



SQL

Gold - Chapter 5 - Topic 4

Selamat datang di **Chapter 5 Topik 4** online
course **Fullstack Web** dari Binar Academy!





Hello bestie!

Di topik sebelumnya kita bahas tentang database, dimana ada 2 tipe database yaitu SQL dan NoSQL. Nah ditopik ini kita akan lebih dalam lagi mengenal **database dengan bahasa SQL**.

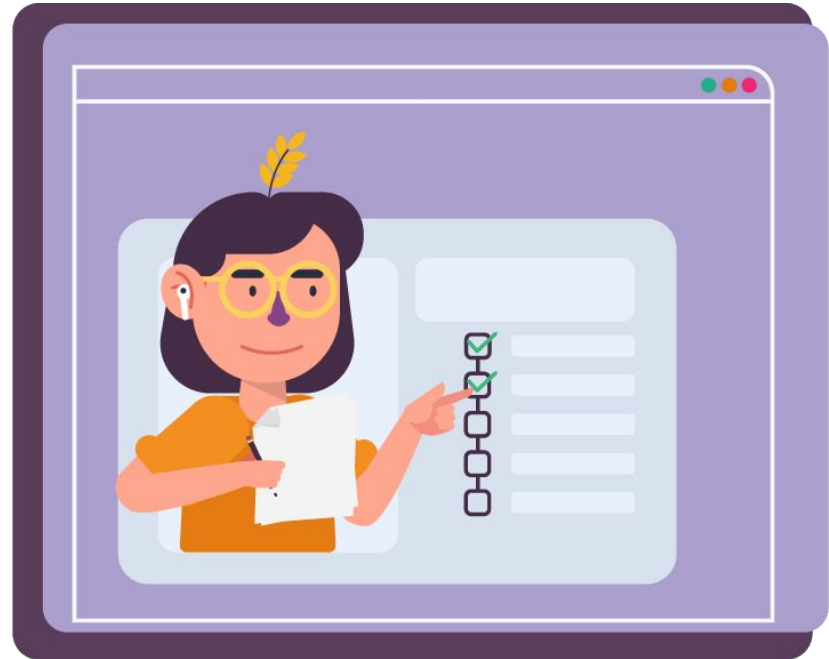
Penasaran gak? Yuk langsung capcus ~





Detailnya, kita bakal bahas hal-hal berikut:

- Memahami query dengan menggunakan SQL
- Memahami cara penggunaan DDL
- Memahami cara penggunaan DML





Di topik sebelumnya, kita mengenal apa itu query dan sekilas penggunaan query, pastinya masih ingat kan?

Nah, sekarang biar makin mantap, kita terusin belajar **query di SQL** yuk!



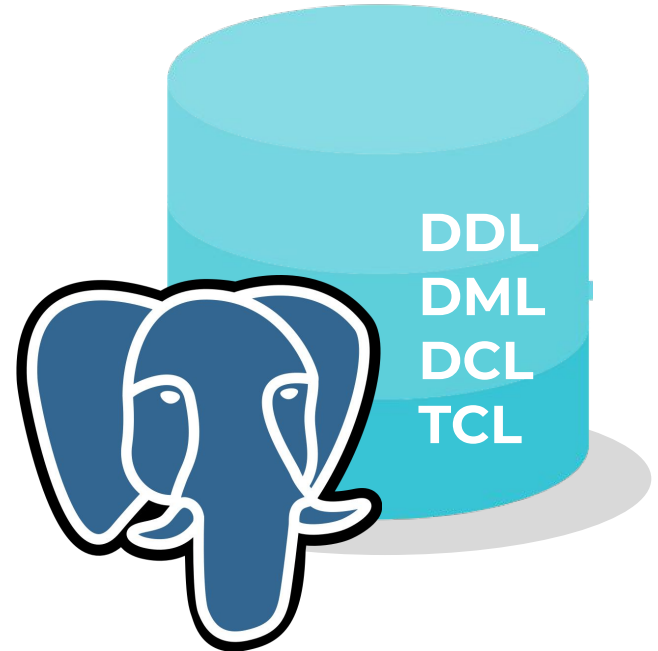
Query dalam SQL

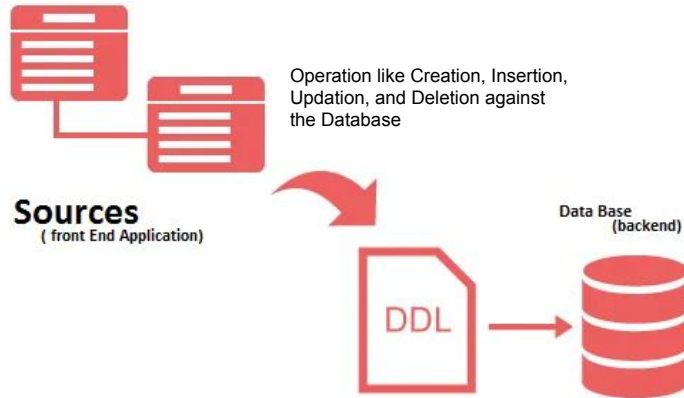
Masih ingat kan ada sebuah bahasa yang bisa dipakai untuk melakukan query database!?

Yap! Bener banget! Bahasa itu bernama **SQL (Structured Query Language)**. **Ciri khasnya tuh query-nya sangat terstruktur dan strict**. Di dalam SQL, query-query ini dapat kita golongan menjadi empat jenis, yaitu:

- **Data Definition Language (DDL)**
- **Data Manipulation Language (DML)**
- **Data Control Language (DCL)**
- **Transaction Control Language (TCL)**

Kita bahas satu per satu yuk!





Data Definition Language (DDL)

Data Definition Language adalah **perintah yang kita gunakan untuk mendefinisikan suatu struktur**, baik itu table, database, dan lain-lain.

Berikut ini contoh perintah DDL:

- ❖ CREATE
- ❖ ALTER
- ❖ DROP



Data Manipulation Language (DML)

Data Manipulation Language adalah **perintah yang kita gunakan untuk melakukan manipulasi data** di dalam suatu database. Manipulasi itu bisa berupa mencari data, membuat data, mengubah data, dan menghapus data.

Berikut ini contoh perintah DML:

- ❖ SELECT
- ❖ INSERT
- ❖ UPDATE
- ❖ DELETE

DML

Select



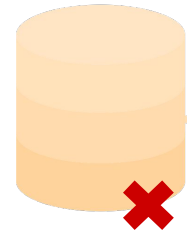
Insert

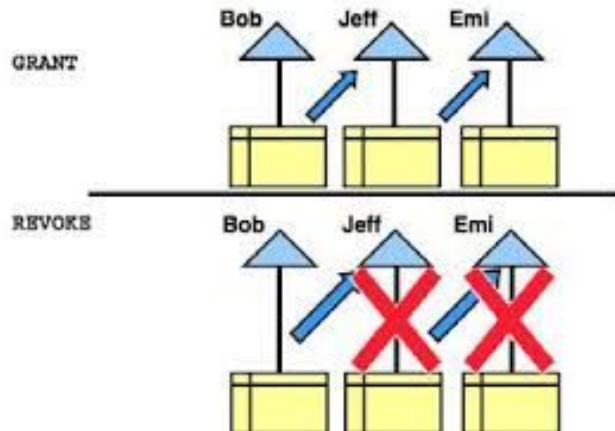


Update



Delete





Data Control Language (DCL)

Data Control Language adalah **perintah yang kita gunakan untuk mengubah hak akses, memberikan roles, dan isu lain yang berhubungan dengan keamanan** database.

Berikut ini contoh perintah DCL:

- ❖ GRANT
- ❖ REVOKE

Namun, kita akan bahas cara penggunaan untuk DDL dan DML saja

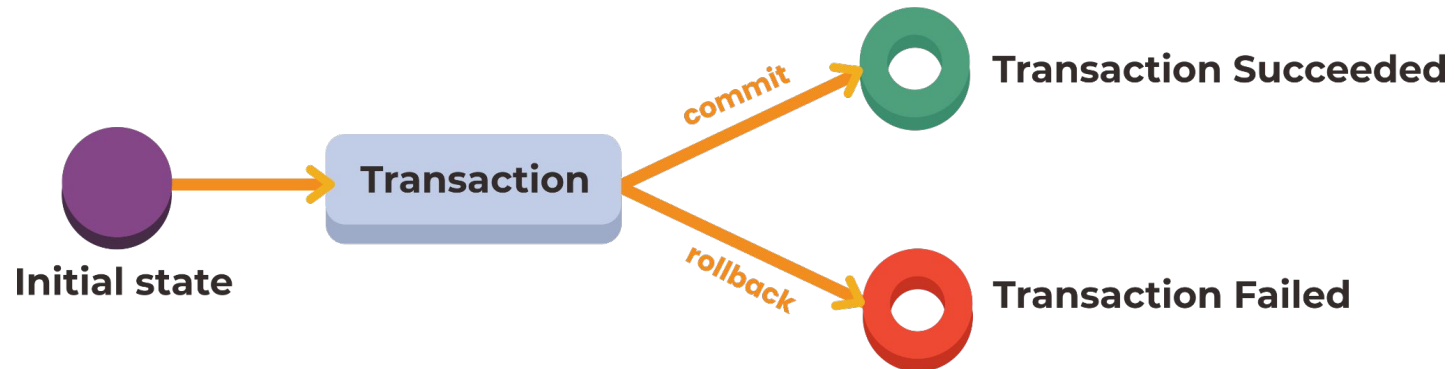


Transaction Control Language (TCL)

Transaction Control Language adalah **perintah yang berurusan dengan transaksi** database. Berikut ini contoh perintah TCL:

- ❖ BEGIN
- ❖ COMMIT
- ❖ ROLLBACK

Namun, di topik ini kita tidak membahas tentang penggunaan TCL, karena ini adalah pembahasan tingkat lanjut.





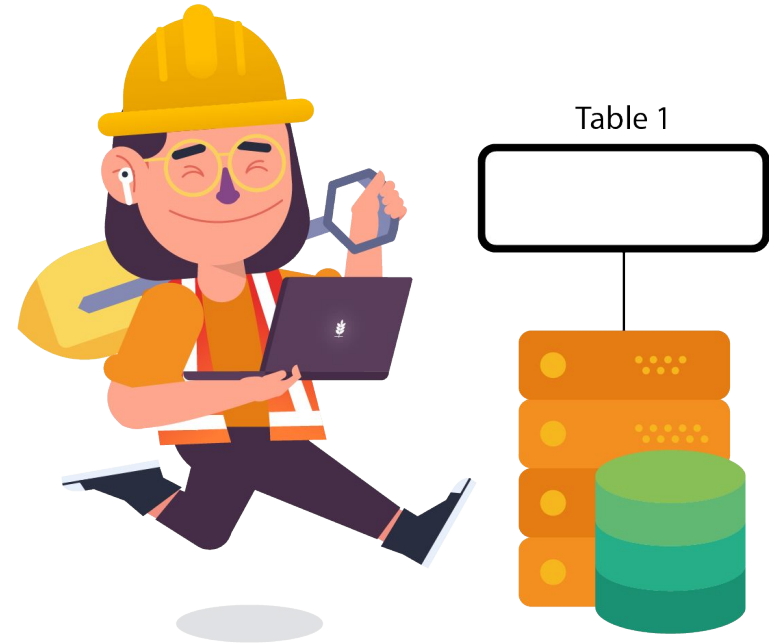
Oke, sekarang kita udah tahu secara sekilas beberapa keyword atau perintah yang bisa digunakan untuk setiap jenis query.

Nah, sekarang kita akan bahas gimana cara menggunakan perintah query untuk **Data Definition Language (DDL)**. Lanjut kuy~

Create

Perintah DDL yang pertama akan kita coba implementasi adalah create. Perintah create ini digunakan untuk **membuat sesuatu di dalam SQL DBMS, baik itu table, database, view, procedure, dan lain-lain.**

Namun, di materi ini kita akan bahas dua hal saja, yaitu **create database** dan **create table**.



1. Create Database

Perintah ini cukup simpel, kita hanya perlu masuk ke psql dengan user yang diberi wewenang untuk membuat database. Cara masuk (login) ke konsol psql dan membuat database ini sudah dijelaskan di topik sebelumnya, kalau lupa boleh cek lagi slide nya ya!

```
# psql <nama-database>
# Pastikan login di terminal sebagai user fnurhidayat
psql postgres

# Untuk penamaan database standar yang dipakai adalah menggunakan kaidah snake_case
CREATE DATABASE nama_database;
```

2. Create Table

Untuk membuat table, kita harus berada di dalam database yang sudah kita buat sebelumnya. Namun, sebelum kita membuat table, kita perlu ketahui dulu beberapa tipe data yang umum digunakan di PostgreSQL.

Tipe Data	Contoh
VARCHAR(n)	"Hello World"
TEXT	"Perintah SQL yang berhubungan dengan manipulasi atau pengolahan data dalam table"
INTEGER	10
FLOAT	3.14
BOOLEAN	true

Kalau barisan kode di bawah ini adalah contoh format default untuk membuat table:

```
CREATE TABLE nama_table (  
  nama_kolom1 TИPEDATA ATURAN,  
  nama_kolom2 TИPEDATA ATURAN,  
  nama_kolom3 TИPEDATA ATURAN  
);
```

Nah, setelah kita tahu formatnya, kita bisa coba buat table untuk artikel, yang di dalamnya memiliki kolom title, body, dan approved.

Oh iya, ada satu kolom yang wajib kita tambahkan yaitu **id**. Kolom id ini biasanya kita gunakan sebagai PRIMARY KEY, yang berarti formatnya harus unik dan bisa menjadi informasi ketika kita ingin mencari data tersebut secara spesifik. Keunikan kolom id ini bisa kita ibaratkan seperti sidik jari suatu data.

Kita coba langsung bikin query-nya aja ya!

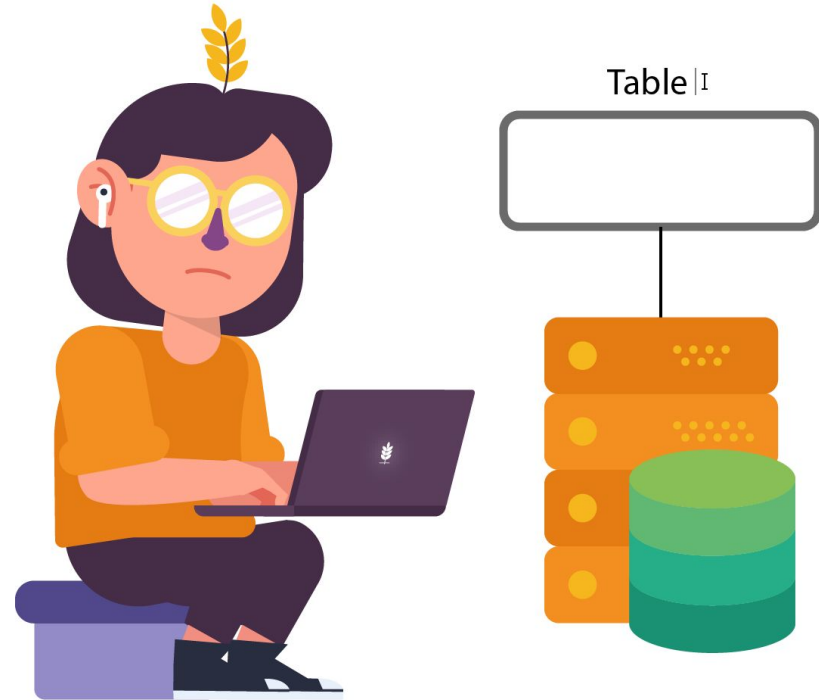
```
CREATE TABLE articles (  
  id BIGSERIAL PRIMARY KEY,  
  title VARCHAR(255) NOT NULL,  
  body TEXT NOT NULL,  
  approved BOOLEAN NOT NULL DEFAULT FALSE,  
);
```

PRIMARY KEY, NOT NULL, dan DEFAULT FALSE adalah sebuah constraint yang mengatur kaidah pembuatan data di dalam tabel tersebut. Untuk mengetahui lebih lanjut mengenai constraint bisa dibaca di link [ini](#).



Alter

Perintah DDL kedua yang akan kita coba implementa adalah alter. Perintah alter ini **digunakan untuk mengubah struktur di dalam suatu database dan table di dalam DBMS.**





1. Alter database

Perintah ini untuk memodifikasi *database*, misalnya kita dapat melakukan penamaan ulang pada suatu *database*.

```
ALTER DATABASE nama_database  
RENAME TO nama_database_baru;
```

2. Alter table

Perintah ini untuk mengganti nama *table*. *Query*-nya sangat simpel, kita hanya perlu menuliskan perintah seperti berikut ini:

```
# Perintah default ALTER  
ALTER TABLE nama_tabel_lama  
RENAME TO nama_tabel_baru;  
  
# Jika ingin menambah kolom baru pada suatu table  
ALTER TABLE articles  
ADD COLUMN created_at TIMESTAMP NOT NULL;
```



Drop

Selanjutnya, perintah DDL ketiga yang akan kita coba implementasi adalah drop. Perintah drop ini **digunakan untuk menghapus sebuah struktur, baik itu database maupun**

table.





Nah, berhubung perintah ini termasuk dalam kategori perintah yang berbahaya, karena bisa menghapus keseluruhan data, maka kita perlu ekstra hati-hati dalam menggunakannya ya! Daripada kenapa-napa kan~



```
# Perintah drop untuk menghapus database  
DROP DATABASE nama_database;  
  
# Perintah drop untuk menghapus table  
DROP TABLE nama_table;
```



Selanjutnya, kita akan bahas gimana cara menggunakan perintah query untuk **Data Manipulation Language (DML)**. Markicuss~



Select

Perintah select ini digunakan untuk **mencari data di dalam database**. Perintah select sebenarnya bisa sangat kompleks, tergantung kita ingin mencari data yang seperti apa. Sekarang kita coba bahas perintah yang simpel dulu saja ya, misalnya mencari data dari table artikel.

```
# Perintah default SELECT
SELECT * FROM nama_tabel;

# Jika table tersebut bernama articles, maka
perintahnya seperti ini:
SELECT * FROM articles;
```

Response dari perintah tersebut akan memunculkan data-data dari table articles, yang berupa informasi di setiap kolomnya. Namun, jika ternyata tidak ada data di dalam table, maka output-nya akan menampilkan table kosong saja dengan judul kolomnya.

Insert

Perintah insert ini digunakan untuk **memasukkan baris data baru di dalam sebuah table.**

```
# Perintah default INSERT
INSERT INTO (
    nama_kolom1,
    nama_kolom2
) VALUES (nilai1, nilai2);

# Perintah INSERT untuk memasukkan data baru di table articles
INSERT INTO (
    title,
    body,
    approved
) VALUES ("Hello World", "Lorem Ipsum Dolor Sit Amet", TRUE);
# output: INSERT 0 1

# Kita bisa validasi atau mengecek datanya sudah masuk dengan
perintah SELECT
SELECT * FROM articles;
```

Update

Perintah update ini digunakan untuk **memperbarui atau update suatu data** di dalam sebuah table.

```
# Perintah default UPDATE
UPDATE nama_tabel
SET
    nama_kolom1 = nilai1
    nama_kolom2 = nilai2
WHERE kondisi;

# Perintah UPDATE untuk mengubah data di table articles
UPDATE articles
SET
    approved = FALSE
WHERE articles.title = 'Hello World';
# output-nya data di table articles yang berjudul "Hello World"
yang memiliki properti approved, nilainya berubah menjadi FALSE
```


Delete

Perintah delete ini digunakan untuk **menghapus suatu data di dalam sebuah table**. Namun, kita perlu ingat untuk selalu berhati-hati ketika ingin menggunakan perintah ini karena bisa menghapus data secara permanen. Oleh karena itu, sebaiknya kita selalu gunakan operator WHERE ketika ingin menggunakan perintah ini. Jika kita lupa menggunakannya, hmm.. perintah DELETE ini akan menghapus semua data di dalam suatu table.

```
# Perintah default DELETE
DELETE FROM nama_tabel WHERE kondisi;

# Perintah DELETE untuk menghapus data di table articles yang
tidak di-approve
DELETE FROM articles WHERE articles.approved = FALSE;
```

Congraaatss!! Kamu udah berhasil menyelesaikan topik ini!

Selanjutnya kamu bisa explore lagi cara penggunaan **query di SQL**, biar makin mantul yaa~ mantap betul!

- Contoh implementasi query untuk DCL bisa lihat materi di topik sebelumnya atau baca referensi di [link ini](#).
- Jika kalian ingin explore tentang TCL bisa baca di [link ini](#).





Referensi buat kamu yang suka kepo-kepo~

- <https://www.journaldev.com/16767/sql>
- <https://www.complexsql.com/basic-sql-commands/>
- <https://www.guru99.com/sql-commands-dbms-query.html#7>
- <https://www.edureka.co/blog/postgresql-tutorial#:~:text=The%20DML%20>





Naahhh sekarang lengkap deh pengetahuan kita tentang database, khususnya SQL.

Selanjutnya kita bakal bahas konsep, fungsi dan bagaimana cara penggunaan **ORM**.



Terima Kasih!



Next Topic

loading...