Messages'. Underneath the tabs is a toolbar with icons for creating new queries, opening existing ones, and other database operations. A dropdown menu labeled 'Read-only result' is open. The main area displays a single row of results from a query. The first column is a small gray square icon, and the second column contains the text 'MAX (harga)' in blue. The value '90000' is displayed in a larger black font below it.

	MAX (harga)
	90000