

# Data Manipulation Language (DML)



**Basis Data**

# DML



- ◉ Menambah data (Insert)
- ◉ Mengubah data (Update)
- ◉ Menghapus data (Delete)
- ◉ Menampilkan data (Select)

# Menambah Data (Insert)

**INSERT INTO** nama\_tabel  
**VALUES** (nilai\_1, nilai\_2, ....., nilai\_n)

Atau

**INSERT INTO** nama\_tabel (field\_1, field\_2, ..., field\_n)  
**VALUES** (nilai\_1, nilai\_2, ..., nilai\_3)

# Contoh Menambah Data (Insert)

```
1 DESC sayuran;  
2 INSERT INTO sayuran VALUES ('5', 'Seledri')  
3  
4 INSERT INTO sayuran (Kode, Nama)  
5 VALUES ('6', 'Daun Bawang');  
6  
7 INSERT INTO sayuran(Nama)  
8 VALUES('Daun Bawang');  
9  
10 INSERT INTO nama_tabel  
11 VALUES (nilai_1, nilai_2,..., nilai_n)  
12  
13 INSERT INTO table name (kolom 1, kolom 2,..., Kolom_n)  
14 VALUES (nilai_1, nilai_2,..., nilai_n);  
15
```

The diagram illustrates the components of an SQL INSERT statement using color-coded boxes and arrows:

- Red:** The VALUES clause and its arguments (e.g., '5', 'Seledri' and 'Daun Bawang').
- Blue:** The INSERT INTO keywords and the table name.
- Yellow:** The column list in parentheses (e.g., (Kode, Nama) and (kolom 1, kolom 2,..., Kolom\_n)).
- Green:** The VALUES keyword and the list of values in parentheses.

Arrows indicate the flow and relationships between these components in different SQL statements.

My Connections

All Labels

---

**CASE** Flow Control  
Create a conditional construct

---

**COMMENTS** Comment  
Create a comment

---

**IF...ELSE...** Flow Control  
Create a IF...ELSE... construct

---

**INSERT Syntax** DML  
Insert new rows into an existing table

---

**LOOP** Flow Control  
Create a simple loop construct

# Mengubah data (Update)

```
UPDATE nama_tabel  
    SET field_1 = nilai_1,  
    field_2=nilai_2,  
    ...  
    field_n = nilai_n  
WHERE kondisi
```

# Contoh Mengubah data (Update)

The screenshot displays a SQL IDE interface with a menu bar (File, Edit, Format, View, Run, Window, Help) and a toolbar (Save, Query Builder, Beautify SQL). The database connection is set to 'localhost' and the schema is 'stt\_pomosda'.

The SQL query being executed is:

```
2 VALUES (nilai_1, nilai_2, ...
3
4 UPDATE sayuran
5 SET
6     Nama = 'KOL'
7 WHERE Kode=1
8
9 SELECT * FROM sayuran;
```

The result of the query is shown in a table with columns 'Kode' and 'Nama':

Kode	Nama
1	KOL
2	Bayam

Annotations in the image include:

- A red arrow pointing from the 'Nama' column in the result table to the 'Nama' field in the SET clause of the UPDATE statement.
- A yellow arrow pointing from the 'Kode' column in the result table to the 'Kode=1' condition in the WHERE clause of the UPDATE statement.
- A green arrow pointing from the 'KOL' value in the result table to the 'KOL' value in the SET clause.
- A blue arrow pointing from the '1' value in the result table to the '1' value in the WHERE clause.
- A green box highlights the first two rows of the result table.
- A blue box highlights the 'KOL' and 'Bayam' values in the result table.

# Menghapus data (Delete)

**DELETE FROM** nama\_tabel

**WHERE** kondisi

# Contoh Menghapus data (Delete)

The screenshot shows a database query editor window titled "Untitled - Query". The menu bar includes File, Edit, Format, View, Run, Window, and Help. The toolbar contains icons for Save, Query Builder, Beautify SQL, Code Snippets, and Run. The query text is as follows:

```
1 DELETE FROM sayuran
2 WHERE Kode = '1'
```

Annotations on the query:

- A green box highlights the word `sayuran`, with a green arrow pointing to the label **nama\_tabel**.
- A red box highlights the word `Kode`, with a green arrow pointing to the text **Kondisi diisi dengan field yang menjadi PK**.
- A yellow box highlights the value `'1'`, with a yellow arrow pointing to the text **1, menunjukkan bahwa data yang akan dihapus adalah data dengan kode 1**.

Below the query editor, there are two result panels. The left panel shows the table structure for the 'sayuran' table:

Field	Type	Null	Key
Kode	int(11)	NO	PRI
Nama	varchar(255)	YES	

The right panel shows the data from the 'sayuran' table:

Kode	Nama
1	KOL
2	Bayam
3	Wortel

At the bottom of the window, the status bar shows the query `SELECT * FROM sayuran` and the elapsed time `Elapsed Time: 0.`



# Menampilkan data (Select)

```
SELECT * FROM nama_tabel
```

atau

```
SELECT field_1, field_2, ..., field_n  
FROM nama_tabel
```

# Contoh Menampilkan Data (Select)

1 `SELECT * FROM sayuran`

Message	Summary	Result 1	Profile	Hasil
Kode	Nama			
1	KOL			
2	Bayam			
3	Wortel			
4	Sawi			
5	Seledri			
6	Daun Bawang			
7	Daun Bawang			

# Contoh Menampilkan Data (Select)

```
1 SELECT Kode, Nama  
2 FROM sayuran
```

Message Summary Result 1 Profile Status

Hasil

Kode	Nama
1	KOL
2	Bayam
3	Wortel
4	Sawi
5	Seledri
6	Daun Bawang
7	Daun Bawang

```
1 SELECT Nama  
2 FROM sayuran
```

Message Summary Re

Menampilkan data hanya pada field tertentu (Mis : Nama)

Nama
KOL
Bayam
Wortel
Sawi
Seledri
Daun Bawang
Daun Bawang