

# Soal & Template Jawaban

Task 5

Nama : Sonia Epifany Sandah

# 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 : Dari segi optimasi lebih bagus pada option (b) dimana waktu yang dibutuhkan untuk menjalankan query tersebut lebih cepat daripada option (a). Pada option (a) terdapat fungsi pada kolomnya, sehingga akan membatalkan tujuan menciptakan indeks pada kolom tersebut.

# 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 : karena filtering pada option (b) lebih cepat, yaitu hanya memfilter satu blok data (BETWEEN). Sedangkan pada option (a), memfilter dua kali (lebih dari sama dengan, kurang dari sama dengan)

## Soal 3: Menentukan Primary Key

### A. Tugas

Tentukan primary key dari table penjualan. jelaskan alasannya

### B. Jawaban & Penjelasan :

ID Invoice. Karena setiap barisnya unik, tidak duplikat, dan berfungsi sebagai identitas untuk membedakan setiap value pada table.

# 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 : .....

No	Nama File	Link
1	Source table	<a href="https://drive.google.com/drive/folders/1lixgJe25Q3f5HH61OTfRevqIPd8BLb5L?usp=sharing">https://drive.google.com/drive/folders/1lixgJe25Q3f5HH61OTfRevqIPd8BLb5L?usp=sharing</a>
2	Base table	<a href="https://drive.google.com/drive/folders/1-AUg2PKsfpl4kV6pkLYuUzzm6XtzfcTb?usp=sharing">https://drive.google.com/drive/folders/1-AUg2PKsfpl4kV6pkLYuUzzm6XtzfcTb?usp=sharing</a>
3	Aggregate table	<a href="https://drive.google.com/drive/folders/1K6xesG8-HHiDGSDSQPXQAwZlsIJZ6fx?usp=sharing">https://drive.google.com/drive/folders/1K6xesG8-HHiDGSDSQPXQAwZlsIJZ6fx?usp=sharing</a>

# Table Base "<<base\_table>>"

```
1  --- Create Base Table
2  CREATE TABLE base_table AS
3  (
4      SELECT
5          pj.tanggal,
6          pj.id_invoice,
7          pj.id_distributor,
8          pl.group,
9          pl.nama AS nama_customer,
10         br.lini,
11         br.nama_barang,
12         br.kemasan,
13         pj.jumlah_barang,
14         pj.harga,
15         (pj.jumlah_barang * pj.harga) AS revenue,
16         pl.cabang_sales
17     FROM penjualan AS pj
18     LEFT JOIN pelanggan AS pl ON pj.id_customer = pl.id_customer
19     LEFT JOIN barang AS br ON pj.id_barang = br.kode_barang
20     ORDER BY 2
21 );
22
23 --- Determine Primary Key
24 ALTER TABLE base_table ADD PRIMARY KEY (id_invoice);
25
```

# Table Base “<<base\_table>>”

column	data type	description	transformation
tanggal	date	Tanggal penjualan	
id_invoice	character varying	Invoice penjualan (primary key)	
id_distributor	character varying	Nama distributor	
group	character varying	Kategori customer	
nama_customer	character varying	Nama pelanggan	Nama AS nama_customer
lini	character varying	Lini produk	
nama_barang	character varying	Nama produk	
kemasan	character varying	Kemasan produk	
jumlah_barang	integer	Jumlah produk yang dibeli	
harga	numeric	Harga produk	
revenue	numeric	Total revenue	(Jumlah barang * harga) AS revenue
cabang_sales	character varying	Kota cabang sales	



## Table Aggregate "<<barang\_revenue>>"

```
1  --- Create Table Aggregate
2  --- Table Barang Revenue
3  CREATE TABLE IF NOT EXISTS barang_revenue AS
4  (
5      SELECT
6          nama_barang,
7          COUNT(id_invoice) AS jumlah_transaksi,
8          SUM(jumlah_barang) AS jumlah_produk_terjual,
9          SUM(revenue) AS total_revenue
10     FROM base_table
11     GROUP BY 1
12     ORDER BY 4 DESC);
13
14 ALTER TABLE barang_revenue ALTER COLUMN jumlah_transaksi type int,
15 ALTER TABLE barang_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<<barang\_revenue>>”

column	data type	description	transformation
nama_barang	character varying	Nama produk	Group by
jumlah_transaksi	Integer	Jumlah transaksi per produk	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<barang\_monthly\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Barang Revenue per bulan
3  CREATE TABLE IF NOT EXISTS barang_monthly_revenue AS
4  (
5      SELECT
6          CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         nama_barang,
14         COUNT(id_invoice) AS jumlah_transaksi,
15         SUM(jumlah_barang) AS jumlah_produk_terjual,
16         SUM(revenue) AS total_revenue
17     FROM base_table
18     GROUP BY 1,2
19     ORDER BY 1,5 DESC);
20
21 ALTER TABLE barang_monthly_revenue ALTER COLUMN bulan type character varying,
22 ALTER TABLE barang_monthly_revenue ALTER COLUMN jumlah_transaksi type int,
23 ALTER TABLE barang_monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<<barang\_monthly\_revenue>>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Extract dari tanggal, group by
nama_barang	character varying	Nama produk	Group by
jumlah_transaksi	integer	Jumlah transaksi per produk	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

## Table Aggregate "<<cabang\_sales\_revenue>>"

```
1  --- Create Table Aggregate
2  --- Table Cabang Sales Revenue
3  CREATE TABLE IF NOT EXISTS cabang_sales_revenue AS
4  (
5      SELECT
6          cabang_sales,
7          COUNT(id_invoice) AS jumlah_transaksi,
8          SUM(jumlah_barang) AS jumlah_produk_terjual,
9          SUM(revenue) AS total_revenue
10     FROM base_table
11     GROUP BY 1
12     ORDER BY 4 DESC);
13
14 ALTER TABLE cabang_sales_revenue ALTER COLUMN jumlah_transaksi type int,
15 ALTER TABLE cabang_sales_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

## Table Aggregate “<< cabang\_sales\_revenue >>”

column	data type	description	transformation
cabang_sales	character varying	Kota cabang sales	Group by
jumlah_transaksi	integer	Jumlah transaksi per cabang sales	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<cabang\_sales\_monthly\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Cabang Sales Revenue per bulan
3  CREATE TABLE IF NOT EXISTS cabang_sales_monthly_revenue AS
4  (
5      SELECT
6          CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         cabang_sales,
14         COUNT(id_invoice) AS jumlah_transaksi,
15         SUM(jumlah_barang) AS jumlah_produk_terjual,
16         SUM(revenue) AS total_revenue
17     FROM base_table
18     GROUP BY 1,2
19     ORDER BY 1,5 DESC);
20
21 ALTER TABLE cabang_sales_monthly_revenue ALTER COLUMN bulan type character varying,
22 ALTER TABLE cabang_sales_monthly_revenue ALTER COLUMN jumlah_transaksi type int,
23 ALTER TABLE cabang_sales_monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< cabang\_sales\_monthly\_revenue >>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Extract dari tanggal, group by
cabang_sales	character varying	Kota cabang sales	Group by
jumlah_transaksi	integer	Jumlah transaksi per cabang sales	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue



## Table Aggregate "<<customer\_revenue>>"

```
1  --- Create Table Aggregate
2  --- Table Customer Revenue
3  CREATE TABLE IF NOT EXISTS customer_revenue AS
4  (
5      SELECT
6          nama_customer,
7          COUNT(id_invoice) AS jumlah_transaksi,
8          SUM(jumlah_barang) AS jumlah_produk_terjual,
9          SUM(revenue) AS total_revenue
10     FROM base_table
11     GROUP BY 1
12     ORDER BY 4 DESC);
13
14 ALTER TABLE customer_revenue ALTER COLUMN jumlah_transaksi type int,
15 ALTER TABLE customer_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< customer\_revenue >>”

column	data type	description	transformation
nama_customer	character varying	Nama customer	Group by
jumlah_transaksi	integer	Jumlah transaksi per customer	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<customer\_monthly\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Customer Revenue per bulan
3  CREATE TABLE IF NOT EXISTS customer_monthly_revenue AS
4  (
5      SELECT
6          CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         nama_customer,
14         COUNT(id_invoice) AS jumlah_transaksi,
15         SUM(jumlah_barang) AS jumlah_produk_terjual,
16         SUM(revenue) AS total_revenue
17     FROM base_table
18     GROUP BY 1,2
19     ORDER BY 1,5 DESC);
20
21 ALTER TABLE customer_monthly_revenue ALTER COLUMN bulan type character varying,
22 ALTER TABLE customer_monthly_revenue ALTER COLUMN jumlah_transaksi type int,
23 ALTER TABLE customer_monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< customer\_monthly\_revenue >>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Extract dari tanggal, group by
nama_customer	character varying	Nama customer	Group by
jumlah_transaksi	integer	Jumlah transaksi per customer	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

## Table Aggregate “<<distributor\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Distributor Revenue
3  CREATE TABLE IF NOT EXISTS distributor_revenue AS
4  (
5      SELECT
6          id_distributor,
7          COUNT(id_invoice) AS jumlah_transaksi,
8          SUM(jumlah_barang) AS jumlah_produk_terjual,
9          SUM(revenue) AS total_revenue
10     FROM base_table
11     GROUP BY 1
12     ORDER BY 1 DESC);
13
14 ALTER TABLE distributor_revenue ALTER COLUMN jumlah_transaksi type int,
15 ALTER TABLE distributor_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

## Table Aggregate “<< distributor\_revenue >>”

column	data type	description	transformation
id_distributor	character varying	Id nama distributor	Group by
jumlah_transaksi	integer	Jumlah transaksi per distributor	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<distributor\_monthly\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Distributor Revenue per bulan
3  CREATE TABLE IF NOT EXISTS distributor_monthly_revenue AS
4  (
5      SELECT
6          CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         id_distributor,
14         COUNT(id_invoice) AS jumlah_transaksi,
15         SUM(jumlah_barang) AS jumlah_produk_terjual,
16         SUM(revenue) AS total_revenue
17     FROM base_table
18     GROUP BY 1,2
19     ORDER BY 1,5 DESC);
20
21 ALTER TABLE distributor_monthly_revenue ALTER COLUMN bulan type character varying,
22 ALTER TABLE distributor_monthly_revenue ALTER COLUMN jumlah_transaksi type int,
23 ALTER TABLE distributor_monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< distributor\_monthly\_revenue >>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Extract dari tanggal, group by
id_distributor	character varying	Id nama distributor	Group by
jumlah_transaksi	integer	Jumlah transaksi per customer	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue



## Table Aggregate “<<lini\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Lini Revenue
3  CREATE TABLE IF NOT EXISTS lini_revenue AS
4  (
5      SELECT
6          lini,
7          COUNT(id_invoice) AS jumlah_transaksi,
8          SUM(jumlah_barang) AS jumlah_produk_terjual,
9          SUM(revenue) AS total_revenue
10     FROM base_table
11     GROUP BY 1
12     ORDER BY 4 DESC);
13
14 ALTER TABLE lini_revenue ALTER COLUMN jumlah_transaksi type int,
15 ALTER TABLE lini_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< lini\_revenue >>”

column	data type	description	transformation
lini	character varying	Lini produk	Group by
jumlah_transaksi	integer	Jumlah transaksi per lini	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<lini\_monthly\_revenue>>”

```
1  --- Create Table Aggregate
2  --- Table Lini Revenue per bulan
3  CREATE TABLE IF NOT EXISTS lini_monthly_revenue AS
4  (
5      SELECT
6          CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         lini,
14         COUNT(id_invoice) AS jumlah_transaksi,
15         SUM(jumlah_barang) AS jumlah_produk_terjual,
16         SUM(revenue) AS total_revenue
17     FROM base_table
18     GROUP BY 1,2
19     ORDER BY 1,5 DESC);
20
21 ALTER TABLE lini_monthly_revenue ALTER COLUMN bulan type character varying,
22 ALTER TABLE lini_monthly_revenue ALTER COLUMN jumlah_transaksi type int,
23 ALTER TABLE lini_monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< distributor\_monthly\_revenue >>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Extract dari tanggal, group by
lini	character varying	Lini produk	Group by
jumlah_transaksi	integer	Jumlah transaksi per customer	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# Table Aggregate “<<monthly\_revenue>>”

```
1 --- Create Table Aggregate
2 --- Table Revenue per bulan
3 CREATE TABLE IF NOT EXISTS monthly_revenue AS
4 (
5     SELECT
6         CASE WHEN (date_part('month', tanggal)= 1) THEN 'Januari'
7              WHEN (date_part('month', tanggal)= 2) THEN 'Februari'
8              WHEN (date_part('month', tanggal)= 3) THEN 'Maret'
9              WHEN (date_part('month', tanggal)= 4) THEN 'April'
10             WHEN (date_part('month', tanggal)= 5) THEN 'Mei'
11             WHEN (date_part('month', tanggal)= 6) THEN 'Juni'
12         END AS bulan,
13         COUNT(id_invoice) AS jumlah_transaksi,
14         SUM(jumlah_barang) AS jumlah_produk_terjual,
15         SUM(revenue) AS total_revenue
16     FROM base_table
17     GROUP BY 1
18     ORDER BY 4 DESC);
19
20 ALTER TABLE monthly_revenue ALTER COLUMN bulan type character varying,
21 ALTER TABLE monthly_revenue ALTER COLUMN jumlah_transaksi type int,
22 ALTER TABLE monthly_revenue ALTER COLUMN jumlah_produk_terjual type int;
```

# Table Aggregate “<< monthly\_revenue >>”

column	data type	description	transformation
bulan	character varying	Bulan penjualan	Group by
jumlah_transaksi	integer	Jumlah transaksi per lini	COUNT(id_invoice) AS jumlah_transaksi
jumlah_produk_terjual	integer	Jumlah produk terjual	SUM(jumlah_barang) AS jumlah_produk_terjual
total_revenue	numeric	Total penjualan	SUM(revenue) AS total_revenue

# 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

## A. Jawaban :

Link visualisasi :

[https://public.tableau.com/app/profile/sonia.epifany.sandah/viz/trial\\_16945811128870/Dashboard2](https://public.tableau.com/app/profile/sonia.epifany.sandah/viz/trial_16945811128870/Dashboard2)

## FILTER PANEL

Month

All

Lini

All

Group

All

City

All

Distributor

All

## Total Transaction

350

## Total Items Sold

9,176

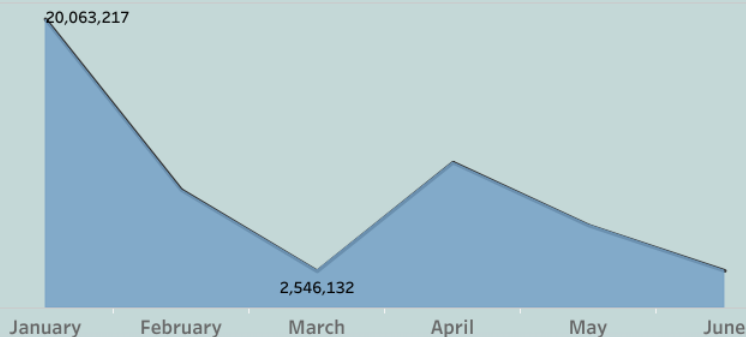
## Total Revenue

Rp49.2M

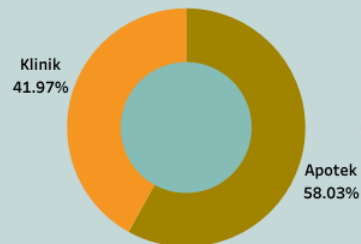
## Total Customer

8

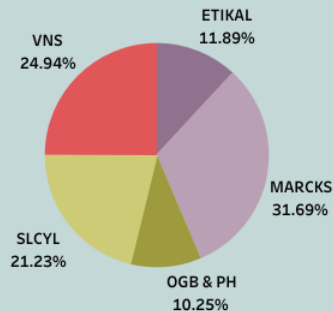
## Revenue by Month



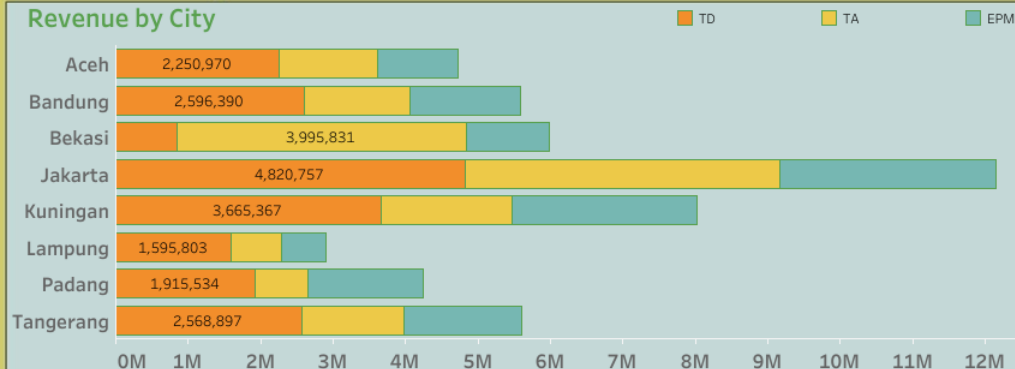
## Revenue Percentage by Group



## Revenue by Lini



## Revenue by City



## Customer Wise Sales Trend

Customer Name	Total Items	Revenue	% of Total Revenue
KLINIK GM	1,751	12,151,455	24.69%
APOTEK MAJA	1,574	8,018,901	16.29%
APOTEK SINAR JAYA	1,062	5,985,146	12.16%
KLINIK SAHABAT	1,127	5,604,114	11.39%
APOTEK MERDEKA	1,267	5,585,452	11.35%
APOTEK TAPAK	1,151	4,730,392	9.61%
APOTEK SAHABAT	783	4,239,980	8.61%
KLINIK DR. ANDRI	461	2,901,026	5.89%



## 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

A. Jawaban :

Iya, Data yang dibutuhkan adalah penggolongan obat berupa kategori atau golongan. Hal ini untuk meningkatkan keamanan dan ketepatan penggunaan serta pengamanan distribusi obat.

Contoh: Obat Bebas Terbatas, Obat Bebas, Obat Keras, dll. Hal ini diperlukan untuk mengetahui distribusi penjualan golongan barang/obat di setiap pelanggan.