

LAPORAN PRAKTIKUM

DATA WAREHOUSE



NAMA : Raihan Daris Ramadhan

NIM : 202431033

KELAS : B

DOSEN : Dine Tiara Kusuma, ST., M.Kom

NO.PC : 29

ASISTEN : 1. Jahraini Saputri

2. Muhammad Bintang Putra Erfin

3. Rama Dinantiar

FAKULTAS TELEMATIKA ENERGI
JURUSAN TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI PLN – JAKARTA
2025/2026

1. Persiapan Database

```
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);
```

#Raihan Daris Ramadhan 202431033

2. Roll UP

Roll Up adalah operasi untuk menaikkan level ringkasan data. Data yang awalnya masih detail berdasarkan tanggal, produk, dan lokasi digabung menjadi total yang lebih umum. Dalam laporan ini, Roll Up digunakan untuk melihat gambaran besar pendapatan berdasarkan tanggal, produk, dan lokasi.

a. Roll Up Berdasarkan tanggal

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

#Raihan Daris Ramadhan 202431033

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

Query ini menggabungkan semua pendapatan dari Produk A dan Produk B di Kota X maupun Kota Y, lalu menjumlahkannya berdasarkan tanggal. Hasilnya menunjukkan bahwa tanggal 2024-01-01 memiliki total pendapatan 5.700, sedangkan tanggal 2024-02-01 memiliki total pendapatan 6.400.

b. Roll Up Berdasarkan Produk

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

produk	total_pendapatan
Produk A	5500
Produk B	6600

#Raihan Daris Ramadhan 202431033

Pada query ini, dimensi tanggal dan lokasi tidak lagi ditampilkan. Fokus analisisnya hanya produk. Dengan cara ini dapat diketahui bahwa total pendapatan Produk B lebih tinggi dibandingkan Produk A.

c. Roll Up berdasarkan lokasi

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

lokasi	total_pendapatan
Kota X	6300
Kota Y	5800

#Raihan Daris Ramadhan 202431033

Query ini melihat pendapatan dari sisi wilayah penjualan. Data Produk A dan Produk B pada dua tanggal digabungkan berdasarkan kota. Dari hasil tersebut, Kota X menghasilkan total pendapatan 6.300 dan Kota Y menghasilkan 5.800.

3. Drill Down

Drill Down adalah kebalikan dari Roll Up. Jika Roll Up membuat data semakin ringkas, maka Drill Down membuat data semakin detail. Operasi ini berguna ketika nilai total sudah ditemukan, tetapi masih ingin diketahui sumber detail yang membentuk total tersebut.

a. Drill Down by Tanggal

```
SELECT
    MONTH(tanggal) AS bulan,
    SUM(pendapatan) AS total_pendapatan
FROM
    penjualan_olap
WHERE
    MONTH(tanggal) = 1
GROUP BY
    MONTH(tanggal),
    produk;
```

bulan	total_pendapatan
1	2500
1	3200

#Raihan Daris Ramadhan 202431033

b. Drill Down by Product

```

SELECT
    produk,
    lokasi,
    tanggal,
    SUM(pendapatan) AS total_pendapatan
FROM
    penjualan_olap
WHERE
    produk = 'Produk A' AND lokasi = 'Kota Y'
GROUP BY
    produk,
    lokasi,
    tanggal;

```

#Raihan Daris Ramadhan 202431033

produk	lokasi	tanggal	total_pendapatan
Produk A	Kota Y	2024-01-01	1500
Produk A	Kota Y	2024-02-01	1800

c. Drill Down by Location

```

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

```

#Raihan Daris Ramadhan 202431033

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

Query ini mendalami data berdasarkan lokasi Kota X. Dari hasil tersebut, pendapatan Kota X tidak lagi hanya terlihat sebagai satu total, tetapi sudah dipisahkan berdasarkan Produk A dan Produk B.

4. Slice

Slice adalah operasi mengambil potongan data berdasarkan satu dimensi saja. Bisa diibaratkan seperti memotong satu lapisan dari kubus data. Karena hanya memakai satu kriteria, hasil Slice masih bisa berisi beberapa produk, beberapa lokasi, atau beberapa tanggal.

a. Slice By Tanggal

```
SELECT tanggal, produk, lokasi, pendapatan
FROM penjualan_olap
WHERE tanggal = '2024-01-01';

#Raihan Daris Ramadhan 202431033
```

tanggal	produk	lokasi	pendapatan
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 hanya menampilkan data penjualan pada tanggal 2024-01-01. Produk dan lokasi tetap beragam karena dimensi yang dibatasi hanya tanggal.

b. Slice By Product

```
SELECT tanggal, produk, lokasi, pendapatan
FROM penjualan_olap
WHERE produk = 'Produk A';

#Raihan Daris Ramadhan 202431033
```

tanggal	produk	lokasi	pendapatan
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 mengambil seluruh transaksi Produk A. Walaupun produknya sama, tanggal dan lokasi tetap bisa berbeda, karena Slice hanya menggunakan satu dimensi filter.

c. Slice By Location

```
SELECT tanggal, produk, lokasi, pendapatan
FROM penjualan_olap
WHERE lokasi = 'Kota X';

#Raihan Daris Ramadhan 202431033
```

tanggal	produk	lokasi	pendapatan
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 mengambil data khusus Kota X. Hasilnya masih menampilkan beberapa produk dan tanggal karena yang dikunci hanya dimensi lokasi.

5. Dice

Dice mirip dengan Slice, tetapi filter yang dipakai lebih dari satu dimensi. Karena syaratnya lebih banyak, hasil Dice biasanya lebih kecil dan lebih spesifik dibandingkan Slice

a. By Tanggal dan produk

```
SELECT tanggal, produk, lokasi, pendapatan
FROM penjualan_olap
WHERE tanggal = '2024-01-01' AND produk = 'Produk A';

#Raihan Daris Ramadhan 202431033
```

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

Data difilter menggunakan dua kriteria, yaitu tanggal dan produk. Karena lokasi tidak dibatasi, maka Kota X dan Kota Y masih muncul pada hasil query.

b. By Tanggal dan lokasi

```
SELECT tanggal, produk, lokasi, pendapatan
FROM penjualan_olap
WHERE tanggal = '2024-02-01' AND lokasi = 'Kota Y';
```

```
#Raihan Daris Ramadhan 202431033
```

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

Query ini membatasi data pada tanggal 2024-02-01 dan lokasi Kota Y. Produk tidak dibatasi, sehingga data Produk A dan Produk B tetap muncul selama berada pada tanggal dan lokasi tersebut.

c. By produk dan lokasi

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

```
#Raihan Daris Ramadhan 202431033
```

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

Query ini hanya menampilkan Produk B yang berlokasi di Kota X. Karena tanggal tidak menjadi filter, data dari Januari dan Februari tetap muncul.

6. Pivot

Pivot digunakan untuk mengubah susunan data agar lebih mudah dibandingkan. Nilai yang awalnya berada dalam baris dapat dipindahkan menjadi kolom baru, contohnya pendapatan Kota X dan Kota Y.

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;
```

```
#Raihan Daris Ramadhan 202431033
```

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

Query ini membuat kolom khusus untuk Kota X dan Kota Y. Dengan bentuk pivot, perbandingan pendapatan antar kota untuk setiap produk dan tanggal menjadi lebih langsung terlihat.

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;
#Raihan Daris Ramadhan 202431033
```

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. Pendapatan dipisahkan berdasarkan kombinasi lokasi dan bulan, sehingga Produk A dan Produk B dapat dibandingkan dalam satu tampilan matriks.

7. Drill Across

Drill Across digunakan untuk melihat data dari dua tabel berbeda yang memiliki dimensi yang sama. Pada contoh ini, tabel penjualan_olap digabung dengan tabel stok_olap berdasarkan tanggal, produk, dan lokasi.

a. Membuat Table Stock

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

#Raihan Daris Ramadhan 202431033

Tabel stok_olap dibuat dengan struktur dimensi yang sama seperti penjualan_olap. Kesamaan ini penting agar kedua tabel dapat disambungkan pada operasi Drill Across.

b. Mengisi Data Stock

```
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);
```

#Raihan Daris Ramadhan 202431033

Data stok diisi sesuai kombinasi tanggal, produk, dan lokasi. Setiap baris stok memiliki pasangan yang sesuai dengan data penjualan.

c. 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;
```

#Raihan Daris Ramadhan 202431033

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 menggabungkan tabel penjualan_olap dan stok_olap. Hasilnya menampilkan pendapatan dan stok secara berdampingan, sehingga analisis tidak hanya melihat nilai penjualan, tetapi juga kondisi stok pada kombinasi data yang sama.

8. Drill Through

Drill Through digunakan untuk menelusuri data ringkasan sampai ke data transaksi mentah. Contohnya, total pendapatan Produk B di Kota X pada 2024-01-01 adalah 2.000, lalu ditelusuri menjadi transaksi yang membentuk angka tersebut.

```
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
);

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);
```

#Raihan Daris Ramadhan 202431033

```
SELECT
  id_transaksi,
  waktu_transaksi,
  kasir,
  tanggal,
  produk,
  lokasi,
  jumlah_barang,
  harga_satuan,
  jumlah_barang * harga_satuan AS total_pendapatan
FROM
  transaksi_oltp_mentah
WHERE
  tanggal = '2024-01-01'
  AND produk = 'Produk B'
  AND lokasi = 'Kota X';
```

#Raihan Daris Ramadhan 202431033

id_transaksi	waktu_transaksi	kasir	tanggal	produk	lokasi	jumlah_barang	harga_satuan	total_pendapatan
TX-001	09:15:00	Andi	2024-01-01	Produk B	Kota X	6	200	1200
TX-002	14:30:22	Siti	2024-01-01	Produk B	Kota X	4	200	800

Dua transaksi tersebut menjelaskan dari mana total 2.000 berasal. TX-001 menghasilkan 1.200 dari 6 barang dengan harga 200, sedangkan TX-002 menghasilkan 800 dari 4 barang dengan harga 200. Jadi, Drill Through membantu menghubungkan data OLAP dengan bukti transaksi di level paling detail.