



Jurusan Teknologi Informasi Politeknik Negeri Malang
Mata Kuliah Data Warehouse
Kuis 1

Nama : Dahniar Davina

Nomor Urut : 09 / SIB – 2A

1. Tuliskan perbandingan star schema dan snowflake schema pada tabel berikut:

	Star Schema	Snowflake Schema
Normalisasi	Menggunakan denormalisasi (efisiensi data) pada tabel dimensi. Tabel dimensi tidak dinormalisasi dan dapat berisi data redundan.	Menggunakan normalisasi pada tabel dimensi (biasanya hingga 3NF). Tabel dimensi dipecah menjadi beberapa tabel yang terhubung untuk mengurangi redundansi.
Kompleksitas desain/skema	Lebih sederhana dengan struktur mirip bintang - satu tabel fakta di tengah yang terhubung langsung ke tabel dimensi.	Lebih kompleks dengan struktur seperti kepingan salju - tabel dimensi terfragmentasi menjadi sub-dimensi yang saling terhubung.
Kompleksitas query	Query lebih sederhana karena memerlukan lebih sedikit join (biasanya hanya antara tabel fakta dan dimensi).	Query lebih kompleks karena memerlukan lebih banyak join untuk menghubungkan tabel fakta dengan sub-dimensi data yang terpecah.
Performa query	Umumnya lebih cepat karena lebih sedikit join dan struktur yang lebih sederhana. Optimal untuk operasi OLAP dan analisis data.	Relatif lebih lambat karena memerlukan lebih banyak join. Namun, untuk query yang hanya mengakses sub-dimensi tertentu bisa lebih efisien.
Storage	Membutuhkan lebih banyak ruang penyimpanan karena redundansi data dalam tabel dimensi yang tidak dinormalisasi.	Lebih hemat ruang penyimpanan karena mengurangi redundansi data melalui normalisasi.
Integritas data	Integritas data lebih rendah karena redundansi dapat menyebabkan inkonsistensi jika pembaruan tidak dilakukan dengan benar.	Integritas data lebih tinggi karena normalisasi mengurangi redundansi dan membantu menjaga konsistensi data.
Maintenance (pengisian data dengan proses ETL dari OLTP)	Proses ETL lebih sederhana karena struktur yang lebih sederhana, tetapi pembaruan dimensi bisa lebih kompleks karena redundansi data.	Proses ETL lebih kompleks karena data harus didistribusikan ke beberapa tabel dimensi, tetapi pembaruan dimensi lebih mudah karena data terpecah dengan baik.

2. Gambar berikut menunjukkan skema OLTP database dari sebuah sistem informasi ekspedisi. Buatlah data warehouse dalam star schema yang digunakan sebagai dasar analisis performa ekspedisi.



A. Buat Database Baru

```

1 CREATE DATABASE dw_ekspedisi;
1 CREATE DATABASE ekspedisi_olap;
  
```

B. Buat Table Dimensi OLAP

1. dimWaktu

```

1 CREATE TABLE DimWaktu (
2     WaktuID INT AUTO_INCREMENT PRIMARY KEY,
3     Tanggal DATE NOT NULL,
4     Hari INT NOT NULL,
5     Bulan INT NOT NULL,
6     Tahun INT NOT NULL,
7     Kuartal INT NOT NULL,
8     NamaBulan VARCHAR(20) NOT NULL,
9     NamaHari VARCHAR(20) NOT NULL,
10    IsAkhirPekan BOOLEAN NOT NULL
11 );
  
```

2. dimLokasi

```

1 CREATE TABLE DimLokasi (
2     LokasiID INT AUTO_INCREMENT PRIMARY KEY,
3     KecamatanID INT NOT NULL,
4     NamaKecamatan VARCHAR(100) NOT NULL,
5     KotaID INT NOT NULL,
6     NamaKota VARCHAR(100) NOT NULL,
7     ProvinsiID INT NOT NULL,
8     NamaProvinsi VARCHAR(100) NOT NULL);
  
```

3. dimPelanggan

```
1 CREATE TABLE DimPelanggan (  
2     PelangganID INT AUTO_INCREMENT PRIMARY KEY,  
3     NamaPelanggan VARCHAR(100) NOT NULL,  
4     AlamatLengkap TEXT);
```

4. dimKurir

```
1 CREATE TABLE DimKurir (  
2     KurirID INT AUTO_INCREMENT PRIMARY KEY,  
3     NamaKurir VARCHAR(100) NOT NULL,  
4     TipeKendaraan VARCHAR(50) NOT NULL,  
5     NoHP VARCHAR(20) NOT NULL);
```

5. dimStatusPengiriman

```
1 CREATE TABLE DimStatusPengiriman (  
2     StatusPengirimanID INT AUTO_INCREMENT PRIMARY KEY,  
3     NamaStatus VARCHAR(50) NOT NULL);
```

6. dimPembayaran

```
1 CREATE TABLE DimPembayaran (  
2     PembayaranID INT AUTO_INCREMENT PRIMARY KEY,  
3     TipePembayaranID INT NOT NULL,  
4     NamaTipePembayaran VARCHAR(50) NOT NULL,  
5     StatusPembayaranID INT NOT NULL,  
6     NamaStatusPembayaran VARCHAR(50) NOT NULL,  
7     Total DECIMAL(10,2) NOT NULL);
```

C. Buat Table Fakta OLAP

1. FactPengiriman

```
1 CREATE TABLE FactPengiriman (  
2     PengirimanID INT NOT NULL,  
3     WaktuPengirimanID INT NOT NULL,  
4     WaktuSampaiPerkiraanID INT,  
5     WaktuSampaiAktualID INT,  
6     LokasiAsalID INT NOT NULL,  
7     LokasiTujuanID INT NOT NULL,  
8     PelangganID INT NOT NULL,  
9     KurirID INT NOT NULL,  
10    StatusPengirimanID INT NOT NULL,  
11    PembayaranID INT NOT NULL,  
12    Berat DECIMAL(10,2) NOT NULL,  
13    JarakKM DECIMAL(10,2),  
14    DurasiPengirimanHari INT,  
15    KetepatanWaktu INT, -- 1: Tepat waktu, 0: Terlambat  
16    BiayaPengiriman DECIMAL(10,2),  
  
17    PRIMARY KEY (PengirimanID, WaktuPengirimanID, LokasiAsalID,  
18    LokasiTujuanID, PelangganID, KurirID, StatusPengirimanID, PembayaranID),  
19    FOREIGN KEY (WaktuPengirimanID) REFERENCES DimWaktu(WaktuID),  
20    FOREIGN KEY (WaktuSampaiPerkiraanID) REFERENCES DimWaktu(WaktuID),  
21    FOREIGN KEY (WaktuSampaiAktualID) REFERENCES DimWaktu(WaktuID),  
22    FOREIGN KEY (LokasiAsalID) REFERENCES DimLokasi(LokasiID),  
23    FOREIGN KEY (LokasiTujuanID) REFERENCES DimLokasi(LokasiID),  
24    FOREIGN KEY (PelangganID) REFERENCES DimPelanggan(PelangganID),  
25    FOREIGN KEY (KurirID) REFERENCES DimKurir(KurirID),  
26    FOREIGN KEY (StatusPengirimanID) REFERENCES  
    DimStatusPengiriman(StatusPengirimanID),  
27    FOREIGN KEY (PembayaranID) REFERENCES DimPembayaran(PembayaranID));
```

D. Buat Stored Procedure untuk mengisi dimWaktu

```
1 DELIMITER //
2 CREATE PROCEDURE PopulateDimWaktu()
3 BEGIN
4     DECLARE start_date DATE DEFAULT '2023-01-01';
5     DECLARE end_date DATE DEFAULT '2025-03-18';
6     DECLARE curr_date DATE DEFAULT start_date;
7     WHILE curr_date <= end_date DO
8         INSERT INTO DimWaktu (Tanggal, Hari, Bulan, Tahun, Kuartal, NamaBulan,>NamaHari, IsAkhirPekan)
9         VALUES (
10             curr_date,
11             DAYOFMONTH(curr_date),
12             MONTH(curr_date),
13             YEAR(curr_date),
14             QUARTER(curr_date),
15             MONTHNAME(curr_date),
16             DAYNAME(curr_date),
17             CASE WHEN DAYOFWEEK(curr_date) IN (1, 7) THEN 1 ELSE 0 END);
18         SET curr_date = DATE_ADD(curr_date, INTERVAL 1 DAY);
19     END WHILE;
20 END //
21 DELIMITER ;
```

E. Insert Data

1. dimWaktu

```
1 CALL PopulateDimWaktu();
2 DROP PROCEDURE PopulateDimWaktu;
```

2. dimLokasi

```
1 INSERT INTO DimLokasi (KecamatanID,>NamaKecamatan, KotaID,>NamaKota, ProvinsiID,>NamaProvinsi)
2 VALUES
3     (1, 'Menteng', 1, 'Jakarta Pusat', 1, 'DKI Jakarta'),
4     (2, 'Kebayoran Baru', 2, 'Jakarta Selatan', 1, 'DKI Jakarta'),
5     (3, 'Kelapa Gading', 3, 'Jakarta Utara', 1, 'DKI Jakarta'),
6     (4, 'Tandes', 4, 'Surabaya', 2, 'Jawa Timur'),
7     (5, 'Gubeng', 4, 'Surabaya', 2, 'Jawa Timur'),
8     (6, 'Lowokwaru', 5, 'Malang', 2, 'Jawa Timur'),
9     (7, 'Bandung Wetan', 6, 'Bandung', 3, 'Jawa Barat'),
10    (8, 'Coblong', 6, 'Bandung', 3, 'Jawa Barat'),
11    (9, 'Bogor Selatan', 7, 'Bogor', 3, 'Jawa Barat'),
12    (10, 'Banyumanik', 8, 'Semarang', 4, 'Jawa Tengah'),
13    (11, 'Denpasar Barat', 9, 'Denpasar', 5, 'Bali'),
14    (12, 'Kuta', 10, 'Badung', 5, 'Bali'),
15    (13, 'Klojen', 5, 'Malang', 2, 'Jawa Timur'),
16    (14, 'Wenang', 11, 'Manado', 6, 'Sulawesi Utara');
```

3. dimPelanggan

```
1 INSERT INTO DimPelanggan (NamaPelanggan,>AlamatLengkap)
2 VALUES
3     ('Budi Santoso', 'Jl. Merdeka No. 123, Jakarta Pusat'),
4     ('Dewi Lestari', 'Jl. Sudirman No. 45, Jakarta Selatan'),
5     ('Ahmad Hidayat', 'Jl. Gatot Subroto No. 67, Jakarta Selatan'),
6     ('Siti Rahayu', 'Jl. Pahlawan No. 89, Surabaya'),
7     ('Eko Prasetyo', 'Jl. Diponegoro No. 12, Bandung'),
8     ('Rina Wulandari', 'Jl. Ahmad Yani No. 34, Semarang'),
9     ('Agus Wijaya', 'Jl. Veteran No. 56, Malang'),
10    ('Lina Susanti', 'Jl. Gajah Mada No. 78, Denpasar'),
11    ('Hendro Gunawan', 'Jl. Imam Bonjol No. 90, Makassar'),
12    ('Maya Anggraini', 'Jl. Thamrin No. 23, Manado');
```

4. dimKurir

```
1 INSERT INTO DimKurir (NamaKurir, TipeKendaraan, NoHP)
2 VALUES
3     ('Joko Sudirman', 'Motor', '081234567890'),
4     ('Budi Hartono', 'Motor', '081234567891'),
5     ('Agus Setiawan', 'Motor', '081234567892'),
6     ('Dedi Kurniawan', 'Mobil', '081234567893'),
7     ('Rudi Hermawan', 'Mobil', '081234567894'),
8     ('Andi Wijaya', 'Motor', '081234567895'),
9     ('Bambang Susilo', 'Motor', '081234567896'),
10    ('Cahyo Purnomo', 'Mobil', '081234567897'),
11    ('Deni Santoso', 'Motor', '081234567898'),
12    ('Eko Prasetyo', 'Mobil', '081234567899');
```

5. dimStatusPengiriman

```
1 INSERT INTO DimStatusPengiriman (NamaStatus
2 VALUES
3     ('Pesanan Dibuat'),
4     ('Dikemas'),
5     ('Dikirim'),
6     ('Dalam Perjalanan'),
7     ('Sampai di Kota Tujuan'),
8     ('Diantar ke Alamat Tujuan'),
9     ('Diterima'),
10    ('Gagal Diantar'),
11    ('Dikembalikan'),
12    ('Dibatalkan');
```

6. dimPembayaran

```
1 INSERT INTO DimPembayaran (TipePembayaranID, NamaTipePembayaran, StatusPembayaranID, NamaStatusPembayaran, Total)
2 VALUES
3     (1, 'Transfer Bank', 1, 'Lunas', 50000),
4     (1, 'Transfer Bank', 2, 'Pending', 75000),
5     (1, 'Transfer Bank', 3, 'Gagal', 100000),
6     (2, 'E-Wallet', 1, 'Lunas', 45000),
7     (2, 'E-Wallet', 2, 'Pending', 60000),
8     (3, 'Kartu Kredit', 1, 'Lunas', 85000),
9     (3, 'Kartu Kredit', 3, 'Gagal', 120000),
10    (4, 'COD', 1, 'Lunas', 40000),
11    (4, 'COD', 4, 'Belum Dibayar', 65000),
12    (5, 'QRIS', 1, 'Lunas', 55000);
```

F. Buat Stored Procedure untuk mengisi table fakta dengan 100 data dummy

```
1 DELIMITER //
2 CREATE PROCEDURE PopulateFactPengiriman()
3 BEGIN
4     DECLARE i INT DEFAULT 1;
5     DECLARE waktu_id INT;
6     DECLARE waktu_perkiraan_id INT;
7     DECLARE waktu_aktual_id INT;
8     DECLARE lokasi_asal_id INT;
9     DECLARE lokasi_tujuan_id INT;
10    DECLARE pelanggan_id INT;
11    DECLARE kurir_id INT;
12    DECLARE status_id INT;
13    DECLARE pembayaran_id INT;
14    DECLARE berat DECIMAL(10,2);
15    DECLARE jarak DECIMAL(10,2);
16    DECLARE durasi INT;
17    DECLARE ketepatan INT;
18    DECLARE biaya DECIMAL(10,2);
```

```

20 -- Mendapatkan jumlah maksimum dari setiap tabel dimensi
21 SELECT COUNT(*) INTO @max_waktu FROM DimWaktu;
22 SELECT COUNT(*) INTO @max_lokasi FROM DimLokasi;
23 SELECT COUNT(*) INTO @max_pelanggan FROM DimPelanggan;
24 SELECT COUNT(*) INTO @max_kurir FROM DimKurir;
25 SELECT COUNT(*) INTO @max_status FROM DimStatusPengiriman;
26 SELECT COUNT(*) INTO @max_pembayaran FROM DimPembayaran;

28 WHILE i <= 100 DO
29     -- Menghasilkan ID acak untuk setiap dimensi
30     SET waktu_id = FLOOR(1 + RAND() * @max_waktu);
31     SET waktu_perkiraan_id = waktu_id + FLOOR(1 + RAND() * 5);
32
33     -- 80% pengiriman sudah sampai, 20% belum
34     IF RAND() <= 0.8 THEN
35         SET waktu_aktual_id = waktu_perkiraan_id + FLOOR(-2 + RAND() * 5); -- Bisa lebih cepat atau lambat
36     ELSE
37         SET waktu_aktual_id = NULL; -- Belum sampai
38     END IF;

40     -- Pastikan lokasi asal dan tujuan berbeda
41     SET lokasi_asal_id = FLOOR(1 + RAND() * @max_lokasi);
42     SET lokasi_tujuan_id = FLOOR(1 + RAND() * @max_lokasi);
43     WHILE lokasi_asal_id = lokasi_tujuan_id DO
44         SET lokasi_tujuan_id = FLOOR(1 + RAND() * @max_lokasi);
45     END WHILE;

46
47     SET pelanggan_id = FLOOR(1 + RAND() * @max_pelanggan);
48     SET kurir_id = FLOOR(1 + RAND() * @max_kurir);
49     SET status_id = FLOOR(1 + RAND() * @max_status);
50     SET pembayaran_id = FLOOR(1 + RAND() * @max_pembayaran);

52     -- Menghasilkan nilai metrik
53     SET berat = ROUND(0.5 + RAND() * 20, 2);
54     SET jarak = ROUND(5 + RAND() * 1000, 2);
55     SET durasi = FLOOR(1 + RAND() * 7);
56
57     -- 70% pengiriman tepat waktu
58     IF RAND() <= 0.7 THEN
59         SET ketepatan = 1;
60     ELSE
61         SET ketepatan = 0;
62     END IF;

63
64     -- Biaya pengiriman berdasarkan berat dan jarak
65     SET biaya = ROUND((berat * 10000) + (jarak * 100), 2);

```

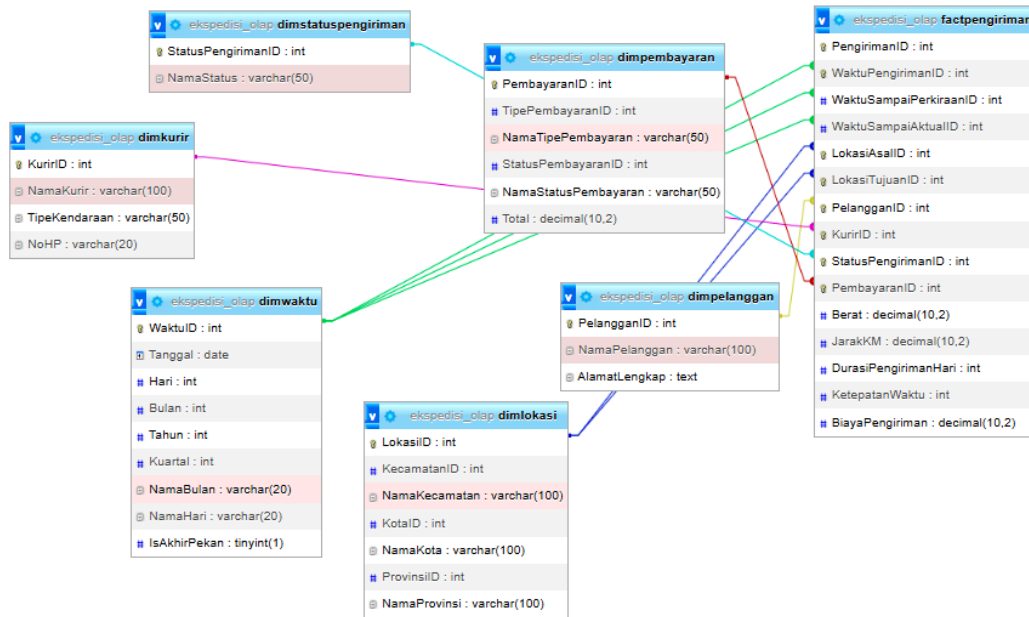
G. Insert Data factPembayaran

```

67 -- Memasukkan data ke tabel fakta
68 INSERT INTO FactPengiriman (
69     PengirimanID, WaktuPengirimanID, WaktuSampaiPerkiraanID, WaktuSampaiAktualID,
70     LokasiAsalID, LokasiTujuanID, PelangganID, KurirID, StatusPengirimanID,
71     PembayaranID, Berat, JarakKM, DurasiPengirimanHari, KetepatanWaktu, BiayaPengiriman
72 ) VALUES (
73     i, waktu_id, waktu_perkiraan_id, waktu_aktual_id,
74     lokasi_asal_id, lokasi_tujuan_id, pelanggan_id, kurir_id, status_id,
75     pembayaran_id, berat, jarak, durasi, ketepatan, biaya
76 );
77 SET i = i + 1;
78 END WHILE;
79 END //

```

H. Struktur database



I. Analisis Performa dengan Query mySQL

1. Analisis Performa Kurir

```
mysql> SELECT
->     dk.KurirID,
->     dk>NamaKurir,
->     dk.TipeKendaraan,
->     COUNT(fp.PengirimanID) AS JumlahPengiriman,
->     ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
->     ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
->     ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalBiayaPengiriman,
->     ROUND(AVG(fp.Berat), 2) AS RataRataBerat
-> FROM
->     FactPengiriman fp
-> JOIN
->     DimKurir dk ON fp.KurirID = dk.KurirID
-> GROUP BY
->     dk.KurirID, dk>NamaKurir, dk.TipeKendaraan
-> ORDER BY
->     PersentaseTepatWaktu DESC;
```

KurirID	NamaKurir	TipeKendaraan	JumlahPengiriman	RataRataDurasi	PersentaseTepatWaktu	TotalBiayaPengiriman	RataRataBerat
1	Joko Sudirman	Motor	3	6.00	100.00	570841.00	1.00
5	Rudi Hermawan	Mobil	11	3.45	81.82	1595747.00	1.00
7	Bambang Susilo	Motor	15	4.07	80.00	2071097.00	1.00
10	Eko Prasetyo	Mobil	14	4.14	78.57	1993176.00	1.00
6	Andi Wijaya	Motor	9	4.11	77.78	1315462.00	1.00
9	Dani Santoso	Motor	8	4.88	75.00	1179745.00	1.00
2	Budi Hartono	Motor	10	4.50	70.00	1389629.00	1.00
4	Dedi Kurniawan	Mobil	8	4.25	62.50	1297556.00	1.00
8	Cahyo Purnomo	Mobil	14	4.86	57.14	2178184.00	1.00
3	Agus Setiawan	Motor	8	3.25	50.00	1428860.00	1.00

```
10 rows in set (0.02 sec)
```

2. Analisis Performa Rute Pengiriman

```
mysql> SELECT
-> dl_asal>NamaKota AS KotaAsal,
-> dl_asal>NamaProvinsi AS ProvinsiAsal,
-> dl_tujuan>NamaKota AS KotaTujuan,
-> dl_tujuan>NamaProvinsi AS ProvinsiTujuan,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(AVG(fp.JarakKM), 2) AS RataRataJarak,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimLokasi dl_asal ON fp.LokasiAsalID = dl_asal.LokasiID
-> JOIN
-> DimLokasi dl_tujuan ON fp.LokasiTujuanID = dl_tujuan.LokasiID
-> GROUP BY
-> dl_asal>NamaKota, dl_asal>NamaProvinsi, dl_tujuan>NamaKota, dl_tujuan>NamaProvinsi
-> ORDER BY
-> JumlahPengiriman DESC
-> LIMIT 10;
```

KotaAsal	ProvinsiAsal	KotaTujuan	ProvinsiTujuan	JumlahPengiriman	RataRataJarak	RataRataDurasi	PersentaseTepatWaktu	TotalPendapatan
Jakarta Pusat	DKI Jakarta	Surabaya	Jawa Timur	3	442.74	4.67	100.00	530221.00
Malang	Jawa Timur	Jakarta Selatan	DKI Jakarta	3	551.08	4.33	33.33	485624.00
Malang	Jawa Timur	Malang	Jawa Timur	3	731.52	5.33	100.00	422355.00
Bandung	Jawa Barat	Surabaya	Jawa Timur	3	613.72	3.67	66.67	648817.00
Bandung	Jawa Barat	Denpasar	Bali	3	621.18	3.00	66.67	545754.00
Semarang	Jawa Tengah	Jakarta Pusat	DKI Jakarta	3	391.32	6.33	66.67	280997.00
Jakarta Utara	DKI Jakarta	Malang	Jawa Timur	3	626.25	3.33	100.00	408476.00
Jakarta Pusat	DKI Jakarta	Jakarta Selatan	DKI Jakarta	2	752.42	4.00	100.00	522184.00
Surabaya	Jawa Timur	Manado	Sulawesi Utara	2	577.11	5.00	100.00	288122.00
Surabaya	Jawa Timur	Denpasar	Bali	2	260.02	3.00	50.00	248503.00

```
10 rows in set (0.03 sec)
```


3. Analisis Performa Tren Pengiriman berdasar Waktu

```
mysql> SELECT
-> dw.Tahun,
-> dw.Bulan,
-> dw>NamaBulan,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(SUM(fp.Berat), 2) AS TotalBerat,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimWaktu dw ON fp.WaktuPengirimanID = dw.WaktuID
-> GROUP BY
-> dw.Tahun, dw.Bulan, dw>NamaBulan
-> ORDER BY
-> dw.Tahun, dw.Bulan;
```

Tahun	Bulan	NamaBulan	JumlahPengiriman	TotalBerat	RataRataDurasi	PersentaseTepatWaktu	TotalPendapatan
2023	1	January	4	32.01	4.00	75.00	594546.00
2023	2	February	6	51.34	5.17	66.67	983173.00
2023	3	March	5	42.78	4.60	100.00	569577.00
2023	4	April	4	45.84	5.25	50.00	645003.00
2023	5	May	2	28.85	3.50	100.00	430690.00
2023	6	June	7	83.48	4.29	85.71	1091556.00
2023	7	July	3	25.39	1.00	66.67	372891.00
2023	8	August	2	26.96	3.00	50.00	363722.00
2023	9	September	4	46.34	3.00	100.00	685130.00
2023	10	October	2	20.79	5.00	100.00	274592.00
2023	11	November	2	15.56	4.50	100.00	201976.00
2023	12	December	5	52.84	4.20	40.00	715167.00
2024	1	January	1	15.19	1.00	100.00	157238.00
2024	2	February	2	33.62	5.50	0.00	464292.00
2024	3	March	1	20.03	6.00	100.00	296067.00
2024	4	April	5	47.09	4.20	60.00	658805.00
2024	5	May	3	25.11	5.33	66.67	384814.00
2024	6	June	5	44.56	4.60	60.00	574200.00
2024	7	July	4	53.36	4.25	75.00	611488.00
2024	8	August	6	63.43	5.00	83.33	956463.00
2024	9	September	4	34.18	3.25	75.00	570518.00
2024	10	October	3	22.31	5.33	100.00	390778.00
2024	11	November	2	26.24	6.00	100.00	350660.00
2024	12	December	8	76.63	4.38	62.50	1173375.00
2025	1	January	3	35.91	2.67	66.67	481801.00
2025	2	February	4	51.36	3.50	50.00	635422.00
2025	3	March	3	21.59	4.00	66.67	386353.00

27 rows in set (0.01 sec)

4. Analisis Kinerja Pengiriman dalam 1minggu

```
mysql> SELECT
-> dw>NamaHari,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimWaktu dw ON fp.WaktuPengirimanID = dw.WaktuID
-> GROUP BY
-> dw>NamaHari
-> ORDER BY
-> FIELD(dw>NamaHari, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday');
```

NamaHari	JumlahPengiriman	RataRataDurasi	PersentaseTepatWaktu	TotalPendapatan
Monday	17	3.88	76.47	2592012.00
Tuesday	10	4.40	70.00	1243571.00
Wednesday	16	4.69	68.75	2763933.00
Thursday	18	4.83	88.89	2858779.00
Friday	14	4.00	64.29	2287846.00
Saturday	7	3.29	57.14	692148.00
Sunday	18	4.06	66.67	2582008.00

7 rows in set (0.00 sec)

5. Analisis Metode Pembayaran

```
mysql> SELECT
-> dp>NamaTipePembayaran,
-> dp>NamaStatusPembayaran,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan,
-> ROUND(AVG(fp.BiayaPengiriman), 2) AS RataRataBiayaPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimPembayaran dp ON fp.PembayaranID = dp.PembayaranID
-> GROUP BY
-> dp>NamaTipePembayaran, dp>NamaStatusPembayaran
-> ORDER BY
-> JumlahPengiriman DESC;
```

NamaTipePembayaran	NamaStatusPembayaran	JumlahPengiriman	TotalPendapatan	RataRataBiayaPengiriman	PersentaseTepatWaktu
Kartu Kredit	Gagal	18	2729043.00	151613.50	72.22
E-Wallet	Pending	16	2098205.00	131137.81	75.00
Transfer Bank	Gagal	14	1988554.00	142039.57	85.71
Transfer Bank	Pending	13	2335857.00	179681.31	61.54
E-Wallet	Lunas	9	1519582.00	168842.44	66.67
Transfer Bank	Lunas	7	931208.00	133029.71	71.43
Kartu Kredit	Lunas	7	995687.00	142241.00	57.14
QRIS	Lunas	7	1136881.00	162411.57	57.14
COD	Belum Dibayar	6	856857.00	142809.50	83.33
COD	Lunas	3	428423.00	142807.67	100.00

10 rows in set (0.00 sec)

6. Analisis Status Pengiriman

```
mysql> SELECT
-> dsp>NamaStatus,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(COUNT(fp.PengirimanID) / (SELECT COUNT(*) FROM FactPengiriman) * 100, 2) AS Persentase,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(AVG(fp.BiayaPengiriman), 2) AS RataRataBiayaPengiriman
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimStatusPengiriman dsp ON fp.StatusPengirimanID = dsp.StatusPengirimanID
-> GROUP BY
-> dsp>NamaStatus
-> ORDER BY
-> JumlahPengiriman DESC;
```

NamaStatus	JumlahPengiriman	Persentase	RataRataDurasi	RataRataBiayaPengiriman
Dibatalkan	19	19.00	3.84	162173.05
Sampai di Kota Tujuan	12	12.00	4.00	167604.33
Gagal Diantar	11	11.00	4.55	148449.27
Dikemas	9	9.00	4.56	144200.44
Dikirim	9	9.00	4.00	167493.11
Diterima	9	9.00	5.00	114148.89
Dikembalikan	9	9.00	4.22	160448.89
Dalam Perjalanan	8	8.00	4.88	118317.00
Pesanan Dibuat	7	7.00	5.43	130423.14
Diantar ke Alamat Tujuan	7	7.00	2.29	165527.86

10 rows in set (0.01 sec)

7. Analisis Keterlambatan Pengiriman - Berdasarkan Jarak

```
mysql> SELECT
-> 'Kurang dari 100 KM' AS KategoriJarak,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.JarakKM < 100
-> UNION ALL
-> SELECT
-> '100-500 KM' AS KategoriJarak,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.JarakKM BETWEEN 100 AND 500
-> UNION ALL
-> SELECT
-> 'Lebih dari 500 KM' AS KategoriJarak,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.JarakKM > 500;
```

KategoriJarak	JumlahPengiriman	JumlahTerlambat	PersentaseKeterlambatan
Kurang dari 100 KM	14	3	21.43
100-500 KM	40	13	32.50
Lebih dari 500 KM	46	12	26.09

3 rows in set (0.00 sec)

8. Analisis Keterlambatan Pengiriman - Berdasarkan Berat

```
mysql> SELECT
-> 'Kurang dari 5 KG' AS KategoriBerat,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.Berat < 5
-> UNION ALL
-> SELECT
-> '5-10 KG' AS KategoriBerat,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.Berat BETWEEN 5 AND 10
-> UNION ALL
-> SELECT
-> 'Lebih dari 10 KG' AS KategoriBerat,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END), 0) AS JumlahTerlambat,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS PersentaseKeterlambatan
-> FROM
-> FactPengiriman fp
-> WHERE
-> fp.Berat > 10;
```

KategoriBerat	JumlahPengiriman	JumlahTerlambat	PersentaseKeterlambatan
Kurang dari 5 KG	22	9	40.91
5-10 KG	25	4	16.00
Lebih dari 10 KG	53	15	28.30

3 rows in set (0.00 sec)

9. Analisis Korelasi Berat dan Jarak dengan Biaya

```
mysql> SELECT
-> CASE
-> WHEN fp.Berat < 5 THEN 'Kurang dari 5 KG'
-> WHEN fp.Berat BETWEEN 5 AND 10 THEN '5-10 KG'
-> ELSE 'Lebih dari 10 KG'
-> END AS KategoriBerat,
-> CASE
-> WHEN fp.JarakKM < 100 THEN 'Kurang dari 100 KM'
-> WHEN fp.JarakKM BETWEEN 100 AND 500 THEN '100-500 KM'
-> ELSE 'Lebih dari 500 KM'
-> END AS KategoriJarak,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(AVG(fp.BiayaPengiriman), 2) AS RataRataBiaya,
-> ROUND(MIN(fp.BiayaPengiriman), 2) AS BiayaMinimum,
-> ROUND(MAX(fp.BiayaPengiriman), 2) AS BiayaMaksimum
-> FROM
-> FactPengiriman fp
-> GROUP BY
-> KategoriBerat, KategoriJarak
-> ORDER BY
-> CASE
-> WHEN KategoriBerat = 'Kurang dari 5 KG' THEN 1
-> WHEN KategoriBerat = '5-10 KG' THEN 2
-> ELSE 3
-> END,
-> CASE
-> WHEN KategoriJarak = 'Kurang dari 100 KM' THEN 1
-> WHEN KategoriJarak = '100-500 KM' THEN 2
-> ELSE 3
-> END;

```

KategoriBerat	KategoriJarak	JumlahPengiriman	RataRataBiaya	BiayaMinimum	BiayaMaksimum
Kurang dari 5 KG	Kurang dari 100 KM	2	25185.00	13438.00	36932.00
Kurang dari 5 KG	100-500 KM	11	57451.27	30098.00	91822.00
Kurang dari 5 KG	Lebih dari 500 KM	9	106580.78	72927.00	136859.00
5-10 KG	Kurang dari 100 KM	2	106303.00	106244.00	106362.00
5-10 KG	100-500 KM	10	102438.00	76850.00	134305.00
5-10 KG	Lebih dari 500 KM	13	138641.77	103140.00	177150.00
Lebih dari 10 KG	Kurang dari 100 KM	10	153754.30	106281.00	205147.00
Lebih dari 10 KG	100-500 KM	19	181690.26	129388.00	229378.00
Lebih dari 10 KG	Lebih dari 500 KM	24	222906.21	173702.00	296385.00

9 rows in set (0.01 sec)

10. Analisis Efisiensi Pengiriman berdasar Tipe Kendaraan

```
mysql> SELECT
-> dk.TipeKendaraan,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(AVG(fp.JarakKM), 2) AS RataRataJarak,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(AVG(fp.JarakKM / NULLIF(fp.DurasiPengirimanHari, 0)), 2) AS KecepatanRataRata,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
-> ROUND(AVG(fp.Berat), 2) AS RataRataBerat,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan,
-> ROUND(SUM(fp.BiayaPengiriman) / SUM(fp.JarakKM), 2) AS PendapatanPerKM
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimKurir dk ON fp.KurirID = dk.KurirID
-> GROUP BY
-> dk.TipeKendaraan
-> ORDER BY
-> KecepatanRataRata DESC;

```

TipeKendaraan	JumlahPengiriman	RataRataJarak	RataRataDurasi	KecepatanRataRata	PersentaseTepatWaktu	RataRataBerat	TotalPendapatan	PendapatanPerKM
Motor	53	479.70	4.26	147.61	73.58	10.21	7955634.00	312.91
Mobil	47	436.16	4.21	139.39	70.21	10.67	7064663.00	344.62

2 rows in set (0.00 sec)

11. Analisis Pengiriman berdasar Kuartal dan Tahun

```
mysql> SELECT
-> dw.Tahun,
-> dw.Kuartal,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(SUM(fp.Berat), 2) AS TotalBerat,
-> ROUND(AVG(fp.DurasiPengirimanHari), 2) AS RataRataDurasi,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan,
-> ROUND(AVG(fp.BiayaPengiriman), 2) AS RataRataBiayaPengiriman
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimWaktu dw ON fp.WaktuPengirimanID = dw.WaktuID
-> GROUP BY
-> dw.Tahun, dw.Kuartal
-> ORDER BY
-> dw.Tahun, dw.Kuartal;

```

Tahun	Kuartal	JumlahPengiriman	TotalBerat	RataRataDurasi	PersentaseTepatWaktu	TotalPendapatan	RataRataBiayaPengiriman
2023	1	15	126.13	4.67	80.00	2147296.00	143153.07
2023	2	13	158.17	4.46	76.92	2167249.00	166711.46
2023	3	9	98.69	2.33	77.78	1421743.00	157971.44
2023	4	9	89.19	4.44	66.67	1191735.00	132415.00
2024	1	4	68.84	4.50	50.00	917597.00	229399.25
2024	2	13	116.76	4.62	61.54	1617819.00	124447.62
2024	3	14	150.97	4.29	78.57	2138469.00	152747.79
2024	4	13	125.18	4.85	76.92	1914813.00	147293.31
2025	1	10	108.86	3.40	60.00	1503576.00	150357.60

9 rows in set (0.00 sec)

12. Analisis Status Pembayaran

```
mysql> SELECT
-> dp>NamaStatusPembayaran,
-> COUNT(fp.PengirimanID) AS JumlahPengiriman,
-> ROUND(SUM(fp.BiayaPengiriman), 2) AS TotalPendapatan,
-> ROUND(SUM(CASE WHEN fp.KetepatanWaktu = 1 THEN 1 ELSE 0 END) / COUNT(fp.PengirimanID) * 100, 2) AS PersentaseTepatWaktu
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimPembayaran dp ON fp.PembayaranID = dp.PembayaranID
-> GROUP BY
-> dp>NamaStatusPembayaran
-> ORDER BY
-> JumlahPengiriman DESC;
```

NamaStatusPembayaran	JumlahPengiriman	TotalPendapatan	PersentaseTepatWaktu
Lunas	33	5011781.00	66.67
Gagal	32	4717597.00	78.13
Pending	29	4434062.00	68.97
Belum Dibayar	6	856857.00	83.33

4 rows in set (0.01 sec)

13. Analisis Biaya Pengiriman per Kilo

```
mysql> SELECT
-> CASE
-> WHEN fp.Berat < 5 THEN 'Kurang dari 5 KG'
-> WHEN fp.Berat BETWEEN 5 AND 10 THEN '5-10 KG'
-> ELSE 'Lebih dari 10 KG'
-> END AS KategoriBerat,
-> COUNT(*) AS JumlahPengiriman,
-> ROUND(AVG(fp.BiayaPengiriman), 2) AS RataRataBiaya,
-> ROUND(AVG(fp.BiayaPengiriman / fp.Berat), 2) AS BiayaPerKilogram
-> FROM
-> FactPengiriman fp
-> GROUP BY
-> KategoriBerat
-> ORDER BY
-> CASE
-> WHEN KategoriBerat = 'Kurang dari 5 KG' THEN 1
-> WHEN KategoriBerat = '5-10 KG' THEN 2
-> ELSE 3
-> END;
```

KategoriBerat	JumlahPengiriman	RataRataBiaya	BiayaPerKilogram
Kurang dari 5 KG	22	74616.41	36424.42
5-10 KG	25	121573.16	16921.72
Lebih dari 10 KG	53	195083.15	13105.74

3 rows in set (0.00 sec)

14. Analisis 10 Pengiriman dengan Biaya Termahal

```
mysql> SELECT
-> fp.PengirimanID,
-> dp>NamaPelanggan,
-> dl_asal>NamaKota AS KotaAsal,
-> dl_tujuan>NamaKota AS KotaTujuan,
-> fp.Berat,
-> fp.JarakKM,
-> fp.BiayaPengiriman,
-> CASE WHEN fp.KetepatanWaktu = 1 THEN 'Tepat Waktu' ELSE 'Terlambat' END AS Status
-> FROM
-> FactPengiriman fp
-> JOIN
-> DimPelanggan dp ON fp.PelangganID = dp.PelangganID
-> JOIN
-> DimLokasi dl_asal ON fp.LokasiAsalID = dl_asal.LokasiID
-> JOIN
-> DimLokasi dl_tujuan ON fp.LokasiTujuanID = dl_tujuan.LokasiID
-> ORDER BY
-> fp.BiayaPengiriman DESC
-> LIMIT 10;
```

PengirimanID	NamaPelanggan	KotaAsal	KotaTujuan	Berat	JarakKM	BiayaPengiriman	Status
57	Budi Santoso	Bandung	Surabaya	20.32	931.85	296385.00	Tepat Waktu
43	Eko Prasetyo	Jakarta Pusat	Jakarta Selatan	20.03	957.67	296067.00	Tepat Waktu
45	Hendro Gunawan	Jakarta Utara	Bandung	19.28	762.51	269051.00	Tepat Waktu
92	Budi Santoso	Makassar	Badung	20.48	597.42	264542.00	Tepat Waktu
30	Siti Rahayu	Bandung	Denpasar	18.29	635.48	246448.00	Tepat Waktu
79	Agus Wijaya	Badung	Jakarta Utara	18.71	580.40	245140.00	Tepat Waktu
8	Siti Rahayu	Denpasar	Manado	18.62	576.50	243850.00	Terlambat
60	Siti Rahayu	Malang	Malang	13.86	973.36	235936.00	Tepat Waktu
71	Rina Wulandari	Manado	Denpasar	15.66	783.14	234914.00	Terlambat
27	Eko Prasetyo	Malang	Jakarta Selatan	15.40	775.44	231544.00	Terlambat

10 rows in set (0.02 sec)