

MYSKILL

FINAL PROJECT

DATA ANALYSIS USING SQL IN POSTGRESQL

Mochamad Syahrizal

MySkill Final Project Data Analysis Batch 18



TABLE OF CONTENTS



Data Understanding



Problem Statements



Analysis & Insight





DATA UNDERSTANDING

Berikut dataset yang akan digunakan:

- 1.order_detail
- 2.sku_detail
- 3.customer_detail
- 4.payment_detail

Data yang digunakan adalah data yang berasal dari Tokopedia (bukan data sesungguhnya).

DATA OVERVIEW

Secara keseluruhan, dataset ini memberikan gambaran lengkap tentang pelanggan, pesanan, produk, dan metode pembayaran yang digunakan, memungkinkan analisis mendalam terkait perilaku pelanggan, preferensi produk, dan tren pembayaran.





DATA UNDERSTANDING

Dataset

order_detail:

- | | |
|--------------------|--|
| 1. id | → angka unik dari order / <i>id_order</i> |
| 2. customer_id | → angka unik dari pelanggan |
| 3. order_date | → tanggal saat dilakukan transaksi |
| 4. sku_id | → angka unik dari produk (sku adalah <i>stock keeping unit</i>) |
| 5. price | → harga yang tertera pada tagging harga |
| 6. qty_ordered | → jumlah barang yang dibeli oleh pelanggan |
| 7. before_discount | → nilai harga total dari produk (<i>price * qty_ordered</i>) |
| 8. discount_amount | → nilai diskon product total |
| 9. after_discount | → nilai harga total produk ketika sudah dikurangi dengan diskon |
| 10. is_gross | → menunjukkan pelanggan belum membayar pesanan |
| 11. is_valid | → menunjukkan pelanggan sudah melakukan pembayaran |
| 12. is_net | → menunjukkan transaksi sudah selesai |
| 13. payment_id | → angka unik dari metode pembayaran |



DATA UNDERSTANDING

Dataset

sku_detail:

- | | |
|---------------|---|
| 1. id | → angka unik dari produk (dapat digunakan untuk <i>key</i> saat join) |
| 2. sku_name | → nama dari produk |
| 3. base_price | → harga barang yang tertera pada tagging harga / <i>price</i> |
| 4. cogs | → <i>cost of goods sold</i> / total biaya untuk menjual 1 produk |
| 5. category | → kategori produk |

customer_detail:

- | | |
|--------------------|---|
| 1. id | → angka unik dari pelanggan |
| 2. registered_date | → tanggal pelanggan mulai mendaftarkan diri sebagai anggota |

Payment_detail:

- | | |
|-------------------|-------------------------------------|
| 1. id | → angka unik dari metode pembayaran |
| 2. payment_method | → metode pembayaran yang digunakan |



DATA UNDERSTANDING

COLUMN DATA TYPE

 customer_detail

General Columns Advanced Constraints Parameters

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/
 	id	text	▾
 	registered_date	date	▾

 payment_detail

General Columns Advanced Constraints Parameters

Inherited from table(s) Select to inherit from...

Columns

	Name	Data type	Length/
 	id	integer	▾
 	payment_method	text	▾



DATA UNDERSTANDING

COLUMN DATA TYPE

 **order_detail**

General Columns Advanced Constraints Parameters Security

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision
id	text	▾
customer_id	text	▾
order_date	date	▾
sku_id	text	▾
price	integer	▾
qty_ordered	integer	▾
before_discount	double precision	▾
discount_amount	double precision	▾
after_discount	double precision	▾
is_gross	integer	▾
is_valid	integer	▾
is_net	integer	▾
payment_id	integer	▾

i ?

 **sku_detail**

General Columns Advanced Constraints Parameters

Inherited from table(s) Select to inherit from...

Columns

Name	Data type	Length/Precision
id	text	▾
sku_name	text	▾
base_price	double precision	▾
cogs	double precision	▾
category	text	▾



DATA UNDERSTANDING

Table Schema

- Tabel **customer_detail** berisi informasi pelanggan seperti ID dan tanggal registrasi.
- Tabel **order_detail** menyimpan data transaksi pesanan, menghubungkan pelanggan dengan produk yang dibeli, serta detail harga dan jumlah pesanan.
- Tabel **sku_detail** mencakup informasi tentang produk yang dijual, termasuk nama produk, harga dasar, dan kategori.
- Tabel **payment_detail** mencatat metode pembayaran yang digunakan dalam transaksi

Skema Tabel

order_detail
id
customer_id
order_date
sku_id
price
qty_ordered
before_discount
discount_amount
after_discount
is_gross
is_valid
is_net
payment_id

customer_detail
id
registered_date

sku_detail
id
sku_name
base_price
cogs
category

payment_detail
id
payment_method



DATA UNDERSTANDING

INITIAL DATA INSPECTION

sku_detail

123 `SELECT * FROM sku_detail LIMIT 10;`

Data Output Messages Notifications

SQL

	<code>id</code> text	<code>sku_name</code> text	<code>base_price</code> double precision	<code>cogs</code> double precision	<code>category</code> text
1	P798444	AT-FSM-35	57631.7	46052	Kids & Baby
2	P938347	AYS_Haier-18HNF	3931789.26	3499256	Appliances
3	P826364	Atalian _DV206A-Brown-41	324597	243426	Men Fashion
4	P467533	Darul_Sakoon_Food_Bundle	2870.42	2378	Superstore
5	P229955	HP_15AY-15-Ay072NIA-ci3	2265625	1631250	Computing
6	P985828	UnzeLondon_GS4666-9	202855	172376	Men Fashion
7	P381119	UnzeLondon_GS5348-10	202855	139954	Men Fashion
8	P499423	UnzeLondon_GS5348-9	202855	158224	Men Fashion
9	P672852	Paramount_97811076773...	182856.6	138968	School & Education
10	P447157	Paramount_97811076976...	182856.6	153584	School & Education

Total rows: 10 of 10 Query complete 00:00:00.101

payment_detail

121 `SELECT * FROM payment_detail LIMIT 10;`

Data Output Messages Notifications

SQL

	<code>id</code> integer	<code>payment_method</code> text
1	1	cod
2	2	jazzvoucher
3	3	customercredit
4	4	Payaxis
5	5	jazzwallet
6	6	easypay_voucher
7	7	Easypay
8	8	ublcreditcard
9	9	mygateway
10	10	mcblite

Keterangan : Mengecek dan melihat sekilas bentuk data dalam table



DATA UNDERSTANDING

INITIAL DATA INSPECTION

order_detail

```

118
119  SELECT * FROM order_detail LIMIT 10;
120

```

Data Output Messages Notifications

SQL

	id text	customer_id text	order_date date	sku_id text	price integer	qty_ordered integer	before_discount double precision	discount_amount double precision	after_discount double precision	is_gross integer	is_valid integer	is_net integer	paym integer
1	ODR9939707760w	C713589L	2021-11-19	P858068	26100	200	5220000	2610000	2610000	1	1	0	
2	ODR7448356649d	C551551L	2021-11-19	P886455	1971942	5	9859710	2464927.5	7394782.5	1	0	0	
3	ODR4011281866z	C685596L	2021-11-25	P678648	7482000	1	7482000	2065344.62	5416655.38	1	0	0	
4	ODR3378927994s	C830683L	2021-11-22	P540013	3593680	1	3593680	1455440.4	2138239.6	1	1	1	
5	ODR4904430099k	C191766L	2021-11-21	P491032	4413220	1	4413220	1059172.8	3354047.2	1	1	1	
6	ODR7618778722h	C299859L	2021-11-20	P886455	1971942	2	3943884	985971	2957913	1	0	0	
7	ODR7610732813d	C313534L	2022-12-01	P849301	2697000	1	2697000	809100	1887900	1	1	1	
8	ODR4415476736l	C215074L	2022-12-01	P918122	2533672	1	2533672	760101.6	1773570.4	1	1	0	
9	ODR6474755771h	C368335L	2021-11-22	P240605	2479790	1	2479790	743937	1735853	1	0	0	
10	ODR9699658949w	C246762L	2022-05-21	P168052	6487648	1	6487648	638000	5849648	1	0	0	

Total rows: 10 of 10 | Query complete 00:00:00.115 | Ln 119, Col 37

customer_detail

```

115  -- cek overview data --
116
117  SELECT * FROM customer_detail LIMIT 10;
118

```

Data Output Messages Notifications

SQL

	id text	registered_date date
1	C996508L	2021-07-10
2	C180415L	2021-07-18
3	C535451L	2021-07-23
4	C177843L	2021-07-12
5	C951682L	2021-07-27
6	C483469L	2021-07-26
7	C346553L	2021-07-11
8	C631361L	2021-07-15
9	C534453L	2021-07-18
10	C404192L	2021-07-06

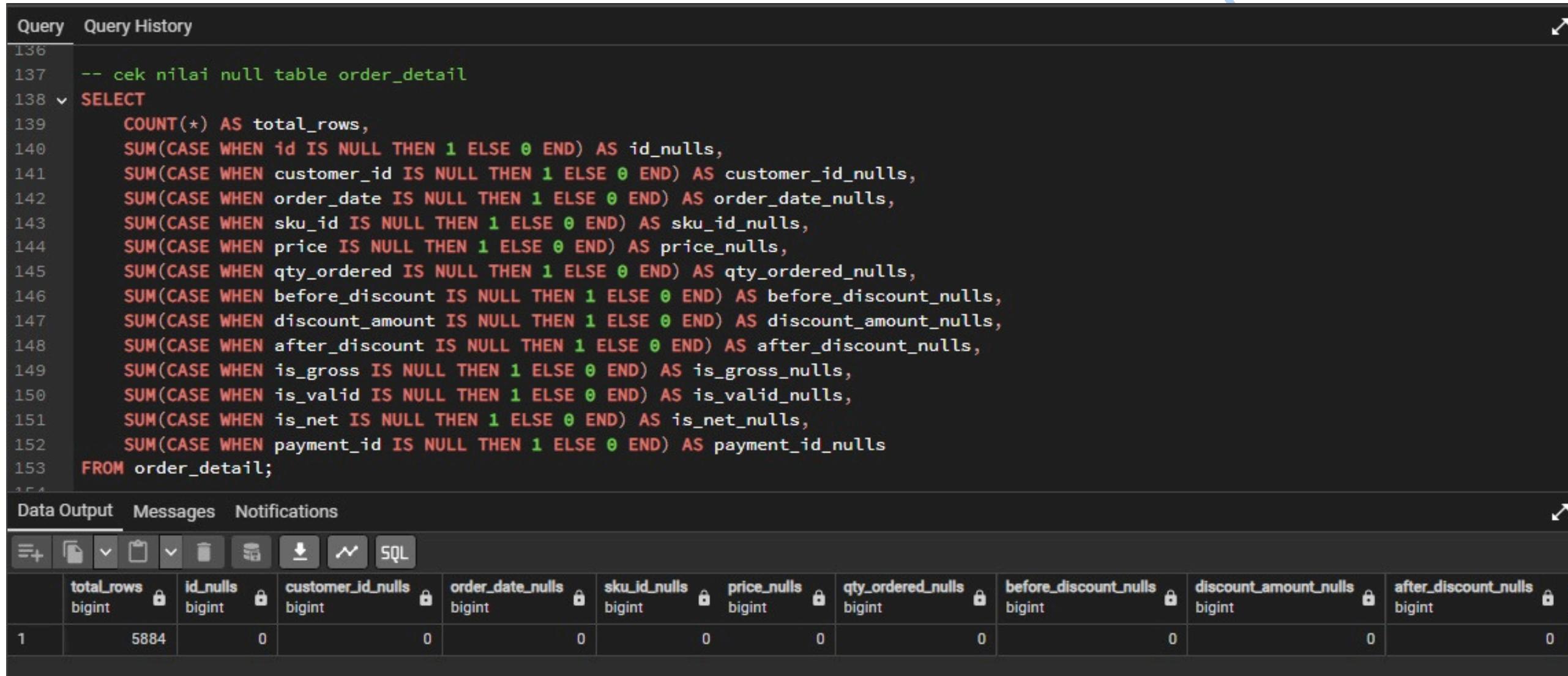
Keterangan : Mengecek dan melihat sekilas bentuk data dalam table



DATA UNDERSTANDING

CHECK NULL VALUE

order_detail



The screenshot shows a SQL query results window. The top section displays the SQL code used to check for null values:

```
136  
137 -- cek nilai null table order_detail  
138 SELECT  
139     COUNT(*) AS total_rows,  
140     SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS id_nulls,  
141     SUM(CASE WHEN customer_id IS NULL THEN 1 ELSE 0 END) AS customer_id_nulls,  
142     SUM(CASE WHEN order_date IS NULL THEN 1 ELSE 0 END) AS order_date_nulls,  
143     SUM(CASE WHEN sku_id IS NULL THEN 1 ELSE 0 END) AS sku_id_nulls,  
144     SUM(CASE WHEN price IS NULL THEN 1 ELSE 0 END) AS price_nulls,  
145     SUM(CASE WHEN qty_ordered IS NULL THEN 1 ELSE 0 END) AS qty_ordered_nulls,  
146     SUM(CASE WHEN before_discount IS NULL THEN 1 ELSE 0 END) AS before_discount_nulls,  
147     SUM(CASE WHEN discount_amount IS NULL THEN 1 ELSE 0 END) AS discount_amount_nulls,  
148     SUM(CASE WHEN after_discount IS NULL THEN 1 ELSE 0 END) AS after_discount_nulls,  
149     SUM(CASE WHEN is_gross IS NULL THEN 1 ELSE 0 END) AS is_gross_nulls,  
150     SUM(CASE WHEN is_valid IS NULL THEN 1 ELSE 0 END) AS is_valid_nulls,  
151     SUM(CASE WHEN is_net IS NULL THEN 1 ELSE 0 END) AS is_net_nulls,  
152     SUM(CASE WHEN payment_id IS NULL THEN 1 ELSE 0 END) AS payment_id_nulls  
153 FROM order_detail;
```

The bottom section shows the resulting data output:

	total_rows	id_nulls	customer_id_nulls	order_date_nulls	sku_id_nulls	price_nulls	qty_ordered_nulls	before_discount_nulls	discount_amount_nulls	after_discount_nulls	is_gross_nulls	is_valid_nulls	is_net_nulls	payment_id_nulls
1	5884	0	0	0	0	0	0	0	0	0	0	0	0	0

Keterangan : Mengecek apakah ada nilai null di setiap kolom dalam table, jika ada satupun nilai null, maka akan bernilai 1 di kolom pengecekan. **Pada hasil di atas, tidak ditemukan nilai null**



DATA UNDERSTANDING

CHECK NULL VALUE

customer_detail

```
125
126 -- mengecek nilai null --
127
128 -- cek nilai null table customer_detail
129 SELECT
130     COUNT(*) AS total_rows,
131     SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS id_nulls,
132     SUM(CASE WHEN registered_date IS NULL THEN 1 ELSE 0 END) AS registered_date_nulls
133 FROM customer_detail;
134
135
```

Data Output Messages Notifications

total_rows id_nulls registered_date_nulls

	total_rows	id_nulls	registered_date_nulls
1	3998	0	0

Keterangan : Mengecek apakah ada nilai null di setiap kolom dalam table, jika ada satupun nilai null, maka akan bernilai 1 di kolom pengecekan. **Pada hasil di atas, tidak ditemukan nilai null**



DATA UNDERSTANDING

CHECK NULL VALUE

payment_detail

```
155
156  -- cek nilai null table payment_detail
157 v SELECT
158      COUNT(*) AS total_rows,
159      SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS id_nulls,
160      SUM(CASE WHEN payment_method IS NULL THEN 1 ELSE 0 END) AS payment_method_nulls
161  FROM payment_detail;
162
163
```

Data Output Messages Notifications

total_rows id_nulls payment_method_nulls

	total_rows bigint	id_nulls bigint	payment_method_nulls bigint
1	16	0	0

Keterangan : Mengecek apakah ada nilai null di setiap kolom dalam table, jika ada satupun nilai null, maka akan bernilai 1 di kolom pengecekan. **Pada hasil di atas, tidak ditemukan nilai null**



DATA UNDERSTANDING

CHECK NULL VALUE

sku_detail

```
163  
164  -- cek nilai null table sku_detail  
165  SELECT  
166      COUNT(*) AS total_rows,  
167      SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS id_nulls,  
168      SUM(CASE WHEN sku_name IS NULL THEN 1 ELSE 0 END) AS sku_name_nulls,  
169      SUM(CASE WHEN base_price IS NULL THEN 1 ELSE 0 END) AS base_price_nulls,  
170      SUM(CASE WHEN cogs IS NULL THEN 1 ELSE 0 END) AS cogs_nulls,  
171      SUM(CASE WHEN category IS NULL THEN 1 ELSE 0 END) AS category_nulls  
172  FROM sku_detail;  
173
```

Data Output Messages Notifications

SQL

	total_rows	id_nulls	sku_name_nulls	base_price_nulls	cogs_nulls	category_nulls
	bigint	bigint	bigint	bigint	bigint	bigint
1	3206	0	0	0	0	0

Keterangan : Mengecek apakah ada nilai null di setiap kolom dalam table, jika ada satupun nilai null, maka akan bernilai 1 di kolom pengecekan. **Pada hasil di atas, tidak ditemukan nilai null**



DATA UNDERSTANDING

CHECK DUPLICATES

customer_detail

```
175
176  -- mengecek duplicate --
177
178  -- cek duplicate customer_detail
179 ✓ SELECT id, registered_date, COUNT(*)
180   FROM customer_detail
181   GROUP BY id, registered_date
182   HAVING COUNT(*) > 1;
183
184
```

Data Output Messages Notifications

SQL

	id text	registered_date date	count bigint

Keterangan : Mengecek apakah ada nilai baris yang sama dengan nilai baris lainnya (duplicate). Jika tidak ada baris yang duplicate, maka hasil query tidak akan memunculkan apapun. **Dalam table ini tidak ditemukan duplicate.**



DATA UNDERSTANDING

CHECK DUPLICATES

payment_detail

```
212  
213    -- cek duplicate payment_detail  
214  
215 ✓ SELECT id, payment_method, COUNT(*)  
216     FROM payment_detail  
217     GROUP BY id, payment_method  
218     HAVING COUNT(*) > 1;  
219  
...
```

Data Output Messages Notifications

SQL

	id integer 	payment_method text 	count bigint 

Keterangan : Mengecek apakah ada nilai baris yang sama dengan nilai baris lainnya (duplicate). Jika tidak ada baris yang duplicate, maka hasil query tidak akan memunculkan apapun. **Dalam table ini tidak ditemukan duplicate.**



DATA UNDERSTANDING

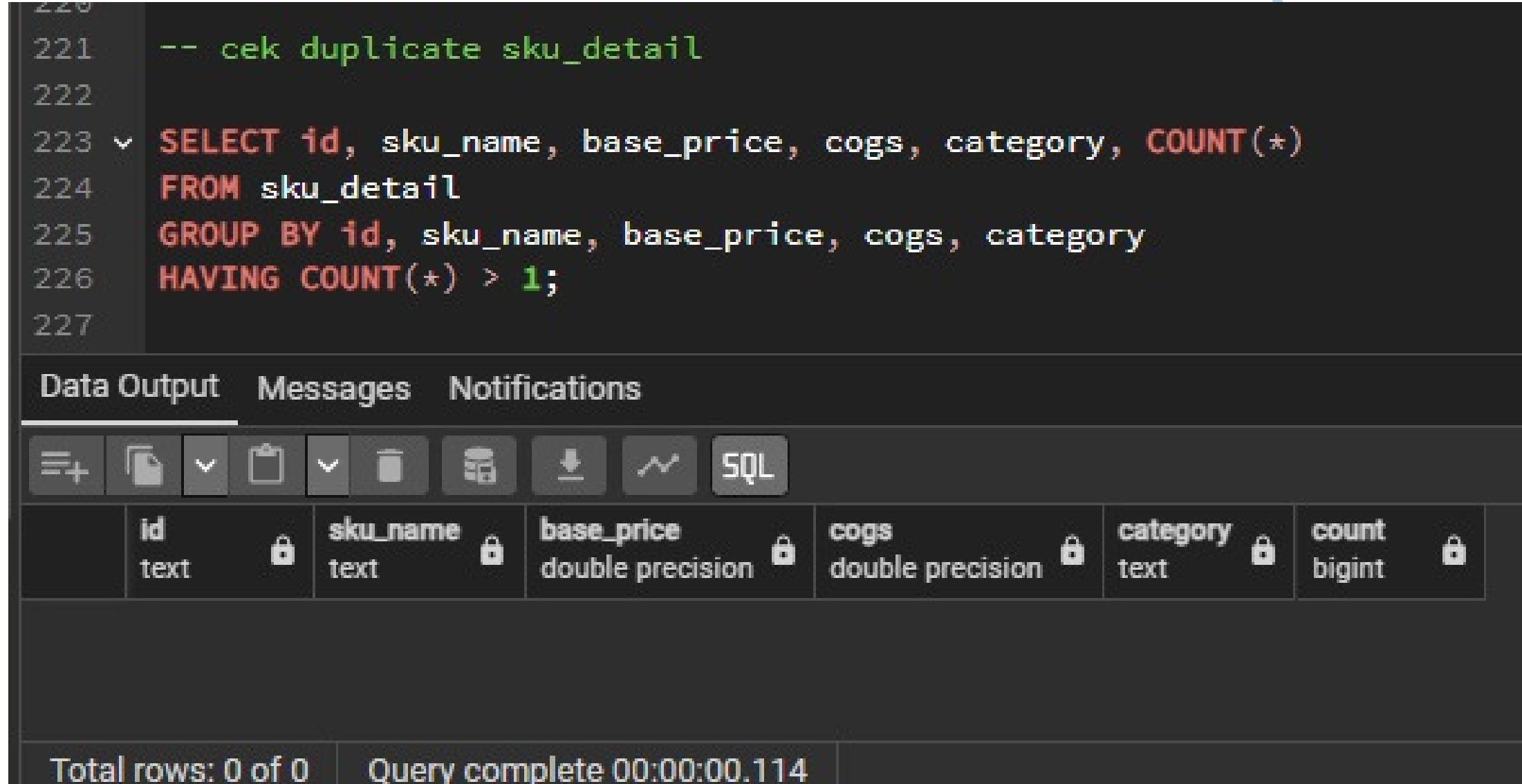
CHECK DUPLICATES

sku_detail

```
220  
221 -- cek duplicate sku_detail  
222  
223 ✓ SELECT id, sku_name, base_price, cogs, category, COUNT(*)  
224   FROM sku_detail  
225   GROUP BY id, sku_name, base_price, cogs, category  
226   HAVING COUNT(*) > 1;  
227
```

Data Output Messages Notifications

Total rows: 0 of 0 | Query complete 00:00:00.114



Keterangan : Mengecek apakah ada nilai baris yang sama dengan nilai baris lainnya (duplicate). Jika tidak ada baris yang duplicate, maka hasil query tidak akan memunculkan apapun. **Dalam table ini tidak ditemukan duplicate.**



DATA UNDERSTANDING

CHECK DUPLICATES

order_detail

The screenshot shows a SQL query being run against a database table named 'order_detail'. The query is designed to check for duplicate entries. It selects various columns including customer_id, order_date, sku_id, price, qty_ordered, before_discount, discount_amount, after_discount, is_gross, is_valid, is_net, payment_id, and a COUNT(*) column. The results are grouped by these columns. A HAVING clause at the end filters the results to only show rows where the count is greater than 1, indicating duplicates.

```
184  
185 -- cek duplicate order_detail  
186 SELECT  
187     id, customer_id,  
188     order_date, sku_id, price,  
189     qty_ordered, before_discount,  
190     discount_amount, after_discount,  
191     is_gross, is_valid,  
192     is_net, payment_id,  
193     COUNT(*)  
194 FROM order_detail  
195 GROUP BY  
196     id,  
197     customer_id,  
198     order_date,  
199     sku_id,  
200     price,  
201     qty_ordered,  
202     before_discount,  
203     discount_amount,  
204     after_discount,  
205     is_gross,  
206     is_valid,  
207     is_net,  
208     payment_id  
209     HAVING COUNT(*) > 1;  
210  
211
```

Data Output Messages Notifications

customer_id	order_date	sku_id	price	qty_ordered	before_discount	discount_amount	after_discount	is_gross	is_valid	is_net	payment_id	count

Keterangan : Mengecek apakah ada nilai baris yang sama dengan nilai baris lainnya (duplicate). Jika tidak ada baris yang duplicate, maka hasil query tidak akan memunculkan apapun.

Dalam table ini tidak ditemukan duplicate.

PROBLEM STATEMENTS?

1

Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after_discount) paling besar?

2

Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?

3

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022

4

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022

5

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya. Samsung, Apple, Sony, Huawei dan Lenovo

ANALYSIS & INSIGHT



1

Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after_discount) paling besar?

QUERY

```
Query Query History
1 -- Untuk menjawab question 1 :
2 SELECT
3     EXTRACT(MONTH FROM order_date) AS Transaction_Month,
4     SUM(after_discount) AS Total_Transaction
5 FROM order_detail
6 WHERE
7     EXTRACT(YEAR FROM order_date) = 2021
8     AND is_valid = 1
9 GROUP BY
10    1
11 ORDER BY
12    2 DESC
13 LIMIT 7
14 ;
```

Query ini digunakan untuk menghitung total transaksi bulanan selama tahun 2021 dari tabel order_detail. Berikut adalah penjelasannya:

Fungsi SELECT dan Agregat:

- **SELECT**: Memilih kolom atau nilai yang akan ditampilkan
- **EXTRACT(MONTH FROM order_date) AS Transaction_Month**: Mengambil nilai bulan dari kolom order_date dan menggantinya dengan alias Transaction_Month.
- **SUM(after_discount) AS Total_Transaction**: Menjumlahkan nilai di kolom after_discount untuk setiap bulan, lalu mengaliasinya sebagai Total_Transaction.

Sumber Data:

- **FROM order_detail**: Menentukan bahwa data yang diambil berasal dari tabel order_detail.

Kondisi di Klause WHERE:

- **WHERE EXTRACT(YEAR FROM order_date) = 2021 AND is_valid = 1**: Membatasi hasil hanya pada transaksi yang terjadi di tahun 2021 dan telah divalidasi (dengan is_valid = 1 menunjukkan pembayaran telah dilakukan).

Fungsi GROUP BY dan Filter:

- **GROUP BY 1**: Mengelompokkan data berdasarkan bulan (Transaction_Month) yang diambil dari kolom order_date.
- **ORDER BY 2 DESC**: Mengurutkan data berdasarkan total transaksi (after_discount) dari yang terbesar ke yang terkecil.
- **LIMIT 7**: Menampilkan hanya 7 hasil teratas.



ANALYSIS & INSIGHT



1

Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after_discount) paling besar?

QUERT RESULT

	transaction_month numeric	total_transaction double precision
1	8	227862744
2	12	217309963
3	10	207603259.9
4	11	180396009.73999998
5	7	148007735
6	9	145943335.2
7	6	43154552

INSIGHT

Berdasarkan data yang diperoleh, **bulan Agustus (bulan ke-8) tahun 2021** mencatat total transaksi tertinggi, yaitu **sebesar 227.862.744 atau 17.22% dari total transaksi 2021 (1.322.726.912.)**

RECOMMENDATIONS

- **Mengidentifikasi Faktor Peningkatan:** Penting untuk memahami apa yang menyebabkan lonjakan transaksi di bulan Agustus 2021. Apakah kampanye promosi yang dilakukan tim marketing berhasil menarik lebih banyak pelanggan? Apakah ada faktor eksternal seperti tren musiman, peringatan hari besar, atau perilisan produk baru yang berperan? Mengetahui penyebabnya dapat membantu dalam merancang strategi pemasaran yang lebih efektif di masa mendatang.
- **Evaluasi Performa Marketing Channel:** Tinjau efektivitas saluran pemasaran yang digunakan pada bulan Agustus. Identifikasi kanal mana yang berkontribusi paling signifikan terhadap peningkatan transaksi, sehingga bisa dioptimalkan untuk kampanye mendatang.
- **Segmentasi Pelanggan:** Analisis data pelanggan lebih lanjut untuk memahami apakah peningkatan transaksi berasal dari segmen pelanggan baru atau pelanggan yang kembali melakukan pembelian. Ini akan membantu dalam personalisasi strategi pemasaran yang lebih tepat sasaran.

ANALYSIS & INSIGHT



2

Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?

QUERY

```
Query Query History
16
17 -- Untuk menjawab question 2
18 v
19 SELECT
20     sd.category AS Category_Product,
21     SUM(od.after_discount) AS Total_Transaction
22 FROM
23     order_detail AS od
24     LEFT JOIN sku_detail AS sd ON od.sku_id = sd.id
25 WHERE
26     EXTRACT(YEAR FROM od.order_date) = 2022
27     AND is_valid = 1
28 GROUP BY 1
29 ORDER BY 2 DESC
30 LIMIT 7;
31
```

Query ini digunakan untuk menghitung total transaksi berdasarkan kategori produk selama tahun 2022 dari gabungan tabel order_detail dan sku_detail. Berikut penjelasan dari query tersebut:

Fungsi SELECT dan Agregasi:

- **SELECT sd.category AS Category_Product:** Mengambil kolom category dari tabel sku_detail dan menggantinya dengan alias Category_Product.
- **SUM(od.after_discount) AS Total_Transaction:** Menjumlahkan nilai dari kolom after_discount di tabel order_detail dan mengaliskan hasilnya sebagai Total_Transaction.

Sumber Data & JOIN:

- **FROM order_detail AS od:** Menggunakan tabel order_detail sebagai sumber data utama dan memberikan alias od.
- **LEFT JOIN sku_detail AS sd ON od.sku_id = sd.id:** Menggabungkan tabel order_detail dan sku_detail menggunakan kolom sku_id dari order_detail dan id dari sku_detail sebagai penghubung.

Kondisi WHERE:

- **WHERE EXTRACT(YEAR FROM order_date) = 2022 AND is_valid = 1:** Membatasi data yang ditampilkan hanya pada transaksi yang terjadi di tahun 2022 dan yang sudah valid (ditandai dengan is_valid = 1).

Fungsi GROUP BY dan Filter:

- **GROUP BY 1:** Mengelompokkan data berdasarkan kolom category dari tabel sku_detail, yang telah diberi alias Category_Product.
- **ORDER BY 2 DESC:** Mengurutkan hasil berdasarkan total transaksi (after_discount), dari yang tertinggi ke yang terendah.
- **LIMIT 7:** Membatasi hasil yang ditampilkan hanya sebanyak 7 baris teratas.



ANALYSIS & INSIGHT



2

Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?

INSIGHT

QUERT RESULT

	Data Output	Messages	Notifications
	category_product text	total_transaction double precision	
1	Mobiles & Tablets	918451576	
2	Entertainment	365344148.9	
3	Appliances	316358100	
4	Computing	214028543.4	
5	Men Fashion	135588253	
6	Women Fashion	93014970.62	
7	Home & Living	79483716.2	

Berdasarkan data yang diperoleh, kategori **Mobile & Tablets** mencatatkan nilai transaksi tertinggi sepanjang tahun 2022, dengan **total transaksi mencapai 918.451.576 atau 39.17% dari total transaksi tahun 2022 (2.344.848.413)**

RECOMMENDATIONS

- **Inovasi Produk dan Peluncuran Terbaru:** Melakukan riset pasar mendalam dan mengidentifikasi tren teknologi terkini dalam kategori Mobile & Tablets akan membantu perusahaan merancang produk baru dan inovatif yang dapat mempertahankan serta meningkatkan loyalitas pelanggan.
- **Kampanye Promosi Terarah:** Perusahaan dapat mengadakan kampanye promosi khusus untuk produk Mobile & Tablets, seperti diskon atau bundling dengan aksesoris, guna menarik lebih banyak pembeli potensial dan meningkatkan volume transaksi.
- **Kolaborasi dengan Brand Terkemuka:** Menjalin kemitraan dengan brand besar di sektor teknologi untuk menyediakan produk eksklusif atau edisi terbatas juga bisa menjadi strategi efektif dalam menarik perhatian pasar.
- **Analisis Segmentasi Pelanggan:** Lakukan segmentasi pelanggan untuk mengetahui preferensi pembeli produk Mobile & Tablets, sehingga kampanye pemasaran dapat dipersonalisasi dan lebih efektif dalam menarik minat pembeli di segmen tertentu.



ANALYSIS & INSIGHT



3

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022

QUERY

```
Query Query History
31
32 -- Menjawab soal 3 --
33
34 v WITH penjualan_tahunan AS (
35     SELECT
36         sd.category AS kategori,
37         SUM(CASE WHEN EXTRACT(YEAR from od.order_date) = 2021 then after_discount END) AS total_sales_2021,
38         SUM(CASE WHEN EXTRACT(YEAR from od.order_date) = 2022 then after_discount END) AS total_sales_2022,
39         SUM(CASE WHEN EXTRACT(YEAR from od.order_date) = 2022 then after_discount END)
40         -
41         SUM(CASE WHEN EXTRACT(YEAR from od.order_date) = 2021 then after_discount END) AS perbedaan
42     FROM order_detail AS od
43     LEFT JOIN sku_detail AS sd
44         ON od.sku_id = sd.id
45     WHERE
46         EXTRACT(YEAR from od.order_date) IN (2021,2022)
47         AND od.is_valid = 1
48     GROUP BY sd.category
49     ORDER BY perbedaan DESC
50 )
51
52     SELECT
53         kategori,
54         total_sales_2021,
55         total_sales_2022,
56         perbedaan,
57         CASE WHEN total_sales_2022 < total_sales_2021 THEN 'Penurunan'
58             ELSE 'Kenaikan'
59         END AS keterangan
60     FROM penjualan_tahunan
61 ;
62
```



ANALYSIS & INSIGHT



PENJELASAN QUERY

3

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022

Query ini digunakan untuk menghitung total penjualan tahunan untuk setiap kategori produk selama tahun 2022 dan 2021 dari tabel order_detail dan sku_detail yang telah digabungkan, kemudian membandingkan kategori produk mana yang mengalami peningkatan atau penurunan.

Common Table Expression (CTE):

- **WITH penjualan_tahunan AS (....)**: CTE digunakan untuk menyederhanakan query yang kompleks dan memudahkan jika kita ingin menggunakan kembali query yang ada di dalam CTE tersebut.

Fungsi SELECT dan Agregat:

- **SELECT sd.category**: Memilih kolom category dari tabel sku_detail.
- **SUM(CASE WHEN EXTRACT(YEAR FROM od.order_date) = '2021' THEN)**: Menghitung penjualan untuk tahun 2021 dan 2022, kemudian membandingkannya untuk mendapatkan nilai perbedaan antara kedua tahun tersebut.

Sumber Data dan Fungsi JOIN:

- **FROM order_detail AS od**: Memilih tabel order_detail sebagai sumber data, dan memberi alias od.

- **LEFT JOIN sku_detail sd ON od.sku_id = sd.id**: Menggabungkan tabel order_detail dan sku_detail menggunakan sku_id dari order_detail dan id dari sku_detail sebagai kunci utama.

Klausula WHERE:

- **WHERE EXTRACT(YEAR FROM od.order_date) IN ('2021', '2022') AND is_valid = 1**: Membatasi hasil hanya untuk tahun 2021 dan 2022, serta hanya untuk transaksi yang valid (is_valid = 1).

Fungsi GROUP BY dan Filter:

- **GROUP BY sd.category**: Mengelompokkan data berdasarkan kolom category dari tabel sku_detail.
- **ORDER BY perbedaan DESC**: Mengurutkan hasil berdasarkan perbedaan penjualan antara tahun 2021 dan 2022, dari yang terbesar ke yang terkecil.

Query:

- **SELECT * , CASE WHEN total_sales_2021 < total_sales_2022 THEN**: Jika penjualan tahun 2021 lebih kecil dibandingkan dengan 2022, tampilkan "Kenaikan"; jika tidak, tampilkan "Penurunan" di kolom Keterangan berdasarkan data dari CTE penjualan_tahunan.



ANALYSIS & INSIGHT



3

Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022. Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami penurunan nilai transaksi dari tahun 2021 ke 2022

QUERT RESULT

	kategori text	total_sales_2021 double precision	total_sales_2022 double precision	perbedaan double precision	keterangan text
1	Mobiles & Tablets	370606718	918451576	547844858	Kenaikan
2	Entertainment	162326357.4	365344148.9	203017791.49999997	Kenaikan
3	Appliances	218550177	316358100	97807923	Kenaikan
4	Men Fashion	58628198	135588253	76960055	Kenaikan
5	Computing	172878860	214028543.4	41149683.400000006	Kenaikan
6	Home & Living	45797873	79483716.2	33685843.2	Kenaikan
7	Health & Sports	33837965.6	54235579.6	20397614	Kenaikan
8	Women Fashion	84045961.4	93014970.62	8969009.219999999	Kenaikan
9	School & Education	11558982.4	17362465.3	5803482.9	Kenaikan
10	Superstore	28828088	32643266.519999996	3815178.519999996	Kenaikan
11	Soghaat	15056202.6	17658332	2602129.400000004	Kenaikan
12	Kids & Baby	23971057.8	25931276.84	1960219.039999999	Kenaikan
13	Beauty & Grooming	46047360	46211019.18	163659.179999997	Kenaikan
14	Books	10124596	6792519.2	-3332076.8	Penurunan
15	Others	40468515.74	21744646.02	-18723869.720000003	Penurunan

Total rows: 15 of 15 | Query complete 00:00:00.242

INSIGHT

Berdasarkan data yang didapat, **kategori yang menunjukkan peningkatan pada tahun 2022 di bandingkan 2021 meliputi Mobiles & Tablets, Entertainment, Appliances, Men Fashion, Computing, Home & Living, Health & Sport, Women Fashion, School & Education, Superstore, Soghaat, Kids & Baby, serta Beauty & Grooming.** Di sisi lain, **kategori yang mengalami penurunan adalah Others dan Books.**

RECOMMENDATIONS

- **Kategori yang mengalami peningkatan dapat dioptimalkan** dengan memperbaiki strategi promosi dan memperluas ragam penawaran produk, sehingga perusahaan dapat mempertahankan atau bahkan meningkatkan pendapatan yang diperoleh.
- **Menggali Penyebab Penurunan: Penting untuk mengidentifikasi faktor-faktor yang menyebabkan penurunan nilai transaksi pada kategori Others dan Books.** Dengan memahami penyebabnya, perusahaan dapat merumuskan strategi pemasaran yang lebih efektif serta meningkatkan kualitas produk untuk mengatasi penurunan tersebut.



ANALYSIS & INSIGHT



4

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022
(berdasarkan total unique order)

QUERY

```
63
64  -- Menjawab Soal No 4 --
65  SELECT
66      pd.payment_method AS Payment_Method,
67      COUNT(DISTINCT od.id) AS Total_Transaksi
68  FROM
69      order_detail AS od
70  LEFT JOIN payment_detail pd ON od.payment_id = pd.id
71  WHERE
72      EXTRACT(YEAR FROM order_Date) = 2022
73      AND is_valid = 1
74  GROUP BY 1
75  ORDER BY 2 DESC
76  LIMIT 5;
77
```



ANALYSIS & INSIGHT



PENJELASAN QUERY

4

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022
(berdasarkan total unique order)

Query ini digunakan untuk menghitung jumlah metode pembayaran yang digunakan selama tahun 2022 dari tabel `order_detail` dan `payment_detail` yang telah digabungkan. Berikut adalah penjelasan rinci dari query tersebut:

Fungsi SELECT dan Agregat:

- **SELECT pd.payment_method AS Payment_Method:** Mengambil kolom `payment_method` dari tabel `payment_detail` untuk ditampilkan.
- **COUNT(DISTINCT od.id) AS Total_Transaksi:** Menghitung jumlah pesanan dari kolom `id` di tabel `order_detail`, memastikan hanya menghitung nilai unik untuk menghindari kesalahan data akibat duplikasi.

Sumber Data dan Fungsi JOIN:

- **FROM order_detail od:** Menggunakan tabel `order_detail` sebagai sumber data dan memberikan alias `od`.
- **LEFT JOIN payment_detail pd ON od.payment_id = pd.id:** Menggabungkan tabel `order_detail` dan `payment_detail` menggunakan kolom `payment_id` dari `order_detail` dan `id` dari `payment_detail` sebagai kunci utama.

Klausa WHERE:

- **WHERE EXTRACT(YEAR FROM order_date) = 2022 AND is_valid = 1:** Membatasi hasil untuk hanya mencakup transaksi yang terjadi pada tahun 2022 dan yang telah divalidasi (`is_valid = 1`).

Fungsi GROUP BY dan Filter:

- **GROUP BY 1:** Mengelompokkan hasil berdasarkan kolom `payment_method` dari tabel `payment_detail`.
- **ORDER BY 2 DESC:** Mengurutkan hasil berdasarkan jumlah transaksi yang dihitung dari kolom `id` di tabel `order_detail`, dari yang terbesar ke yang terkecil.
- **LIMIT 5:** Membatasi hasil yang ditampilkan hanya sebanyak 5 baris teratas.

ANALYSIS & INSIGHT

4

Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022
(berdasarkan total unique order)

QUERT RESULT

Data Output			Messages	Notifications
	payment_method	total_transaksi		
1	cod	1809		
2	Payaxis	181		
3	customercredit	75		
4	Easypay	69		
5	jazzwallet	26		

Total rows: 5 of 5 Query complete 00:00:00.110

INSIGHT

Menurut data yang dikumpulkan, **metode pembayaran** yang paling umum digunakan oleh konsumen **sepanjang tahun 2022 adalah :**

1. cod: 1809 (86.56%)
2. Payaxis: 181 (8.67%)
3. customercredit: 75(3.59%)
4. Easypay: 69 (3.30%)
5. jazzwallet: 26 (1.24%)

RECOMMENDATIONS

- **Menganalisis penyebab mengapa metode pembayaran non-tunai** seperti Payaxis, Customer Credit, Easypay, dan Jazzwallet tidak menjadi pilihan utama bagi konsumen dibandingkan dengan COD (tunai). Apakah faktor efisiensi, kurangnya promosi dan manfaat, atau kurangnya pemahaman dari konsumen yang menjadi alasan utama rendahnya popularitas metode pembayaran non-tunai?
- **Melanjutkan dan memperbaiki strategi promosi** yang telah diterapkan untuk metode pembayaran COD.



ANALYSIS & INSIGHT



5

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya : 1. Samsung, 2. Apple, 3. Sony, 4. Huawei, 5. Lenovo

Query Query History

```
78
79  -- Menjawab Soal No 5 --
80  WITH
81      top_produk AS (
82          SELECT
83              CASE
84                  WHEN LOWER(sd.sku_name) LIKE '%samsung%' THEN 'samsung'
85                  WHEN LOWER(sd.sku_name) LIKE '%apple%'
86                  OR LOWER(sd.sku_name) LIKE '%iphone%'
87                  OR LOWER(sd.sku_name) LIKE '%macbook%' THEN 'apple'
88                  WHEN LOWER(sd.sku_name) LIKE '%sony%' THEN 'sony'
89                  WHEN LOWER(sd.sku_name) LIKE '%huawei%' THEN 'huawei'
90                  WHEN LOWER(sd.sku_name) LIKE '%lenovo%' THEN 'lenovo'
91              END AS nama_produk,
92              SUM(after_discount) AS nilai_transaksi
93          FROM
94              order_detail AS od
95              LEFT JOIN sku_detail AS sd ON od.sku_id = sd.id
96          WHERE
97              is_valid = 1
98          GROUP BY
99              1
100         ORDER BY
101             2 DESC
102     )
103    SELECT
104        *
105    FROM
106        top_produk
107    WHERE
108        nama_produk IS NOT NULL
109    ;
```



ANALYSIS & INSIGHT



PENJELASAN QUERY

5

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya : 1. Samsung, 2. Apple, 3. Sony, 4. Huawei, 5. Lenovo

Query ini berguna untuk menghitung jumlah total transaksi dari produk Samsung, Apple, Sony, Huawei, dan Lenovo yang ada di kolom sku_name, kemudian mengurutkannya dari yang tertinggi ke yang terendah.

Common Table Expression (CTE):

- **WITH top_produk AS (....)** : CTE berfungsi untuk mempermudah penulisan query yang rumit dan memungkinkan penggunaan ulang query yang ada dalam CTE.

Fungsi Select & Agregasi:

- **SELECT CASE WHEN LOWER(SD.sku_name) LIKE '%samsung%' THEN 'samsung' (.....)** : Menyeleksi data yang berisi nama produk seperti "Samsung," "Apple," "Sony," "Huawei," dan "Lenovo" di kolom sku_name.
- **SUM(after_discount) AS nilai_transaksi**: Menghitung total transaksi di kolom after_discount dan mengaliasinya sebagai Nilai_Transaksi.

Sumber Data & Fungsi Join:

- **FROM order_detail AS od**: Memilih tabel order_detail sebagai sumber data, lalu mengaliasinya sebagai od.

- **LEFT JOIN sku_detail AS sd ON od.sku_id = sd.id**: Digunakan untuk menggabungkan tabel order_detail dan sku_detail (dialiasi sebagai sd) dengan menggunakan kolom sku_id dari tabel order_detail dan kolom id dari sku_detail sebagai kunci utama.

Klausa Where:

- **WHERE is_valid = 1**: Digunakan untuk memfilter data hanya pada baris yang memiliki is_valid = 1 (pembayaran sudah dilakukan oleh pelanggan).

Fungsi Group By & Filter:

- **GROUP BY 1**: Mengelompokkan data berdasarkan nama produk (sku_name) dari tabel sku_detail.
- **ORDER BY 2 DESC**: Mengurutkan hasil berdasarkan kolom after_discount yang telah dijumlahkan, dari nilai terbesar hingga terkecil.

Query 2:

- **SELECT * FROM Top_Produk WHERE nama_produk IS NOT NULL**: Memilih semua kolom dari tabel Top_Produk (CTE) dan memfilter baris yang tidak memiliki nilai null di kolom nama_produk.



ANALYSIS & INSIGHT



5

Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya : 1. Samsung, 2. Apple, 3. Sony, 4. Huawei, 5. Lenovo

QUERT RESULT

	nama_produk	nilai_transaksi
1	samsung	588764148
2	apple	444855360
3	sony	63960718
4	huawei	63160260
5	lenovo	62379800.4

Total rows: 5 of 5 Query complete 00:00:00.237

INSIGHT

Urutan ke 5 produk berdasarkan nilai transaksi yaitu :

1. **Samsung: 588.764.148 (45.98%)**
2. **Apple: 444.855.360 (34.74%)**
3. **Sony: 63.960.718 (4.99%)**
4. **Huawei: 63.160.260 (4.93%)**
5. **Lenovo: 62.379.800,4 (4.86%)**

RECOMMENDATIONS

- **Analisis Sentimen Pelanggan:** Lakukan analisis data ulasan dan feedback pelanggan untuk mengidentifikasi kekuatan dan kelemahan produk. Dengan data tersebut, bisa memberikan rekomendasi kepada tim pemasaran atau produk terkait peningkatan fitur atau penanganan masalah yang sering dikeluhkan oleh pelanggan.
- **Diversifikasi Promosi Berdasarkan Produk:** Sarankan strategi promosi yang tersegmentasi berdasarkan performa dan demografi konsumen untuk setiap produk. Gunakan data penjualan sebelumnya untuk mengidentifikasi produk yang memerlukan lebih banyak dorongan melalui kampanye atau diskon khusus.



ANALYSIS & INSIGHT



CONCLUSION

Proyek Tokopedia ini berhasil memberikan insight yang mendalam mengenai transaksi penjualan selama tahun 2021 dan 2022. Dari proyek ini ditemukan bahwa Agustus 2021 merupakan bulan dengan transaksi tertinggi pada tahun tersebut, yang mungkin dapat dijadikan dasar perencanaan promosi musiman. Selain itu, kategori Mobile & Tablets terus mendominasi penjualan nilai transaksi tertinggi pada tahun 2022, menekankan kenaikan minat produk teknologi seiring waktu.

Dalam hal metode pembayaran, COD tetap menjadi favorit pelanggan, yang menunjukkan perlunya mempertahankan metode ini untuk memaksimalkan konversi penjualan. Sementara itu, evaluasi lebih lanjut terhadap performa produk seperti Samsung sebagai produk dengan nilai transaksi tertinggi perlu dipertahankan dan Lenovo sebagai produk dengan nilai transaksi terendah perlu dilakukan evaluasi untuk meningkatkan daya saing di pasar.

Dengan demikian, hasil proyek ini memberikan panduan yang jelas mengenai strategi produk, pemasaran, dan pengelolaan pembayaran di masa mendatang, sekaligus membuka peluang untuk pengoptimalan lebih lanjut melalui data yang lebih kaya dan analisis lanjutan, meskipun diperlukan adanya data yang lebih banyak dan beragam dari yang dihadirkan project ini.



THANK YOU!

MOCHAMAD SYAHRIZAL



[Mochamad Syahrizal](#)



[MochSyahrizal](#)