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

Nama : Yonanda Mayla Rusdiaty

Nomor Urut : 30

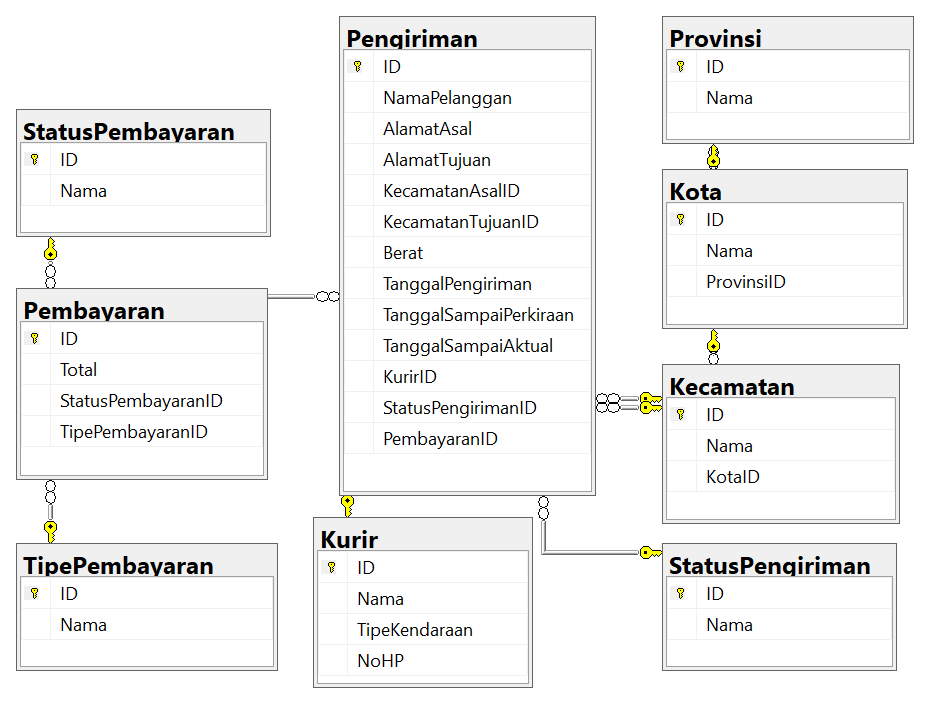
1. Tuliskan perbandingan star schema dan snowflake schema pada tabel berikut:

**Jawab:**

* Star schema: merupakan skema paling sederhana yang memiliki satu tabel tengah yaitu tabel fakta (berisikan key) dimana tabel tersebut terhubung dengan beberapa tabel dimensi lainnya.
* Snowflake schema: merupakan perpanjangan dari Skema Star dengan menambahkan tabel dimensi lainnya dimana tabel dimensi lain tersebut merupakan hasil normalisasi tabel-tabel dimensi yang telah ada sebelumnya

|  |  |  |
| --- | --- | --- |
|  | Star Schema | Snowflake Schema |
| Normalisasi | Tidak dinormalisasi (denormalized), data disimpan secara redundan untuk efisiensi. | Dinormalisasi, data dipecah ke tabel-tabel tambahan untuk mengurangi redundansi. |
| Kompleksitas desain/skema | Desain/skema lebih sederhana dan mudah dipahami, yakni hanya terdiri dari tabel fakta dan tabel dimensi langsung | Desain/skema lebih kompleks karena ada hierarki dimensi yang dinormalisasi ke tabel tambahan. |
| Kompleksitas query | Query lebih sederhana karena jumlah join yang lebih sedikit (hanya fakta dan dimensi) | Query lebih kompleks karena membutuhkan lebih banyak join kibat normalisasi tabel dimensi |
| Performa query | Performa query lebih cepat karena mengurangi jumlah join | Performa query lebih lambat dibanding star schema karena membutuhkan lebih banyak join |
| Storage | Membutuhkan lebih banyak storage karena redundansi data | Lebih hemat storage karena data dinormalisasi dan redundansi dikurangi. |
| Integritas data | Rentan terhadap inkonsistensi data (rendah) karena redundansi. | Lebih baik menjaga integritas data (tinggi) karena normalisasi. |
| Maintenance (pengisian data dengan proses ETL dari OLTP) | Lebih mudah karena struktur sederhana, tapi bisa lebih lambat akibat redundansi. | Lebih kompleks karena perlu mengisi banyak tabel tambahan, tapi lebih efisien |

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



**Jawab:**

Langkah-langkah Membuat Data Warehouse dalam Star Schema untuk Analisis Performa Ekspedisi:

1. Membuat Tabel Fakta (Fact\_ExpeditionPerformance) yang digunakan untuk analisis performa ekspedisi. Atribut-atributnya diperoleh dari tabel Pengiriman dan Pembayaran di skema OLTP, dengan beberapa dihitung atau diturunkan melalui transformasi. Berikut rinciannya:
2. ID\_Pengiriman 🡪 Diambil langsung dari kolom ID di tabel Pengiriman. Digunakan sebagai primary key untuk mengidentifikasi setiap transaksi pengiriman secara unik
3. ID\_Waktu 🡪 Diturunkan dari kolom TanggalPengiriman, TanggalSampaiPerkiraan, atau TanggalSampaiAktual di tabel Pengiriman. Foreign key ke Dim\_Waktu, dibuat berdasarkan tanggal-tanggal tersebut untuk analisis berdasarkan waktu (misalnya, hari, bulan, tahun)
4. ID\_Pelanggan 🡪 Diturunkan dari kolom NamaPelanggan di tabel Pengiriman. Foreign key ke Dim\_Pelanggan, dibuat untuk mengidentifikasi pelanggan berdasarkan nama mereka sebagai pengganti ID unik (karena skema OLTP tidak memiliki ID pelanggan terpisah).
5. ID\_Kurir 🡪 Diambil langsung dari kolom KurirID di tabel Pengiriman. Foreign key ke Dim\_Kurir, menghubungkan pengiriman dengan detail kurir yang menanganinya.
6. ID\_LokasiAsal 🡪 Diturunkan dari kolom KecamatanAsalID di tabel Pengiriman, yang terkait dengan tabel Kecamatan, Kota, dan Provinsi. Foreign key ke Dim\_Lokasi, mewakili lokasi asal pengiriman setelah denormalisasi (menggabungkan kecamatan, kota, provinsi)
7. ID\_LokasiTujuan 🡪 Diturunkan dari kolom KecamatanTujuanID di tabel Pengiriman, yang terkait dengan tabel Kecamatan, Kota, dan Provinsi. Foreign key ke Dim\_Lokasi, mewakili lokasi tujuan pengiriman setelah denormalisasi.
8. ID\_StatusPengiriman 🡪 Diambil langsung dari kolom StatusPengirimanID di tabel Pengiriman. Foreign key ke Dim\_StatusPengiriman, menghubungkan status pengiriman (misalnya, "Sukses" atau "Tertunda")
9. ID\_TipePembayaran 🡪 Diturunkan dari kolom PembayaranID di tabel Pengiriman, yang terkait dengan tabel Pembayaran, lalu ke TipePembayaranID di tabel Pembayaran. Foreign key ke Dim\_TipePembayaran, menghubungkan tipe pembayaran (misalnya, "Tunai" atau "Transfer")
10. Waktu\_Pengiriman 🡪 Dihitung dari selisih antara TanggalSampaiAktual dan TanggalPengiriman di tabel Pengiriman. Metrik kuantitatif yang menunjukkan durasi pengiriman dalam hari untuk mengukur efisiensi.
11. Keterlambatan 🡪 Dihitung dari selisih antara TanggalSampaiAktual dan TanggalSampaiPerkiraan di tabel Pengiriman. Metrik kuantitatif yang menunjukkan apakah pengiriman terlambat (nilai positif) atau lebih cepat (nilai negatif)
12. Berat\_Pengiriman 🡪 Diambil langsung dari kolom Berat di tabel Pengiriman. Metrik kuantitatif untuk analisis beban pengiriman
13. Total\_Biaya 🡪 Diambil langsung dari kolom Total di tabel Pembayaran, yang dihubungkan melalui PembayaranID. Metrik kuantitatif untuk analisis finansial pengiriman
14. Menambahkan Tabel Dimensi:
15. Dim\_Waktu 🡪 Memungkinkan analisis performa ekspedisi berdasarkan dimensi waktu
16. Dim\_Pelanggan 🡪 Memungkinkan analisis performa ekspedisi dari perspektif pelanggan
17. Dim\_Kurir 🡪 Memungkinkan analisis performa ekspedisi dari perspektif kurir
18. Dim\_Lokasi 🡪 Memungkinkan analisis performa ekspedisi berdasarkan lokasi asal dan tujuan pengiriman
19. Dim\_StatusPengiriman 🡪 Memungkinkan analisis performa berdasarkan status pengiriman
20. Dim\_TipePembayaran 🡪 Memungkinkan analisis performa ekspedisi berdasarkan metode pembayaran

