TASK 5

Data Mart and Data Visualization
Big Data Analytics

SALSABILA RANI INDIRAWATI

Questions

1

(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*

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'

3

A. Tugas
Tentukan primary key dari table penjualan. jelaskan alasannya

B. Jawaban & Penjelasan : .....

Questions

4	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)
5	buatlah data visualiasasi nya, dan cantumkan linknya di bawah (pastikan bisa diakses publik). Lalu cantumkan juga screenshot visualisasinya
6	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

Tools

Data Mart
Query

Data
Visualization

Answers 1

Query A

```
1 •   SELECT * FROM sales_kimfar.pelanggan
2     WHERE SUBSTR(cabang_sales, 1, 3) = 'Lam';
3
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_customer	level	nama	id_cabang_sales	cabang_sales	id_group	group
CUST55394	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55409	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55424	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55439	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55454	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55469	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik

Query B

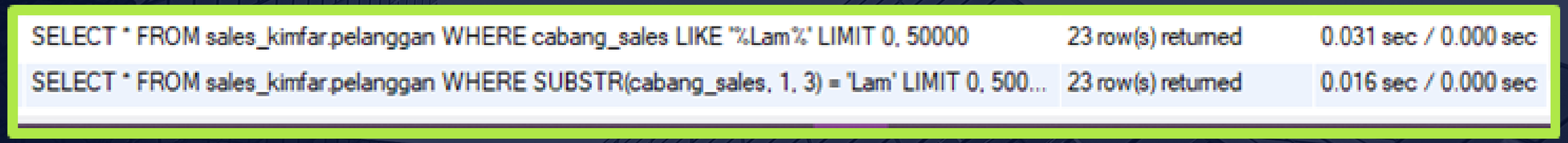
```
1 •   SELECT * FROM sales_kimfar.pelanggan;
2 •   select*from sales_kimfar.pelanggan where cabang_sales like 'Lam%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_customer	level	nama	id_cabang_sales	cabang_sales	id_group	group
CUST55394	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55409	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55424	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55439	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55454	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55469	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55484	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik
CUST55499	Company	KLINIK DR. ANDRI	CAB08	Lampung	Z31	Klinik

Menurut saya,

Query A dan **Query B** sama-sama bekerja dengan baik dalam mencapai tujuan yang diinginkan, yaitu menampilkan *features* yang berkaitan dengan cabang sales di Lampung; mengembalikan *features*, jumlah *rows* dan *values* yang sama. Namun, **Query B** lebih nyaman digunakan tanpa harus mendefinisikan secara detail



SELECT * FROM sales_kimfar.pelanggan WHERE cabang_sales LIKE '%Lam%' LIMIT 0, 50000	23 row(s) returned	0.031 sec / 0.000 sec
SELECT * FROM sales_kimfar.pelanggan WHERE SUBSTR(cabang_sales, 1, 3) = 'Lam' LIMIT 0, 500...	23 row(s) returned	0.016 sec / 0.000 sec

Jika dihitung dari proses *running* **Query A** mengembalikan tabel dengan proses *running* 50% **lebih cepat** daripada Query B, namun proses *running* bersifat fluktuatif sehingga tidak dapat dijadikan acuan.

Answers 2

Query A

```
1 •   SELECT * FROM penjualan
2     WHERE tgl >= '2022-01-21' and tgl <= '2022-01-28';
3
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_distributor	id_cabang	id_invoice	id_customer	id_barang	jumlah_barang	unit	harga	mata_uang	brand_id	lini	tgl
TA	CAB02	IN6155	CUST55382	BRG0003	9	DUS	10690.6	IDR	BRND003	MARCKS	2022-01-21
EPM	CAB03	IN6144	CUST55383	BRG0004	13	DUS	8700.7	IDR	BRND004	VNS	2022-01-22
TD	CAB04	IN6280	CUST55384	BRG0005	1	DUS	5648.3	IDR	BRND005	SLCYL	2022-01-23
TA	CAB05	IN6052	CUST55385	BRG0006	5	DUS	2819.2	IDR	BRND006	OGB & PH	2022-01-23
EPM	CAB06	IN6089	CUST55386	BRG0007	9	DUS	4592.1	IDR	BRND007	ETIKAL	2022-01-25
TD	CAB07	IN6251	CUST55387	BRG0008	1	DUS	3991.9	IDR	BRND008	MARCKS	2022-01-23

Query B

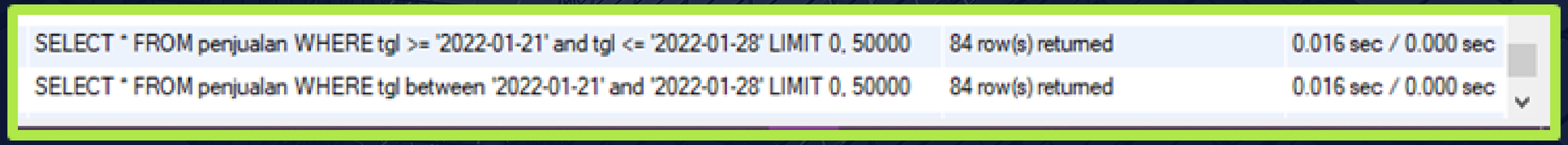
```
5 •   SELECT * FROM penjualan
6     WHERE tgl between '2022-01-21' and '2022-01-28';
7
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_distributor	id_cabang	id_invoice	id_customer	id_barang	jumlah_barang	unit	harga	mata_uang	brand_id	lini	tgl
TA	CAB02	IN6155	CUST55382	BRG0003	9	DUS	10690.6	IDR	BRND003	MARCKS	2022-01-21
EPM	CAB03	IN6144	CUST55383	BRG0004	13	DUS	8700.7	IDR	BRND004	VNS	2022-01-22
TD	CAB04	IN6280	CUST55384	BRG0005	1	DUS	5648.3	IDR	BRND005	SLCYL	2022-01-23
TA	CAB05	IN6052	CUST55385	BRG0006	5	DUS	2819.2	IDR	BRND006	OGB & PH	2022-01-23
EPM	CAB06	IN6089	CUST55386	BRG0007	9	DUS	4592.1	IDR	BRND007	ETIKAL	2022-01-25
TD	CAB07	IN6251	CUST55387	BRG0008	1	DUS	3991.9	IDR	BRND008	MARCKS	2022-01-23

Menurut saya,

Query A dan **Query B** sama-sama bekerja dengan tepat dalam mencapai tujuan yang diinginkan, yaitu menampilkan *features* yang berkaitan dengan tanggal transaksi penjualan yang berlangsung diantara tanggal 21 - 28 Januari 2022; mengembalikan *features*, jumlah *rows* dan *values* yang sama. Di sisi lain, perintah query ***between*** merepresentasikan *value* yang berada diantara lebih besar dari sama dengan *value1* dan lebih kecil sama dengan *value2*. Sehingga **Query B** mengandung arti yang **sama** dengan Query A. Dapat dikatakan **Query B** lebih nyaman digunakan karena tidak harus mendefinisikan secara detail.



SELECT * FROM penjualan WHERE tgl >= '2022-01-21' and tgl <= '2022-01-28' LIMIT 0, 50000	84 row(s) returned	0.016 sec / 0.000 sec
SELECT * FROM penjualan WHERE tgl between '2022-01-21' and '2022-01-28' LIMIT 0, 50000	84 row(s) returned	0.016 sec / 0.000 sec

Answers 3

Primary Key

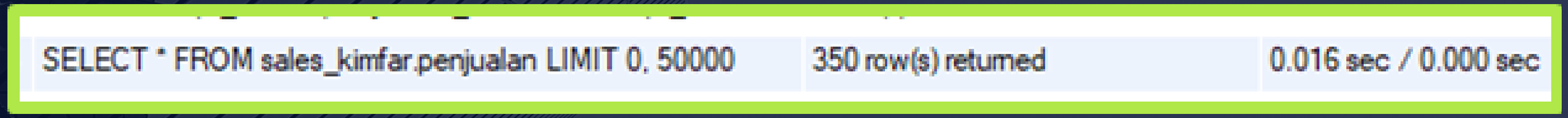
```
5 •    SELECT * FROM sales_kimfar.penjualan;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_distributor	id_cabang	id_invoice	id_customer	id_barang	jumlah_barang	unit	harga	mata_uang	brand_id	lini	tgl
TD	CAB01	IN5997	CUST55380	BRG0001	1	DUS	1169.91	IDR	BRND001	OGB & PH	2022-01-20
TD	CAB01	IN6297	CUST55381	BRG0002	5	DUS	2337.5	IDR	BRND002	ETIKAL	2022-01-20
TA	CAB02	IN6155	CUST55382	BRG0003	9	DUS	10690.6	IDR	BRND003	MARCKS	2022-01-21
EPM	CAB03	IN6144	CUST55383	BRG0004	13	DUS	8700.7	IDR	BRND004	VNS	2022-01-22
TD	CAB04	IN6280	CUST55384	BRG0005	1	DUS	5648.3	IDR	BRND005	SLCYL	2022-01-23

Primary Key adalah kolom atau grup kolom yang mengidentifikasi secara unik dan tidak dapat menerima duplikat. Sehingga primary key pada tabel Penjualan adalah ***id_invoice*** dan ***id_customer***.

Bukti



SELECT * FROM sales_kimfar.penjualan LIMIT 0, 50000	350 row(s) returned	0.016 sec / 0.000 sec

```
6 •   select count(distinct id_distributor) as jumlah_dist,
7     count(distinct id_cabang) as jumlah_cabang,
8     count(distinct id_invoice) as jumlah_invoice,
9     count(distinct id_customer) as jumlah_customer,
10    count(distinct id_barang) as jumlah_barang
11    from penjualan;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

jumlah_dist	jumlah_cabang	jumlah_invoice	jumlah_customer	jumlah_barang
3	10	350	350	10

Tabel penjualan memiliki **350 *rows*** yang berarti seharusnya memiliki **350 nilai unik** pada suatu kolom. Nilai unik ini dimiliki oleh *id_invoice* dan *id_customer*, sehingga berperan sebagai *primary key*. Sedangkan id yang lain berperan sebagai foreign key karena mengandung duplikat *value*.

Answers 4

Accessible Query Links

No.	File Name	Link
1.	Aggregration Table Query - Salsabila Rani	*https://s.id/Aggregation-Table-SalsabilaRani*
2.	Table Base Query - Salsabila Rani	*https://s.id/Table-Base-Query-SalsabilaRani*

Table Base: Business Case #1

```sql
-- business case #1:
-- melihat total pendapatan dari penjualan masing-masing produk
select penj.id_barang,
brg.nama_barang,
penj.harga
from penjualan penj
left join barang brg on penj.id_barang = brg.kode_barang;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_barang	nama_barang	harga
BRG0001	ACYCLOVIR DUS	1169.91
BRG0002	ALERGINE TABLET SALUT	2337.5
BRG0003	AMPICILLIN	10690.6
BRG0004	TRAMADOL KAPSUL 50 MG	8700.7
BRG0005	KLORPROMAZINA TABLET SALUT SELAPUT 100 ...	5648.3
BRG0006	KETOCONAZOLE TABLET 200 MG	2819.2

Table Base: Business Case #2

```sql
-- business case #2:
-- melihat total pendapatan dari penjualan masing-masing lini berdasarkan pelanggan
select penj.id_cabang,
pel.nama,
penj.lini,
penj.harga
from penjualan penj
left join pelanggan pel
on penj.id_cabang = pel.id_cabang_sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_cabang	nama	lini	harga
CAB01	APOTEK TAPAK	OGB & PH	1169.91
CAB01	APOTEK TAPAK	OGB & PH	1169.91
CAB01	APOTEK TAPAK	OGB & PH	1169.91
CAB01	APOTEK TAPAK	OGB & PH	1169.91
CAB01	APOTEK TAPAK	OGB & PH	1169.91
CAB01	APOTEK TAPAK	OGB & PH	1169.91

Table Base: Business Case #3

```sql
-- business case #3:
-- melihat segmentasi customer setiap daerah cabang
select penj.id_barang,
brg.nama_barang,
pel.id_cabang_sales,
pel.cabang_sales,
penj.jumlah_barang,
penj.harga
from penjualan penj
left join pelanggan pel on penj.id_cabang = pel.id_cabang_sales
left join barang brg on penj.id_barang = brg.kode_barang;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_barang	nama_barang	id_cabang_sales	cabang_sales	jumlah_barang	harga
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91
BRG0001	ACYCLOVIR DUS	CAB01	Aceh	1	1169.91

Table Base: Business Case #4

```sql
-- business case #4
-- melihat informasi perkembangan sales setiap produk setiap bulan
select (str_to_date(tanggal,'%m/%d/%Y')) as tgl,
penj.id_barang,
brg.nama_barang,
penj.jumlah_barang,
penj.harga
from penjualan penj
left join barang brg on penj.id_barang = brg.kode_barang
order by tgl;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

tgl	id_barang	nama_barang	jumlah_barang	harga
2022-01-20	BRG0001	ACYCLOVIR DUS	1	1169.91
2022-01-20	BRG0002	ALERGINE TABLET SALUT	5	2337.5
2022-01-21	BRG0003	AMPICILLIN	9	10690.6
2022-01-22	BRG0004	TRAMADOL KAPSUL 50 MG	13	8700.7
2022-01-23	BRG0005	KLORPROMAZINA TABLET SALUT SELAPUT 100 ...	1	5648.3
2022-01-23	BRG0006	KETOCONAZOLE TABLET 200 MG	5	2819.2

Keterangan

column	data type	description
id_barang	varchar	kode produk yang dijūal
nama_barang	varchar	nama produk yang dijual
harga	float	harga setiap produk
id_cabang	varchar	kode cabang apotek atau klinik pada tabel penjualan

column	data type	description
nama	varchar	nama pelanggan berupa nama apotek ḏan klinik
lini	varchar	kategori kelompok produk
id_cabang _sales	float	kode cabang apotek atau klinik pada tabel pelanggan
cabang _sales	varchar	area cabang apotek atau klinik

column	data type	description
jumlah _barang	varchar	jumlah barang yang terjual setiap transaksi
tgl	date	tanggal berlangsungnya transaksi

Aggregation Table:
Data Mart #1

```sql
-- data mart #1
-- diambil dari tabel business case #1
-- melihat total pendapatan dari penjualan masing-masing produk
select distinct(nama_barang) as nama_produk,
round(sum(harga), 2) as total_penjualan
from `business case #1`
group by nama_produk;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nama_produk	total_penjualan
ACYCLOVIR DUS	91459.95
ALERGINE TABLET SALUT	125723.4
AMPICILLIN	437758.7
TRAMADOL KAPSUL 50 MG	380867.8
KLORPROMAZINA TABLET SALUT SELAPUT 100 ...	197690.5
KETOCONAZOLE TABLET 200 MG	98672
ERGOTAMINE COFFEINE	110210.4
TETRACYCLINE KAPSUL 250 MG	95805.6
AMBROXOL HC	138736.8
PARACETAMOL	166567.2

Aggregation Table:
Data Mart #2

```sql
-- data mart #2
-- diambil dari tabel business case #2
-- melihat total pendapatan dari penjualan masing-masing lini berdasarkan pelanggan
select distinct(lini),
nama as nama_pelanggan,
round(sum(harga), 2) as pendapatan_pelanggan
from `business case #2`
where nama is not null
group by 1,2
order by 1 asc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

lini	nama_pelanggan	pendapatan_pelanggan
ETIKAL	APOTEK SINAR JAYA	211236.6
ETIKAL	APOTEK TAPAK	3802328.2
ETIKAL	KLINIK SAHABAT	3798431.9
MARCKS	APOTEK MAJA	502458.2
MARCKS	APOTEK SAHABAT	183627.4
MARCKS	APOTEK SINAR JAYA	7612972.4

Aggregation Table: Data Mart #3

```sql
-- data mart #3
-- diambil dari tabel business case #3
-- melihat segmentasi customer setiap daerah cabang
select distinct(cabang_sales),
nama_barang as nama_produk,
sum(jumlah_barang) as produk_terjual,
round(sum(harga), 2) as pendapatan_produk
from `business case #3`
where cabang_sales is not null
group by 1,2
order by 1,2 desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

cabang_sales	nama_produk	produk_terjual	pendapatan_produk
Aceh	TRAMADOL KAPSUL 50 MG	14711	4498261.9
Aceh	TETRACYCLINE KAPSUL 250 MG	10904	2063812.3
Aceh	AMPICILLIN	29140	11556538.6
Aceh	AMBROXOL HC	16074	2988621.9
Aceh	ALERGINE TABLET SALUT	21573	3492024.8
Aceh	ACYCLOVIR DUS	15604	2429101.47

Aggregation Table: Data Mart #4

```sql
-- data mart #4
-- diambil dari tabel business case #4
-- melihat informasi perkembangan sales setiap produk setiap bulan
select tanggal,
nama_barang as nama_produk,
sum(jumlah_barang) as produk_terjual
from `business case #4`
group by 1,2
order by 1,2,3 desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

tanggal	nama_produk	produk_terjual
1/20/2022	ACYCLOVIR DUS	1
1/20/2022	ALERGINE TABLET SALUT	5
1/21/2022	AMPICILLIN	9
1/22/2022	TRAMADOL KAPSUL 50 MG	13
1/23/2022	ACYCLOVIR DUS	227
1/23/2022	AMPICILLIN	289

Aggregation Table: Data Mart #5

```sql
-- data mart #5
-- diambil dari tabel business case #4
-- melihat revenue growth setiap bulan
select bulan,
(round(sum(harga),2)) as pendapatan_bulan,
(sum(harga)) - lag(sum(harga)) over (order by bulan asc) as revenue_growth,
(sum(harga) - LAG (sum(harga)) OVER (ORDER BY bulan ASC))/LAG (sum(harga)) OVER (ORDER BY bulan ASC)*100
AS revenue_percentage_growth
from `business case #4`
group by bulan;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

bulan	pendapatan_bulan	revenue_growth	revenue_percentage_growth
1	732989.37	NULL	NULL
2	268933.53	-464055.8399999995	-63.31003681540427
3	160351.13	-108582.4	-40.375181183246276
4	298581.85	138230.72000000012	86.20501770084195
5	251014.95	-47566.90000000014	-15.930941549193333
6	131621.52	-119393.43	-47.56427057432236

Keterangan

column	data type	description	transformation
nama_produk	varchar	nama produk yang dijual	as: nama_barang diubah menjadi nama_produk
total_ penjualan	float	total keseluruhan penjualan per produk	sum(): menjumlahkan total penjualan per produk
lini	varchar	kategori produk yang dijual. 1 lini terdapat beberapa produk	group by: mengelompokkan value berdasarkan kelompok lini

column	data type	description	transformation
nama_ pelanggan	varchar	nama pelanggan kimia farma, berupa klinik dan apotek	as: nama diubah menjadi nama_pelanggan
pendapatan_ pelanggan	float	pendapatan pelanggan setiap lini	sum(): menjumlahkan total penjualan per lini
cabang_sales	varchar	cabang apotek atau klinik	-
produk_ terjual	int	jumlah produk terjual	sum(): menjumlahkan produk terjual per produk

column	data type	description	transformation
tanggal	varchar	tanggal transaksi penjualan	memanggil tanggal dengan select str_to_date()
bulan	int	bulan transaksi penjualan	select month pada tanggal
pendapatan_bulan	float	total pendapatan setiap bulan	sum(): menjumlahkan total penjualan per bulan

column	data type	description	transformation
revenue_growth	float	nominal pertumbuhan pendapatan	mengurangi row saat ini dengan row sebelum menggunakan lag() dan over()
revenue_percentage_growth	float	presentase pertumbuhan pendapatan	menerapkan lag() dan over() pada rumus presentase

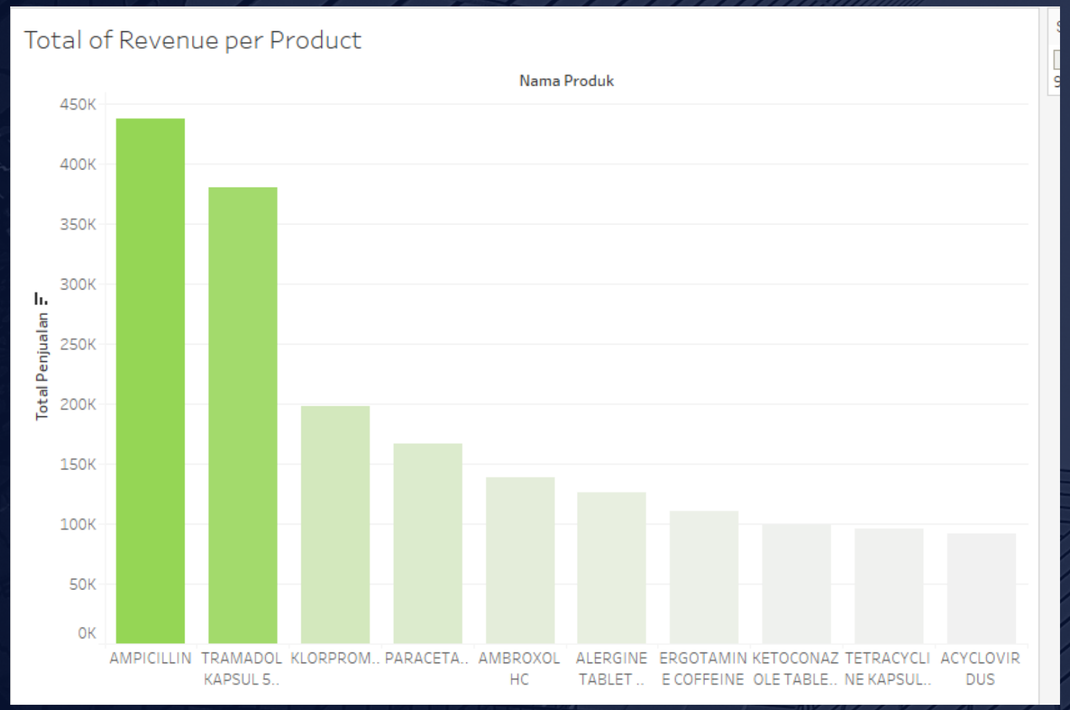
Answers 5

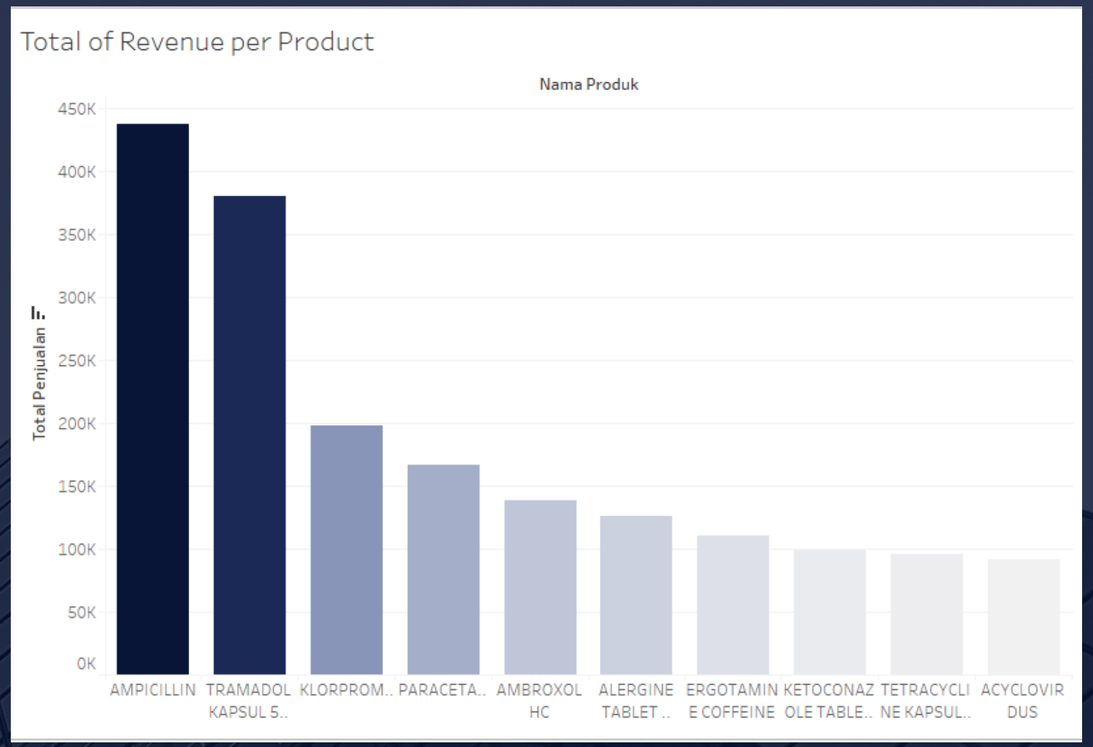
Accessible Viz Links

No.	Title	Link
1.	Total of Revenue per Product	*https://s.id/*Visualization-1
2.	Revenue of Each Customer per Lini	*https://s.id/*Visualization-2
3.	Total of Product Sold per Sales Area	*https://s.id/*Visualization-3

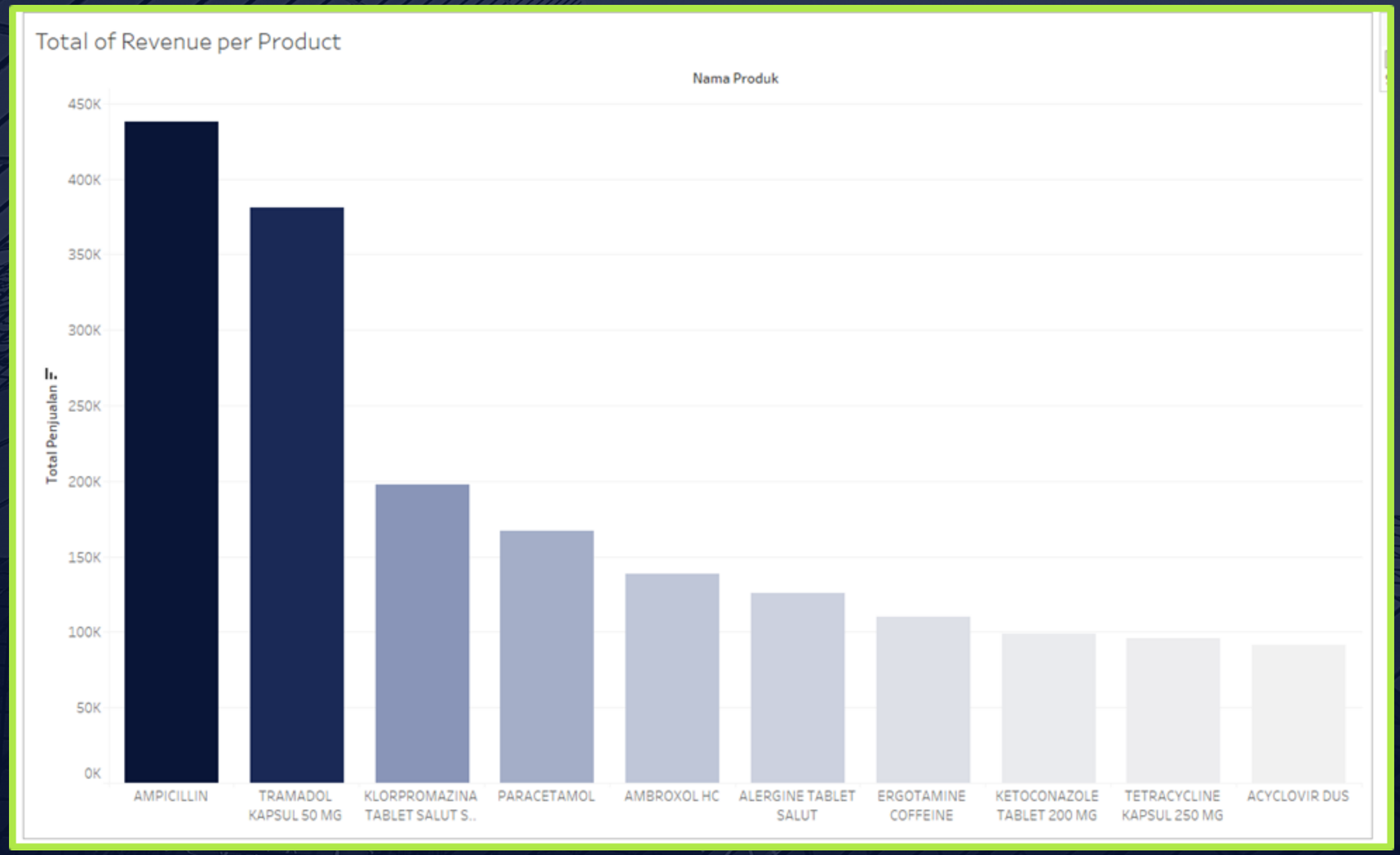
No.	Title	Link
4.	Sales Growth of Product per Month	*https://s.id/Visualization-4*
5.	Revenue Growth per Month	*https://s.id/Visualization-5*

Total of Revenue per Product

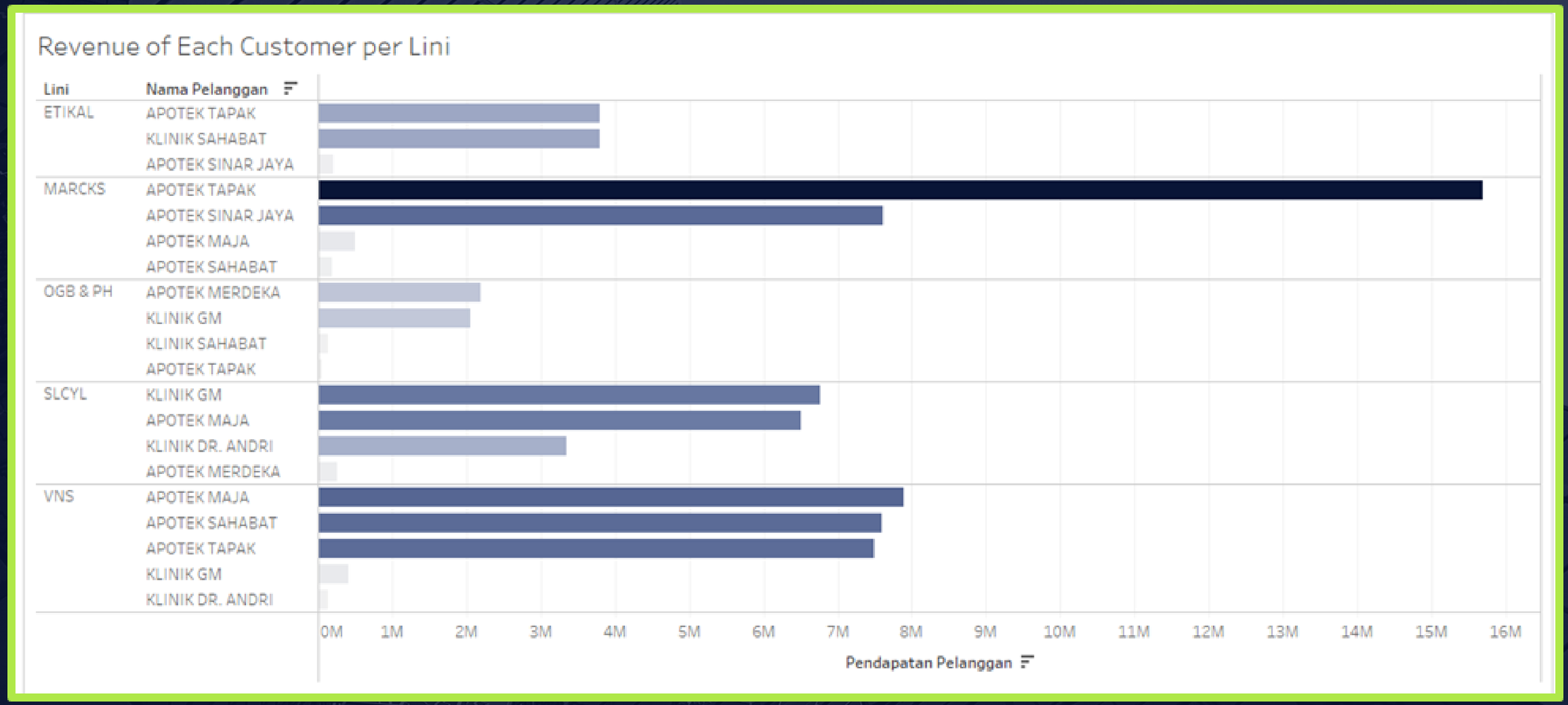




Total of Revenue per Product

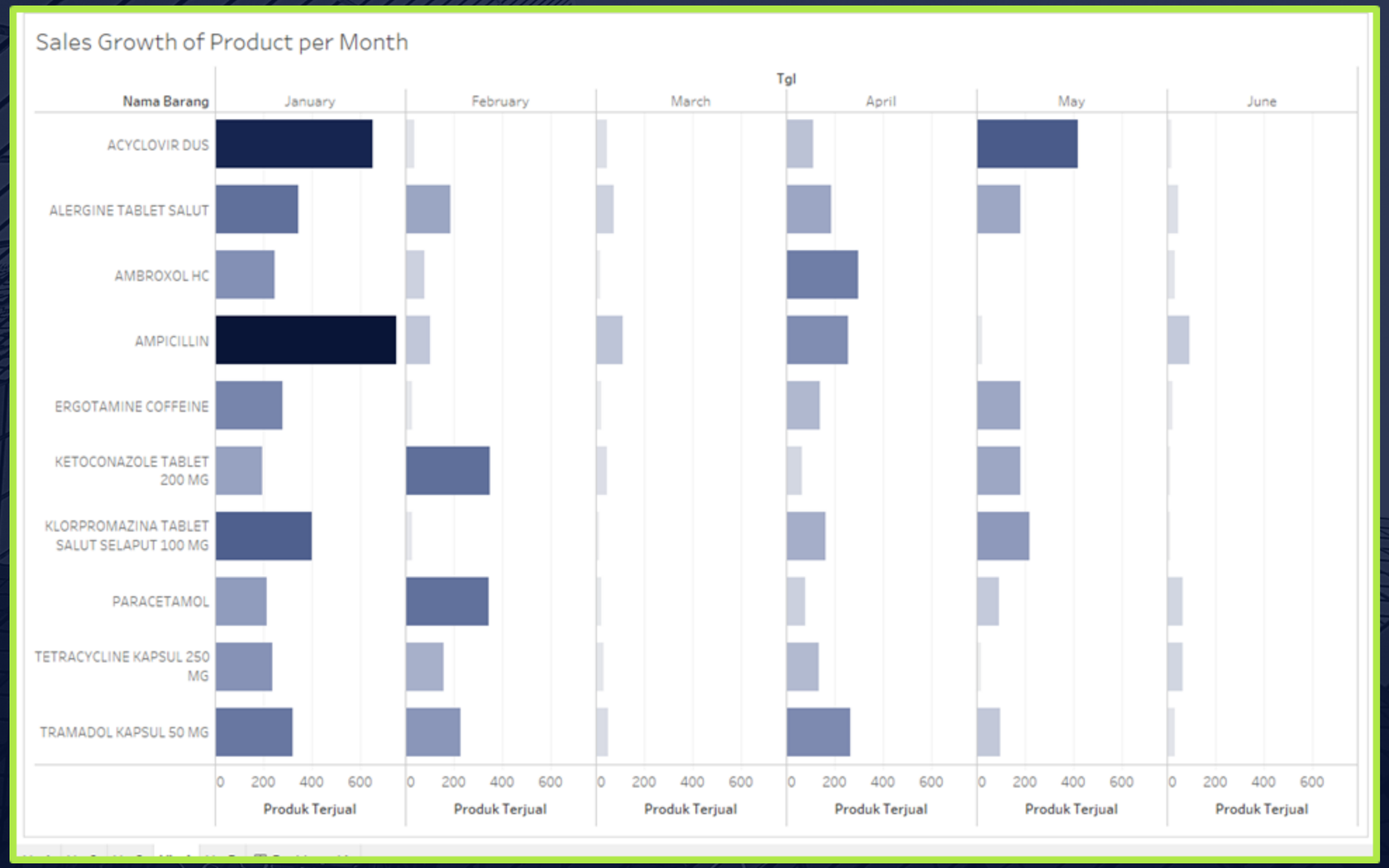


Revenue of Each Customer per Lini



Total of Product Sold per Sales Area

Sales Growth of Product per Sales Month



Answers 6

Menurut saya,

- Diperlukan data stok barang untuk memantau presentase barang yang sudah dan belum terjual, sehingga dapat segera mengambil tindakan sebelum barang mengalami kadaluarsa.
- Diperlukan data lengkap dari distributor agar tim pusat dapat turut memantau perkembangan sales di area distributor


Thank You