

# Soal & Template Jawaban

Task 5

Nama : Arya Vito Fauzal

# Petunjuk

**Silahkan merujuk pada Data Source Task 5 yang telah disediakan untuk mengerjakan soal soal di bawah ini**

Pada bagian data analytics, terdiri dari 4 soal dengan use case & tabel yang sama. Bayangkan kamu memiliki database erp yang terdiri dari 3 tabel: penjualan, pelanggan, barang. Tabel tersebut akan dibuat menjadi sebuah datamart yang nantinya digunakan untuk visualisasi.

# Query

## Soal 1\*:

Dari 2 query ini, mana yang bekerja lebih baik? Jelaskan mengapa.

- (a) `SELECT * FROM pelanggan WHERE SUBSTR(alamat, 1, 3) = Mat;`
- (b) `SELECT * FROM pelanggan WHERE alamat LIKE 'Mat%'`

*\*disclaimer: soal ini tidak terkait dengan data source*

## Jawaban : B

Alasan : Karena jawaban A seharusnya `SUBSTR(alamat, 1,3) = "Ala"` apabila ingin mencari alamat dengan awalan ala atau apabila ingin mat maka diganti `SUBSTR(alamat, 3,6)`. Pilihan B tepat karena `'Mat%'` akan mencari seluruh alamat yang berawalan dengan kata Mat dan ditambah apapun kata, simbolik, atau numerik yang ada setelahnya seperti Matoga 5, Mataram, Mato Kopi 3A, dsb

# Query

## Soal 2 \*:

Anggap kita memiliki tabel pelanggan dengan kolom: id, nama, tanggal\_lahir, alamat. Bagaimana cara yang lebih tepat dalam menulis query untuk mendapatkan data pelanggan yang tanggal\_lahir nya ada di antara 2000-01-01 sampai 2008-12-31? Pilihlah salah satu jawaban dan berikan alasannya.

- (a) `SELECT * FROM pelanggan WHERE tanggal_lahir >= '2000-01-01' AND tanggal_lahir <= '2008-12-31'`
- (b) `SELECT * FROM pelanggan WHERE tanggal_lahir BETWEEN '2000-01-01' AND '2008-12-31'`

*\*disclaimer: soal ini tidak terkait dengan data source*

**Jawaban : B**

Alasan : Diantara, berarti 2000-01-01 dan 2008-12-31 tidak termasuk kriteria sehingga opsi A menjadi kurang tepat, pilihan B menjadi tepat karena menggunakan kata between yang menunjukkan rentang waktunya 2000-01-01 dan 2008-12-31 terekslusi

## Soal 3: Menentukan Primary Key

### A. Tugas

Tentukan primary key dari table penjualan. jelaskan alasannya

### B. Jawaban & Penjelasan :

- a. id\_customer
- b. karena pada table 'penjualan' ke 3 field ini terhubung dengan table lain seperti 'pelanggan' dan 'barang' dan mempunyai unique identifier dimana tidak mungkin ada kesamaan terhadap ID yang ada sehingga tidak memungkinkan adanya duplikasi record apabila disambungkan antar table dengan menggunakan inner join, outer join, join lainnya.

## Soal 4: Design Datamart

### A. Tugas

Buatlah design datamart (Terdiri dari tabel base, dan tabel aggregate). Upload file query dalam gdrive mu (pastikan dapat diakses public). Lalu masukkan linknya di tabel di bawah, dan cantumkan juga screenshoot query nya (jika lebih dari 1 file, maka masing masing file di-screenshoot)

Silahkan tambah halaman jika dibutuhkan

### B. Jawaban : Terlampir

No	Nama File	Link
1	Task 5 Rakamin	<a href="https://drive.google.com/drive/folders/18KMK6RgLszzJp7Qk5HAF3kB59wlv mDGJ?usp=drive_link">https://drive.google.com/drive/folders/18KMK6RgLszzJp7Qk5HAF3kB59wlv mDGJ?usp=drive_link</a>

# Table Base "<<Task5.barang>>"

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows the database structure: 'Languages', 'Publications', 'Schemas (1)' (expanded to 'public'), and 'Tables (3)' (expanded to show 'Task5.barang', 'Task5.pelanggan', and 'Task5.penjualan'). The 'Task5.barang' table is selected. The main pane shows the SQL editor with the following query:

```
1  -- Table: public.Task5.barang
2
3  -- DROP TABLE IF EXISTS public."Task5.barang";
4
5  CREATE TABLE IF NOT EXISTS public."Task5.barang"
6  (
7      kode_barang character varying(20) COLLATE pg_catalog."default",
8      sektor character varying(20) COLLATE pg_catalog."default",
9      nama_barang text COLLATE pg_catalog."default",
10     tipe character varying(20) COLLATE pg_catalog."default",
11     nama_tipe character varying(20) COLLATE pg_catalog."default",
12     kode_lini integer,
13     lini character varying(20) COLLATE pg_catalog."default",
14     kemasan character varying(20) COLLATE pg_catalog."default"
15 )
16
17 TABLESPACE pg_default;
18
19 ALTER TABLE IF EXISTS public."Task5.barang"
20     OWNER to postgres;
```

The status bar at the bottom indicates the current location: 'Servers > PostgreSQL 16 > Databases > postgres > Schemas > public > Tables > Task5.barang' and the cursor position 'Ln 1, Col 1'.

## Table Base “<<Task5.barang>>”

column	data type	description	transformation
kode_barang	varchar	kode dari obat	-
sektor	varchar	sektor terhadap obat tersebut	-
nama_barang	text	nama obat	-
tipe	varchar	kode dari tipe obat	-
nama_tipe	varchar	jenis produk obat	-
kode_lini	int	kode lini obat	-
lini	varchar	lini obat-obatan	-



# Table Base "<<Task5.pelanggan>>"

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane shows the database structure: 'Languages', 'Publications', 'Schemas (1)' (expanded to 'public'), and various database objects like 'Aggregates', 'Collations', 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Operators', 'Procedures', 'Sequences', 'Tables (3)' (expanded to 'Task5.barang', 'Task5.pelanggan', 'Task5.penjualan'), 'Trigger Functions', 'Types', 'Views', and 'Subscriptions'. The 'Task5.pelanggan' table is selected and highlighted.

The main pane shows the SQL editor for the selected table. The query is as follows:

```
1  -- Table: public.Task5.pelanggan
2
3  -- DROP TABLE IF EXISTS public."Task5.pelanggan";
4
5  CREATE TABLE IF NOT EXISTS public."Task5.pelanggan"
6  (
7      id_customer character varying(20) COLLATE pg_catalog."default" NOT NULL,
8      level character varying(20) COLLATE pg_catalog."default" NOT NULL,
9      nama character varying(20) COLLATE pg_catalog."default" NOT NULL,
10     id_cabang_sales character varying(20) COLLATE pg_catalog."default" NOT NULL,
11     cabang_sales character varying(20) COLLATE pg_catalog."default" NOT NULL,
12     id_group character varying(20) COLLATE pg_catalog."default" NOT NULL,
13     "group" character varying(20) COLLATE pg_catalog."default" NOT NULL
14 )
15
16 TABLESPACE pg_default;
17
18 ALTER TABLE IF EXISTS public."Task5.pelanggan"
19     OWNER to postgres;
```

The status bar at the bottom indicates the current location: 'Servers > PostgreSQL 16 > Databases > postgres > Schemas > public > Tables > Task5.pelanggan'. The bottom right corner shows 'Ln 1, Col 1'.

## Table Base “<<Task5.pelanggan>>”

column	data type	description	transformation
id_customer	varchar	ID customer pembeli obat	-
level	varchar	perusahaan atau bukan	-
nama	varchar	nama apotek/klinik pembeli obat	-
id_cabang_sales	varchar	ID cabang apotek/klinik	-
cabang_sales	varchar	cabang apotek/klinik	-
id_group	varchar	ID penentu apotek/klinik	-
group	varchar	apotek/klinik yang memesan obat	-

# Table Base "<<Task5.penjualan>>"

The screenshot displays the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, with the 'public' schema expanded and the 'Task5.penjualan' table selected. The main pane shows the SQL editor with the following code:

```
1 -- Table: public.Task5.penjualan
2
3 -- DROP TABLE IF EXISTS public."Task5.penjualan";
4
5 CREATE TABLE IF NOT EXISTS public."Task5.penjualan"
6 (
7     id_distributor character varying(20) COLLATE pg_catalog."default",
8     id_cabang character varying(20) COLLATE pg_catalog."default",
9     id_invoice character varying(20) COLLATE pg_catalog."default",
10    tanggal date,
11    id_customer character varying(20) COLLATE pg_catalog."default",
12    id_barang character varying(20) COLLATE pg_catalog."default",
13    jumlah_barang integer,
14    unit character varying(20) COLLATE pg_catalog."default",
15    harga money,
16    mata_uang character varying(20) COLLATE pg_catalog."default",
17    brand_id character varying(20) COLLATE pg_catalog."default",
18    lini character varying(20) COLLATE pg_catalog."default"
19 )
20
21 TABLESPACE pg_default;
```

The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Ln 1, Col 1'.

## Table Base “<<Task5.pelanggan>>”

column	data type	description	transformation
id_distributor	varchar	ID distributor obat	-
id_cabang	varchar	ID cabang pembeli	-
id_invoice	varchar	ID resi pembelian obat	-
tanggal	date	tanggal pembelian obat	-
id_customer	varchar	ID customer pembeli	-
id_barang	varchar	ID barang yang dibeli	-
jumlah_barang	int	jumlah barang yang dibeli	-

## Table Base “<<Task5.pelanggan>>”

column	data type	description	transformation
unit	varchar	berapa unit yang dibeli (Dus/Kardus)	-
harga	money	Harga obat/unit	-
mata_uang	varchar	mata uang yang digunakan untuk pembelian	-
brand_id	varchar	ID brand yang dibeli	-
lini	varchar	lini dari obat	-

# Table Aggregate “<<Data Analysis Rakamin>>”

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the 'public' schema with 'Tables (3)' selected. The main pane displays a SQL query in the 'Query' tab. The query is an aggregate query that selects product names, customer IDs, and the maximum unit and price for each product. The results are shown in the 'Data Output' tab below the query editor.

```
1 select TPL.nama, TPL.cabang_sales, TB.nama_barang, TPL.id_customer, max(TPJ.jumlah_barang)
2 as jumlah_barang_tertinggi, TPJ.unit, max(TPJ.harga) as harga_tertinggi
3 from "Task5.pelanggan" as TPL
4 inner join "Task5.penjualan" as TPJ on TPL.id_customer = TPJ.id_customer
5 inner join "Task5.barang" as TB on TB.kode_barang = TPJ.id_barang
6 where TPJ.harga > 'Rp 5000' and TPJ.jumlah_barang > 50
7 group by TPL.nama, TPL.cabang_sales, TB.nama_barang, TPL.id_customer, TPJ.unit
8 order by jumlah_barang_tertinggi desc
9
10
11
12
```

	nama_barang text	id_customer character varying (20)	jumlah_barang_tertinggi integer	unit character varying (20)	harga_tertinggi money
1	AMPICILLIN	CUST55452	130	DUS	Rp10.691
2	PARACETAMOL	CUST55549	130	DUS	Rp6.940
3	AMPICILLIN	CUST55468	120	DUS	Rp5.781
4	AMPICILLIN	CUST55662	120	DUS	Rp10.691
5	AMBROXOL HC	CUST55718	100	DUS	Rp5.781
6	KLORPROMAZINA TABLET SALUT SELAPUT 100 MG	CUST55524	100	DUS	Rp5.648

Total rows: 27 of 27    Query complete 00:00:00.090    Ln 2, Col 42

# Table Aggregate “<<Data Analysis Rakamin>>”

column	data type	description	transformation
Nama	varchar	nama dari apotek dan klinik	Nama Apotek
cabang_sales	varchar	cabang sales apotek	Cabang Sales
nama_barang	text	nama dari obat-obatan yang dipesan	Nama Obat
id_customer	varchar	ID dari customer yang memesan obat	Customer ID
jumlah_barang_tertinggi	int	Jumlah pemesanan obat tertinggi dari tiap cabang apotek	Jumlah Obat (Dus)
unit	varchar	satuan Dus (Kardus) tiap obat	Unit
harga_tertinggi	money	harga per 1 dus tiap pemesanan obat di cabang tertentu	Harga Tertinggi

# Soal 5 : Data Visualization

## A. Tugas

buatlah data visualiasasi nya, dan cantumkan linknya di bawah (pastikan bisa diakses publik).  
Lalu cantumkan juga screenshot visualisasinya

Silahkan tambah halaman jika dibutuhkan

## B. Jawaban :

Link visualisasi (ex link Google Data Studio) :

<https://lookerstudio.google.com/reporting/936a258b-8750-4d21-bfb6-0f634642a3d9>



## Jumlah Pembelian Obat Tertinggi Skala Dus Customer Tiap Apotek dan Klinik di Indonesia

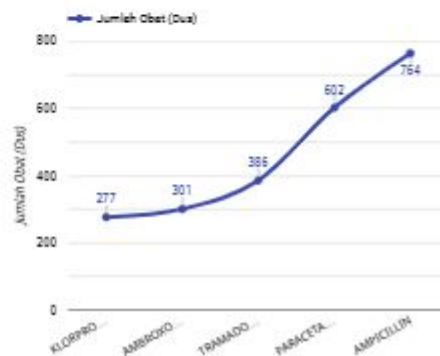
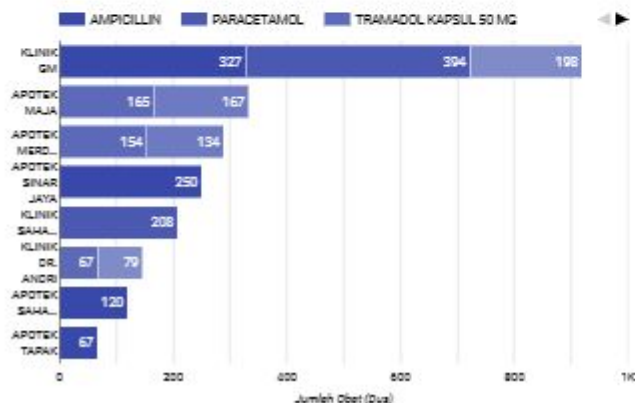
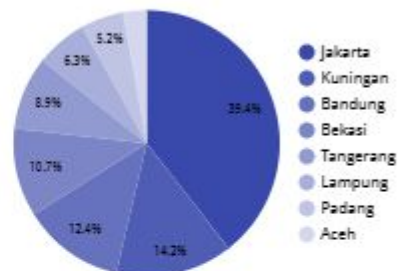
Total Customer  
27

Total Jumlah Obat  
2,330

Kondisi: Harga Obat > Rp 5000 dan Jumlah Pemesanan Obat > 50 Dus

Nama Apotek/Klinik	Cabang Sales	Customer ID	Jumlah Obat (Dus) -
1. APOTEK SINAR JAYA	Bekasi	CUST55452	
2. KLINIK SAHABAT	Tangerang	CUST55549	
3. APOTEK SAHABAT	Padang	CUST55468	
4. APOTEK SINAR JAYA	Bekasi	CUST55662	
5. APOTEK MAJA	Kuningan	CUST55718	

1 - 27 / 27 < >



## Soal 6 : Additional Complementary Data

A. Tugas :

Dari data yang tersedia, menurut kamu untuk melengkapi analisis nya apakah diperlukan data lain juga? jika iya, sebutkan data apa yang kamu maksud dan mengapa memerlukan data tersebut

B. Jawaban : Menurut saya sudah cukup lengkap bagi saya yang pemula dan masih belajar. Untuk memperlengkap analisa, dapat ditambahkan di beberapa wilayah yang tersebar agar saat menambahkan geo chart tidak terkesan sepi dan untuk di tabel penjualan dapat diberikan keuntungan agar dapat dilihat berapa keuntungan yang didapatkan pada apotek X di cabang X. Dapat ditambahkan juga waktu (seperti tahun atau bulan atau batch) agar dapat memasukkan chart berupa tren yang dapat dianalisa dari berbagai aspek nantinya