

LAPORAN TUGAS KELOMPOK



MATAKULIAH DATA WAREHOUSE SEMESTER GENAP TA. 2025 /2026

Kelas : D

Anggota Kelompok

No	NIM	Name	Photo	Peran	Sign
1.	202431005	Muhammad Hafizh Wijdan		Operasi OLAP DRILL DOWN, DICE, PIVOT KOTA, DRILL THROUGH	
2.	202431033	Raihan Daris Ramadhan		Operasi Roll UP, Slice, Pivot A/B, Drill Across,	
3.					
4.					
5.					

PERSIAPAN DATABASE

CREATE DATABASE tugas_data_warehouse;

USE tugas_data_warehouse;

CREATE TABLE penjualan_olap (
tanggal DATE,
produk VARCHAR(50),
lokasi VARCHAR(50),
pendapatan INT
);

INSERT INTO penjualan_olap (tanggal, produk, lokasi, pendapatan) VALUES
('2024-01-01', 'Produk A', 'Kota X', 1000),
('2024-01-01', 'Produk A', 'Kota Y', 1500),
('2024-01-01', 'Produk B', 'Kota X', 2000),
('2024-01-01', 'Produk B', 'Kota Y', 1200),
('2024-02-01', 'Produk A', 'Kota X', 1200),
('2024-02-01', 'Produk A', 'Kota Y', 1800),
('2024-02-01', 'Produk B', 'Kota X', 2100),
('2024-02-01', 'Produk B', 'Kota Y', 1300);

1. ROLL UP

Roll Up adalah proses penggabungan data dari level yang lebih rendah ke level yang lebih tinggi.

a. Tanggal + Total Pendapatan

```
SELECT tanggal, SUM(pendapatan) AS total_pendapatan
FROM penjualan_olap
GROUP BY tanggal;
```

tanggal	total_pendapatan
2024-01-01	5700
2024-02-01	6400

Query ini melakukan proses Roll Up dengan mengelompokkan data berdasarkan tanggal. Pendapatan yg awalnya masih berdasarkan produk dan lokasi dijumlahkan menjadi total pendapatan per tanggal

b. Produk + Total Pendapatan

```
SELECT produk, SUM(pendapatan) AS total_pendapatan
FROM penjualan_olap
GROUP BY produk;
```

produk	total_pendapatan
Produk A	5500
Produk B	6600

Query ini menjumlahkan pendapatan berdasarkan produk , sehingga dapat diketahui total pendapatan dari Produk A dan Produk B

c. Location + Total Pendapatan

```
SELECT lokasi, SUM(pendapatan) AS total_pendapatan
FROM penjualan_olap
GROUP BY lokasi;
```

lokasi	total_pendapatan
Kota X	6300
Kota Y	5800

Query ini menjumlahkan pendapatan berdasarkan lokasi , sehingga dapat diketahui total pendapatan dari Lokasi A dan Lokasi B

2. DRILL DOWN (Mendalami Data)

By tanggal

Kebalikan dari Roll Up. Memecah data agregasi tingkat tinggi menjadi level yang lebih detail (menambah dimensi). Misalnya, kita jabarkan total pendapatan tadi menjadi detail per bulan januari saja, atau februari

QUERY:

```
1 SELECT |
2     MONTH(tanggal) AS bulan,
3     SUM(pendapatan) AS total_pendapatan
4 FROM
5     penjualan_olap
6 WHERE
7     MONTH(tanggal) = 1
8 GROUP BY
9     MONTH(tanggal),
10    produk;
```

SELECT

MONTH(tanggal) AS bulan,

SUM(pendapatan) AS total_pendapatan

FROM

penjualan_olap

WHERE

MONTH(tanggal) = 1 -- Fokus drill-down hanya pada bulan Januari

GROUP BY

MONTH(tanggal),

produk;

bulan	total_pendapatan
1	2500
1	3200

Data yang sebelumnya hanya total per produk sekarang dipecah lagi ke level yang lebih detail, yaitu dipecah berdasarkan bulan (bulan 1 dan bulan 2), hari dan tahun, dikarenakan hari sama juga dengan tahun, maka bulan menjadi urutan 1 dan 2.

By product

Kita akan melihat berdasarkan produk a atau b yang kemudian akan ditambah berdasarkan kota x atau y, yang artinya drill down adalah mencari pada secara mendetail dan tidak secara umum dengan syarat atau kondisi tertentu

QUERY:

```

1 SELECT
2     produk,
3     lokasi,
4     tanggal, -- Drill-down ke tingkat paling detail dari data
5     SUM(pendapatan) AS total_pendapatan
6 FROM
7     penjualan_olap
8 WHERE
9     produk = 'Produk A' AND lokasi = 'Kota Y' //bisa ganti produk b, dan kota x
10 GROUP BY
11     produk,
12     lokasi,
13     tanggal;

```

SELECT

produk,

SUM(pendapatan) AS total_pendapatan

FROM

penjualan_olap

WHERE

produk = 'Produk A' //bisa ganti produk b

GROUP BY

produk,

lokasi,

tanggal;

output

produk	total_pendapatan
Produk A	1000
Produk A	1200
Produk A	1500
Produk A	1800

By location

Kita akan melihat berdasarkan kota x atau y, yang artinya drill down adalah mencari pada secara mendetail dan tidak secara umum dengan syarat atau kondisi tertentu

QUERY:

```
SELECT
  lokasi,
  produk,
  SUM(pendapatan) AS total_pendapatan
FROM
  penjualan_olap
WHERE
  lokasi = 'Kota X'
GROUP BY
  lokasi,
  produk;
```

lokasi	produk	total_pendapatan
Kota X	Produk A	2200
Kota X	Produk B	4100

Maka hasilnya akan menghasilkan kota x saja

Jika digabungkan maka

```
1 SELECT
2     MONTH(tanggal) AS bulan,
3     lokasi,
4     produk,
5     SUM(pendapatan) AS pendapatan
6 FROM penjualan_olap
7 WHERE MONTH(tanggal) = 1
8     AND lokasi = 'Kota X'
9     AND produk = 'Produk A' -- Mengunci Produk A saja
10 GROUP BY MONTH(tanggal), lokasi, produk
```

```
SELECT
  MONTH(tanggal) AS bulan,
  lokasi,
  produk,
  SUM(pendapatan) AS pendapatan
FROM penjualan_olap
WHERE MONTH(tanggal) = 1
```

AND lokasi = 'Kota X'
AND produk = 'Produk A'
GROUP BY MONTH(tanggal), lokasi, produk;

bulan	lokasi	produk	pendapatan
1	Kota X	Produk A	1000

3. Slice

Slice digunakan untuk mengambil sebagian data berdasarkan satu dimensi tertentu.

a. By Tanggal

	tanggal	produk	lokasi	pendapatan
<pre>SELECT tanggal, produk, lokasi, pendapatan FROM penjualan_olap WHERE tanggal = '2024-01-01';</pre>	2024-01-01	Produk A	Kota X	1000
	2024-01-01	Produk A	Kota Y	1500
	2024-01-01	Produk B	Kota X	2000
	2024-01-01	Produk B	Kota Y	1200

Query ini melakukan operasi slice berdasarkan dimensi tanggal. Data yang ditampilkan hanya data penjualan pada 2024-01-01.

b. By Product

	tanggal	produk	lokasi	pendapatan
<pre>SELECT tanggal, produk, lokasi, pendapatan FROM penjualan_olap WHERE produk = 'Produk A';</pre>	2024-01-01	Produk A	Kota X	1000
	2024-01-01	Produk A	Kota Y	1500
	2024-02-01	Produk A	Kota X	1200
	2024-02-01	Produk A	Kota Y	1800

Query ini melakukan operasi slice berdasarkan dimensi produk. Data yang ditampilkan hanya data penjualan pada produk A.

c. By Location

	tanggal	produk	lokasi	pendapatan
<pre>SELECT tanggal, produk, lokasi, pendapatan FROM penjualan_olap WHERE lokasi = 'Kota X';</pre>	2024-01-01	Produk A	Kota X	1000
	2024-01-01	Produk B	Kota X	2000
	2024-02-01	Produk A	Kota X	1200
	2024-02-01	Produk B	Kota X	2100

Query ini melakukan operasi slice berdasarkan dimensi lokasi. Data yang ditampilkan hanya data penjualan pada Lokasi kota X.

4. Dice

- By tanggal dan produk

Mirip dengan Slice, namun mengambil "sub-kubus" dengan memfilter dua atau lebih dimensi. Misalnya, kita ingin melihat transaksi pada **Bulan Januari (2024-01-01)**, tetapi hanya untuk **Produk A**.

Query:

SELECT

tanggal,

produk,

lokasi,

pendapatan

FROM

penjualan_olap

WHERE

tanggal = '2024-01-01'

AND produk = 'Produk A';

tanggal	produk	lokasi	pendapatan
2024-01-01	Produk A	Kota X	1000
2024-01-01	Produk A	Kota Y	1500

Data difilter menggunakan lebih dari satu kriteria, hanya data yang memenuhi syarat tanggal dan produk tersebut tanpa memedulikan di kota

- **By tanggal dan lokasi**

kita ingin melihat transaksi pada **Bulan Februari (2024-02-01)**, tetapi hanya yang berlokasi di **Kota Y**.

SELECT

tanggal,

produk,

lokasi,

pendapatan

FROM

penjualan_olap

WHERE

tanggal = '2024-02-01'

AND lokasi = 'Kota Y';

tanggal	produk	lokasi	pendapatan
2024-02-01	Produk A	Kota Y	1800
2024-02-01	Produk B	Kota Y	1300

Data difilter secara spesifik menggunakan lebih dari satu kriteria, menyisakan hanya data yang memenuhi syarat tanggal dan lokasi tersebut tanpa memedulikan produk apa saja yang terjual.

- **By produk dan lokasi**

mengambil "sub-kubus" dengan memfilter dua atau lebih dimensi. Misalnya, kita ingin melihat transaksi untuk **Produk B**, tetapi hanya yang berlokasi di **Kota X**.

SELECT

```

    tanggal,
    produk,
    lokasi,
    pendapatan
FROM
    penjualan_olap
WHERE
    produk = 'Produk B'
    AND lokasi = 'Kota X';

```

tanggal	produk	lokasi	pendapatan
2024-01-01	Produk B	Kota X	2000
2024-02-01	Produk B	Kota X	2100

Data difilter secara spesifik menggunakan lebih dari satu kriteria, menyisakan hanya data yang memenuhi syarat produk dan lokasi tersebut tanpa memedulikan kapan (tanggal) transaksi itu

- **By tanggal produk dan lokasi**

mengambil "sub-kubus" dengan memfilter dua atau lebih dimensi. Misalnya, kita ingin melihat transaksi pada **Bulan Januari (2024-01-01)**, khusus untuk **Produk A**, dan hanya yang berlokasi di **Kota X**.

tanggal	produk	lokasi	pendapatan
2024-01-01	Produk A	Kota X	1000

5. Pivot

a. X, dan Y

```

SELECT
    tanggal,
    produk,
    SUM(CASE WHEN lokasi = 'Kota X' THEN pendapatan ELSE 0 END) AS
    Pendapatan_Kota_X,
    SUM(CASE WHEN lokasi = 'Kota Y' THEN pendapatan ELSE 0 END) AS Pendapatan_Kota_Y
FROM
    penjualan_olap
GROUP BY
    tanggal,
    produk
ORDER BY
    tanggal,
    produk;

```

tanggal	produk	Pendapatan_Kota_X	Pendapatan_Kota_Y
2024-01-01	Produk A	1000	1500
2024-01-01	Produk B	2000	1200
2024-02-01	Produk A	1200	1800
2024-02-01	Produk B	2100	1300

b. Pivot Table Product A/B

```
SELECT
CASE
    WHEN produk = 'Produk A' THEN 'A'
    WHEN produk = 'Produk B' THEN 'B'
    END AS Product,

SUM(CASE
    WHEN lokasi = 'Kota X' AND MONTH(tanggal) = 1
    THEN pendapatan ELSE 0
    END) AS KotaX_Bulan1,

SUM(CASE
    WHEN lokasi = 'Kota X' AND MONTH(tanggal) = 2
    THEN pendapatan ELSE 0
    END) AS KotaX_Bulan2,

SUM(CASE
    WHEN lokasi = 'Kota Y' AND MONTH(tanggal) = 1
    THEN pendapatan ELSE 0
    END) AS KotaY_Bulan1,

SUM(CASE
    WHEN lokasi = 'Kota Y' AND MONTH(tanggal) = 2
    THEN pendapatan ELSE 0
    END) AS KotaY_Bulan2S
```

```
FROM penjualan_olap
GROUP BY produk;
```

Product	KotaX_Bulan1	KotaX_Bulan2	KotaY_Bulan1	KotaY_Bulan2S
A	1000	1200	1500	1800
B	2000	2100	1200	1300

Query ini membuat tabel Pivot dengan Produk sebagai baris, lalu pendapatan dipisahkan menjadi kolom berdasarkan kombinasi Kota X/Kota Y dan Bulan 1/Bulan 2.

6. Drill Across

a. Membuat Table stok

```
CREATE TABLE stok_olap (
    tanggal DATE,
    produk VARCHAR(50),
    lokasi VARCHAR(50),
    stok INT
);
```

b. Mengisi data Stok

```
INSERT INTO stok_olap (tanggal, produk, lokasi, stok) VALUES
('2024-01-01', 'Produk A', 'Kota X', 50),
('2024-01-01', 'Produk A', 'Kota Y', 40),
('2024-01-01', 'Produk B', 'Kota X', 60),
('2024-01-01', 'Produk B', 'Kota Y', 35),
('2024-02-01', 'Produk A', 'Kota X', 55),
('2024-02-01', 'Produk A', 'Kota Y', 45),
('2024-02-01', 'Produk B', 'Kota X', 65),
('2024-02-01', 'Produk B', 'Kota Y', 38);
```

c. Cek isi table

`SELECT * FROM stok_olap;`

☐ Show all | Number of rows:

Extra options

tanggal	produk	lokasi	stok
2024-01-01	Produk A	Kota X	50
2024-01-01	Produk A	Kota Y	40
2024-01-01	Produk B	Kota X	60
2024-01-01	Produk B	Kota Y	35
2024-02-01	Produk A	Kota X	55
2024-02-01	Produk A	Kota Y	45
2024-02-01	Produk B	Kota X	65
2024-02-01	Produk B	Kota Y	38

d. Main Query

```
SELECT
    p.tanggal,
    p.produk,
    p.lokasi,
    p.pendapatan,
    s.stok
FROM penjualan_olap p
JOIN stok_olap s
    ON p.tanggal = s.tanggal
    AND p.produk = s.produk
    AND p.lokasi = s.lokasi;
```

tanggal	produk	lokasi	pendapatan	stok
2024-01-01	Produk A	Kota X	1000	50
2024-01-01	Produk A	Kota Y	1500	40
2024-01-01	Produk B	Kota X	2000	60
2024-01-01	Produk B	Kota Y	1200	35
2024-02-01	Produk A	Kota X	1200	55
2024-02-01	Produk A	Kota Y	1800	45
2024-02-01	Produk B	Kota X	2100	65
2024-02-01	Produk B	Kota Y	1300	38

Query ini melakukan operasi Drill Across dengan menggabungkan tabel penjualan_olap dan tabel stok_olap. Kedua tabel tersebut digabungkan berdasarkan tanggal, produk dan lokasi yang sama. Hasilnya menampilkan pendapatan dan stok produk secara bersamaan, sehingga dapat dianalisis

7. Drill Through

melihat total pendapatan Produk B di Kota X pada 2024-01-01 adalah **2,000**, lalu melakukan *drill-through* untuk melihat jam berapa saja transaksi itu terjadi, siapa kasirnya, atau ID transaksi terkait.

Buat tabel baru

-- 1. Buat tabel transaksi operasional (OLTP) untuk menampung data mentah CREATE TABLE transaksi_oltp_mentah (id_transaksi VARCHAR(10), waktu_transaksi TIME, kasir VARCHAR(50), tanggal DATE, produk VARCHAR(50), lokasi VARCHAR(50), jumlah_barang INT, harga_satuan INT);

-- 2. Masukkan data detail transaksi asli (struk) -- Jika dijumlah, total pendapatan dari 2 transaksi ini adalah $1.200 + 800 = 2.000$ -- Angka ini cocok dengan baris ('2024-01-01', 'Produk B', 'Kota X', 2000) di tabel OLAP-mu. INSERT INTO transaksi_oltp_mentah VALUES ('TX-001', '09:15:00', 'Andi', '2024-01-01', 'Produk B', 'Kota X', 6, 200), ('TX-002', '14:30:22', 'Siti', '2024-01-01', 'Produk B', 'Kota X', 4, 200);

☐ Show all

Number of rows: 25

Filter rows:

Extra options

id_transaksi	waktu_transaksi	kasir	jumlah_barang	harga_satuan	sub_total
TX-001	09:15:00	Andi	6	200	1200
TX-002	14:30:22	Siti	4	200	800

☐ Show all

Number of rows: 25

Filter rows:

Query results operations

