



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

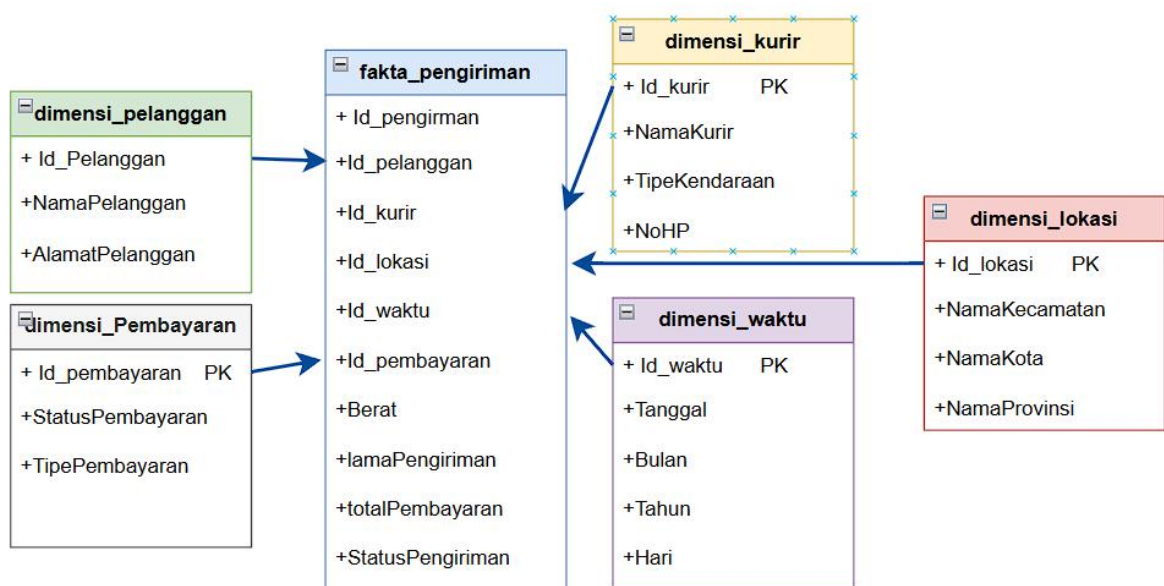
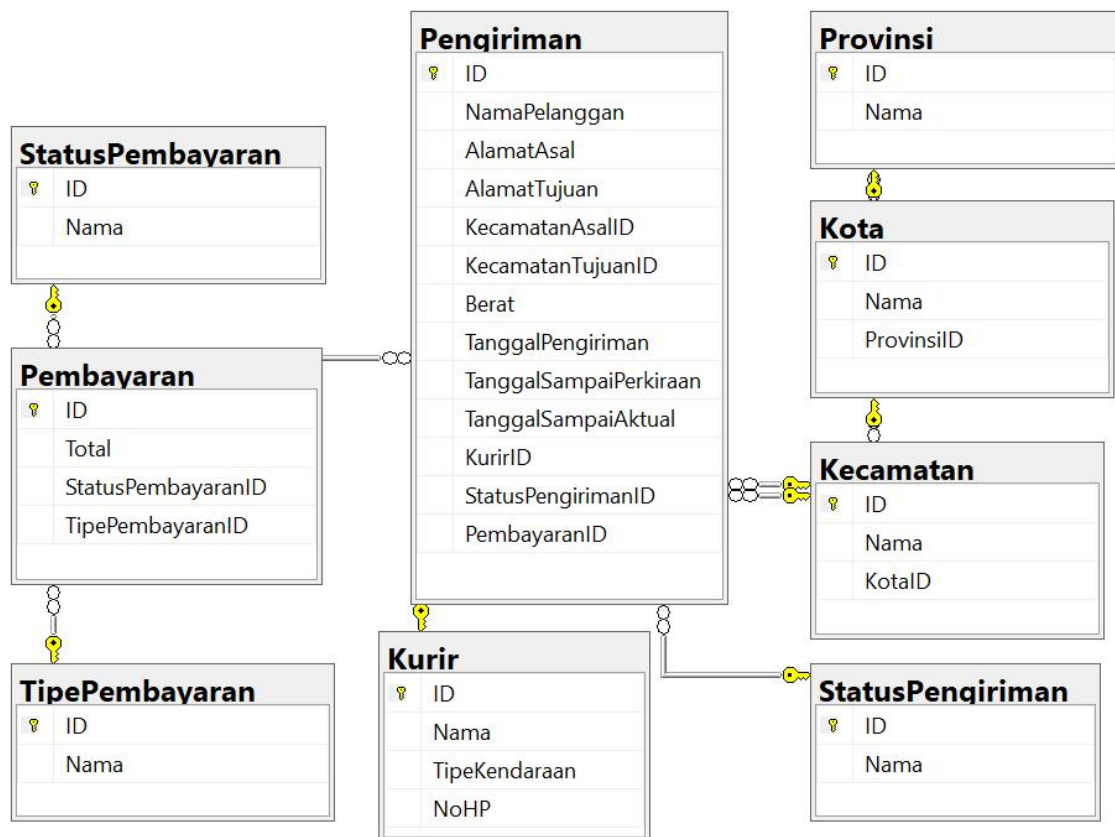
Nama : Khuzaima Filla Januarta

Nomor Urut : 17

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

	Star Schema	Snowflake Schema
Normalisasi	Tidak/Rendah (Denormalisasi)	Tinggi (Ternormalisasi)
Kompleksitas desain/skema	Simple / Direct	Kompleks(Tabel memiliki subdimensi)
Kompleksitas query	Lebih simple karena querynya langsung merujuk ke tabel yang ada, tanpa memerlukan integrasi dengan tabel yang lain.	Perlu banyak join, karena menghubungkan banyak tabel
Performa query	Lebih cepat untuk menganalisa dan agregasi, karena tidak banyak relasi tabel	Lebih lambat/berat dikarenakan banyak query penghubung antar tabel.
Storage	Lebih boros (besar) karena terdapat redundansi data	Lebih hemat (ringan) karena tidak ada redundansi data
Integritas data	Lebih rendah karena ada duplikasi data dalam tabel	Lebih tinggi karena di normalisasikan
Maintenance (pengisian data dengan proses ETL dari OLTP)	Lebih mudah karena tabel yang perlu di edit lebih sedikit atau bahkan hanya 1	Lebih kompleks dan ribet, karena banyaknya tabel yang saling berhubungan/memiliki relationship.

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.



Analisis ekspedisi :

Struktur database ini dirancang untuk mendukung sistem yang melibatkan pelanggan, pembayaran, pengiriman, kurir, waktu, dan lokasi. Performa keseluruhan sistem akan sangat bergantung pada bagaimana tabel-tabel ini diindeks dan dioptimalkan.

Untuk tabel **dimensi_pelanggan**, performa pencarian dan pengambilan data pelanggan akan cepat karena adanya **Id_Pelanggan** sebagai kunci unik. Namun, jika sering dilakukan pencarian berdasarkan **NamaPelanggan** atau **AlamatPelanggan**, perlu dipertimbangkan untuk menambahkan indeks pada kolom tersebut agar query tidak melambat, terutama jika jumlah data pelanggan sangat besar.

Pada tabel **dimensi_Pembayaran**, **Id_pembayaran** sebagai kunci utama akan memastikan performa query yang melibatkan pembayaran tetap cepat. Namun, kolom **StatusPembayaran** dan **TipePembayaran** yang sering digunakan dalam filter atau analisis perlu diindeks untuk menghindari bottleneck, terutama jika tabel ini sering di-join dengan tabel lain.

Tabel **fakta_pengiriman** adalah inti dari sistem ini, karena menghubungkan berbagai entitas seperti pelanggan, kurir, lokasi, waktu, dan pembayaran. Performa query pada tabel ini akan sangat bergantung pada indeks yang tepat pada kunci asing seperti **Id_pelanggan**, **Id_kurir**, **Id_lokasi**, **Id_waktu**, dan **Id_pembayaran**. Selain itu, kolom **Berat** dan **StatusPengiriman** yang sering digunakan dalam perhitungan atau filter juga perlu diindeks untuk memastikan query tetap efisien.

Untuk tabel **dimensi_kurir**, **Id_kurir** sebagai kunci utama akan memastikan performa query yang melibatkan kurir tetap cepat. Namun, jika sering dilakukan pencarian berdasarkan **NamaKurir**, **TipeKendaraan**, atau **NoHP**, perlu dipertimbangkan untuk menambahkan indeks pada kolom tersebut agar query tidak melambat.

Tabel **dimensi_waktu** akan sering digunakan dalam analisis temporal, seperti melihat pengiriman berdasarkan bulan atau tahun. Performa query pada tabel ini akan sangat baik jika kolom **Tanggal**, **Bulan**, **Tahun**, dan **Hari** diindeks dengan tepat. Partisi data berdasarkan waktu juga bisa sangat membantu dalam meningkatkan performa query yang melibatkan data historis.

Terakhir, tabel **dimensi_lokasi** akan digunakan untuk analisis geografis. **Id_lokasi** sebagai kunci utama akan memastikan performa query yang melibatkan lokasi tetap cepat. Namun, jika sering dilakukan pencarian berdasarkan **NamaKecamatan**, **NamaKota**, atau **NamaProvinsi**, perlu dipertimbangkan untuk menambahkan indeks pada kolom tersebut agar query tidak melambat.

Secara keseluruhan, performa database ini akan sangat bergantung pada optimasi indeks dan kunci asing. Query yang melibatkan join antara tabel fakta dan dimensi perlu dioptimalkan dengan indeks yang tepat. Selain itu, partisi data berdasarkan waktu atau lokasi bisa sangat membantu dalam meningkatkan performa query yang melibatkan data historis atau geografis. Dengan optimasi yang tepat, sistem ini dapat menangani volume data yang besar dan query yang kompleks dengan efisien.