

# DATA ANALYST USING SQL

Kelompok 7

Learning With My Skill

FINAL PROJECT 17



# KELOMPOK 7

- Dita Nur Rahma Sari
- Ahmad Alfaridhi
- Ilham Fadillah
- Niro Adzikri
- Syauqi Ghufuran Lubis
- Anel Fuad Abiyyu
- Elizabeth Feodora
- lora luayya
- Naufal Syahfikri
- ST. Mega Dzulfahra Manguluang

# DATASET

Berikut dataset yang akan digunakan:

1. order\_detail
2. sku\_detail
3. customer\_detail
4. payment\_detail

## OVERVIEW:

Dataset digunakan untuk mendapatkan informasi data pelanggan serta mengoptimalkan strategi penjualan produk



# DATASET

## order\_detail:

- |                    |  |
|--------------------|--|
| 1. id              | → angka unik dari order / id_order                                       |
| 2. customer_id     | → angka unik dari pelanggan  |
| 3. order_date      | → tanggal saat dilakukan transaksi                                       |
| 4. sku_id          | → angka unik dari produk (sku adalah stock keeping unit)                 |
| 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 ( $\text{price} * \text{qty\_ordered}$ ) |
| 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                                      |

# DATASET

## sku\_detail:

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



# DATASET

## customer\_detail:

- 1. id → angka unik dari pelanggan
- 2. register\_date → tanggal pelanggan mulai mendaftarkan diri sebagai anggota

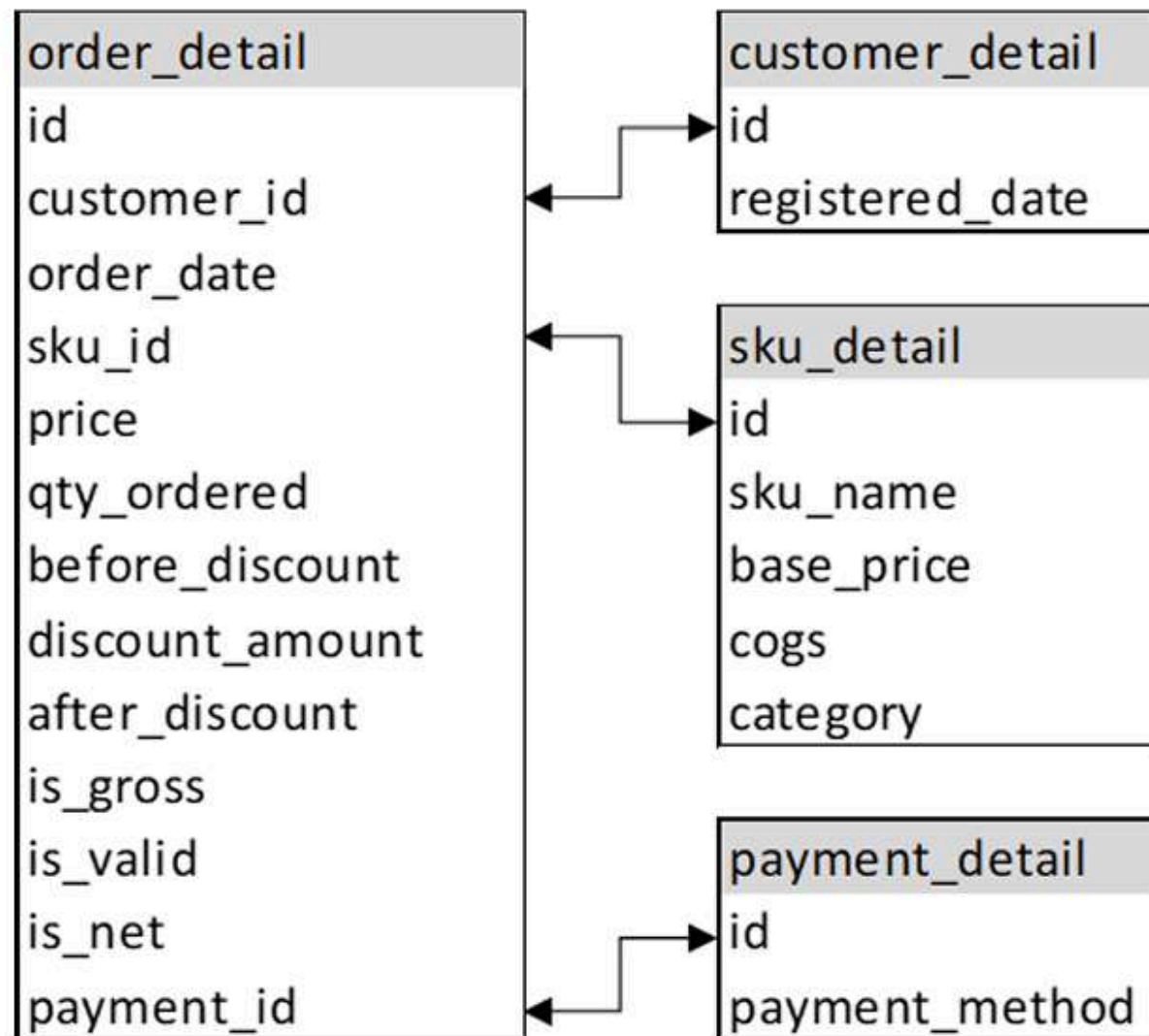
## payment\_detail:

- 1. id → angka unik dari metode pembayaran
- 2. payment\_method → metode pembayaran yang digunakan



# DATA Schema

Skema Tabel



# PROBLEM





Q: | Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after\_discount) paling besar? (Gunakan is\_valid = 1 untuk memfilter data transaksi).

Q: | Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar? (Gunakan is\_valid = 1 untuk memfilter data transaksi).

Q: | 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. (Gunakan is\_valid = 1 untuk memfilter data transaksi).

Q: | Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order). (Gunakan is\_valid = 1 untuk memfilter data transaksi).

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

# Let's solve this problem



# ANSWER



Selama transaksi yang terjadi selama 2021, pada bulan  
**Q:** apa total nilai transaksi (after\_discount) paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

# SYNTAX

```
SELECT
    EXTRACT(MONTH FROM order_date) AS MONTH,
    SUM(after_discount) AS total_transaction
FROM
    order_detail
WHERE
    EXTRACT(YEAR FROM order_date) = 2021
    AND is_valid = 1
GROUP BY MONTH
ORDER BY total_transaction DESC
```

*using SQL Online*

## SELECT

**EXTRACT(MONTH from order\_date) as month,**

- Bagian ini mengekstrak bulan dari kolom order\_date, yang menyimpan tanggal transaksi. EXTRACT(MOTNH FROM order\_date) akan mengembalikan angka bulan
- Hasil ekstraksi bulan ini diberi alias month.
- Menggunakan tanda (,) untuk memanggil lebih dari satu column.

**SUM(after\_discount) AS total\_transaction**

- Bagian ini menghitung total nilai transaksi untuk setiap bulan setelah diskon diterapkan.
- Hasilnya diberi alias total\_transaction, yang mewakili jumlah total transaksi setelah diskon.

**FROM order\_detail**

- Bagian ini menentukan tabel sumber data, yaitu order\_detail, yang berisi data transaksi.

Selama transaksi yang terjadi selama 2021, pada bulan  
**Q:** apa total nilai transaksi (after\_discount) paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

# SYNTAX

```
SELECT
    EXTRACT(MONTH FROM order_date) AS MONTH,
    SUM(after_discount) AS total_transaction
FROM
    order_detail
WHERE
    EXTRACT(YEAR FROM order_date) = 2021
    AND is_valid = 1
GROUP BY MONTH
ORDER BY total_transaction DESC
```

*using SQL Online*

## WHERE

**EXTRACT(YEAR FROM order\_date) = 2021**  
**AND is\_valid = 1**

- Kondisi WHERE digunakan untuk memfilter data transaksi.
- Memastikan bahwa hanya transaksi yang terjadi pada tahun 2021 yang diambil.
- is\_valid = 1 memastikan bahwa hanya transaksi yang valid yang dipertimbangkan

## GROUP BY MONTH

- Bagian ini mengelompokkan hasil berdasarkan bulan yang telah diekstrak sebelumnya (month). Jadi, semua transaksi dalam bulan yang sama akan digabungkan menjadi satu grup, dan nilai transaksi mereka akan dijumlahkan.

## ORDER BY total\_transaction DESC

- Bagian ini mengurutkan hasil berdasarkan total\_transaction dalam urutan menurun (DESC).

# RESULT

**Q:** Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi (after\_discount) paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

month	total_transaction
8	227862744
12	217309963
10	207603259.9
11	180396009.73999998
7	148007735



## INSIGHT:

Data transaksi selama tahun 2021 menunjukkan bulan Agustus (8) sebagai bulan dengan total transaksi paling besar dengan nilai **227.862.744**



**Q:** Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

# SYNTAX

```
SELECT
    s.category,
    SUM(o.after_discount) AS total_transaction
FROM
    order_detail AS o
    LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE
    EXTRACT(YEAR FROM o.order_date) = 2022
    AND o.is_valid = 1
GROUP BY s.category
ORDER BY total_transaction DESC
LIMIT 5
```

*using SQL Online*

## **SELECT**

**s.category,**  
**SUM(o.after\_discount) AS total\_transaction**

- **s.category** untuk menampilkan column category dari tabel sku\_detail yang disingkat dengan 's'.
- Menggunakan tanda (,) untuk memanggil lebih dari satu column.
- Bagian **SUM(o.after\_discount)** ini menghitung total nilai transaksi untuk setiap bulan setelah diskon diterapkan dari tabel order\_detail yang disingkat 'o'.
- Hasilnya diberi alias total\_transaction, yang mewakili jumlah total transaksi setelah diskon.

## **FROM**

**order\_detail AS o**  
**LEFT JOIN sku\_detail AS s ON o.sku\_id=s.id**

- Bagian ini menentukan tabel sumber data, yaitu tabel order\_detail yang digabung dengan tabel sku\_detail dengan primary key dan foreign key berupa o.sku\_id dan s.id



**Q:** Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

# SYNTAX

```
SELECT
    s.category,
    SUM(o.after_discount) AS total_transaction
FROM
    order_detail AS o
    LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE
    EXTRACT(YEAR FROM o.order_date) = 2022
    AND o.is_valid = 1
GROUP BY s.category
ORDER BY total_transaction DESC
LIMIT 5
```

*using SQL Online*

## WHERE

**EXTRACT(YEAR FROM o.order\_date) = 2022**  
**AND o.is\_valid = 1**

- **WHERE** digunakan untuk memfilter data yang akan dimunculkan.
- **EXTRACT(YEAR FROM o.order\_date) = 2022** menfilter tahun dari column o.order\_date pada tabel order\_detail pada tahun 2022 saja.
- **AND o.is\_valid = 1** memfilter data yang hanya data yang valid yang dipertimbangkan dari tabel order\_detail.

## GROUP BY s.category

## ORDER BY total\_transaction DESC

## LIMIT 5

- **GROUP BY s.category** ini mengelompokkan data berdasarkan category dari tabel sku\_detail.
- **ORDER BY total\_transaction DESC** mengurutkan hasil berdasarkan total\_transaction dalam urutan menurun (DESC).
- **LIMIT 5** untuk hanya menampilkan lima data saja.

# RESULT

**Q:** Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling besar?  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

category	total_transaction
Mobiles & Tablets	918451576
Entertainment	365344148.9
Appliances	316358100
Computing	214028543.4
Men Fashion	135588253



## INSIGHT:

Data transaksi selama tahun 2022 menunjukkan kategori dengan nilai transaksi paling besar adalah ***Mobiles & Tablets*** dengan total transaksi senilai ***918.451.576***

**Q:** 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. (Gunakan is\_valid = 1 untuk memfilter data transaksi).

## SYNTAX

```
WITH data_transaksi AS(
SELECT
    s.category,
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_date) = 2021 THEN o.after_discount END) AS t_21,
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_date) = 2022 THEN o.after_discount END) AS t_22,
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_date) = 2022 THEN o.after_discount END)
    -
    SUM(CASE WHEN EXTRACT(YEAR FROM o.order_date) = 2021 THEN o.after_discount END) AS selisih
FROM
    order_detail AS o
    LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE
    EXTRACT(YEAR FROM o.order_date) IN (2021,2022)
    AND o.is_valid=1
GROUP BY s.category
ORDER BY selisih DESC
)
SELECT *,
CASE WHEN t_21 > t_22 THEN 'Penurunan' ELSE 'Peningkatan' END AS status
FROM data_transaksi
```

*using SQL Online*

## WITH data\_transaksi AS ()

- Menggunakan CTE untuk menyederhanakan query.
- CTE dinamai sebagaia data\_transaksi.

## SELECT

**s.category,  
SUM(CASE WHEN EXTRACT(YEAR FROM o.order\_date)  
= 2021 THEN o.after\_discount END) AS t\_21, .....**

- Menampilkan category dari tabel sku\_detail
- Menampilkan nilai total transaksi selama tahun yang ditentukan dengan menggunakan kombinasi Agregat SUM dan SQL CASE.
- Hasil nilai total transaksi tahun 2021 dinamai t\_21 dan tahun 2022 dinamai t\_22.
- Menampilkan selisih antara nilai total transaksi tahun 2021 dan 2022 dengan menggunakan kombinasi Agregat dan SQL CASE.
- Hasil dinamai sebagai selisih.

## FROM

**order\_detail AS o**

**LEFT JOIN sku\_detail AS s on o.sku\_id=s.id**

- Menunjukkan asal data dari gabungan antara tabel order\_Detail dan sku\_detail dengan *primary key* dan *foreign key* adalah o.sku\_id dan s.id.

## WHERE

**EXTRACT(YEAR FROM o.order\_date) IN (2021,2022)  
AND o.is\_valid=1**

- Where digunakan untuk memfilter data.
- Extract(Year From o.order\_date) In (2021,2022) memfilter data hanya yang terjadi di tahun 2021 dan tahun 2022.
- And digunakan digunakan untuk data memenuhi kedua filter yang diminta.
- o.is\_valid = 1 memastikan bahwa hanya transaksi yang valid yang dipertimbangkan.



### **GROUP by s.category**

- Bagian ini mengelompokkan hasil berdasarkan category. Jadi, semua transaksi dalam bulan yang sama akan digabungkan menjadi satu grup, dan nilai transaksi mereka akan dijumlahkan.

### **order by selisih DESC**

- Mengurutkan data berdasarkan besaran nilai selisih yang diurutkan kebawah.

```
SELECT *,  
CASE WHEN t_21 > t_22 THEN 'Penurunan'  
ELSE 'Peningkatan' END AS status  
FROM data_transaksi
```

- **SELECT\*** Menampilkan keseluruhan data dari CTE data\_transaksi.
- **SQL CASE** digunakan untuk menampilkan status ketika:
  - (1) nilai transaksi tahun 2021 lebih besar dari tahun 2022 maka status bernilai **‘Penurunan’**.
  - (2) jika bernilai sebaliknya maka status bernilai **‘Peningkatan’**
- **SQL CASE** dinamai sebagai **‘status’**.
- **FROM data\_transaksi** menunjukkan data berasal dari data\_transaksi yang merukana CTE yang sudah dibuat.

# RESULT

**Q:** 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.  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

category	t_21	t_22	selisih	status
Mobiles & Tablets	370606718	918451576	547844858	Peningkatan
Entertainment	162326357.4	365344148.9	203017791.49999997	Peningkatan
Appliances	218550177	316358100	97807923	Peningkatan
Men Fashion	58628198	135588253	76960055	Peningkatan
Computing	172878860	214028543.4	41149683.400000006	Peningkatan
Home & Living	45797873	79483716.2	33685843.2	Peningkatan
Health & Sports	33837965.6	54235579.6	20397614	Peningkatan
Women Fashion	84045961.4	93014970.62	8969009.219999999	Peningkatan
School & Education	11558982.4	17362465.3	5803482.9	Peningkatan
Superstore	28828088	32643266.519999996	3815178.519999996	Peningkatan
Soghaat	15056202.6	17658332	2602129.4000000004	Peningkatan
Kids & Baby	23971057.8	25931276.84	1960219.0399999999	Peningkatan
Beauty & Grooming	46047360	46211019.18	163659.17999999997	Peningkatan
Books	10124596	6792519.2	-3332076.8	Penurunan
Others	40468515.74	21744646.02	-18723869.720000003	Penurunan



## INSIGHT:

Data perbandingan nilai transaksi antara tahun 2021 dan 2022 menunjukkan terdapat 13 dari 15 kategori yang mengalami peningkatan nilai transaksi yang dipimpin oleh ***Mobiles & Tablets***.

Terdapat 2 dari 15 kategori yang mengalami penurunan, diantaranya:

- ***Books***
- ***Others***



**Q:** Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order).  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

# SYNTAX

```
SELECT
    p.payment_method,
    COUNT(DISTINCT o.id) AS total_order
FROM
    order_detail AS o
    LEFT JOIN payment_detail AS p ON o.payment_id=p.id
WHERE
    EXTRACT(YEAR FROM o.order_date) = 2022
    AND o.is_valid = 1
GROUP BY p.payment_method
ORDER BY total_order DESC
LIMIT 5
```

*using SQL Online*

## SELECT

**p.payment\_method,**  
**COUNT(DISTINCT o.id) AS total\_order**

- **p.payment\_method** untuk menampilkan column payment\_method dari tabel payment\_detail yang disingkat dengan 'p'.
- Menggunakan tanda (,) untuk memanggil lebih dari satu column.
- Bagian **COUNT(DISTINCT o.id)** ini menghitung jumlah data id yang berbeda dari tabel order\_detail yang disingkat 'o'.
- Hasilnya diberi alias total\_order.

## FROM

**order\_detail AS o**  
**LEFT JOIN payment\_detail AS s ON o.payment\_id=p.id**

- Bagian ini menentukan tabel sumber data, yaitu tabel order\_detail yang digabung dengan tabel payment\_detail dengan primary key dan foreign key berupa o.payment\_id dan p.id

**Q:** Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order).  
(Gunakan is\_valid = 1 untuk memfilter data transaksi).

## SYNTAX

```
SELECT
    P.payment_method,
    COUNT(DISTINCT o.id) AS total_order
FROM
    order_detail AS o
    LEFT JOIN payment_detail AS P ON o.payment_id=P.id
WHERE
    EXTRACT(YEAR FROM o.order_date) = 2022
    AND o.is_valid = 1
GROUP BY P.payment_method
ORDER BY total_order DESC
LIMIT 5
```

*using SQL Online*

## WHERE

**EXTRACT(YEAR FROM o.order\_date) = 2022**  
**AND o.is\_valid = 1**

- **WHERE** digunakan untuk memfilter data yang akan dimunculkan.
- **EXTRACT(YEAR FROM o.order\_date) = 2022** menfilter tahun dari column o.order\_date pada tabel order\_detail pada tahun 2022 saja.
- **AND o.is\_valid = 1** memfilter data yang hanya data yang valid yang dipertimbangkan dari tabel order\_detail.

**GROUP BY p.payment\_method**  
**ORDER BY total\_order DESC**  
**LIMIT 5**

- **GROUP BY p.payment\_method** ini mengelompokkan data berdasarkan metode pembayaran dari tabel payment\_detail.
- **ORDER BY total\_order DESC** mengurutkan hasil berdasarkan total\_order dalam urutan menurun (DESC).
- **LIMIT 5** untuk hanya menampilkan lima data saja.

# RESULT

**Q:** Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022 (berdasarkan total unique order). (Gunakan is\_valid = 1 untuk memfilter data transaksi).

payment_method	total_order
cod	1809
Payaxis	181
customercredit	75
Easypay	69
jazzwallet	26



## INSIGHT:

Data transaksi selama tahun 2022 menunjukkan top 5 metode pembayaran yang paling populer digunakan adalah:

**#1 COD**

**#2 Payaxis**

**#3 customercredit**

**#4 Easypay**

**#5 jazzwallet**

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

# SYNTAX

```
SELECT
  CASE
    WHEN s.sku_name LIKE '%Samsung%' THEN 'Samsung'
    WHEN s.sku_name LIKE '%Apple%' THEN 'Apple'
    WHEN s.sku_name LIKE '%Sony%' THEN 'Sony'
    WHEN s.sku_name LIKE '%Huawei%' THEN 'Huawei'
    WHEN s.sku_name LIKE '%Lenovo%' THEN 'Lenovo'
    END AS product,
  SUM(o.after_discount) AS total_transaction
FROM order_detail AS o
  LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE o.is_valid = 1
  AND (s.sku_name LIKE '%Samsung%'
    OR s.sku_name LIKE '%Apple%'
    OR s.sku_name LIKE '%Sony%'
    OR s.sku_name LIKE '%Huawei%'
    OR s.sku_name LIKE '%Lenovo%')
GROUP BY product
ORDER BY total_transaction DESC
```

*using SQL Online*

## SELECT CASE

```
WHEN s.sku_name LIKE '%Samsung%' THEN 'Samsung'
WHEN s.sku_name LIKE '%Apple%' Then 'Apple'
WHEN s.sku_name LIKE '%Sony%' THEN 'Sony'
WHEN s.sku_name like '%Huawei%' THEN 'Huawei'
WHEN s.sku_name LIKE '%Lenovo%' THEN 'Lenovo'
END as product,
SUM(o.after_discount) AS total_transaction
```

- **SQL CASE** digunakan untuk mengelompokkan sku\_name ke dalam brand yang sesuai (Samsung, Apple, Sony, Huawei, Lenovo). Hasilnya diberi alias product.
- Tanda ‘%’ digunakan untuk mendeteksi kata (Samsung, Apple, Sony, Huawei, Lenovo) dari column s.sku\_name.
- Menggunakan tanda (,) untuk memanggil lebih dari satu column.
- Bagian **SUM(o.after\_discount) AS total\_transaction** ini menghitung jumlah nilai transaksi dari tabel order\_detail yang disingkat ‘o’.
- Hasilnya diberi alias total\_transaction untuk mewakili total transaksi.



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

# SYNTAX

```
SELECT
  CASE
    WHEN s.sku_name LIKE '%Samsung%' THEN 'Samsung'
    WHEN s.sku_name LIKE '%Apple%' THEN 'Apple'
    WHEN s.sku_name LIKE '%Sony%' THEN 'Sony'
    WHEN s.sku_name LIKE '%Huawei%' THEN 'Huawei'
    WHEN s.sku_name LIKE '%Lenovo%' THEN 'Lenovo'
  END AS product,
  SUM(o.after_discount) AS total_transaction
FROM order_detail AS o
  LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE o.is_valid = 1
  AND (s.sku_name LIKE '%Samsung%'
    OR s.sku_name LIKE '%Apple%'
    OR s.sku_name LIKE '%Sony%'
    OR s.sku_name LIKE '%Huawei%'
    OR s.sku_name LIKE '%Lenovo%')
GROUP BY product
ORDER BY total_transaction DESC
```

*using SQL Online*

**FROM order\_detail as o**  
**LEFT JOIN sku\_detail as s on o.sku\_id=s.id**  
**WHERE**

**o.is\_valid = 1**

**AND (s.sku\_name like '%Samsung%'**  
**OR s.sku\_name LIKE '%Apple%'**  
**OR s.sku\_name LIKE '%Sony%'**  
**OR s.sku\_name like '%Huawei%'**  
**OR s.sku\_name LIKE '%Lenovo%')**

- **FROM order\_detail AS o LEFT JOIN sku\_detail AS s on o.sku\_id=s.id** Bagian ini menentukan tabel sumber data, yaitu tabel order\_detail yang digabung dengan tabel sku\_detail dengan primary key dan foreign key berupa o.sku\_id dan s.id
- **WHERE o.is\_valid = 1** memfilter data yang hanya data yang valid yang dipertimbangkan dari tabel order\_detail.
- **AND & OR** memfilter produk yang namanya mengandung salah satu dari lima brand yang dicari dari column s.sku\_name.

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

# SYNTAX

```
SELECT
  CASE
    WHEN s.sku_name LIKE '%Samsung%' THEN 'Samsung'
    WHEN s.sku_name LIKE '%Apple%' THEN 'Apple'
    WHEN s.sku_name LIKE '%Sony%' THEN 'Sony'
    WHEN s.sku_name LIKE '%Huawei%' THEN 'Huawei'
    WHEN s.sku_name LIKE '%Lenovo%' THEN 'Lenovo'
    END AS product,
  SUM(o.after_discount) AS total_transaction
FROM order_detail AS o
  LEFT JOIN sku_detail AS s ON o.sku_id=s.id
WHERE o.is_valid = 1
  AND (s.sku_name LIKE '%Samsung%'
    OR s.sku_name LIKE '%Apple%'
    OR s.sku_name LIKE '%Sony%'
    OR s.sku_name LIKE '%Huawei%'
    OR s.sku_name LIKE '%Lenovo%')
GROUP BY product
ORDER BY total_transaction DESC
```

*using SQL Online*

## GROUP BY product

- Bagian ini mengelompokkan hasil berdasarkan produk yang dicari. Jadi, semua transaksi dalam bulan yang sama akan digabungkan menjadi satu grup, dan nilai transaksi mereka akan dijumlahkan.

## ORDER BY total\_transaction DESC

- Bagian ini mengurutkan hasil berdasarkan total\_transaction dalam urutan menurun (DESC).



# RESULT

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

product	total_transaction
Samsung	380818894
Apple	190980080
Huawei	43092260
Sony	31127440
Lenovo	8199402



## INSIGHT:

Data transaksi menunjukkan urutan 5 product berdasarkan nilai total transaksi terbesar adalah:

**#1 Samsung**

**#2 Apple**

**#3 Huawei**

**#4 Sony**

**#5 Lenovo**

An illustration of a person's hands holding a black rectangular sign with the words 'THANK YOU!' in large, green, sans-serif capital letters. The sign is decorated with several small, colorful polka dots in shades of green, grey, and brown. The person holding the sign is wearing a green long-sleeved shirt. The background is a light green, stylized shape resembling a person's head and shoulders.

THANK  
YOU!

Kelompok 7

Learning With My Skill

FINAL PROJECT 17