Essay Report

PacCars Database

Rizky Bagaskara

rzkybgs11@gmail.com

Daftar Isi

Langkah #1 – Designing the Database

1. Mission Statement

Website “PacCars” merupakan website yang mempertemukan pengguna untuk jual-beli mobil bekas. Pada website tersebut dapat dilakukan:

- Pengguna dapat memposting iklan mobil bekas lebih dari satu dalam website
- Pengguna dapat melakukan penawaran (jika penjual mengizinkan) terkait iklan yang terpasang di dalam website
- Transaksi pembelian dilakukan di luar website PacCars sehingga tidak dalam scope project ini

Berdasarkan kebutuhan-kebutuhan di atas, dibutuhkan sebuah database untuk website “PacCars”yang memiliki fungsi transaksional dan analitik.

2. Creating Table Structures

Berdasarkan *mission statement* yang ada di atas, berikut adalah tabel-tabel yang ada pada database tersebut:

	Tabel Pengguna	**Tabel Iklan**	**Tabel Detail Iklan**	**Tabel Mobil**	**Tabel Penawaran**
Fungsi	Menyimpan data diri pengguna	Menyimpan data iklan di dalam website	Menyimpan data detail iklan (deskripsi dll)	Menyimpan data mobil bekas (master data) di dalam website	Menyimpan data penawaran yang dilakukan oleh pengguna

Kolom/field dari masing-masing tabel adalah sebagai berikut:

1. Tabel Pengguna

Nama Kolom	**Tipe Data**	**Panjang Data**	**Keterangan**
id_pengguna	INT		Primary Key
nama	VARCHAR	200	
notelp	VARCHAR	20	

email	VARCHAR	200	
alamat	VARCHAR	200	
domisili	VARCHAR	200	
latitude	DECIMAL		
longitude	DECIMAL		
tanggal_bergabung	DATE		

2. Tabel Iklan

Nama Kolom	Tipe Data	Panjang Data	Keterangan
id_iklan	INT		Primary Key
id_mobil	INT		Foreign Key
id_pengguna	INT		Foreign Key
id_detail	INT		Foreign Key
judul	VARCHAR	200	
is_bid	BOOLEAN	2	
id_penawaran	INT		Foreign Key
harga	DECIMAL		
tanggal_dibuat	DATE		

3. Tabel Detail Iklan

Nama Kolom	Tipe Data	Panjang Data	Keterangan
id_detail	INT	20	Primary Key

deskripsi	VARCHAR	200	
alamat	VARCHAR	200	
domisili	VARCHAR	200	
latitude	DECIMAL		
longitude	DECIMAL		

4. Tabel Mobil

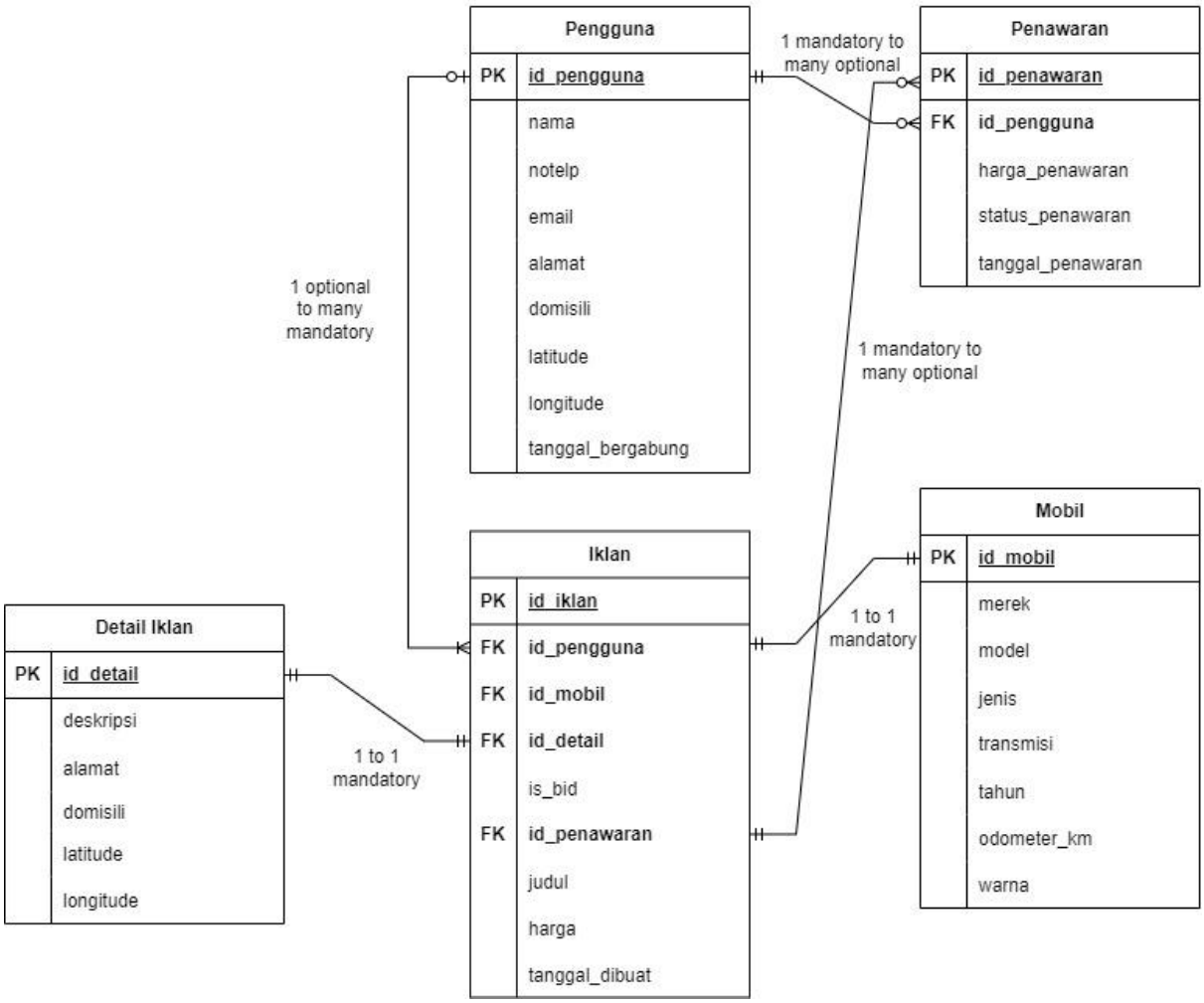
Nama Kolom	Tipe Data	Panjang Data	Keterangan
id_mobil	INT		Primary Key
merek	VARCHAR	20	
model	VARCHAR	50	
jenis	VARCHAR	15	
transmisi	VARCHAR	15	
tahun	INT		
odometer_km	DECIMAL	20	
warna	VARCHAR	20	

5. Tabel Penawaran

Nama Kolom	Tipe Data	Panjang Data	Keterangan

id_penawaran	INT		Primary Key
id_pengguna	INT		Foreign Key
harga_penawaran	DECIMAL		
status_penawaran	VARCHAR	20	CHECK_CONSTRAINT (Pending, Diterima, Ditolak)
tanggal_penawaran	DATE		

3. Determine Table Relationships



Relationship Review

1. Pengguna -> Iklan

Pengguna boleh saja tidak memasang iklan (hanya berstatus sebagai pembeli), tetapi iklan yang muncul pada website harus dimasukkan oleh pengguna, dimana mobil yang diiklankan bisa > 1 (data mobil harus distinct/unique). (1 optional to many mandatory)

2. Pengguna -> Penawaran

Pengguna boleh saja tidak melakukan penawaran yang berarti bahwa suatu iklan tidak ada yang menawar, tetapi dalam suatu penawaran harus ada data pengguna yang menawar (1 mandatory to many optional)

3. Iklan -> Penawaran

Sebuah iklan bisa saja tidak ada yang menawar, tetapi iklan tersebut harus ada di dalam website. (1 mandatory to many optional)

4. Iklan -> Mobil

Setiap iklan yang terpampang pada website wajib untuk mempunyai detail mobil yang diiklankan dan iklan yang sudah muncul tidak boleh ada duplikasi (1 to 1 mandatory).

5. Detail Iklan -> Iklan

Tiap iklan harus punya detailnya sebanyak exact 1 (1 to 1 mandatory)

4. Determine Business Rules

	Tabel Pengguna	**Tabel Iklan**	**Tabel Detail Iklan**	**Tabel Mobil**	**Tabel Penawaran**
Field	id_pengguna nama notelp email alamat domisili latitude longitude tanggal_bergabung	id_iklan id_pengguna id_mobil id_detail judul is_bid id_penawaran harga tanggal_dibuat	id_detail deskripsi alamat domisili latitude longitude	id_mobil merek model jenis transmisi tahun odometer_km warna	id_penawaran id_pengguna harga_penawaran status_penawaran tanggal_penawaran
Rule	Alamat boleh	- Setiap kolom	Setiap kolom	Setiap kolom	- Setiap kolom

	kosong	tidak boleh kosong - Nilai harga tidak boleh < 0	tidak boleh kosong	tidak boleh kosong	tidak boleh kosong - Nilai harga_penawaran tidak boleh < 0 - Nilai status_penawaran hanya bisa (‘Pending, ‘Diterima’, ‘Ditolak’)
Constraint	Field alamat bisa NULL, selain itu wajib NOT NULL	- Semua field wajib NOT NULL - Penambahan CHECK > 0 pada field harga	- Semua field wajib NOT NULL	- Semua field wajib NOT NULL	- Semua field wajib NOT NULL - Penambahan CHECK > 0 pada field harga_penawaran - Penambahan CHECK IN (‘Pending, ‘Diterima’, ‘Ditolak’) pada status_penawaran

Delete Action:

	Tabel Pengguna	Tabel Iklan	Tabel Detail Iklan	Tabel Mobil	Tabel Penawaran
Tabel Pengguna	-	NO ACTION	-	-	NO ACTION
Tabel Iklan	NO ACTION	-	NO ACTION	NO ACTION	NO ACTION
Tabel Detail Iklan	-	NO ACTION	-	-	-

Tabel Mobil	-	NO ACTION	-	-	-
Tabel Penawaran	NO ACTION	NO ACTION	-	-	-

5. Implementing a Relational Database

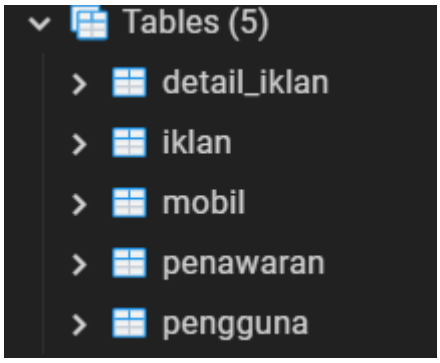
Menggunakan query sebagai berikut:

```
--Tabel Mobil
CREATE TABLE Mobil (
        id_mobil INT PRIMARY KEY,
        merek VARCHAR(20) NOT NULL,
   model VARCHAR(50) NOT NULL,
   jenis VARCHAR(15) NOT NULL,
   transmisi VARCHAR(15) NOT NULL,
   tahun INT NOT NULL,
   odometer_km DECIMAL(20) NOT NULL,
   warna VARCHAR(20) NOT NULL
);

--Tabel Pengguna
CREATE TABLE Pengguna (
   id_pengguna INT PRIMARY KEY NOT NULL,
   nama VARCHAR(200) NOT NULL,
   notelp VARCHAR(20) NOT NULL,
   email VARCHAR(200) UNIQUE NOT NULL,
        alamat VARCHAR(200) NOT NULL,
   domisili VARCHAR(200) NOT NULL,
   tanggal_bergabung DATE NOT NULL,
        latitude float8 NOT NULL,
        longitude float8 NOT NUL
);
```

```
--Tabel Detail Iklan
CREATE TABLE Detail_Iklan(
        id_detail INT PRIMARY KEY,
        deskripsi VARCHAR(200) NOT NULL,
        alamat VARCHAR(200) NOT NULL,
        domisili VARCHAR(200) NOT NULL,
        latitude float8 NOT NULL,
        longitude float8 NOT NUL
);

--Tabel Penawaran
CREATE TABLE Penawaran (
   id_penawaran INT PRIMARY KEY,
   id_pengguna INT NOT NULL,
   harga_penawaran DECIMAL(20) CHECK(harga_penawaran >= 0) NOT NULL,
        status_penawaran VARCHAR(40) CHECK(status_penawaran IN ('Pending',
'Diterima', 'Ditolak')) NOT NULL,
        tanggal_penawaran DATE NOT NULL,
        CONSTRAINT fk_pengguna
                FOREIGN KEY(id_pengguna)
                REFERENCES pengguna(id_pengguna)
);

--Tabel Iklan
CREATE TABLE Iklan (
   id_iklan INT PRIMARY KEY NOT NULL,
        id_detail INT NOT NULL,
   id_mobil INT NOT NULL,
        id_pengguna INT NOT NULL,
        judul VARCHAR(200) NOT NULL,
   is_bid BOOLEAN NOT NULL,
        id_penawaran INT,
   harga DECIMAL(20) CHECK(harga >= 0) NOT NULL,
```

```
    tanggal_dibuat DATE NOT NULL,
        CONSTRAINT fk_detail
            FOREIGN KEY(id_detail)
            REFERENCES detail_iklan(id_detail),
        CONSTRAINT fk_mobil
            FOREIGN KEY(id_mobil)
            REFERENCES mobil(id_mobil),
        CONSTRAINT fk_pengguna
            FOREIGN KEY(id_pengguna)
            REFERENCES pengguna(id_pengguna),
        CONSTRAINT fk_penawaran
            FOREIGN KEY(id_penawaran)
            REFERENCES penawaran(id_penawaran)
);
```

Hasil dalam database PacCars:

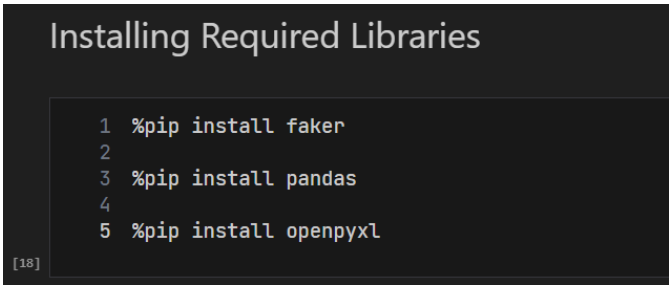


Langkah #2 – Populating the Database

1. Pembuatan Dummy Data

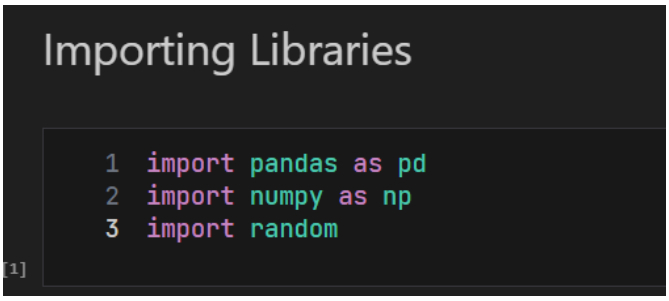
Dummy data di *generate* menggunakan Python dengan bantuan library Faker dengan **locale =id_ID** yang berarti menggunakan domain Indonesia. Code dibuat dengan menggunakan Code Editor Visual Studio Code dengan extension Jupyter Notebook. Berikut adalah codenya:

1. Install library-library yang dibutuhkan



   - faker untuk generate random data
   - pandas untuk menyimpan data dalam format DataFrame yang akan memudahkan dalam proses export data
   - openpyxl untuk menyimpan data dalam format XLSX (akan dijelaskan di bawah lebih lanjut)

2. Import library



   - Import library pandas, numpy, dan random (untuk menggunakan method random.choice)

3. Deklarasi global faker

Global Faker Declaration

```
[2]
from faker import Faker
from datetime import date
f = Faker('id_ID') #using Indonesian Locale
```

- Faker dideklarasikan secara Global agar bisa digunakan oleh function-function yang ada di bawah (untuk generate data random tiap table)
- Faker menggunakan localization Indonesia yang ditandai dengan ‘id_ID’

4. Membuat primary keys

Generate Primary Keys

```
def generate_unique_ids(num_entries):
    return list(range(1, num_entries + 1)) #simpan ke dalam list

pengguna_ids = generate_unique_ids(40)
mobil_ids = generate_unique_ids(80)
iklan_ids = generate_unique_ids(100)
detail_iklan_ids = generate_unique_ids(100)
penawaran_ids = generate_unique_ids(50)
```

5. Generate data tiap table
   a. Pengguna

Generate Data Pengguna

```
def generate_data_pengguna(entries = 40):
  data = []
  for i in range(entries):
    random_date = f.date_between_dates(date_start=date(2020, 1, 1), date_end=date(2024, 6, 6))
    nama_depan = f.first_name()
    nama_belakang = f.last_name()
    nama = f"{nama_depan} {nama_belakang}" #menggunakan F-String untuk string interpolation (gabungan first dan last name)
    email = f"{nama.lower().replace(' ', '')}@{f.free_email_domain()}" #match email dengan nama dari pengguna
    fake_data_pengguna = {
        #"ID_Pengguna": np.random.randint(1, 100),
        "ID_Pengguna": pengguna_ids[i],
        "Nama": nama,
        "NoTelp": f.phone_number(),
        "Email": email,
        "Alamat": f.address(),
        "Domisili": f.city(),
        'Latitude': f.latitude(),
        'Longitude': f.longitude(),
        'Tanggal_Bergabung': random_date #set tahunnya antara 2020-2024
    }
    data.append(fake_data_pengguna)
  return pd.DataFrame(data)

generate_data_pengguna()
```

```
[5]
generate_data_pengguna().to_excel('pengguna.xlsx', index=False)
```

b. Mobil

Generate Data Mobil

Dictionary untuk matching merek mobil dengan namanya

Ex: Toyota -> Toyota Corolla

```python
merek_model_dict = {
        'Toyota': ['Toyota Camry', 'Toyota Corolla Altis', 'Toyota Vios', 'Toyota Camry Hybrid', 'Toyota RAV4', 'Toyota Fortuner',
                   'Toyota Hilux', 'Toyota Yaris', 'Toyota Avanza', 'Toyota Land Cruiser'],

        'Honda': ['Honda Civic', 'Honda Accord', 'Honda City', 'Honda CR-V', 'Honda HR-V', 'Honda Jazz',
                  'Honda BR-V', 'Honda Pilot', 'Honda Odyssey', 'Honda Fit'],

        'Suzuki': ['Suzuki Swift', 'Suzuki Ertiga', 'Suzuki Baleno', 'Suzuki Ignis', 'Suzuki XL7', 'Suzuki Jimny',
                   'Suzuki Vitara', 'Suzuki Ciaz', 'Suzuki S-Cross', 'Suzuki Alto'],

        'Volkswagen': ['Volkswagen Golf', 'Volkswagen Tiguan', 'Volkswagen Passat', 'Volkswagen Polo', 'Volkswagen Touareg',
                       'Volkswagen Jetta', 'Volkswagen Arteon', 'Volkswagen ID.4', 'Volkswagen T-Roc', 'Volkswagen Atlas'],

        'BMW': ['BMW 3 Series', 'BMW 5 Series', 'BMW X1', 'BMW X5', 'BMW Z4', 'BMW 7 Series',
                'BMW 2 Series', 'BMW X3', 'BMW i3', 'BMW i8'],

        'Mercedes-Benz': ['Mercedes-Benz C-Class', 'Mercedes-Benz E-Class', 'Mercedes-Benz GLC', 'Mercedes-Benz GLE',
                          'Mercedes-Benz A-Class', 'Mercedes-Benz S-Class', 'Mercedes-Benz B-Class', 'Mercedes-Benz GLA',
                          'Mercedes-Benz GLS', 'Mercedes-Benz CLA'],

        'Hyundai': ['Hyundai Tucson', 'Hyundai Santa Fe', 'Hyundai Elantra', 'Hyundai Kona', 'Hyundai Palisade', 'Hyundai Ioniq',
```

```python
def generate_data_mobil(entries = 80):
  data = []
  jenis_lists = ['MPV', 'SUV', 'Sedan', 'Hatchback', 'Van', 'Station Wagon']
  years = [ 2005, 2010, 2011, 2020]
  transmisi_lists = ['manual', 'matic']

  for i in range(entries):
    merek = random.choice(list(merek_model_dict.keys())) #ubah jadi list kemudian ambil key dari dictionarnya (merek)
    model = random.choice(merek_model_dict[merek]) #ambil value berdasarkan key dict tersebut (merek)
    fake_data_mobil = {
        #"ID_Mobil": np.random.randint(1, 100),
        'ID_Mobil': mobil_ids[i],
        "Merek": merek,
        "Model": model,
        "Jenis": random.choice(jenis_lists),
        "Transmisi": random.choice(transmisi_lists),
        "Tahun": random.choice(years),
        "Odometer_KM": np.random.randint(8000, 200_000),
        "Warna": f.color_name(),
    }
    data.append(fake_data_mobil)
  return pd.DataFrame(data)

generate_data_mobil()
```

```python
generate_data_mobil().to_excel('mobil.xlsx', index=False)
```

c. Detail_Iklan

Generate Data Detail Iklan

```python
deskripsi_mobil_bekas = [
    "Full original, STNK dan BPKB sesuai",
    "STNK dan BPKB sesuai, nomor mesin nomor rangka akur",
    "STNK dan BPKB sesuai, bodi ngaleng",
    "BPKB only, body lecet pemakaian",
    "FULL 100% GRESS ORIGINAL, bekas pemakaian dokter",
    "Lecet pemakaian, mesin sehat dan kering, surat-surat lengkap",
    "Kondisi mesin prima, interior bersih, surat-surat komplit",
    "Bekas pemakaian pribadi, tidak pernah kecelakaan, surat-surat lengkap",
    "Mesin halus, ac dingin, surat-surat lengkap",
    "Mobil mulus, ban baru, surat-surat lengkap",
    "Siap pakai, surat-surat lengkap, kondisi terawat",
    "Interior bersih, mesin terawat, surat-surat lengkap",
    "Mobil langka, koleksi pribadi, surat-surat lengkap",
    "Cat orisinil, tidak pernah tabrakan, surat-surat lengkap",
    "Mesin bagus, kaki-kaki empuk, surat-surat lengkap",
    "Kondisi masih baru, pemakaian pribadi, surat-surat lengkap",
    "Ban baru, oli baru ganti, surat-surat lengkap",
    "Tidak bekas banjir, bodi mulus, surat-surat lengkap",
    "Mesin sehat, ac dingin, surat-surat lengkap",
    "Mobil antik, kondisi terawat, surat-surat lengkap",
    "Kondisi siap pakai, tidak ada PR, surat-surat lengkap",
    "Harga nego, mesin sehat, surat-surat lengkap",
    "Bekas pakai sendiri, terawat, surat-surat lengkap",
    "Bodi mulus, mesin kering, surat-surat lengkap",
    "Mobil hobi, perawatan rutin, surat-surat lengkap",
    "Mesin standar, tidak ada modifikasi, surat-surat lengkap",
    "Surat-surat lengkap, pajak hidup, mesin terawat",
```

```python
def generate_data_detail_iklan(entries = 100):
    data = []
    for i in range(entries):
        fake_data_detail_iklan = {
            "ID_Detail": detail_iklan_ids[i],
            "Deskripsi": random.choice(deskripsi_mobil_bekas),
            "Alamat": f.address(),
            'Latitude': f.latitude(),
            'Longitude': f.longitude(),
            "Domisili": f.city(),
        }
        data.append(fake_data_detail_iklan)
    return pd.DataFrame(data)

generate_data_detail_iklan()
```

```python
generate_data_detail_iklan().to_excel('detail_iklan.xlsx', index=False)
```

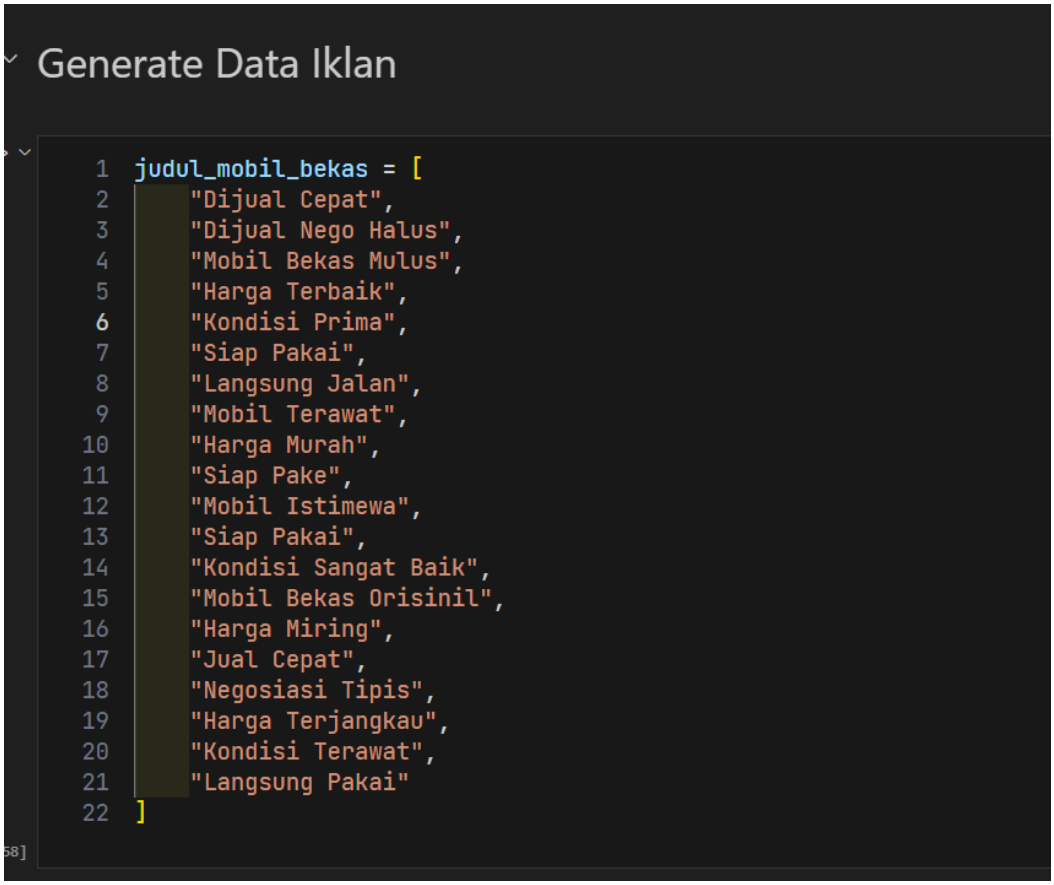
d. Penawaran

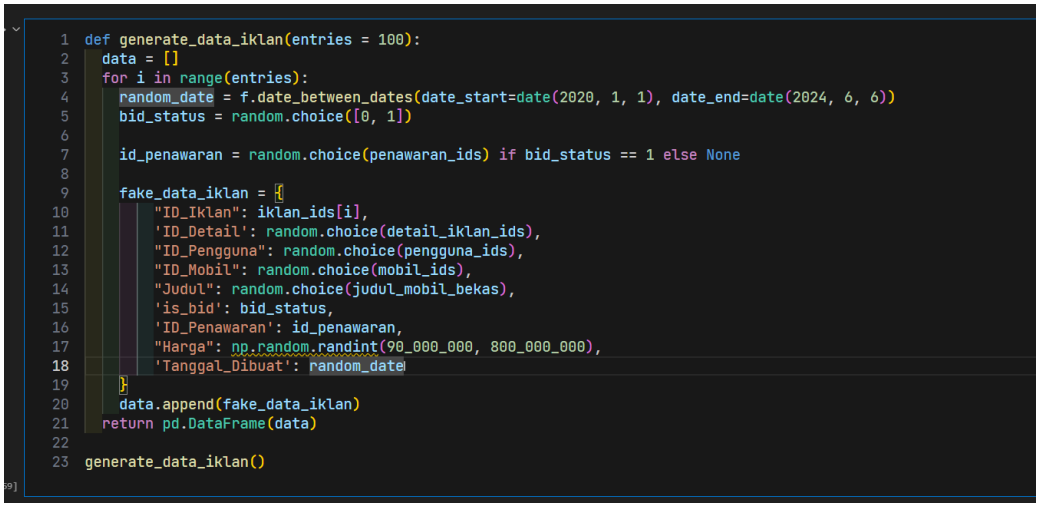
Generate Data Penawaran

```python
def generate_data_penawaran(entries = 50):
  data = []
  status = ['Pending', 'Diterima', 'Ditolak']
  for i in range(entries):
    random_date = f.date_between_dates(date_start=date(2020, 1, 1), date_end=date(2024, 6, 6))
    fake_data_penawaran = {
        'ID_Penawaran': penawaran_ids[i],
        'ID_Pengguna': random.choice(pengguna_ids),
        "Harga_Penawaran": np.random.randint(70_000_000, 450_000_000),
        "Status_Penawaran": random.choice(status),
        'Tanggal_Penawaran': random_date
    }
    data.append(fake_data_penawaran)
  return pd.DataFrame(data)

generate_data_penawaran()
```

```python
generate_data_penawaran().to_excel('penawaran.xlsx', index=False)
```

e. Iklan



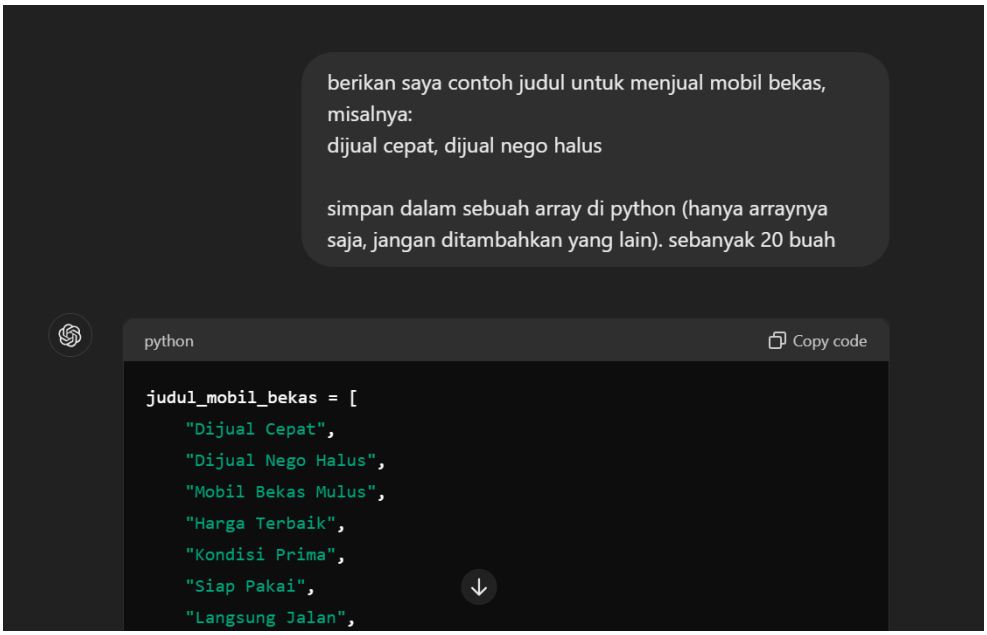


Catatan:

- Dummy data yang ada pada tiap function di export menjadi format XLSX (Excel) bukan CSV karena ketika di-import ke dalam PostgreSQL menggunakan format

CSV, datanya berantakan (meskipun sudah diberikan delimiter ;, sehingga harus di save as CSV secara manual).

- Dictionary dan array yang memuat *catchphrase* serta merek dan model mobil di-*generate* dengan bantuan Chat GPT untuk mempersingkat waktu.



2. Penginputan Dummy Data ke dalam Database

Penginputan data ke dalam database dilakukan berdasarkan urutan *dependency* terendah atau tingkat ketergantungan rendah dengan tabel lain.

Nama Tabel	Tingkatan Dependency
pengguna	0
mobil	0
detail_iklan	0
penawaran	1 (pengguna)
iklan	4 (pengguna, mobil, detail_iklan, dan penawaran)

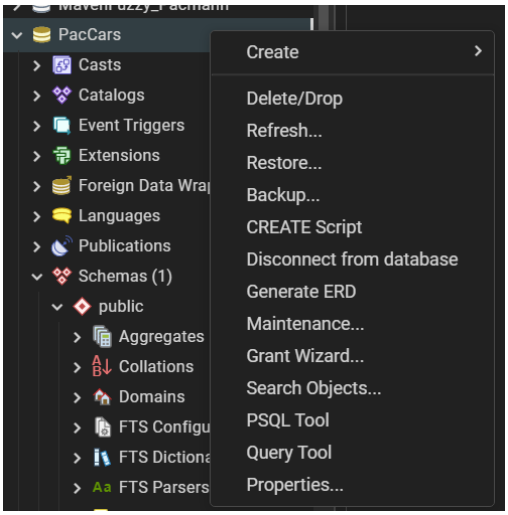
Penginputan data ke dalam database menggunakan keyword **COPY** dari PostgreSQL. Berikut adalah query-nya:

```
COPY pengguna(id_pengguna, nama, notelp, email, alamat, domisili, latitude, longitude, tanggal_bergabung)
FROM 'E:\Pacmann Final Project - Database\Dummy Data\CSV\pengguna.csv'
DELIMITER ';'
CSV HEADER;

select * from pengguna

COPY mobil(id_mobil, merek, model, jenis, transmisi, tahun, odometer_km, warna)
FROM 'E:\Pacmann Final Project - Database\Dummy Data\CSV\mobil.csv'
DELIMITER ';'
CSV HEADER;

select * from mobil

COPY detail_iklan(id_detail, deskripsi, alamat, latitude, longitude, domisili)
FROM 'E:\Pacmann Final Project - Database\Dummy Data\CSV\detail_iklan.csv'
DELIMITER ';'
CSV HEADER;

select * from detail_iklan

COPY penawaran(id_penawaran, id_pengguna, harga_penawaran, status_penawaran, tanggal_penawaran)
FROM 'E:\Pacmann Final Project - Database\Dummy Data\CSV\penawaran.csv'
DELIMITER ';'
CSV HEADER;

select * from penawaran

COPY iklan(id_iklan, id_detail, id_pengguna, id_mobil, judul, is_bid, id_penawaran, harga, tanggal_dibuat)
```

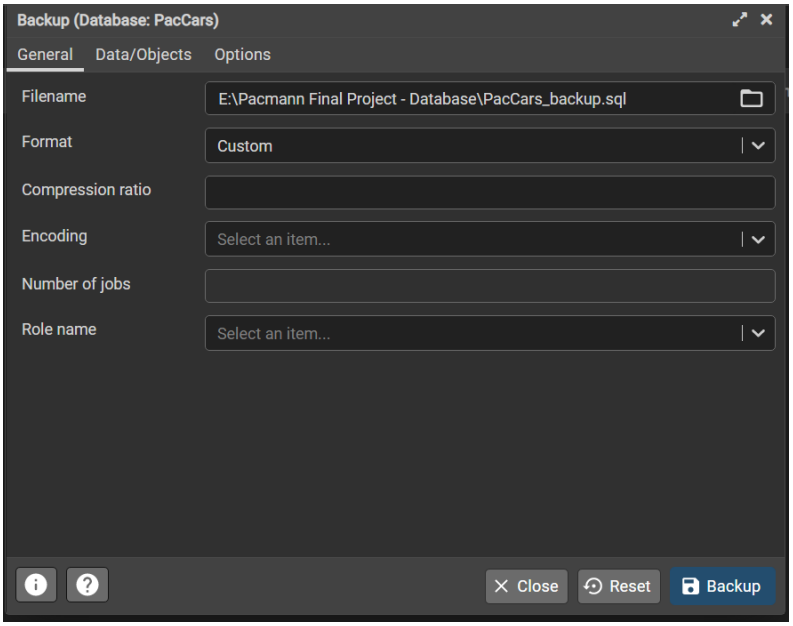
```
FROM 'E:\Pacmann Final Project - Database\Dummy Data\CSV\iklan.csv'
DELIMITER ';'
CSV HEADER;

select * from iklan
```

#Langkah 3 – Database Backup

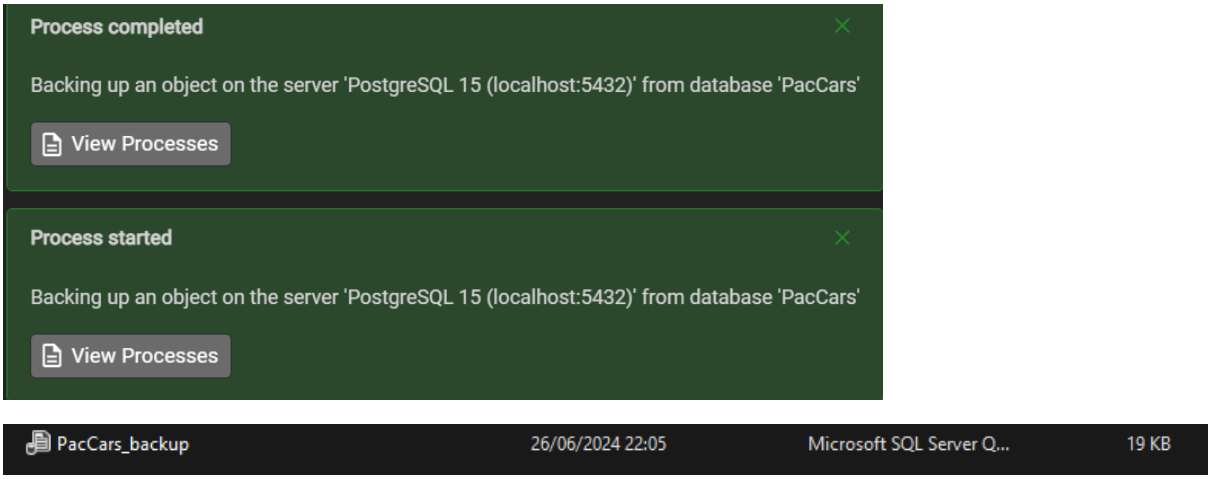
1. Klik kanan pada database PacCars, kemudian pilih Backup



2. Pada Backup Dialog, beri nama file dan file path yang diinginkan



3. Klik Backup, maka file backup akan muncul pada filepath


Process completed
Backing up an object on the server 'PostgreSQL 15 (localhost:5432)' from database 'PacCars'
View Processes
Process started
Backing up an object on the server 'PostgreSQL 15 (localhost:5432)' from database 'PacCars'
View Processes
PacCars_backup
26/06/2024 22:05
Microsoft SQL Server Q...
19 KB

#Langkah 4 - Transactional and Analytical Queries

I. Transactional

1. Mencari mobil keluaran >= 2015

```
Query  Query History
1  --Transactional Queries
2  --1. Cari mobil keluaran 2015 ke atas
3  SELECT * FROM mobil
4  WHERE tahun >= 2015
```

	id_mobil [PK] integer	merek character varying (20)	model character varying (50)	jenis character varying (15)	transmisi character varying (15)	tahun integer	odometer_km numeric (20)	warna character varying (20)
1	3	Honda	Honda CR-V	Van	manual	2020	25960	Putih
2	4	Suzuki	Suzuki Jimny	Station Wagon	matic	2020	14650	Zaitun
3	5	Mazda	Mazda RX-8	Hatchback	matic	2020	54632	Hitam
4	15	Volkswagen	Volkswagen T-Roc	SUV	matic	2020	141953	Putih
5	21	BMW	BMW 3 Series	Station Wagon	manual	2020	42911	Biru dongker
6	24	Mercedes-Benz	Mercedes-Benz C-Cla...	Sedan	matic	2020	178853	Biru dongker
7	28	Suzuki	Suzuki Alto	SUV	manual	2020	69775	Biru muda
8	35	Hyundai	Hyundai Tucson	MPV	manual	2020	125429	Merah jambu
9	38	Mercedes-Benz	Mercedes-Benz GLC	Van	manual	2020	182996	Merah marun
10	39	Isuzu	Isuzu MU-X	Sedan	manual	2020	195181	Putih
11	42	BMW	BMW 5 Series	Station Wagon	manual	2020	160350	Ungu tua
12	45	Land Rover	Land Rover Discovery	Hatchback	matic	2020	26411	Biru muda
13	49	Mazda	Mazda CX-9	Station Wagon	matic	2020	123045	Magenta
14	52	KIA	KIA Carnival	Hatchback	matic	2020	158102	Biru dongker
15	56	Mercedes-Benz	Mercedes-Benz B-Cla...	Van	manual	2020	118744	Merah bata
16	60	Mercedes-Benz	Mercedes-Benz GLC	Sedan	matic	2020	20173	Hijau muda
17	66	BMW	BMW 5 Series	Sedan	matic	2020	102138	Hijau tua

2. Memasukkan data penawaran

   Before:

Data Output Messages Notifications

	id_penawaran [PK] integer	id_pengguna integer	harga_penawaran numeric (20)	status_penawaran character varying (40)	tanggal_penawaran date
39	39	32	430608239	Ditolak	2020-10-28
40	40	10	217375882	Diterima	2023-08-04
41	41	17	421765615	Pending	2021-01-27
42	42	1	143918948	Diterima	2020-04-17
43	43	26	131834533	Diterima	2021-06-28
44	44	35	420667774	Pending	2022-11-18
45	45	36	422461421	Pending	2021-02-13
46	46	21	325987894	Diterima	2021-07-17
47	47	11	427097572	Diterima	2020-04-09
48	48	33	130702446	Ditolak	2024-04-11
49	49	7	387868637	Diterima	2020-04-05
50	50	14	282011411	Pending	2024-05-13

Total rows: 50 of 50 Query complete 00:00:00.088

   Proses memasukkan data:

```
--2. tambah satu data bid (penawaran) produk baru
INSERT INTO penawaran (id_penawaran, id_pengguna, harga_penawaran, status_penawaran, tanggal_penawaran)
VALUES (51, 19, 100000000, 'Pending', '2024-01-01')
```

After:

Data Output | Messages | Notifications

	id_penawaran [PK] integer	id_pengguna integer	harga_penawaran numeric (20)	status_penawaran character varying (40)	tanggal_penawaran date
46	46	21	325987894	Diterima	2021-07-17
47	47	11	427097572	Diterima	2020-04-09
48	48	33	130702446	Ditolak	2024-04-11
49	49	7	387868637	Diterima	2020-04-05
50	50	14	282011411	Pending	2024-05-13
51	51	19	100000000	Pending	2024-01-01

3. Melihat semua mobil yang dijual oleh 1 akun (random) dari yang paling baru (earliest post)

   Pilih random nama:

Data Output | Messages | Notifications

	id_pengguna [PK] integer	nama character varying (200)
9	40	Bakiono Melani
10	13	Kamila Prabowo
11	19	Luwes Hutapea
12	36	Respati Wibisono
13	12	Tugiman Tamba
14	37	Praseiyo Uyainah
15	31	Adikara Mangunsong
16	16	Paulin Samosir
17	12	Tugiman Tamba
18	11	Najam Purwanti
19	11	Najam Purwanti
20	14	Raden Dongoran
21	35	Nadia Hariyah

   Misalnya dengan nama = ‘Tugiman Tamba’ dengan id_pengguna = 12

```sql
--3. Melihat semua mobil yang dijual oleh 1 akun (random) dari yang paling baru (earliest post)
-- cari pengguna bernama "Tugiman Tamba"
SELECT m.id_mobil,
       i.id_iklan,
       m.merek,
       m.model,
       m.tahun,
       i.harga,
       i.tanggal_dibuat
FROM mobil m
JOIN iklan i USING(id_mobil)
JOIN pengguna p USING(id_pengguna)
WHERE p.nama = 'Tugiman Tamba'
ORDER BY i.tanggal_dibuat
```

	id_mobil integer	id_iklan integer	merek character varying (20)	model character varying (50)	tahun integer	harga numeric (20)	tanggal_dibuat date
1	53	1	Lexus	Lexus LX	2005	146687153	2020-03-04
2	76	73	Toyota	Toyota Corolla Altis	2020	194843630	2021-10-19
3	74	68	Hyundai	Hyundai Ioniq	2020	706416464	2021-11-08
4	28	17	Suzuki	Suzuki Alto	2020	467808546	2022-05-14
5	57	13	KIA	KIA Seltos	2010	418448485	2022-07-19

4. Mencari harga mobil bekas termurah berdasarkan keyword = ‘Honda’

```sql
--4. Mencari mobil bekas termurah berdasarkan keyword
SELECT m.id_mobil,
       m.merek,
       m.model,
       m.tahun,
       MIN(i.harga) as harga
FROM mobil m
JOIN iklan i USING(id_mobil)
WHERE model LIKE 'Honda%'
GROUP BY m.id_mobil, m.merek, m.model, m.tahun
ORDER BY harga
```

	id_mobil [PK] integer	merek character varying (20)	model character varying (50)	tahun integer	harga numeric
1	80	Honda	Honda Pilot	2011	168076658
2	33	Honda	Honda Fit	2010	213419222
3	3	Honda	Honda CR-V	2020	232698796
4	13	Honda	Honda Pilot	2011	245175479
5	8	Honda	Honda City	2005	313926126
6	67	Honda	Honda City	2005	370189242
7	46	Honda	Honda Fit	2010	533336026

5. Mencari mobil bekas terdekat berdasarkan sebuah id kota/nama kota, dihitung berdasarkan latitude dan longitude menggunakan rumus Euclidean:

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

where,

- (x_1, y_1) are the coordinates of one point.
- (x_2, y_2) are the coordinates of the other point.
- d is the distance between (x_1, y_1) and (x_2, y_2).

a. Buat function untuk rumus Euclidean

```
CREATE OR REPLACE FUNCTION euclidean_distance(lat1 float, lon1 float, lat2 float, lon2 float)
RETURNS float AS $$
BEGIN
    RETURN sqrt(pow(lat2 - lat1, 2) + pow(lon2 - lon1, 2));
END;
$$ LANGUAGE plpgsql;
```

b. Buat CTE untuk mendapatkan data mobil (Toyota Yaris) dan asal domisili dari iklan tersebut

```
-- misalnya mencari merk Toyota Yaris
WITH yaris_data AS(
    SELECT m.id_mobil,
           m.merek,
           m.model,
           m.tahun,
           di.domisili,
           di.latitude,
           di.longitude,
           MIN(i.harga) as harga
    FROM mobil m
    JOIN iklan i USING(id_mobil)
    JOIN detail_iklan di USING(id_detail)
    WHERE m.model = 'Toyota Yaris'
    GROUP BY m.id_mobil, m.merek, m.model, m.tahun, di.domisili, di.latitude, di.longitude
    ORDER BY harga
)
```

Data Output | Messages | Notifications

	id_mobil integer	merek character varying (20)	model character varying (50)	tahun integer	domisili character varying (200)	latitude double precision	longitude double precision	harga numeric
1	77	Toyota	Toyota Yaris	2010	Madiun	-30.190006	172.879396	448809590
2	16	Toyota	Toyota Yaris	2005	Depok	75.0292115	-137.091796	730549286

c. Masukkan latitude dan longitude ke dalam function yang sudah dibuat di atas

```sql
SELECT yd1.id_mobil,
       yd1.merek,
       yd1.model,
       yd1.tahun,
       yd1.harga,
       yd1.domisili,
       euclidean_distance(yd1.latitude, yd1.longitude, yd2.latitude, yd2.longitude) AS distance
FROM yaris_data yd1, yaris_data yd2
WHERE euclidean_distance(yd1.latitude, yd1.longitude, yd2.latitude, yd2.longitude) != 0
```

Data Output | Messages | Notifications

	id_mobil integer	merek character varying (20)	model character varying (50)	tahun integer	harga numeric	domisili character varying (200)	distance double precision
1	77	Toyota	Toyota Yaris	2010	448809590	Madiun	327.34266999768477
2	16	Toyota	Toyota Yaris	2005	730549286	Depok	327.34266999768477

II. Analytical

1. Ranking popularitas model mobil berdasarkan jumlah bid

```sql
--Analytical Queries
--1.Ranking popularitas model mobil berdasarkan jumlah bid
SELECT m.model,
       COUNT(m.model) as count_product,
       COUNT(i.id_penawaran) as count_bid
FROM mobil m
JOIN iklan i USING(id_mobil)
GROUP BY m.model
HAVING COUNT(i.id_penawaran) > 0
ORDER BY count_product DESC
```

Data Output | Messages | Notifications

	model character varying (50)	count_product bigint	count_bid bigint
1	Mazda CX-9	6	3
2	Subaru BRZ	6	4
3	Honda Pilot	5	3
4	Honda City	4	3
5	Mazda 2	4	1
6	Suzuki Jimny	4	2
7	Mazda RX-8	4	2
8	BMW 2 Series	3	2
9	BMW 5 Series	3	3
10	Honda Fit	3	2
11	KIA Carnival	3	1
12	Isuzu D-Max	2	1
13	Mercedes-Benz A-Cla...	2	2
14	Toyota Yaris	2	1
15	Isuzu N-Series	2	2

2. Perbandingan harga mobil berdasarkan harga rata-rata per kota

```
12  --2. Membandingkan harga mobil berdasarkan harga rata-rata per kota
13  SELECT di.domisili,
14         m.merek,
15         m.model,
16         m.tahun,
17         i.harga,
18         AVG(i.harga) OVER(PARTITION BY di.domisili) as avg_car_city
19  FROM mobil m
20  JOIN iklan i USING(id_mobil)
21  JOIN detail_iklan di USING(id_detail)
22  GROUP BY di.domisili, m.merek, m.model, m.tahun, i.harga
```

Data Output | Messages | Notifications

	domisili character varying (200)	merek character varying (20)	model character varying (50)	tahun integer	harga numeric (20)	avg_car_city numeric
1	Ambon	Suzuki	Suzuki Alto	2020	537597629	537597629.000
2	Balikpapan	Suzuki	Suzuki Jimny	2020	777400553	777400553.000
3	Banda Aceh	Honda	Honda Fit	2010	337823680	337823680.000
4	Bandung	BMW	BMW 7 Series	2005	658818596	590243806.000
5	Bandung	Honda	Honda City	2005	521669016	590243806.000
6	Banjarmasin	Isuzu	Isuzu MU-X	2020	371531380	371531380.000
7	Bogor	Honda	Honda Pilot	2011	168076658	168076658.000
8	Bukittinggi	Mazda	Mazda CX-30	2005	225188069	316966073.333
9	Bukittinggi	Mazda	Mazda RX-8	2010	361964130	316966073.333
10	Bukittinggi	Subaru	Subaru Forester	2011	363746021	316966073.333
11	Denpasar	BMW	BMW 7 Series	2005	458114691	318306387.333
12	Denpasar	Mazda	Mazda CX-9	2005	388334631	318306387.333
13	Denpasar	Subaru	Subaru BRZ	2011	108469840	318306387.333
14	Depok	Toyota	Toyota Yaris	2005	730549286	730549286.000
15	Dumai	Isuzu	Isuzu Giga	2005	627900208	404895804.666
16	Dumai	Isuzu	Isuzu MU-X	2020	98604348	404895804.666

3. Dari penawaran suatu model mobil, cari perbandingan tanggal user melakukan bid dengan bid selanjutnya beserta harga tawar yang diberikan

   Contoh: Bid untuk mobil “Mazda CX-9”

```
--3. Membandingkan tanggal user melakukan bid terhadap suatu mobil
-- Mazda CX-9
-- Join 4 tabel (mobil, iklan, pengguna, dan penawaran)
-- CLUE = gunakan fungsi LEAD() pada tanggal_penawaran dan harga_penawaran
SELECT m.model,
       pg.id_pengguna as user_id,
       pe.tanggal_penawaran AS first_bid_date,
       LEAD(pe.tanggal_penawaran, 1) OVER(
            ORDER BY pe.tanggal_penawaran
       ) AS next_bid,
       pe.harga_penawaran AS first_bid_price,
       LEAD(pe.harga_penawaran, 1) OVER(
            ORDER BY pe.harga_penawaran
       ) AS next_bid_price
FROM mobil m
JOIN iklan i USING(id_mobil)
JOIN penawaran pe ON i.id_penawaran = pe.id_penawaran --harus di specified joinnya make kolom yang mana
JOIN pengguna pg ON pe.id_pengguna = pg.id_pengguna
WHERE m.model = 'Mazda CX-9';
```

Data Output | Messages | Notifications

	model character varying (50)	user_id integer	first_bid_date date	next_bid date	first_bid_price numeric (20)	next_bid_price numeric
1	Mazda CX-9	11	2020-10-02	2021-10-06	281169398	[null]
2	Mazda CX-9	25	2021-10-06	2022-04-01	235983438	267676686
3	Mazda CX-9	23	2022-04-01	[null]	267676686	281169398

4. Membandingkan persentase perbedaan rata-rata harga mobil berdasarkan modelnya dan rata-rata harga bid yang ditawarkan oleh customer pada 6 bulan terakhir
    - Difference adalah selisih antara rata-rata harga model mobil(avg_price) dengan rata-rata harga bid yang ditawarkan terhadap model tersebut(avg_bid_6month)
    - Difference dapat bernilai negatif atau positif
    - Difference_percent adalah persentase dari selisih yang telah dihitung, yaitu dengan cara difference dibagi rata-rata harga model mobil(avg_price) dikali 100%
    - Difference_percent dapat bernilai negatif atau positif

```sql
--4. Perbandingan persentase perbedaan rata-rata harga mobil berdasarkan modelnya dari rata-rata
--harga bid yang ditawarkan oleh customer selama 6 bulan terakhir
SELECT m.model,
       AVG(i.harga) as avg_price,
       AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '6' MONTH PRECEDING) as avg_bid_6month,
        AVG(i.harga) -  AVG(pe.harga_penawaran) OVER(
                    PARTITION BY m.model
                    ORDER BY pe.tanggal_penawaran
                    RANGE INTERVAL '6' MONTH PRECEDING) as difference,
        (AVG(i.harga) -  AVG(pe.harga_penawaran) OVER(
                    PARTITION BY m.model
                    ORDER BY pe.tanggal_penawaran
                    RANGE INTERVAL '6' MONTH PRECEDING)) / AVG(i.harga) * 100 as difference_percent
FROM mobil m
JOIN iklan i USING(id_mobil)
JOIN penawaran pe ON i.id_penawaran = pe.id_penawaran
GROUP BY m.model, i.harga, pe.harga_penawaran, i.tanggal_dibuat, pe.tanggal_penawaran
```

Data Output | Messages | Notifications

	model character varying (50)	avg_price numeric	avg_bid_6month numeric	difference numeric	difference_percent numeric
1	BMW 2 Series	531580843.00000000	531580843.00000000	214451444.00000000	40.342206989577312
2	BMW 2 Series	476403450.00000000	476403450.00000000	41048585.00000000	8.6163492308882313
3	BMW 3 Series	372064748.00000000	372064748.00000000	239680143.000000000000	64.418933609910283
4	BMW 5 Series	550351669.00000000	550351669.00000000	258909210.50000000	47.044321855231804
5	BMW 5 Series	441216282.00000000	495783975.50000000	14118710.00000000	3.1999521722092749
6	BMW 5 Series	534127887.00000000	534127887.00000000	103519648.00000000	19.381060326475707
7	BMW X3	397684367.00000000	397684367.00000000	-24777054.00000000	-6.230331402491363
8	Honda City	393361686.00000000	393361686.00000000	200435642.000000000000	50.954541109018939
9	Honda City	521669016.00000000	521669016.00000000	279895196.00000000	53.653789551496000
10	Honda City	313926126.00000000	313926126.00000000	-85531791.00000000	-27.245833945021829
11	Honda Fit	533336026.00000000	533336026.00000000	106238454.00000000	19.919609555871254
12	Honda Fit	337823680.00000000	337823680.00000000	144897636.000000000000	42.891497718573191
13	Honda Pilot	759690533.00000000	759690533.00000000	477679122.00000000	62.878119609264631
14	Honda Pilot	168076658.000000000	168076658.000000000	11896487.000000000000	7.0780125816161813
15	Honda Pilot	245175479.00000000	245175479.00000000	36055339.50000000	14.705932113219201
16	Hyundai Elantra	391010944.00000000	391010944.00000000	69138326.00000000	17.681941403665673
17	Hyundai Palisade	509237838.00000000	509237838.00000000	329203960.000000000000	64.646405949119593
18	Hyundai Tucson	338609107.00000000	338609107.00000000	83411845.00000000	24.633668520911931
19	Isuzu D-Max	322180869.00000000	322180869.00000000	-21852323.00000000	-6.782625879626639
20	Isuzu Giga	627900208.00000000	627900208.00000000	402793761.00000000	64.149327531358295
21	Isuzu MU-X	98604348.0000000000	98604348.0000000000	-218525051.000000000000	-221.61806799838076
22	Isuzu MU-X	371531380.00000000	371531380.00000000	81936626.50000000	22.053756670567099
23	Isuzu N-Series	529550474.00000000	529550474.00000000	131040819.00000000	24.745671174680130
24	Isuzu N-Series	375748944.00000000	375748944.00000000	182822900.000000000000	48.655599149202133

Untuk output lebih lengkapnya dapat diakses di link berikut:

[PacCars-Database/Hasil Query No 4 Analytical.csv at main · rzkybagaskara/PacCars-Database · GitHub]

5. Membuat window function rata-rata harga bid sebuah merk dan model mobil selama 6 bulan terakhir

   Contoh: Mobil Mazda CX-9

```sql
--5. Window function untuk rata-rata harga bid sebuah merek dan model mobil selama 6 bulan terakhir
-- Mazda CX-9
SELECT m.merek,
       m.model,
        AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '6' MONTH PRECEDING) as m_min_6,
         AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '5' MONTH PRECEDING) as m_min_5,
         AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '4' MONTH PRECEDING) as m_min_4,
         AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '3' MONTH PRECEDING) as m_min_3,
         AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '2' MONTH PRECEDING) as m_min_2,
         AVG(i.harga) OVER(
                    PARTITION BY m.model
                    ORDER BY i.tanggal_dibuat
                    RANGE INTERVAL '1' MONTH PRECEDING) as m_min_1
FROM mobil m
JOIN iklan i USING(id_mobil)
JOIN penawaran pe ON i.id_penawaran = pe.id_penawaran --harus di specified joinnya make kolom yang mana
WHERE m.model = 'Mazda CX-9'
GROUP BY m.merek, m.model, i.harga, i.tanggal_dibuat
```

	merek character varying (20)	model character varying (50)	m_min_6 numeric	m_min_5 numeric	m_min_4 numeric	m_min_3 numeric	m_min_2 numeric	m_min_1 numeric
1	Mazda	Mazda CX-9	480548015	480548015	480548015	480548015	480548015	480548015
2	Mazda	Mazda CX-9	434441323	434441323	434441323	388334631	388334631	388334631
3	Mazda	Mazda CX-9	408643100.5	408643100.5	428951570	428951570	428951570	428951570

Referensi

https://dev.to/iftach/how-i-sorted-places-by-geographical-distance-in-postgres-3a17, diakses pada tanggal 30 Juni 2024

https://stackoverflow.com/questions/40284929/how-to-get-average-value-in-the-last-6-months, diakses pada tanggal 04 Juli 2024