

## **BAB III**

### **PERINTAH DDL (Data Definition Language)**

#### **Tujuan Dan Dasar Teori**

##### **2.1 Tujuan**

##### **2.2 Dasar Teori**

DDL adalah suatu bentuk dari SQL yang dapat dipergunakan untuk menciptakan ataupun membuat suatu database, struktur tabel, tabel, dan lainnya.

- Perintah-Perintah DDL Utama

##### **A. CREATE:**

Digunakan untuk membuat objek-objek database baru.

Contoh:

```
CREATE DATABASE mydatabase;
```

```
CREATE TABLE customers (id INT PRIMARY KEY, name VARCHAR(50), email  
VARCHAR(100));
```

```
CREATE INDEX idx_name ON customers (name);
```

##### **B. ALTER:**

Digunakan untuk mengubah struktur objek-objek database yang sudah ada.

Contoh:

```
ALTER TABLE customers ADD COLUMN phone VARCHAR(20);
```

```
ALTER TABLE customers MODIFY COLUMN name VARCHAR(100);
```

##### **C. DROP:**

Digunakan untuk menghapus objek-objek database.

Contoh:

```
DROP TABLE customers;
```

```
DROP DATABASE mydatabase;
```

##### **2.3 Praktikum**

###### **2.3.1 Praktik menggunakan aplikasi postgre**

##### **A. MODIFIKASI TABEL dan DATABASE**

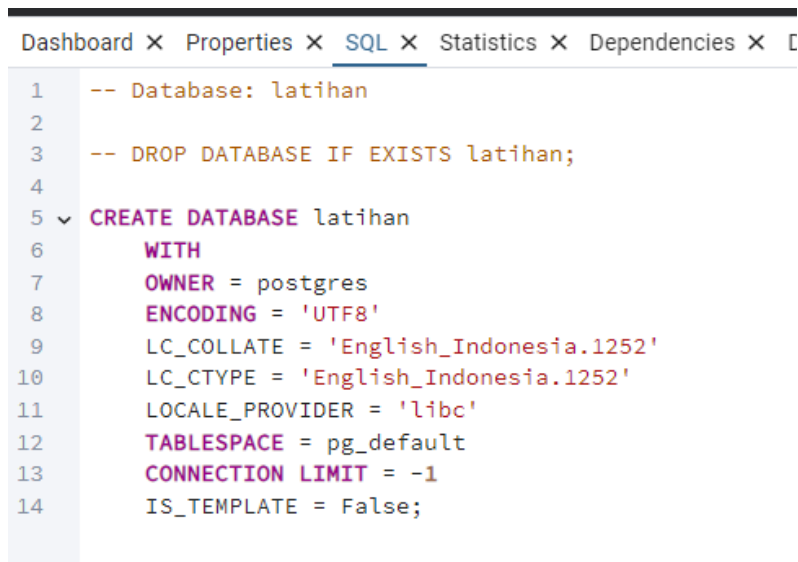
###### **• MEMBUAT DATABASE**

1. Masuk pada postgresQL. Jangan lupa memasukkan password yang sudah diatur saat penginstalan

2. Pada tampilan jendela sebelah kanan, klik server → PostgreSQL → Databases (1). Pada databases(1) klik kanan kemudian pilih create databases
3. Beri nama database sesuai dengan kebutuhan kemudian klik save. Misalkan saya membuat database dengan nama “latihan”. Sampai disini pembuatan database sudah selesai

### **TAMBAHAN:**

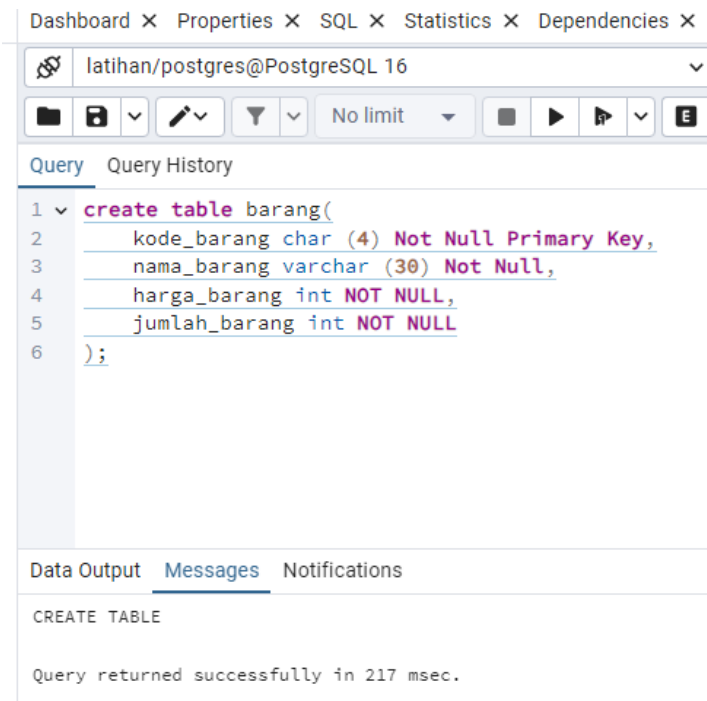
Apabila ingin melihat query SQL pembuatan database maka bisa dengan memilih pada bagian menu SQL seperti tampilan di bawah:



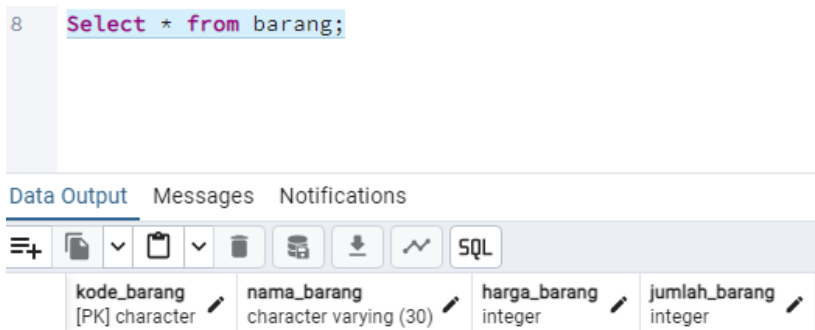
```
Dashboard × Properties × SQL × Statistics × Dependencies × [
1  -- Database: latihan
2
3  -- DROP DATABASE IF EXISTS latihan;
4
5  CREATE DATABASE latihan
6      WITH
7      OWNER = postgres
8      ENCODING = 'UTF8'
9      LC_COLLATE = 'English_Indonesia.1252'
10     LC_CTYPE = 'English_Indonesia.1252'
11     LOCALE_PROVIDER = 'libc'
12     TABLESPACE = pg_default
13     CONNECTION LIMIT = -1
14     IS_TEMPLATE = False;
```

- **CREATE TABLE**

1. Buka aplikasi postgresqlnya dan jangan lupa memilih database yang akan digunakan. Pada database yang terpilih klik kanan → Query Tool
2. Buat tabel sesuai dengan kebutuhan dengan memasukkan perintah:



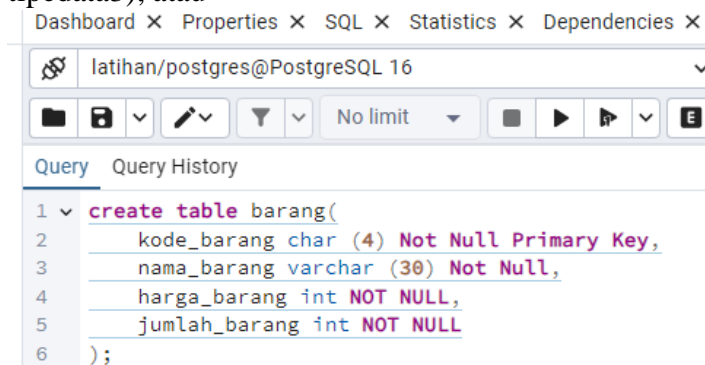
- Untuk mengecek apakah tabel sudah benar-benar berhasil terbentuk, masukkan perintah “Select \* from nama\_tabel”;



## • Menambah primary key

Saat pembuatan tabel

- Create tabel namatabel(field1 typedata1 primary key, field2 typedata 2, field 3 typedata3); atau



- Create tabel namatabel(field1 typedata1, field2 typedata2, field3 typedata3, PRIMARY KEY(field1));

Query Query History

```

1 create table jurusan (kode_jurusan int NOT NULL,
2   nama_jurusan Varchar (20) NOT NULL, jumlah_kelas int NOT NULL,
3   jumlah_siswa int NOT NULL, PRIMARY KEY (kode_jurusan));

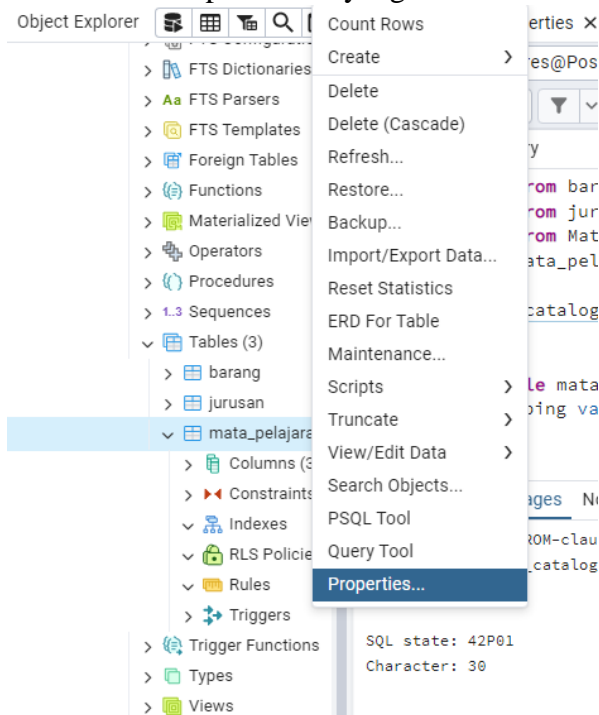
```

Data Output Messages Notifications

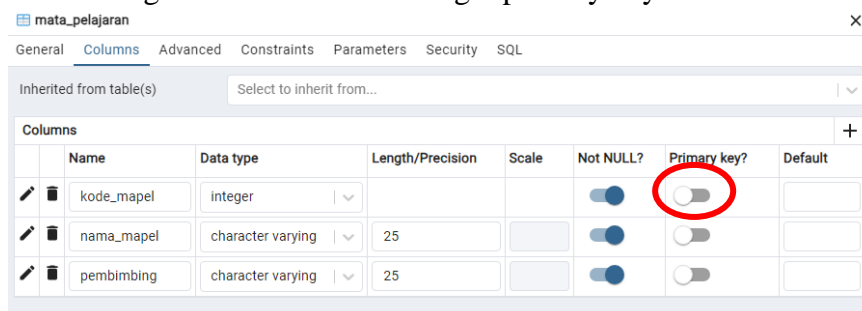
CREATE TABLE

Query returned successfully in 156 msec.

- Edit langsung menggunakan fasilitas yang ada pada postgresQL
  - a. Klik kanan pada tabel yang akan ditambahkan primary key → properties



- b. Buka bagian menu columns → pilih kolom yang akan dijadikan primary key kemudian geser tombol ke kanan agar primary key aktif → klik save



- **ALTER TABLE**

- Hapus kolom

Alter tabel nama\_tabel drop column judul-kolom-yang-akan-dihapus;

```
11 alter table mata_pelajaran drop column pembimbing;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 149 msec.

- Menambah kolom

Alter table judul\_tabel add column judul-kolom tipedata;

```
13 alter table mata_pelajaran add column pembimbing varchar (25) NOT NULL;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 199 msec.

- Mengganti judul kolom tertentu

Alter table nama-tabel rename column judul-kolom-awal to judul-kolom-baru;

```
15 alter table mata_pelajaran rename column pembimbing to pembimbing_mapel;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 150 msec.

- Mengubah default

Alter table nama tabel alter column judul-kolom set default 'set default';

```
17 alter table mata_pelajaran alter column pembimbing_mapel set default 'Wahyu';
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 156 msec.

Untuk mengecek default yang sudah diubah, klik kanan pada tabel → propertis dan buka bagian menu (lihat bagian default (tanda merah))

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
kode_mapel	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
nama_mapel	character varying	25		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
pembimbing_mapel	character varying	25		<input checked="" type="checkbox"/>	<input type="checkbox"/>	'Wahyu'

- Mengatur isi kolom tertentu

Alter table nama-tabel add check (judul-kolom in (isian));

```
19 alter table mata_pelajaran add check (pembimbing_mapel in('Nur','Ria','Vina'));
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 190 msec.

- Menambah kolom baru

Alter table nama-tabel add column judul-kolom-tambahan tipe-data;

```
22 alter table mata_pelajaran add column jadwal varchar (20);
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 187 msec.

- Mengatur kolom bersifat UNIQUE

UNIQUE constraint ini digunakan untuk memastikan isi data tidak ada yang sama atau hampir sama dengan Primary Key.

Alter table nama-tabel add constraint unique\_judul-kolom unique(judul-kolom);

```
23 alter table mata_pelajaran add constraint unique_jadwal unique (jadwal);
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 356 msec.

- Mengubah nama tabel

Alter table nama-tabel-awal rename to tabel-baru;

```
25 alter table mata_pelajaran rename to mapel;
```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 182 msec.

- **RENAME**

1. Buat database baru dengan nama mydb
2. Buat tabel baru dengan judul tabel vendor seperti di bawah

Data Output	Messages	Notifications
<div><div><div>≡+</div><div></div><div>▼</div><div></div><div>▼</div><div></div><div></div><div></div><div></div><div>SQL</div></div><div><div>id</div><div>[PK] integer</div><div></div></div><div><div>nama</div><div>character varying (50)</div><div></div></div></div>		

3. Mengganti nama tabel vendor menjadi supplier  
Alter table vendor RENAME to supplier

4

```
alter table vendor RENAME to suplier;
```

## B. MENGHAPUS DATABASE dan MENGHAPUS TABEL

- Menghapus database  
Drop database namadatabase;
- Menghapus tabel  
Drop table namatabel;

## C. MEMBUAT DAN MENAMBAH FOREIGN KEY

Foreign key adalah sebuah kolom atau sekumpulan kolom dalam satu tabel yang merujuk ke primary key pada tabel lain. Tujuannya untuk menjaga integritas referensial data, artinya data yang ada di tabel yang memiliki foreign key harus sesuai dengan data yang ada di tabel yang menjadi referensi.

- **Perintah SQL untuk Menambahkan Foreign Key**

Ada dua cara utama untuk menambahkan foreign key pada PostgreSQL:

- a. Saat Membuat Tabel

Saat membuat sebuah tabel baru, bisa dengan langsung menambahkan constraint foreign key dalam perintah CREATE TABLE.

```
CREATE TABLE table_name (  
    column1 data_type,  
    column2 data_type,  
    column3 data_type,  
    FOREIGN KEY (column3) REFERENCES other_table (primary_key_column)  
);
```

**Keterangan:**

- table\_name: Nama tabel yang akan dibuat.
- column1, column2, column3: Kolom-kolom dalam tabel.
- FOREIGN KEY (column3): Mendefinisikan kolom column3 sebagai foreign key.
- REFERENCES other\_table (primary\_key\_column): Menentukan tabel referensi dan kolom primary key-nya.

CONTOH:

```
CREATE TABLE customers ( customer_id SERIAL PRIMARY KEY, name  
VARCHAR(100) );
```

```
CREATE TABLE orders ( order_id SERIAL PRIMARY KEY, customer_id INTEGER  
REFERENCES customers(id) );
```

Pada Contoh di atas:

- customers adalah tabel dengan kolom customer\_id sebagai primary key.
- orders adalah tabel dengan kolom customer\_id sebagai foreign key yang mereferensikan customer\_id pada tabel customers.

### **Opsi Tambahan**

- **ON DELETE:** Menentukan tindakan yang dilakukan ketika baris yang direferensikan dihapus. Opsi:
  - **CASCADE:** Hapus juga baris yang terkait di tabel ini.
  - **RESTRICT:** Tidak mengizinkan penghapusan baris yang direferensikan.
  - **SET NULL:** Set kolom foreign key menjadi NULL.
  - **SET DEFAULT:** Set kolom foreign key ke nilai default.
  - **ON UPDATE:** Menentukan tindakan yang dilakukan ketika nilai dalam kolom yang direferensikan diubah. Opsi yang sama dengan ON DELETE tersedia.

Contoh dengan Opsi:

```
CREATE TABLE orders (  
  order_id SERIAL PRIMARY KEY,  
  customer_id INTEGER REFERENCES customers(customer_id) ON  
  DELETE CASCADE ON UPDATE RESTRICT  
);
```

#### **b. Setelah Tabel Dibuat**

Jika tabel sudah ada, kita bisa menggunakan perintah **ALTER TABLE** untuk menambahkan foreign key.

```
ALTER TABLE orders  
ADD CONSTRAINT fk_orders_customers  
FOREIGN KEY (customer_id)  
REFERENCES customers(id);
```

#### **Keterangan:**

**ALTER TABLE orders:** Menunjukkan bahwa kita akan mengubah struktur tabel orders.

**ADD CONSTRAINT fk\_orders\_customers:** Menambahkan sebuah constraint dengan nama fk\_orders\_customers. Nama constraint ini opsional, tetapi disarankan untuk diberikan nama yang jelas.

**FOREIGN KEY (customer\_id):** Menentukan kolom customer\_id sebagai foreign key.

**REFERENCES customers(id):** Menentukan bahwa foreign key customer\_id merujuk ke kolom id pada tabel customers.