

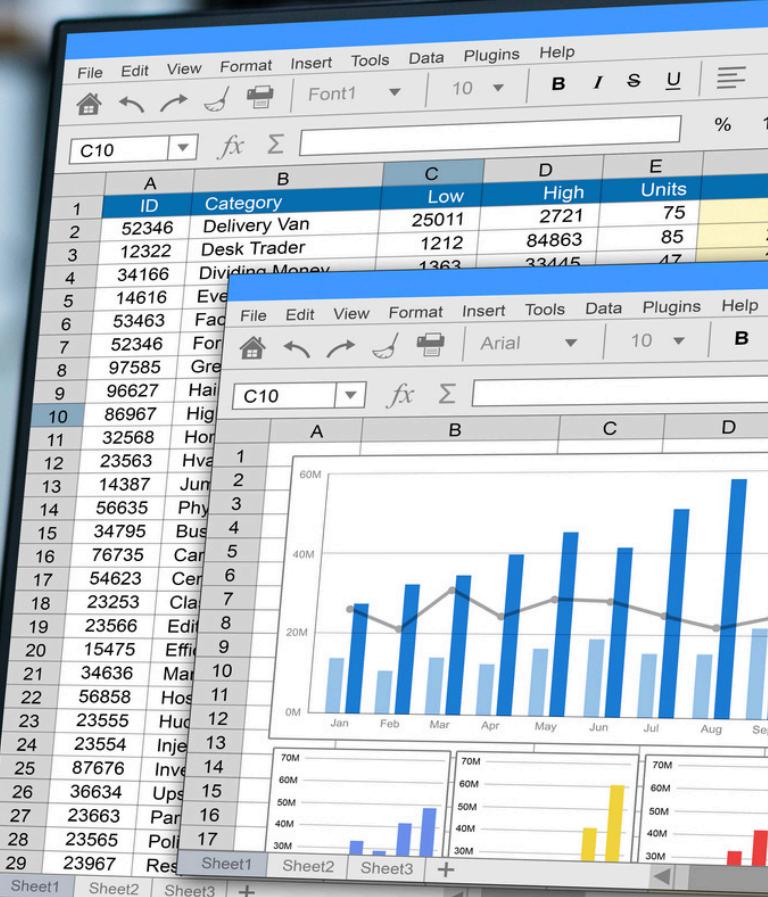


FINAL PROJECT BATCH 23

MySkill

# “TOKOPAEDI BUSINESS PERFORMANCE ANALYSIS REPORT”

DISUSUN OLEH : GABERIA SINAGA



# Project Overview

Final project ini disusun sebagai bentuk penerapan dari seluruh materi yang telah dipelajari selama masa pelatihan, mulai dari kemampuan dalam problem solving, pengolahan dan analisis data menggunakan SQL dan Python, hingga pembuatan Visualisasi Data yang informatif.

Dengan menggabungkan seluruh elemen tersebut, project ini juga berfungsi sebagai portofolio yang merepresentasikan kemampuan dan pemahaman dalam bidang data analysis secara menyeluruh.

Final project ini juga bertujuan untuk mensimulasikan situasi nyata di industri, khususnya dalam menjawab permintaan data yang umum terjadi di perusahaan. Melalui pendekatan analisis yang berbasis data, hasil dari project ini diharapkan dapat memberikan dukungan informasi yang relevan untuk perumusan kebijakan atau pengambilan keputusan yang lebih strategis.

# SQL EXERCISE



## Dataset Overview

Gambaran umum tentang dataset yang digunakan.



## Skema Tabel

struktur tabel dalam database, termasuk nama tabel, nama kolom, serta relasi antar tabel.



## Data Understanding

Proses untuk memastikan kualitas data dengan memeriksa adanya duplikasi dan nilai kosong pada kolom.



## Questions and Query

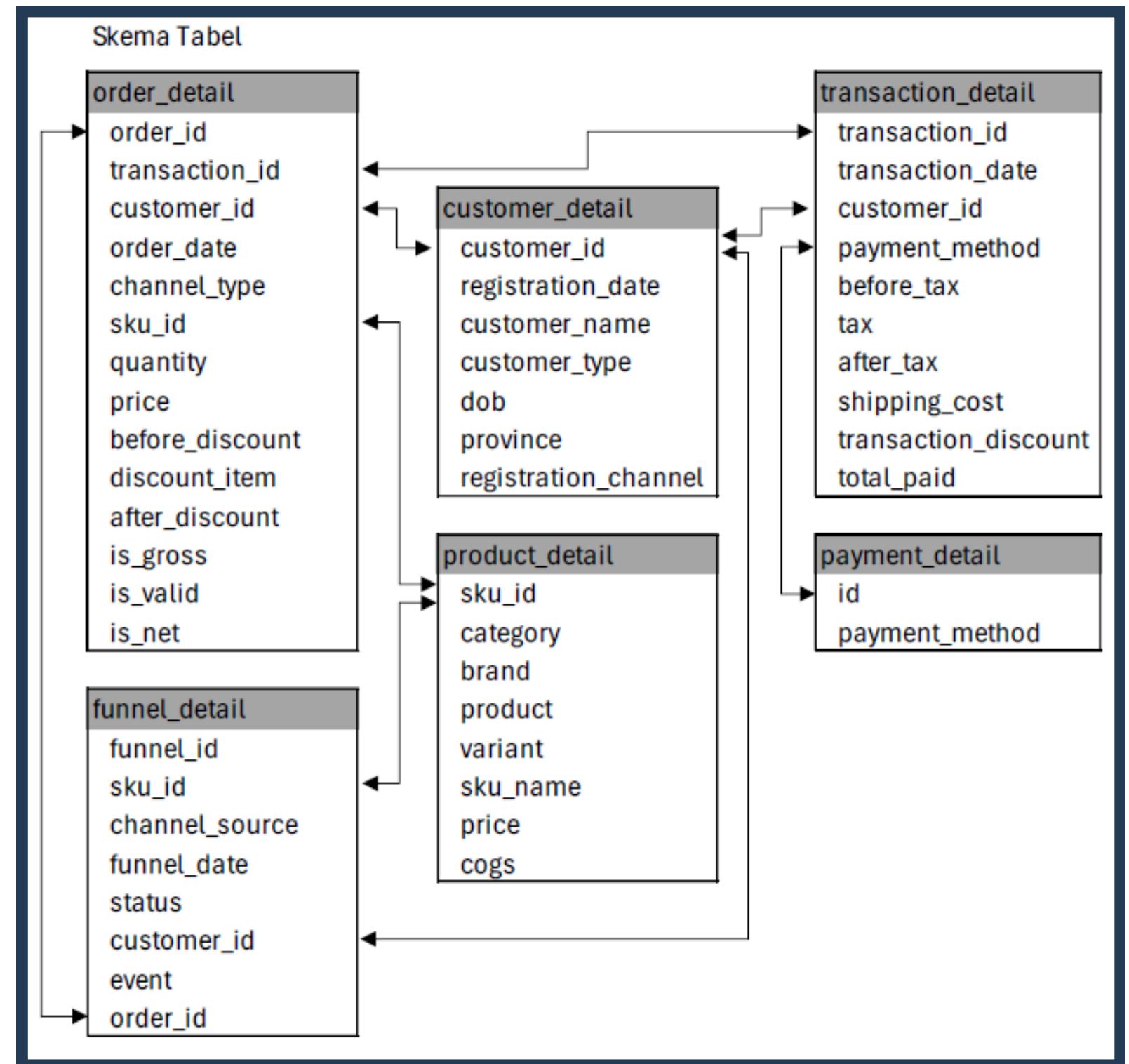
Daftar pertanyaan bisnis yang ingin dijawab serta query SQL yang digunakan untuk menjawabnya.

# DATASET OVERVIEW DAN SKEMA TABEL

Berikut dataset yang digunakan:

- > Order\_detail
- > Payment\_detail
- > Product\_detail
- > Funnel\_detail
- > Customer\_detail
- > Transaction\_detail

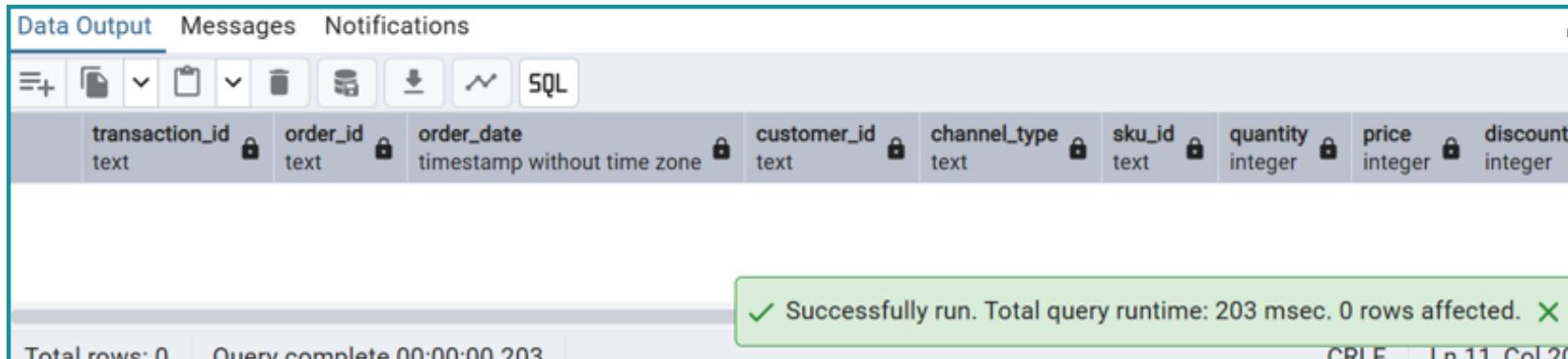
Data yang digunakan adalah dataset dummy dari suatu perusahaan, kami sebagai data analisis akan menyediakan insight yang dibutuhkan sebagai dasar rekomendasi untuk pengambilan keputusan.



# DATA UNDERSTANDING [PENGECEKAN DATA DUPLIKAT]

## Tabel customer\_detail

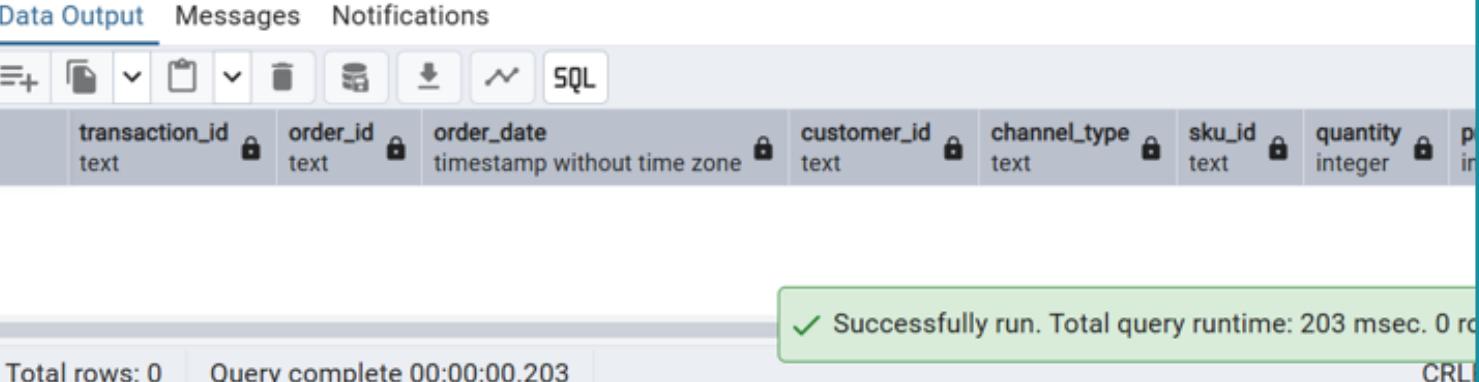
```
1 select
2     customer_id, registration_date, customer_name, customer_type, dob,
3     province, registration_channel,
4     count (*) as jumlah_duplikat
5 from customer_detail
6 group by
7     customer_id, registration_date, customer_name, customer_type, dob,
8     province, registration_channel
9 having count (*)>1;
```



The screenshot shows a database interface with a SQL query window. The query is identical to the one above. Below it, the results pane shows a table structure with columns: transaction\_id, order\_id, order\_date, customer\_id, channel\_type, sku\_id, quantity, price, discount\_item. A message at the bottom indicates "Successfully run. Total query runtime: 203 msec. 0 rows affected.".

## Tabel order\_detail

```
1 select
2     transaction_id, order_id, order_date, customer_id, channel_type,
3     sku_id, quantity, price, discount_item, is_gross, is_valid,
4     is_valid, is_net,
5     count (*) as jumlah_duplikat
6 from order_detail
7 group by
8     transaction_id, order_id, order_date, customer_id, channel_type,
9     sku_id, quantity, price, discount_item, is_gross, is_valid,
10    is_valid, is_net
11 having count (*)>1;
```



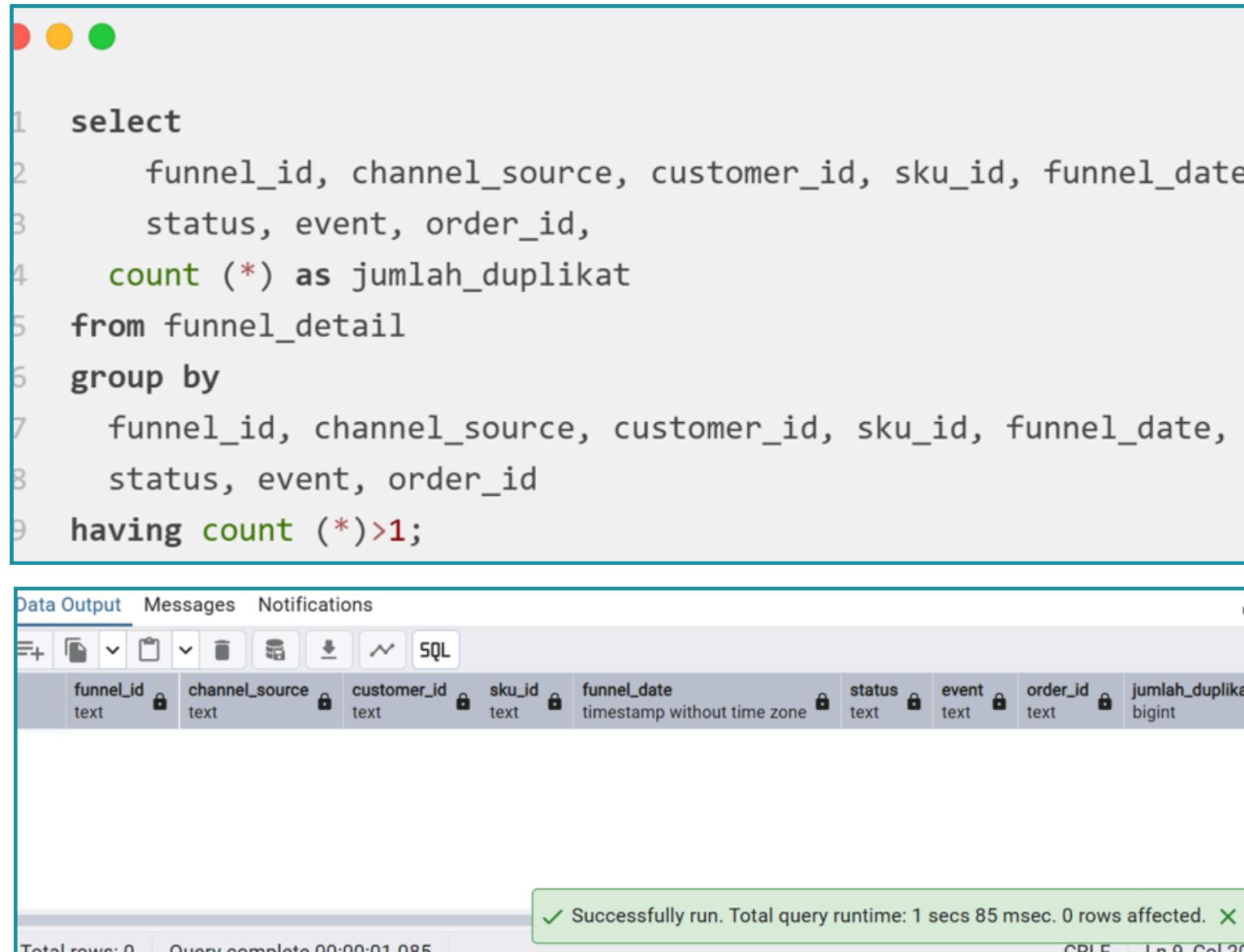
The screenshot shows a database interface with a SQL query window. The query is identical to the one above. Below it, the results pane shows a table structure with columns: transaction\_id, order\_id, order\_date, customer\_id, channel\_type, sku\_id, quantity, price, discount\_item. A message at the bottom indicates "Successfully run. Total query runtime: 203 msec. 0 rows affected.".

Proses pengecekan duplikasi pada tabel customer\_detail dan order\_detail menggunakan query SQL, pengecekan data duplikat dilakukan menggunakan syntax “**having count (\*) > 1**” pada setiap kolom. Hasil output menampilkan 0 rows affected, artinya tidak ditemukan data duplikat pada kedua tabel, sehingga dataset dapat dipastikan bersih dari duplikasi dan siap dipakai untuk analisis lebih lanjut.

# DATA UNDERSTANDING [PENGECEKAN DATA DUPLIKAT]

## Tabel funnel\_detail

```
1 select
2     funnel_id, channel_source, customer_id, sku_id, funnel_date,
3     status, event, order_id,
4     count (*) as jumlah_duplikat
5 from funnel_detail
6 group by
7     funnel_id, channel_source, customer_id, sku_id, funnel_date,
8     status, event, order_id
9 having count (*)>1;
```



Data Output Messages Notifications

SQL

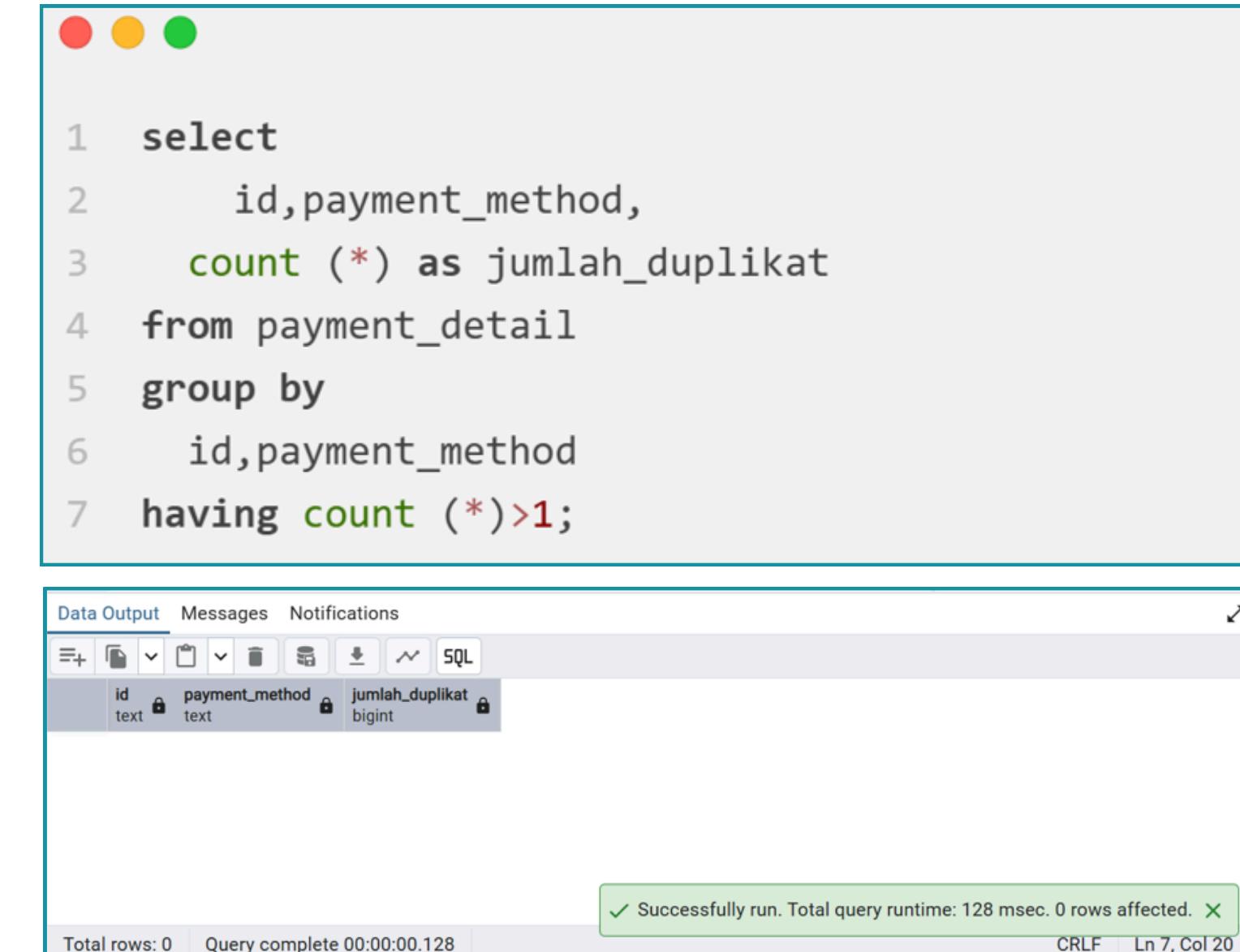
funnel_id	channel_source	customer_id	sku_id	funnel_date	status	event	order_id	jumlah_duplikat
-----------	----------------	-------------	--------	-------------	--------	-------	----------	-----------------

Successfully run. Total query runtime: 1 secs 85 msec. 0 rows affected.

Total rows: 0 Query complete 00:00:01.085 CRLF Ln 9 Col 20

## Tabel payment\_detail

```
1 select
2     id,payment_method,
3     count (*) as jumlah_duplikat
4 from payment_detail
5 group by
6     id,payment_method
7 having count (*)>1;
```



Data Output Messages Notifications

SQL

id	payment_method	jumlah_duplikat
----	----------------	-----------------

Successfully run. Total query runtime: 128 msec. 0 rows affected.

Total rows: 0 Query complete 00:00:00.128 CRLF Ln 7 Col 20

Proses pengecekan duplikasi pada tabel funnel\_detail dan payment\_detail menggunakan query SQL, pengecekan data duplikat dilakukan menggunakan syntax “**having count (\*) > 1**” pada setiap kolom. Hasil output menampilkan 0 rows affected, artinya tidak ditemukan data duplikat pada kedua tabel, sehingga dataset dapat dipastikan bersih dari duplikasi dan siap dipakai untuk analisis lebih lanjut.

# DATA UNDERSTANDING [PENGECEKAN DATA DUPLIKAT]

## Tabel transaction\_detail

```
1 select
2     transaction_id, transaction_date, customer_id, payment_method,
3     before_tax, tax, after_tax, shipping_cost, transaction_discount,
4     total_paid,
5     count (*) as jumlah_duplikat
6 from transaction_detail
7 group by
8     transaction_id, transaction_date, customer_id, payment_method,
9     before_tax, tax, after_tax, shipping_cost, transaction_discount,
10    total_paid
11   having count (*)>1;
```

The screenshot shows a database interface with a query editor and a results viewer. The query editor contains the SQL code provided above. The results viewer shows the schema of the transaction\_detail table with columns: transaction\_id, transaction\_date, customer\_id, payment\_method, before\_tax, tax, after\_tax, shipping\_cost, transaction\_discount, and total\_paid. A message at the bottom indicates the query was successfully run with 0 rows affected.

	transaction_id	transaction_date	customer_id	payment_method	before_tax	tax	after_tax	shipping_cost	transaction_discount	total_paid

✓ Successfully run. Total query runtime: 148 msec. 0 rows affected. X

Total rows: 0 Query complete 00:00:00.148

Proses pengecekan duplikasi pada tabel transaction\_detail menggunakan query SQL, pengecekan data duplikat dilakukan menggunakan syntax “**having count (\*) > 1**” pada setiap kolom. Hasil output menampilkan 0 rows affected, artinya tidak ditemukan data duplikat pada tabel, sehingga dataset dapat dipastikan bersih dari duplikasi dan siap dipakai untuk analisis lebih lanjut.

# DATA UNDERSTANDING (PENGECEKAN DATA NULL)

## Tabel order\_detail

```
1 select
2     sum(case when transaction_id is null then 1 else 0 end) as null_transaction_id,
3     sum(case when order_id is null then 1 else 0 end) as null_order_id,
4     sum(case when order_date is null then 1 else 0 end) as null_order_date,
5     sum(case when customer_id is null then 1 else 0 end) as null_customer_id,
6     sum(case when channel_type is null then 1 else 0 end) as null_channel_type,
7     sum(case when sku_id is null then 1 else 0 end) as null_sku_id,
8     sum(case when quantity is null then 1 else 0 end) as null_quantity,
9     sum(case when price is null then 1 else 0 end) as null_price,
10    sum(case when before_discount is null then 1 else 0 end) as null_before_discount,
11    sum(case when discount_item is null then 1 else 0 end) as null_discount_item,
12    sum(case when after_discount is null then 1 else 0 end) as null_after_discount,
13    sum(case when is_gross is null then 1 else 0 end) as null_is_gross,
14    sum(case when is_valid is null then 1 else 0 end) as null_is_valid,
15    sum(case when is_net is null then 1 else 0 end) as null_is_net
16 from order_detail;
```

The screenshot shows a SQL query execution interface with two tabs: 'Data Output' and 'Messages'. The 'Data Output' tab displays the results of the query. The table has nine columns: null\_transaction\_id, null\_order\_id, null\_order\_date, null\_customer\_id, null\_channel\_type, null\_sku\_id, null\_quantity, null\_price, and null\_before\_discount. All columns have a type of bigint and a value of 0. The message bar at the bottom indicates 'Successfully run. Total query runtime: 80 msec. 1 rows affected.' and 'CRLF Ln 16, Col 19'.

	null_transaction_id bigint	null_order_id bigint	null_order_date bigint	null_customer_id bigint	null_channel_type bigint	null_sku_id bigint	null_quantity bigint	null_price bigint	null_before_discount bigint
1	0	0	0	0	0	0	0	0	0

Query mengecek jumlah nilai kosong (NULL) di setiap kolom tabel order\_detail dan product\_detail.

- **SUM(CASE WHEN kolom IS NULL THEN 1 ELSE 0 END)** untuk menghitung berapa banyak NULL di tiap kolom.

## Tabel product\_detail

```
1 select
2     sum(case when sku_id is null then 1 else 0 end) as null_sku_id,
3     sum(case when category is null then 1 else 0 end) as null_category,
4     sum(case when brand is null then 1 else 0 end) as null_brand,
5     sum(case when product is null then 1 else 0 end) as null_product,
6     sum(case when variant is null then 1 else 0 end) as null_variant,
7     sum(case when sku_name is null then 1 else 0 end) as null_sku_name,
8     sum(case when price is null then 1 else 0 end) as null_price,
9     sum(case when cogs is null then 1 else 0 end) as null_cogs
10    from product_detail;
```

The screenshot shows a SQL query execution interface with three tabs: 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab displays the results of the query. The table has eight columns: null\_sku\_id, null\_category, null\_brand, null\_product, null\_variant, null\_sku\_name, null\_price, and null\_cogs. All columns have a type of bigint and a value of 0. The message bar at the bottom indicates 'Successfully run. Total query runtime: 73 msec. 1 rows affected.', 'Total rows: 1', 'Query complete 00:00:00.073', 'CRLF Ln 10, Col 21'.

	null_sku_id bigint	null_category bigint	null_brand bigint	null_product bigint	null_variant bigint	null_sku_name bigint	null_price bigint	null_cogs bigint
1	0	0	0	0	0	0	0	0

- Alias **(AS)** diberikan agar hasil lebih jelas, misalnya null\_transaction\_id, null\_order\_id, null\_order\_date, dst.
- Query melakukan pengecekan pada setiap kolom dalam tabel customer\_Detail dan tabel order\_Detail
- Hasil query menampilkan satu baris berisi total jumlah NULL pada masing-masing kolom tersebut, **terbukti bahwa tidak ada data yang kosong (Null)** pada kedua tabel dimana jumlah yang menjadi keluaran dari query adalah **0**.

# DATA UNDERSTANDING (PENGECEKAN DATA NULL)

## Tabel transaction\_detail

Query Query History

```
1 select
2     sum(case when transaction_id is null then 1 else 0 end) as null_transaction_id,
3     sum(case when transaction_date is null then 1 else 0 end) as null_transaction_date,
4     sum(case when customer_id is null then 1 else 0 end) as null_customer_id,
5     sum(case when payment_method is null then 1 else 0 end) as null_payment_method,
6     sum(case when before_tax is null then 1 else 0 end) as null_before_tax,
7     sum(case when tax is null then 1 else 0 end) as null_tax,
8     sum(case when after_tax is null then 1 else 0 end) as null_after_tax,
9     sum(case when shipping_cost is null then 1 else 0 end) as null_shipping_cost,
10    sum(case when transaction_discount is null then 1 else 0 end) as null_transaction_discount,
11    sum(case when total_paid is null then 1 else 0 end) as null_total_paid
12 from transaction_detail;
```

Data Output Messages Notifications

null_transaction_id	null_transaction_date	null_customer_id	null_payment_method	null_before_tax	null_tax	null_after_tax	null_shipping_cost
0	0	0	0	0	0	0	0

Showing rows: 1 to 1 Page No: 1 of 1 CRLF Ln 12, Col 24

✓ Successfully run. Total query runtime: 68 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.068 CRLF Ln 12, Col 24

## Tabel payment\_detail

```
1 select
2     sum(case when id is null then 1 else 0 end) as null_id,
3     sum(case when payment_method is null then 1 else 0 end) as null_payment_method
4 from payment_detail;
```

Data Output Messages Notifications

null_id	null_payment_method
0	0

Showing rows: 1 to 1 Page No: 1 of 1 CRLF Ln 4, Col 21

✓ Successfully run. Total query runtime: 66 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.066 CRLF Ln 4, Col 21

## Tabel Funnel\_detail

```
1 select
2     sum(case when funnel_id is null then 1 else 0 end) as null_funnel_id,
3     sum(case when channel_source is null then 1 else 0 end) as null_channel_source,
4     sum(case when customer_id is null then 1 else 0 end) as null_customer_id,
5     sum(case when sku_id is null then 1 else 0 end) as null_sku_id,
6     sum(case when funnel_date is null then 1 else 0 end) as null_funnel_date,
7     sum(case when status is null then 1 else 0 end) as null_status,
8     sum(case when event is null then 1 else 0 end) as null_event,
9     sum(case when order_id is null then 1 else 0 end) as null_order_id
10 from funnel_detail;
```

Data Output Messages Notifications

null_funnel_id	null_channel_source	null_customer_id	null_sku_id	null_funnel_date	null_status	null_event	null_order_id
0	0	0	0	0	0	0	0

Showing rows: 1 to 1 Page No: 1 of 1 CRLF Ln 9, Col 30

✓ Successfully run. Total query runtime: 122 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.122 CRLF Ln 9, Col 30

## Tabel customer\_detail

```
1 select
2     sum(case when customer_id is null then 1 else 0 end) as null_customer_id,
3     sum(case when registration_date is null then 1 else 0 end) as null_registration_date,
4     sum(case when customer_name is null then 1 else 0 end) as null_customer_name,
5     sum(case when customer_type is null then 1 else 0 end) as null_customer_type,
6     sum(case when dob is null then 1 else 0 end) as null_dob,
7     sum(case when province is null then 1 else 0 end) as null_province,
8     sum(case when registration_channel is null then 1 else 0 end) as null_registration_channel
9 from customer_detail;
```

Data Output Messages Notifications

null_customer_id	null_registration_date	null_customer_name	null_customer_type	null_dob	null_province	null_registration_channel
0	0	0	0	0	0	0

Showing rows: 1 to 1 Page No: 1 of 1 CRLF Ln 8, Col 95

✓ Successfully run. Total query runtime: 72 msec. 1 rows affected. X

Total rows: 1 Query complete 00:00:00.072 CRLF Ln 8, Col 95

# LIST OF QUESTIONS

## > Laporan Pendapatan Bulanan 2024

Mohon dibuatkan laporan total sales per bulan untuk tahun 2024, berdasarkan tabel transaction\_detail (kolom total\_paid). Hasilnya akan dipresentasikan di rapat dewan minggu depan.

## > Tren Penjualan Perkategori tahun 2020 - 2024

Untuk perencanaan restock, tolong tampilkan volume (quantity) terjual per kategori setiap tahun dari 2020 s.d. 2024.

## > Performa Chanel Perbulan

Dibutuhkan analisis performa channel (web, app, offline) di 2024:

- Total orders (distinct) dan revenue (after\_discount) per bulan.
- Hitung MoM growth revenue per bulan vs 2023 dalam bulan yang sama

## > Analisis Funnel - Organic Event 2024

Mohon dibuatkan laporan kinerja funnel untuk event “Organic” di funnel\_detail periode 1 Jan–31 Des 2024:

- Total jumlah event organic per channel\_source.
- Total unique order\_id (“converted”) dari event organic.
- Conversion rate =  $\text{total\_orders} \div \text{total\_events} \times 100\%$ .

Data ini untuk memetakan efektivitas jalur organik.

## > Laporan Registrasi Perbulan tahun 2024

Mohon dibuatkan laporan per bulan selama 2024 untuk:

- Jumlah pelanggan baru (distinct customer\_id) yang registrasi per registration\_channel.
- Rata-rata selisih hari antara registration\_date dan tanggal transaksi pertama (order\_date).

Hanya hitung pelanggan yang sudah melakukan minimal satu pembelian. Hasil akan dipakai untuk optimasi onboarding.

# SOAL 1

## Query

```
1 select
2     extract(month from transaction_date) as bulan,
3     sum(total_paid) as total_sales
4 from transaction_detail
5 where extract(year from transaction_date) = 2024
6 group by extract(month from transaction_date)
7 order by bulan;
```

Penjelasan query:

- select extract(month from transaction\_date) as bulan : Mengambil bulan dari kolom transaction\_date lalu memberi alias nama kolom hasilnya sebagai bulan.
- sum(total\_paid) as total\_sales : Menjumlahkan nilai total\_paid (total pembayaran) untuk setiap bulan, lalu hasilnya dinamai total\_sales.
- from transaction\_detail : Data diambil dari tabel transaction\_detail.
- where extract(year from transaction\_date) = 2024 : Hanya memilih baris (row) yang tahun pada transaction\_date adalah 2024.
- group by extract(month from transaction\_date) : Mengelompokkan data berdasarkan bulan dari transaction\_date. Jadi semua transaksi di bulan yang sama digabung.
- order by bulan : Mengurutkan hasil akhir berdasarkan kolom bulan (dari Januari = 1 sampai Desember = 12)

## Output

	bulan numeric	total_sales double precision
1	1	2161697433.600001
2	2	1456930591.2000005
3	3	1205309439.499998
4	4	1534158480.099992
5	5	2728874644.700003
6	6	1748000034.499993
7	7	1107040709.799997
8	8	1404748223.700017
9	9	1978043381.499998
10	10	1724092983.600004
11	11	2136925833.600004
12	12	4116290971.799988

# SOAL 1

## Visualisasi Data [Total Penjualan Perbulan pada tahun 2024]



### Insight:

- Terlihat fluktuasi total sales dari bulan Januari hingga Desember, meskipun terdapat kenaikan namun tidak selalu konsisten tiap bulan.
- Penjualan paling rendah terjadi di Juli ( $\pm 1,1$  B) dan maret ( $\pm 1,1$  Miliar).
- Penjualan tertinggi ada di Desember ( $\pm 4,1$  B), menandakan adanya lonjakan permintaan di akhir tahun (mungkin karena liburan, akhir tahun, atau promo besar).
- Ada tren peningkatan sales yang cukup konsisten dimulai dari Juli, hingga terjadi puncak penjualan pada bulan Desember, menunjukkan periode paruh kedua tahun 2024 lebih produktif dibanding paruh pertama.

### Rekomendasi:

- Optimalkan strategi pemasaran pada quarter 1 dan quarter 2 agar dapat mengantisipasi tren penurunan penjualan di setiap akhir quarter.
- Tingkatkan kapasitas stok & distribusi menjelang akhir tahun (Oktober–Desember) untuk mengantisipasi lonjakan permintaan.
- Analisis faktor musiman seperti liburan, promo, atau event khusus yang mempengaruhi penjualan tinggi di bulan-bulan tertentu, lalu terapkan strategi serupa di bulan dengan performa rendah.

# SOAL 2

## Query

```
1 select
2     p.category,
3     sum(case when extract(year from o.order_date) = 2020 then o.quantity else 0 end) as quantity_2020,
4     sum(case when extract(year from o.order_date) = 2021 then o.quantity else 0 end) as quantity_2021,
5     sum(case when extract(year from o.order_date) = 2022 then o.quantity else 0 end) as quantity_2022,
6     sum(case when extract(year from o.order_date) = 2023 then o.quantity else 0 end) as quantity_2023,
7     sum(case when extract(year from o.order_date) = 2024 then o.quantity else 0 end) as quantity_2024
8 from order_detail o
9 join product_detail p on o.sku_id = p.sku_id
10 where extract(year from o.order_date) between 2020 and 2024
11     and o.is_valid = 1
12 group by p.category
13 order by p.category;
```

## Output

	category text	quantity_2020 bigint	quantity_2021 bigint	quantity_2022 bigint	quantity_2023 bigint	quantity_2024 bigint
1	Beauty & Personal Care	5275	4106	2094	4921	4384
2	Books & Stationery	1733	2468	2851	3130	3389
3	Electronics	333	321	362	348	389
4	Fashion & Footwear	3609	4565	3752	2775	5143
5	Food & Beverage	7105	9368	6505	4055	4973
6	Gaming Gear	387	410	401	350	359
7	Health & Wellness	3465	3431	2397	1678	2366
8	Home Decor	751	1531	893	1921	2104
9	Kitchen Appliances	615	960	1267	993	1419
10	Sports Equipment	1166	1214	1153	738	1262
11	Toys & Games	3519	2988	2535	2517	1987

## Penjelasan query

- **select p.category** : memilih kolom kategori produk dari tabel product\_detail.
- **sum(case when extract(year from o.order\_date) = 2020 then o.quantity else 0 end) as qty\_2020** : menghitung total quantity khusus untuk tahun 2020. Jika tahun order\_date = 2020, ambil nilai quantity, jika tidak maka 0. Logika yang sama diterapkan untuk tahun 2021, 2022, 2023, dan 2024, sehingga setiap tahun punya kolom hasil sendiri.
- **from order\_detail o join product\_detail p on o.sku\_id = p.sku\_id** : menggabungkan tabel order\_detail dan product\_detail berdasarkan sku\_id supaya bisa tahu kategori dari setiap pesanan.
- **where extract(year from o.order\_date) between 2020 and 2024 and o.is\_valid = 1** : hanya mengambil data order tahun 2020–2024 yang statusnya valid (is\_valid = 1).
- **group by p.category** : mengelompokkan hasil per kategori produk, supaya total quantity tiap tahun dihitung per kategori.
- **order by p.category** : mengurutkan hasil akhir berdasarkan kategori produk.

# SOAL 2

## Visualisasi Data [Tren Penjualan Perkategori Pertahun [2020 - 2024]]



### Insight:

- Food & Beverage selalu menjadi kategori dengan quantity tertinggi, meskipun mengalami fluktuasi: puncak di 2021 (9368), turun di 2022–2023, lalu naik kembali di 2024.
- Fashion & Footwear stabil tinggi dengan kenaikan signifikan di 2024 (5143) setelah penurunan di 2023.
- Beauty & Personal Care cenderung fluktuatif, sempat turun di 2022, lalu meningkat kembali hingga 2024.
- Books & Stationery menunjukkan tren naik secara konsisten dari 2020 hingga 2024, meskipun volumenya tidak sebesar Food & Beverage.
- Electronics dan Gaming Gear relatif stabil dengan volume penjualan rendah (sekitar 300–400 tiap tahun). Tidak ada lonjakan berarti.
- Home Decor dan Kitchen Appliances menunjukkan tren kenaikan setelah 2022, mengindikasikan potensi pasar yang semakin membaik.
- Sports Equipment cenderung menurun sejak 2021, meskipun sedikit pulih di 2024.
- Health & Wellness turun tajam di 2023, namun mulai naik kembali di 2024.
- Toys & Games justru menurun konsisten sejak 2020, menunjukkan tren yang kurang positif.

# SOAL 2

## Visualisasi Data [Tren Penjualan Perkategori Pertahun [2020 - 2024]]



### Rekomendasi:

- **Perkuat kategori unggulan** : fokus strategi pada Food & Beverage dan Fashion & Footwear karena kontribusinya paling besar dan masih menunjukkan potensi pertumbuhan.
- **Dorong kategori yang sedang naik** : Books & Stationery, Home Decor, dan Kitchen Appliances bisa dipromosikan lebih agresif karena menunjukkan tren peningkatan.
- **Revitalisasi kategori menurun** : lakukan strategi promosi, bundling, atau kolaborasi brand untuk Sports Equipment, Toys & Games, dan Health & Wellness agar bisa kembali menarik minat konsumen.
- **Eksperimen inovasi di kategori stabil rendah** : Electronics dan Gaming Gear mungkin butuh diferensiasi produk (misalnya aksesoris unik, limited edition, atau diskon seasonal).
- **Pantau pola musiman** : perlu dicek lebih detail apakah penurunan di beberapa kategori (misalnya 2022-2023) terkait faktor musiman, harga, atau perilaku konsumen pasca-pandemi.

# SOAL 3

## Query

```
1 with revenue_summary as (
2     select
3         extract(year from order_date) as tahun,
4         extract(month from order_date) as bulan,
5         channel_type,
6         count(distinct order_id) as total_orders,
7         sum(after_discount) as revenue
8     from order_detail
9     where extract(year from order_date) in (2023, 2024)
10        and is_valid = 1
11    group by extract(year from order_date), extract(month from order_date), channel_type
12 )
13
14 select
15     a.bulan,
16     a.channel_type,
17     b.total_orders as total_orders_2023,
18     b.revenue as revenue_2023,
19     a.total_orders as total_orders_2024,
20     a.revenue as revenue_2024,
21     case
22         when b.revenue = 0 or b.revenue is null then null
23         else round(((a.revenue::numeric - b.revenue::numeric) / b.revenue::numeric) * 100, 2)
24     end as mom_growth_pct
25 from revenue_summary a
26 left join revenue_summary b
27     on a.bulan = b.bulan
28     and a.channel_type = b.channel_type
29     and b.tahun = 2023
30 where a.tahun = 2024
31     and a.channel_type in ('Website', 'App Store', 'Offline Store')
32 order by a.bulan, a.channel_type;
```

## Penjelasan query

- **with revenue\_summary as (...)** : membuat tabel sementara berisi ringkasan data per tahun, bulan, dan channel\_type.
- **extract(year/month from order\_date)** : mengambil tahun dan bulan dari kolom order\_date supaya bisa dianalisis per periode.
- **count(distinct order\_id) as total\_orders** : menghitung jumlah order unik tiap channel, bulan, dan tahun.
- **sum(after\_discount) as revenue** : total pendapatan dihitung setelah diskon, bukan harga awal.
- **where extract(year from order\_date) in (2023, 2024) and is\_valid = 1** : hanya ambil transaksi tahun 2023-2024 dan yang valid (sudah dibayar).
- **left join revenue\_summary b on ...** : menggabungkan data tahun 2024 (a) dengan tahun 2023 (b) berdasarkan bulan dan channel\_type → tujuannya biar bisa dibandingkan.
- **select ...** : mengambil data bulan, channel\_type, total\_orders dan revenue untuk 2023 dan 2024.
- **case ... end as mom\_growth\_pct** : menghitung persentase pertumbuhan revenue 2024 dibanding 2023 per bulan & channel. kalau revenue 2023 kosong maka hasil null (biar tidak error bagi nol).
- **order by a.bulan, a.channel\_type** : mengurutkan hasil akhir berdasarkan bulan lalu channel.

# SOAL 3

## Output

1	bulan	channel_type	# total_orders_2023	# revenue_2023	# total_orders_2024	# revenue_2024	# mom_growth_pct
2	January	App Store	75	\$ 508,083,211	95	\$ 397,890,503.00	-21.69
3	January	Offline Store	56	\$ 104,590,183	89	\$ 530,720,310.00	407.43
4	January	Website	57	\$ 196,780,594	90	\$ 305,793,868.00	55.40
5	February	App Store	68	\$ 255,891,614	47	\$ 152,669,395.00	-40.34
6	February	Offline Store	49	\$ 94,525,054	70	\$ 575,514,088.00	508.85
7	February	Website	64	\$ 92,669,518	81	\$ 171,770,951.00	85.36
8	March	App Store	74	\$ 603,862,323	86	\$ 349,314,723.00	-42.15
9	March	Offline Store	54	\$ 100,390,535	61	\$ 175,464,934.00	74.78
10	March	Website	86	\$ 668,851,772	80	\$ 360,812,092.00	-46.05
11	April	App Store	88	\$ 168,611,565	117	\$ 455,693,914.00	170.26
12	April	Offline Store	83	\$ 537,240,313	94	\$ 262,864,763.00	-51.07
13	April	Website	102	\$ 477,466,790	104	\$ 393,380,572.00	-17.61
14	May	App Store	139	\$ 595,769,033	101	\$ 682,950,468.00	14.63
15	May	Offline Store	112	\$ 168,212,584	76	\$ 569,079,242.00	238.31
16	May	Website	113	\$ 408,827,404	103	\$ 387,316,373.00	-5.26
17	June	App Store	123	\$ 1,025,648,134	84	\$ 184,449,527.00	-82.02
18	June	Offline Store	95	\$ 264,364,175	74	\$ 110,051,705.00	-58.37
19	June	Website	91	\$ 145,561,618	133	\$ 707,227,656.00	385.86
20	July	App Store	55	\$ 172,960,734	55	\$ 116,337,642.00	-32.74
21	July	Offline Store	54	\$ 322,428,581	65	\$ 360,725,352.00	11.88
22	July	Website	67	\$ 233,452,480	80	\$ 178,960,791.00	-23.34
23	August	App Store	107	\$ 685,396,195	77	\$ 159,524,373.00	-76.73
24	August	Offline Store	108	\$ 467,716,006	106	\$ 266,053,772.00	-43.12
25	August	Website	124	\$ 264,860,197	98	\$ 271,023,168.00	2.33
26	September	App Store	87	\$ 200,027,011	117	\$ 659,794,210.00	229.85
27	September	Offline Store	104	\$ 757,525,023	94	\$ 439,463,671.00	-41.99
28	September	Website	97	\$ 377,814,748	100	\$ 183,282,557.00	-51.49
29	October	App Store	78	\$ 577,875,077	70	\$ 201,502,187.00	-65.13
30	October	Offline Store	52	\$ 119,052,993	55	\$ 153,245,857.00	28.72
31	October	Website	75	\$ 319,594,093	108	\$ 901,004,887.00	181.92
32	November	App Store	110	\$ 302,390,431	100	\$ 481,973,951.00	59.39
33	November	Offline Store	37	\$ 186,311,929	79	\$ 909,356,710.00	388.08
34	November	Website	88	\$ 177,586,485	74	\$ 302,212,359.00	70.18
35	December	App Store	134	\$ 685,838,705	147	\$ 891,230,899.00	29.95
36	December	Offline Store	116	\$ 209,043,589	174	\$ 1,543,304,328.00	638.27
37	December	Website	150	\$ 316,203,361	131	\$ 623,533,578.00	97.19

### Insight:

#### 1. App Store

- Revenue fluktuatif: sempat turun tajam di Februari–Maret 2024 (-40% sampai -42%), kemudian rebound besar di April (+170%), tapi kembali drop di pertengahan tahun (Juni -82%, Agustus -77%).
- Jumlah order relatif stabil, tapi revenue tidak konsisten : kemungkinan harga produk, diskon, atau average order value (AOV) berubah signifikan.
- Akhir tahun (Nov–Des) recovery, revenue naik +59% dan +30%.

#### 2. Offline Store

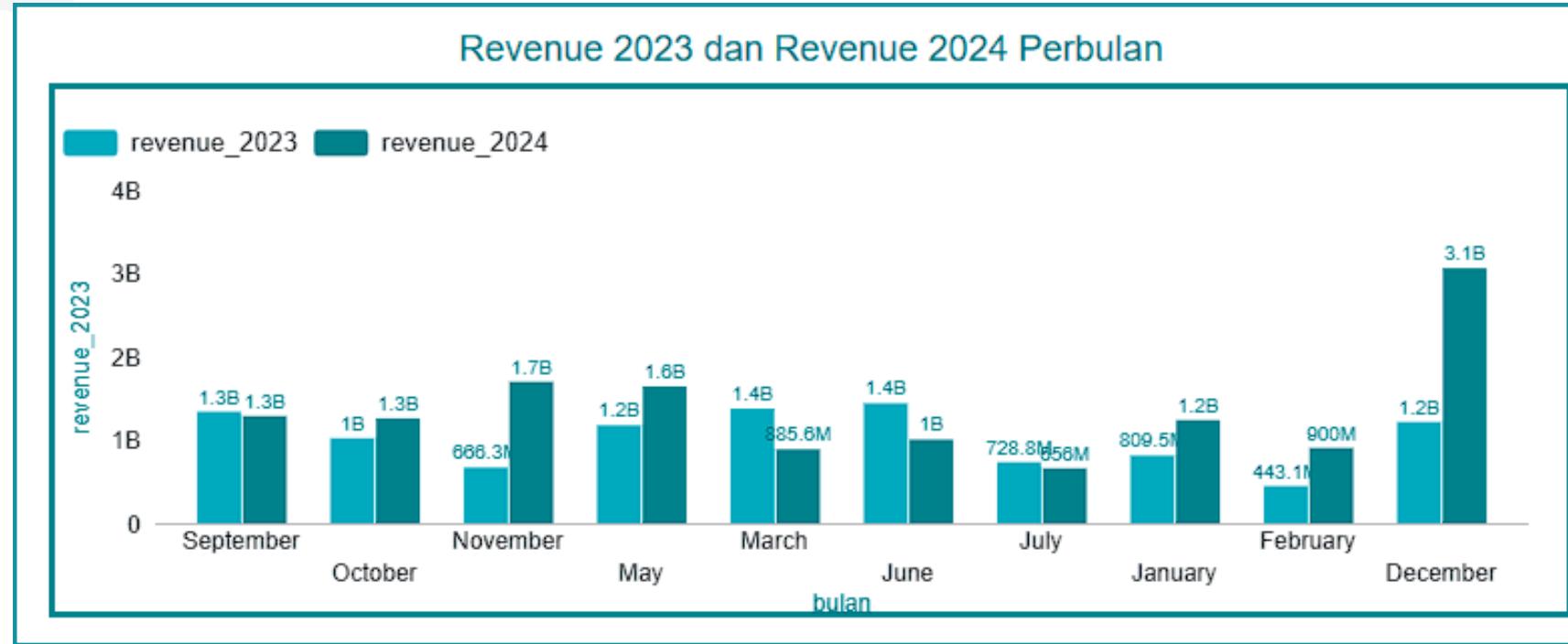
- Pertumbuhan sangat agresif di awal 2024, Januari (+407%), Februari (+509%), Mei (+238%).
- Tapi ada penurunan besar di beberapa bulan (Juni -58%, Agustus -43%, September -42%).
- Desember 2024 jadi puncak dengan revenue 6x lipat (+638%) dibanding Desember 2023 : kemungkinan efek holiday season atau kampanye offline besar.

#### 3. Website

- Cenderung lebih stabil dibanding dua channel lain.
- Ada penurunan di beberapa bulan (Maret -46%, September -51%), tapi juga lonjakan besar (Juni +386%, Oktober +182%, Desember +97%).
- Kenaikan signifikan di paruh akhir tahun 2024 menunjukkan perbaikan strategi website (mungkin promosi digital atau UX optimization).

# SOAL 3

Visualisasi Data[Total Revenue 2023 dan Revenue 2024 Perbulan]]



Visualisasi Data[Total Orders 2023 dan 2024 Perbulan]]



## Rekomendasi:

### App Store

- Perlu evaluasi penyebab turunnya revenue di pertengahan tahun : kemungkinan masalah pricing, promosi tidak konsisten, atau loyalitas pelanggan rendah.
- Terapkan program loyalti khusus mobile user untuk menjaga konsistensi transaksi.
- Optimalkan promosi di bulan yang historically lemah (Februari, Juni, Agustus).

### Offline Store

- Channel offline menunjukkan potensi besar (lonjakan revenue drastis di beberapa bulan).
- Bisa diperkuat dengan event musiman, bundling produk, dan kerja sama merchant lokal.
- Desember terbukti sangat kuat : manfaatkan holiday campaign lebih awal (Nov–Des) agar tren positif makin besar.

### Website

- Website konsisten tumbuh di akhir tahun : optimalkan SEO, iklan digital, dan UX improvement.
- Gunakan personalized promotion untuk menjaga pelanggan tetap berbelanja meski di bulan-bulan lemah (Maret, September).
- Integrasi dengan kampanye online dan offline agar pelanggan bisa beralih channel tanpa hambatan.

# SOAL 4

## Query



```
1 select
2     f.channel_source,
3     count(*) as total_events,
4     count(distinct f.order_id) as total_orders,
5     round(count(distinct f.order_id)::numeric / count(*)::numeric * 100, 2)
6     as conversion_rate
7 from funnel_detail f
8 where f.event = 'Organic'
9     and f.funnel_date between '2024-01-01' and '2025-01-01'
10 group by f.channel_source
11 order by conversion_rate desc;
```

## Output

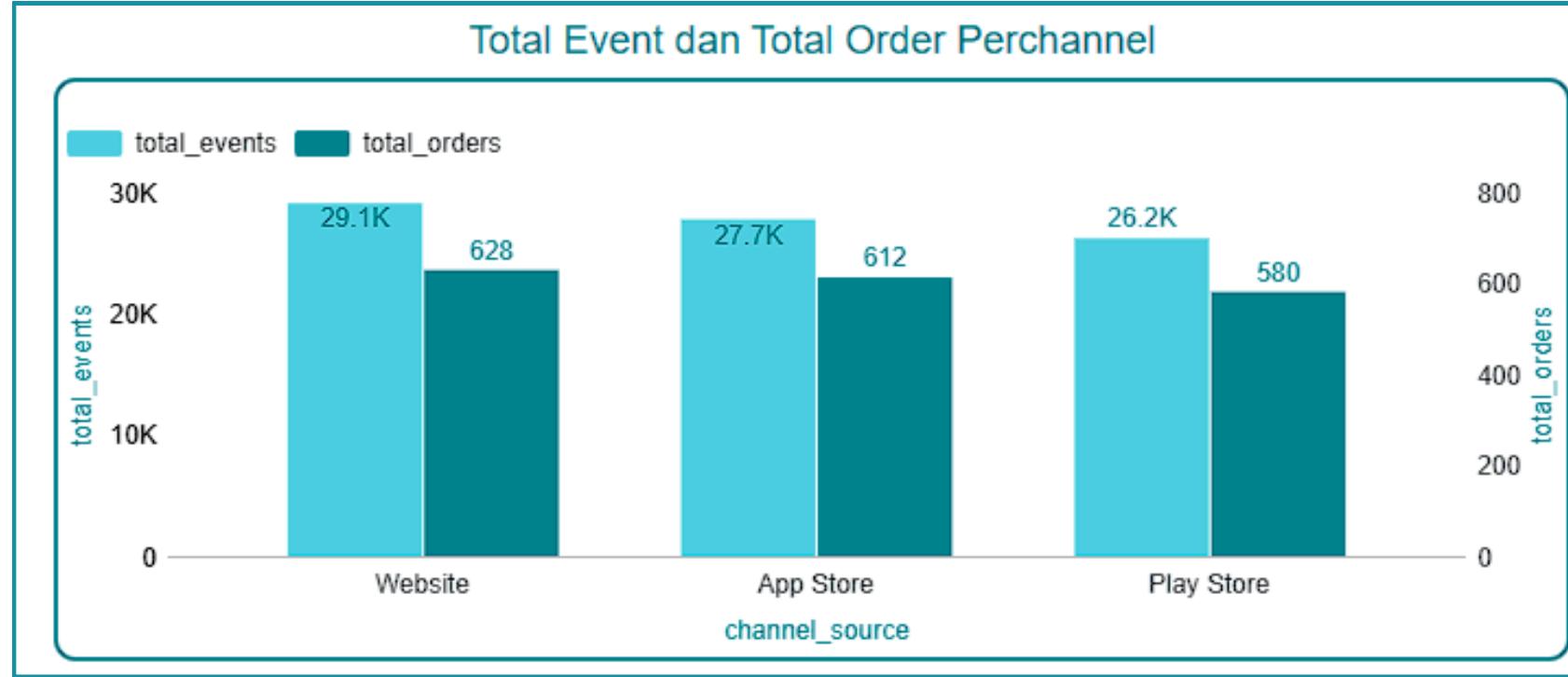
	channel_source	total_events	total_orders	conversion_rate
1	Play Store	26165	580	2.22
2	App Store	27717	612	2.21
3	Website	29062	628	2.16

## Penjelasan query

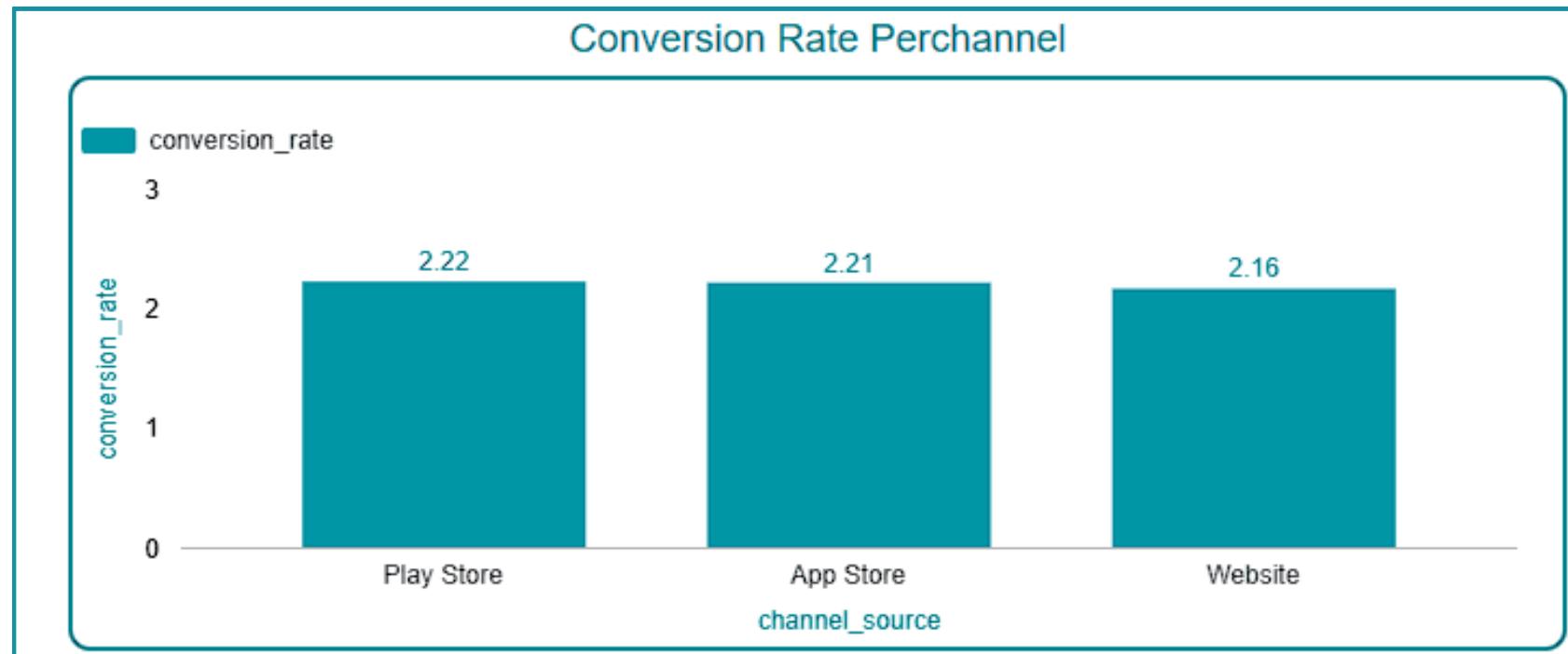
- **select f.channel\_source** : Mengambil sumber channel dari tabel funnel\_detail.
- **count(\*) as total\_events** : Menghitung jumlah total event dengan kondisi yang ditentukan.
- **count(distinct f.order\_id) as total\_orders** : Menghitung jumlah unik order\_id (pesanan yang berbeda).
- **round(count(distinct f.order\_id)::numeric / count(\*)::numeric \* 100, 2) as conversion\_rate** : Menghitung rasio jumlah pesanan unik dibandingkan dengan total event. Hasilnya dikalikan 100 untuk mendapatkan persentase dan dibulatkan hingga 2 angka di belakang koma. dinamai sebagai conversion rate.
- **from funnel\_detail f** : Data diambil dari tabel funnel\_detail dengan alias f.
- **where f.event = 'organic'** : Hanya memilih baris data dengan event bernama organic.
- **and f.funnel\_date between '2024-01-01' and '2025-01-01'** : Membatasi data berdasarkan periode tanggal dari 1 Januari 2024 sampai 1 Januari 2025.
- **group by f.channel\_source** : Mengelompokkan data berdasarkan channel\_source.
- **order by conversion\_rate desc** : Mengurutkan hasil akhir berdasarkan nilai conversion\_rate dari yang tertinggi ke terendah.

# SOAL 4

Visualisasi Data[Total Event dan Total Order tahun 2024 Perchannel]



Visualisasi Data[Conversion Rate Perchannel]

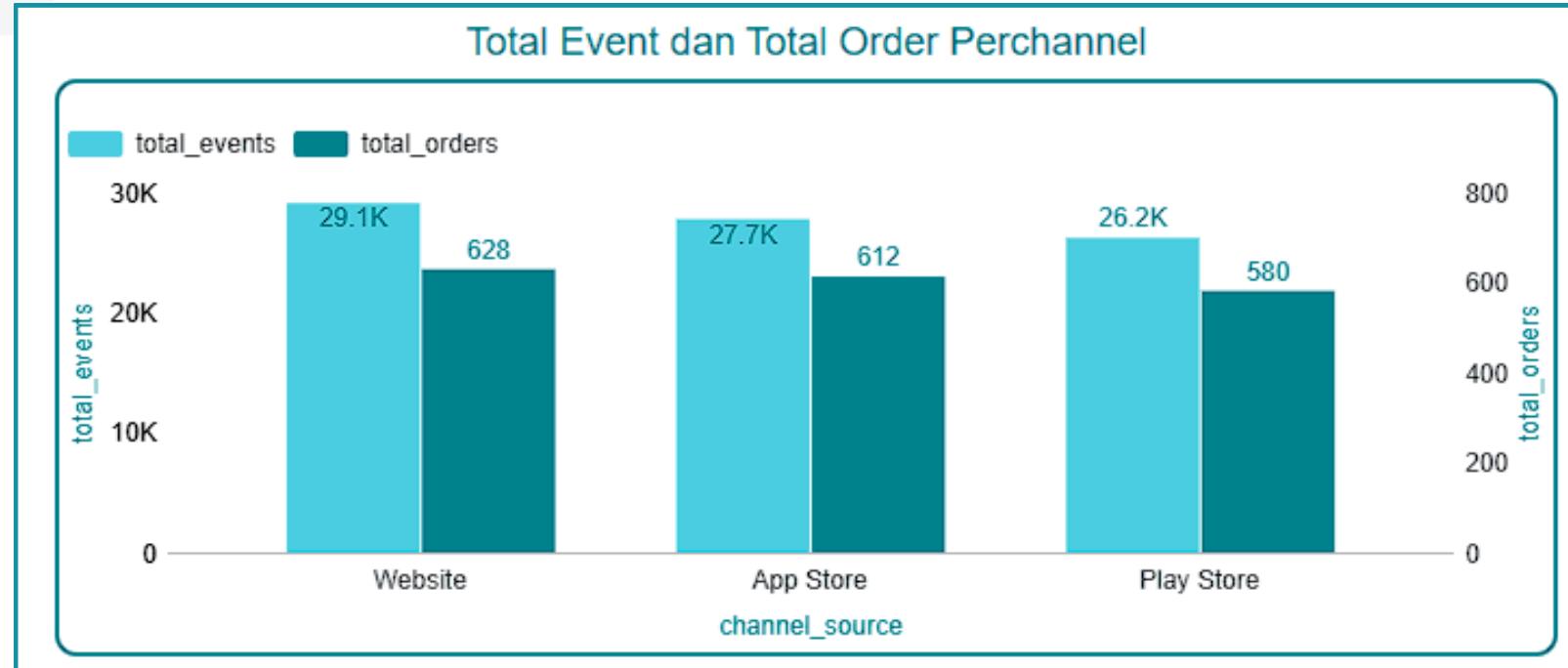


## Insight:

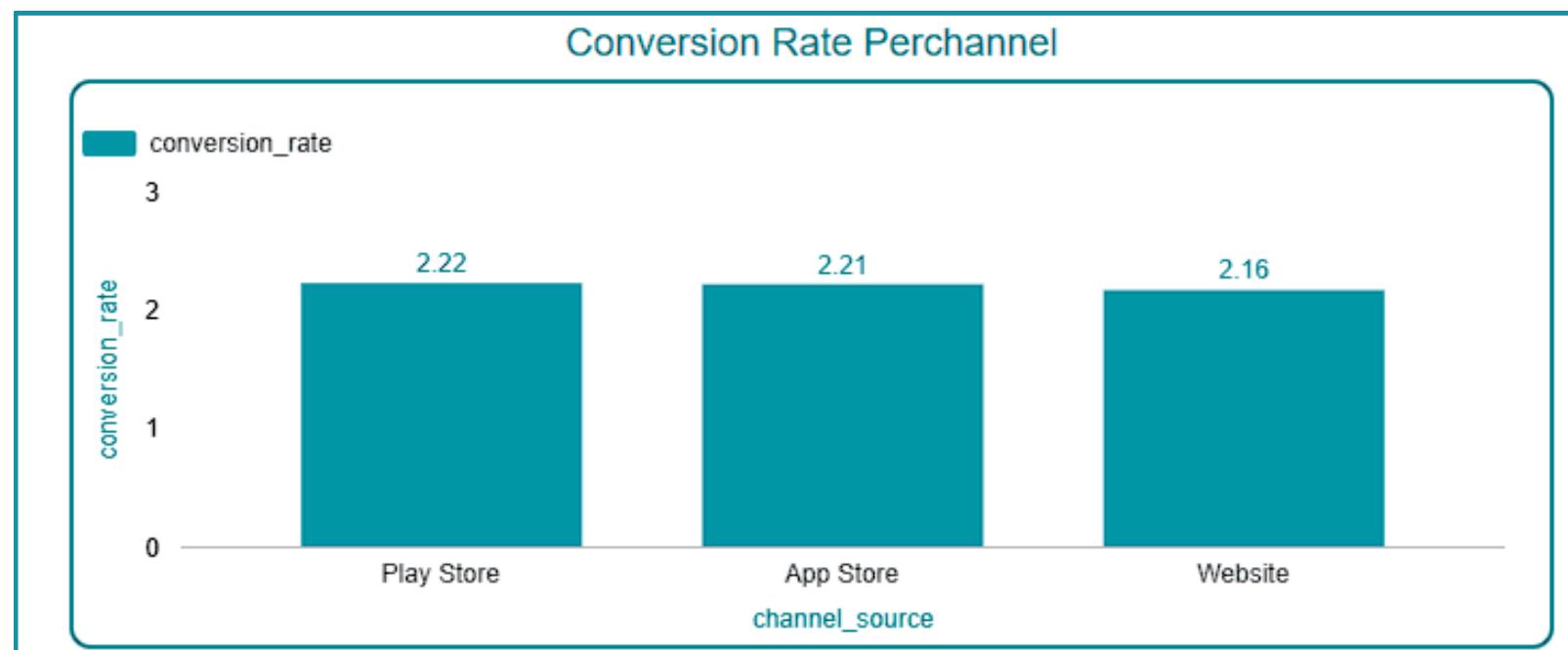
- Jumlah Event Tertinggi adalah Website (29,062 events) : Website menghasilkan traffic paling besar dibanding channel lain, tapi conversion rate paling rendah (2.16%). Artinya banyak user datang lewat website, tapi tidak semua melakukan order.
- Conversion Rate Tertinggi adalah Play Store (2.22%) : Meskipun jumlah event lebih rendah daripada Website dan App Store, Play Store memiliki persentase konversi tertinggi. Artinya user Play Store lebih “niat” untuk melakukan order.
- App Store diposisi Tengah (2.21%) App Store memiliki total events lebih tinggi dari Play Store dan conversion rate hampir sama. Menunjukkan user iOS cukup engaged, meski tidak jauh berbeda dari Android.
- Perbedaan Conversion Rate Kecil (2.22% vs 2.21% vs 2.16%) : Gap antar channel relatif kecil, sekitar 0.06%. Ini menandakan strategi promosi antar platform sudah cukup seimbang.

# SOAL 4

Visualisasi Data[Total Event dan Total Order tahun 2024 Perchannel]



Visualisasi Data[Conversion Rate Perchannel]



## Rekomendasi:

### Optimalkan Website

- Website punya traffic tinggi tapi conversion rate rendah : evaluasi UX/UI website (form order, checkout flow, kecepatan loading).
- Lakukan A/B testing untuk CTA (Call to Action) agar lebih banyak visitor jadi order.

### Fokus di Play Store & App Store

- Walaupun event lebih sedikit, user di mobile app lebih cenderung order, tingkatkan campaign untuk mendorong lebih banyak install app.
- Pertahankan kenyamanan experience di aplikasi (UI/UX, kecepatan, notifikasi push).

### Perbaiki Funnel Awareness → Order di Website

- Identifikasi titik drop-off di funnel website.
- Mungkin banyak pengunjung hanya browsing tanpa niat beli, coba strategi remarketing / reminder email.

### Strategi Cross-Channel

- Gunakan website untuk akuisisi traffic, lalu dorong user install aplikasi (karena conversion rate di aplikasi lebih tinggi). Misalnya pop-up “Download App untuk diskon khusus”.

# SOAL 5

## Query

```
1 with reg as (
2     select
3         customer_id,
4         registration_channel,
5         date_trunc('month', registration_date)::date as reg_month
6     from customer_detail
7     where registration_date between '2024-01-01'::date
8         and '2025-01-01'::date
9 ),
10 first_tx as (
11     select
12         od.customer_id,
13         min(od.order_date)::date as first_order_date
14     from order_detail od
15     where od.is_valid = 1
16     group by od.customer_id
17 )
18 select
19     r.reg_month as tanggal_registrasi,
20     r.registration_channel as channel_registrasi,
21     count(distinct r.customer_id) as Pelanggan_baru,
22     round(
23         avg((f.first_order_date - r.reg_month)::numeric),
24         2
25     ) as avg_days_to_first_order
26     from reg r
27     join first_tx f
28     on r.customer_id = f.customer_id
29     group by r.reg_month, r.registration_channel
30     order by r.reg_month, r.registration_channel;
```

## Penjelasan query

- with reg as : Mulai membuat CTE (tabel sementara) bernama reg.
- select : Memilih kolom yang akan masuk ke CTE reg.
- customer\_id : Ambil id pelanggan.
- registration\_channel : Ambil channel tempat pelanggan mendaftar
- date\_trunc('month', registration\_date)::date as reg\_month : Potong registration\_date ke awal bulan dan diberi nama reg\_month.
- from customer\_detail : Sumber data dari tabel customer\_detail.
- where registration\_date between '2024-01-01'::date and '2025-01-01'::date : Batasi data mulai tanggal 1 Jan 2024 (inklusif) Sampai tanggal 1 Jan 2025 (inklusif). Jadi hanya registrasi dalam rentang ini yang diambil.
- first\_tx as : Mulai membuat CTE lain bernama first\_tx.
- select : Memilih kolom untuk first\_tx.
- od.customer\_id : Ambil id pelanggan dari tabel order (alias od).
- min(od.order\_date)::date as first\_order\_date : Ambil tanggal order paling awal per pelanggan. Jadi ini adalah tanggal transaksi pertama tiap pelanggan.
- from order\_detail od : Sumber data: tabel order\_detail (alias od).
- where od.is\_valid = 1 : Hanya hitung order yang valid (filter validitas).
- group by od.customer\_id : Kelompokkan per pelanggan agar min(order\_date) berlaku untuk tiap pelanggan.

# SOAL 5

## Query

```
1 with reg as (
2     select
3         customer_id,
4         registration_channel,
5         date_trunc('month', registration_date)::date as reg_month
6     from customer_detail
7     where registration_date between '2024-01-01'::date
8         and '2025-01-01'::date
9 ),
10 first_tx as (
11     select
12         od.customer_id,
13         min(od.order_date)::date as first_order_date
14     from order_detail od
15     where od.is_valid = 1
16     group by od.customer_id
17 )
18 select
19     r.reg_month as tanggal_registrasi,
20     r.registration_channel as channel_registrasi,
21     count(distinct r.customer_id) as Pelanggan_baru,
22     round(
23         avg((f.first_order_date - r.reg_month)::numeric),
24         2
25     ) as avg_days_to_first_order
26     from reg r
27     join first_tx f
28     on r.customer_id = f.customer_id
29     group by r.reg_month, r.registration_channel
30     order by r.reg_month, r.registration_channel;
```

## Penjelasan query

- **select** : Mulai query utama yang menghasilkan output akhir.
- **r.reg\_month as tanggal\_registrasi**, : Ambil reg\_month dari CTE reg dan beri nama kolom output tanggal\_registrasi.
- **r.registration\_channel as channel\_registrasi**, : Ambil channel registrasi dan beri nama channel\_registrasi.
- **count(distinct r.customer\_id) as pelanggan\_baru**, : Hitung jumlah pelanggan unik (baru) pada kombinasi bulan + channel tersebut.
- **round**( : Mulai fungsi pembulatan untuk nilai berikutnya.
- **avg((f.first\_order\_date - r.reg\_month)::numeric)** : Hitung rata-rata selisih hari antara first\_order\_date dan reg\_month.
- **f.first\_order\_date - r.reg\_month** : hasilnya jumlah hari (berapa hari dari awal bulan registrasi sampai transaksi pertama).
- **as avg\_days\_to\_first\_order** : Beri nama kolom hasil pembulatan: avg\_days\_to\_first\_order.
- **from reg r** : Sumber utama adalah CTE reg, diberi alias r.
- **join first\_tx f** : Lakukan inner join dengan CTE first\_tx (alias f).
- **on r.customer\_id = f.customer\_id** : satukan baris yang punya customer\_id sama di kedua CTE , artinya hanya pelanggan yang punya transaksi pertama yang masuk ke hasil.
- **group by r.reg\_month, r.registration\_channel** : Agregasi dilakukan per kombinasi bulan registrasi dan channel registrasi.
- **order by r.reg\_month, r.registration\_channel**; Urutkan hasil berdasarkan bulan registrasi lalu channel (ascending).

# SOAL 5

## Output

1	tanggal_daftar	channel_daftar	# pelanggan_baru	# avg_days_to_first_order
2	1/1/2024	Mobile App	15	107.53
3	1/1/2024	Offline Store	8	99.25
4	1/1/2024	Website	7	75.29
5	1/2/2024	Mobile App	9	105.78
6	1/2/2024	Offline Store	7	96.29
7	1/2/2024	Website	10	93.80
8	1/3/2024	Mobile App	5	81.60
9	1/3/2024	Offline Store	8	86.75
10	1/3/2024	Website	10	102.50
11	1/4/2024	Mobile App	5	61.40
12	1/4/2024	Offline Store	4	65.75
13	1/4/2024	Website	7	83.29
14	1/5/2024	Mobile App	6	83.00
15	1/5/2024	Offline Store	1	151.00
16	1/5/2024	Website	8	67.50
17	1/6/2024	Mobile App	6	115.67
18	1/6/2024	Offline Store	1	62.00
19	1/6/2024	Website	4	65.00
20	1/7/2024	Mobile App	9	92.78
21	1/7/2024	Offline Store	2	152.50
22	1/7/2024	Website	7	80.71
23	1/8/2024	Mobile App	10	102.00
24	1/8/2024	Offline Store	4	85.25
25	1/8/2024	Website	3	91.00
26	1/9/2024	Mobile App	6	78.83
27	1/9/2024	Offline Store	2	33.50
28	1/9/2024	Website	2	90.00
29	1/10/2024	Mobile App	3	83.00
30	1/10/2024	Offline Store	4	61.00
31	1/10/2024	Website	2	43.50
32	1/11/2024	Mobile App	2	40.00
33	1/11/2024	Website	2	55.00
34	1/12/2024	Offline Store	1	6.00
35	1/12/2024	Website	1	22.00

### Insight:

#### Mobile App

- Konsisten jadi channel dengan jumlah pelanggan baru paling tinggi. 15 di Jan, 10 di Agustus, rata-rata 5-10 tiap bulan.
- Namun, rata-rata hari ke order pertama cukup tinggi sering di atas 90-110 hari, misalnya Juni = 115,67 hari.
- Interpretasi: User banyak daftar lewat aplikasi, tapi mereka lama sekali melakukan transaksi pertama, onboarding app perlu diperkuat (misalnya reminder, promo first order, edukasi fitur).

#### Website

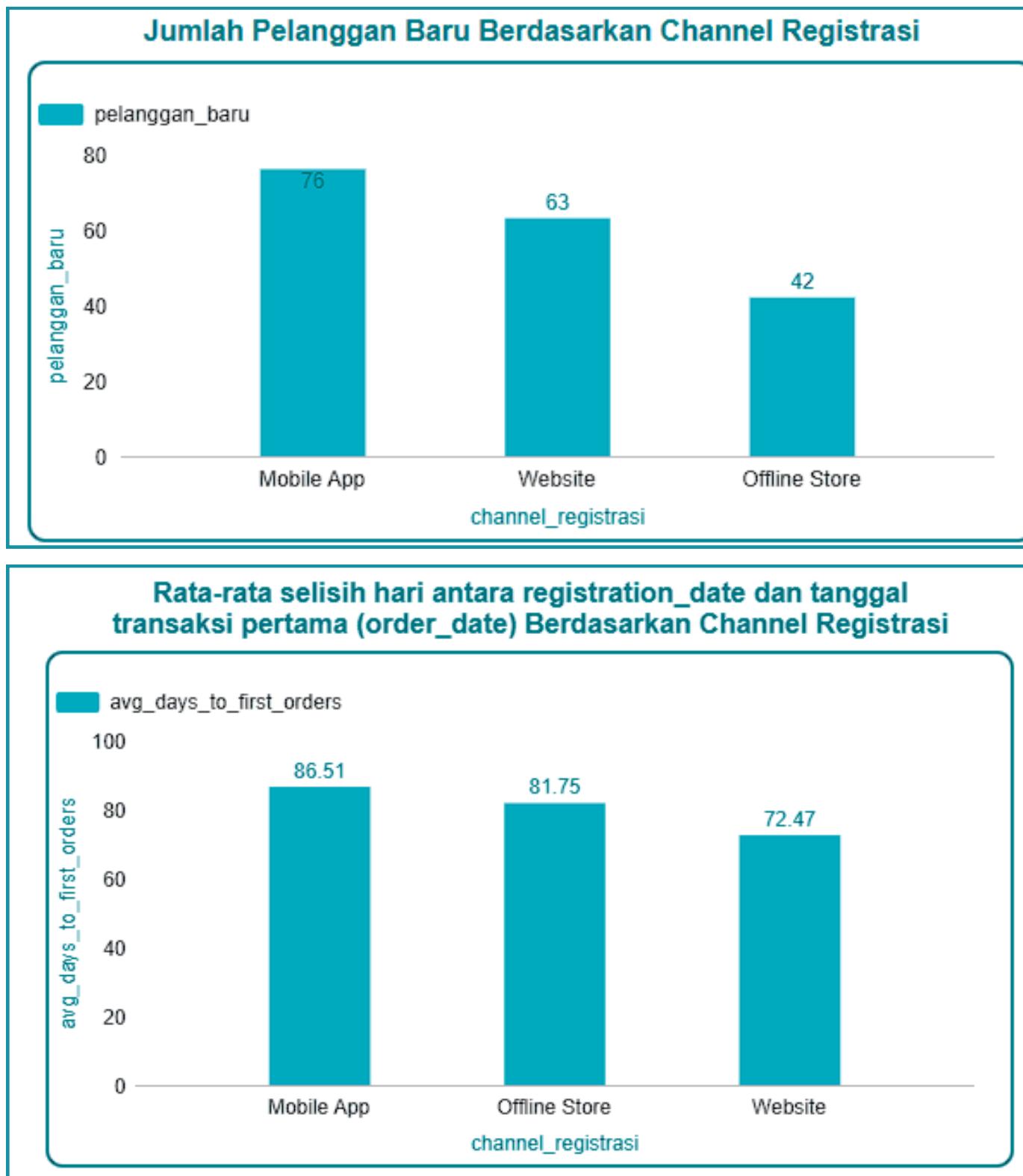
- Jumlah pelanggan baru cukup stabil (sekitar 7-10 di awal tahun, turun jadi 2-4 di akhir tahun).
- Waktu ke transaksi pertama relatif lebih cepat dibanding App (sering di bawah 90 hari, bahkan ada yang 43-55 hari di Okt-Nov).
- Interpretasi: Walaupun akuisisi tidak sebesar Mobile App, customer dari Website lebih cepat dikonversi, funnel website lebih efisien.

#### Offline Store

- Volume pelanggan baru paling kecil dan fluktuatif (kadang cuma 1-2 orang per bulan).
- Tapi variabilitas tinggi di waktu order pertama: Ada bulan yang sangat lama (Mei = 151 hari, Juli = 152,5 hari). Ada juga yang sangat cepat (Desember = 6 hari saja).
- Interpretasi: Experience offline store sangat tergantung situasi, perlu standarisasi proses pendaftaran dan aktivasi supaya konsisten.

# Visualisasi Data

## SOAL 5



### Rekomendasi:

Fokus Akuisisi → Mobile App & Website

- Mobile App : kuat untuk menarik banyak user baru, tapi perlu strategi agar mereka lebih cepat order.
- Website : lebih kecil volume, tapi konversi lebih cepat. Bisa dijadikan jalur utama untuk kampanye cepat konversi.

Percepat Konversi Mobile App

- Gunakan push notification, onboarding, dan voucher khusus first order untuk mempercepat pembelian pertama.
- Analisis UX App (apakah ada hambatan di checkout?)

Manfaatkan Website untuk Kampanye Cepat

- Karena pelanggan website lebih cepat konversi, jalankan promo “Flash Sale” atau “First Purchase Discount” di Website.

Optimalkan Offline Store

- Jumlah pelanggan baru kecil : lebih cocok dijadikan channel retensi atau cross-sell.
- Bisa digabungkan dengan program membership agar pelanggan yang datang langsung tetap aktif secara online.

Evaluasi Kampanye April & Akhir Tahun

- Ada bukti kampanye/seasonal promo mempercepat konversi, strategi tersebut bisa diulang di bulan lain.
- Pertimbangkan replikasi promo serupa di kuartal lain.

# RINGKASAN

- Penjualan menunjukkan fluktuasi sepanjang tahun dengan titik terendah pada bulan Juli–Maret dan puncak di Desember. Hal ini menegaskan adanya pola musiman yang dipengaruhi liburan dan promo akhir tahun. Strategi pemasaran perlu diperkuat pada Q1–Q2 untuk menahan tren penurunan, sementara kapasitas stok dan distribusi harus ditingkatkan menjelang akhir tahun guna mengantisipasi lonjakan permintaan.
- Dari sisi kategori, Food & Beverage dan Fashion & Footwear menjadi penyumbang terbesar meski fluktuatif, sementara Books & Stationery, Home Decor, dan Kitchen Appliances menunjukkan tren positif yang dapat dipacu lebih agresif. Sebaliknya, kategori seperti Sports Equipment, Toys & Games, dan Health & Wellness mengalami penurunan dan memerlukan strategi promosi, bundling, atau kolaborasi untuk memulihkannya. Kategori dengan volume rendah seperti Electronics dan Gaming Gear membutuhkan diferensiasi produk agar lebih menarik.
- Berdasarkan channel, performa Offline Store sangat kuat di periode tertentu (terutama Desember), sehingga perlu kampanye musiman lebih dini. Website menunjukkan pertumbuhan stabil, terutama di paruh akhir tahun, menandakan perbaikan strategi digital yang bisa diperkuat dengan SEO, personalisasi, dan remarketing. App Store mengalami fluktuasi tajam dengan masalah konsistensi revenue, sehingga diperlukan program loyalitas dan optimasi promosi di bulan-bulan lemah.
- Dalam hal funnel, Website menjadi penyumbang traffic terbesar tetapi memiliki conversion rate terendah, sedangkan Play Store mencatatkan conversion rate tertinggi. Hal ini mengindikasikan bahwa strategi lintas channel dapat dioptimalkan: website sebagai akuisisi traffic dan aplikasi sebagai jalur konversi.
- Dari sisi akuisisi pelanggan baru, Mobile App unggul dalam jumlah registrasi, namun waktu menuju transaksi pertama sangat lama, menunjukkan perlunya penguatan onboarding, reminder, dan promo first order. Website walaupun jumlah pelanggan baru lebih sedikit, lebih cepat menghasilkan transaksi, sehingga cocok untuk kampanye cepat konversi. Offline Store memiliki volume kecil namun hasil sangat variatif, sehingga lebih efektif difokuskan untuk retensi dan cross-sell melalui integrasi membership.
- Secara keseluruhan, fokus utama adalah memperkuat strategi promosi di Q1–Q2, menjaga stok di Q4, mengoptimalkan kategori unggulan, memulihkan kategori menurun, serta memanfaatkan sinergi lintas channel untuk meningkatkan konversi dan percepatan transaksi pelanggan baru.

# SQL EXERCISE

[Link Query Final Project SQL - Gaberia Sinaga](#)



**THANK YOU**  
**MySkill**

DISUSUN OLEH : GABERIA SINAGA- BATCH 23

