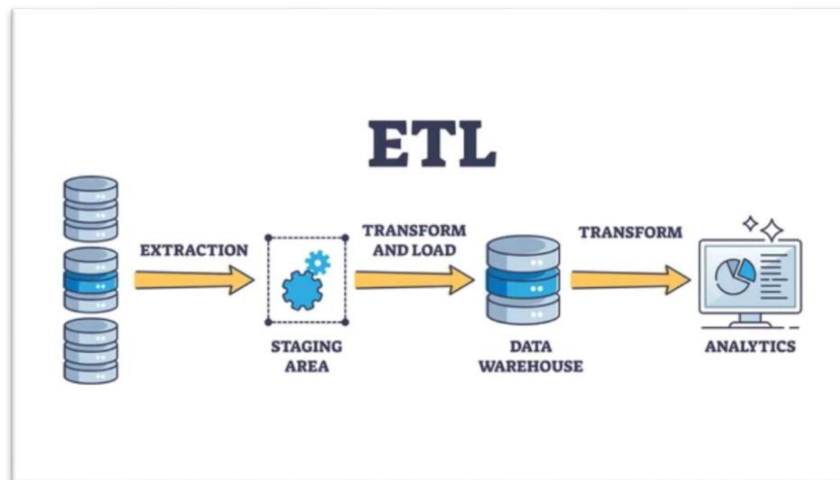


BAB III

LANDASAN TEORI

3.1 *Data Warehouse*

Data warehouse merupakan suatu penyimpanan sekumpulan data yang diperoleh dari berbagai sumber (*resource*) yang berasal dari berbagai sistem dan aplikasi, kemudian data-data tersebut digabungkan dan disesuaikan baik format dan isinya, setelah itu dilakukan proses *ETL* (*Extract, Transform* dan *Load*).



Gambar 3.1 Proses *ETL* (*Extract Transform Loading*)

a. *Extract*

Merupakan proses pengambilan data dari sumber-sumber data baik dari server SQL ataupun file dokumen seperti excel. Dikarenakan data-data tersebut masih dalam keadaan format yang berbeda dan dimungkinkan data tersebut

masih terdapat corrupt, sehingga data tersebut perlu dikumpulkan di suatu penyimpanan khusus yang dinamakan area *staging*.

b. Transform

Proses *ETL* selanjutnya adalah *Transform* yaitu proses untuk menyamakan format yang berbeda dari data-data tersebut. Di dalam proses *transform* ini meliputi:

- *filtering*, menyaring data dengan *filter* tertentu
- *cleaning*, menyesuaikan format penulisan, misalnya “Pria” diubah menjadi “P”
- *joining*, ciri data yang serupa menjadi satu
- *splitting*, memecah ciri data yang berbeda menjadi dua atau lebih dengan begitu data yang sudah rapi, siap disimpan ke dalam *data warehouse*.

c. Loading

Proses yang terakhir adalah proses untuk memasukkan data ke dalam *data warehouse* atau yang disebut *Loading*.

Data warehouse ini memiliki karakteristik yaitu berorientasi pada subjek, terintegrasi, koleksi tetap (*Non-volatile*), dan memiliki dimensi waktu.

Tabel-tabel dalam *data warehouse* ini biasanya ditransformasikan menjadi tabel fakta dan tabel dimensi. Dari beberapa tabel tersebut kemudian bisa diolah lagi agar dapat digunakan untuk berbagai keperluan seperti analisa dan laporan. Pemodelan *data warehouse* diantaranya: *star schema*, *snowflake schema*, *fact constellation schema*.

3.2 MySQL

MySQL merupakan suatu perangkat lunak sistem manajemen basis data atau DBMS (*Data Base Management System*) yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.



Gambar 3.2 Logo MySQL

MySQL pada awalnya diciptakan pada tahun 1979, oleh Michael "Monty" Widenius, seorang programmer komputer asal Swedia. Monty mengembangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi *low-level* ISAM database *engine* dengan *indexing*. Pada saat itu Monty bekerja pada perusahaan bernama TcX di Swedia.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya. SQL (*Structured Query Language*) adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi

dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

3.3 OLAP (*OnLine Analytical Processing*)

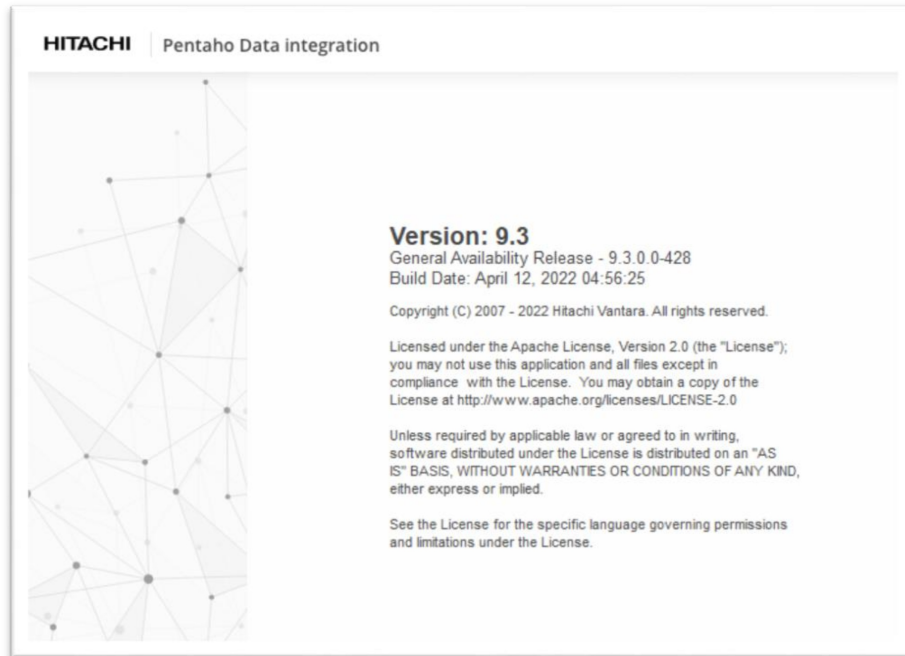
OLAP (*OnLine Analytical Processing*) merupakan teknologi dalam menganalisa multidimensional suatu database yang berukuran besar. Database OLAP dibagi menjadi satu atau beberapa kubus, dan setiap kubus diatur dan dirancang oleh *administrator* kubus agar sesuai dengan cara mengambil dan menganalisis data sehingga lebih mudah membuat dan menggunakan laporan *PivotTabel* dan laporan *PivotChart* yang diperlukan. Aplikasi ini mendukung dalam proses analisis yang kompleks untuk pengambilan suatu keputusan.

Database OLAP dirancang untuk mempercepat pengambilan data. Karena jika dibandingkan antara server OLAP dan Microsoft Office Excel, menghitung nilai data yang diringkas, lebih sedikit data yang perlu dikirim ke Excel saat membuat atau mengubah laporan. Pendekatan ini memungkinkan kita bekerja dengan jumlah data sumber yang jauh lebih besar daripada yang kita dapatkan jika data ditata dalam database tradisional, di mana Excel mengambil semua catatan individu lalu menghitung nilai yang diringkas.

3.4 Pentaho Data Integration (PDI)/Kettle

Pentaho *Data Integration (PDI)* atau *Kettle* merupakan perangkat lunak *Business Intelligence* yang menyediakan fitur untuk digunakan sebagai perangkat lunak pengolah data berukuran besar dan kompleks dengan fitur

diantaranya integrasi data, layanan *OLAP*, dashboard informasi, penggalian informasi dan ekstraksi data, serta untuk pelaporan.



Gambar 3.3 Aplikasi *Pentaho Data Integration*

Selain sifatnya gratis dan adopsi yang semakin hari semakin luas, dukungan Pentaho bisa didapatkan dari Pentaho corp dalam bentuk *Service Level Agreement (SLA)* dan dipaketkan dalam versi *Enterprise Edition* yang sifatnya *annual subscription* atau perlu kontrak tahunan. Selain itu jika Anda tetap menggunakan *community edition* yang gratis, maka bisa mendapatkan dukungan dari banyak *system integrator* Pentaho di seluruh dunia termasuk di Indonesia.

Kumpulan aplikasi Pentaho diantaranya:

- a. Pentaho *Reporting*
- b. Pentaho *Analys / Mondrian OLAP Engine*
- c. Pentaho *Data Integration / Kettle*
- d. Pentaho *Data Mining / Weka*
- e. Pentaho *BI Server / Platform*

3.5 *Nine Step Methodology*

Metodologi Kimball & Ross (2010) yaitu metode *Nine-Step* dimana terdapat 9 langkah dalam membangun sebuah *data warehouse*. Adapun tahapan-tahapannya sebagai berikut:

a. *Choose the Process*

Tahapan pertama kali yaitu menentukan subyek permasalahan yang dihadapi dengan tujuan untuk merancang proses yang akan digunakan.

b. *Choose the Grain*

Memilih *grain* berarti menentukan apa yang akan diwakili atau dipresentasikan oleh sebuah tabel fakta. Dalam hal ini tabel faktanya adalah klasifikasi.

c. *Identify and Conform the Dimension*

Identifikasi dan menghubungkan antara tabel fakta dan dimensi.

d. *Choose the Facts*

Grain dari suatu tabel fakta menentukan fakta-fakta yang bisa digunakan. Pada tahap ini, tentukan *measure* yang dibutuhkan pada tabel fakta. Informasi apa saja yang ingin dibutuhkan.

e. *Store Precalculations in the Fact Table*

Pada tahap ini, hasil perhitungan pada suatu atribut perlu dipertimbangkan untuk disimpan di database. Hal ini untuk mengurangi resiko kesalahan pada program setiap kali melakukan perhitungan pada atribut-atribut tersebut.

f. *Round Out the Dimension Tables*

Dari dimensi-dimensi yang telah diidentifikasi, selanjutnya dibuat deskripsi yang memuat informasi terstruktur mengenai atribut-atribut pada tabel dimensi. Tabel dimensi tersebut harus diberi keterangan secara lengkap dan mudah dipahami oleh pengguna.

g. *Choose the Durations of the Database*

Durasi waktu dari data-data yang akan dimasukkan ke dalam *data warehouse* akan ditentukan pada tahap ini. Misalnya, data perusahaan dua tahun lalu atau lebih diambil dan dimasukkan ke dalam tabel fakta.

h. *Determine the Need to Track Slowly Changing Dimensions*

Dimensi dapat berubah dengan lambat dan menjadi sebuah masalah. Terdapat tiga tipe dasar dari perubahan dimensi yang lambat, yaitu: menulis

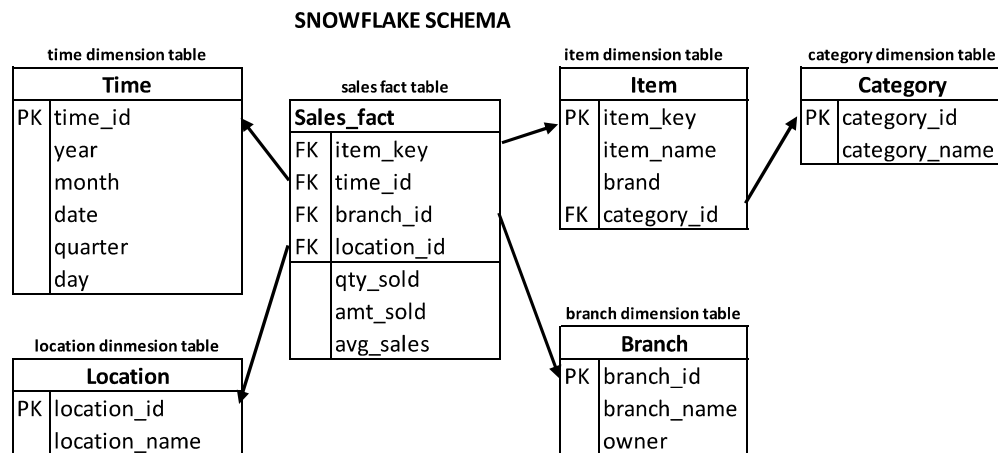
ulang atribut yang berubah, membuat *record* baru pada dimensi, membuat suatu atribut alternatif untuk menampung nilai yang baru.

i. *Decide the Physical Design*

Pada tahap ini, dilakukan perancangan fisik dari *data warehouse*. Selain itu, penentuan masalah-masalah yang mungkin ada pada perancangan fisik.

3.6 Skema Kepingan Salju (*Snowflake Schema*)

Snowflake schema merupakan desain skema tabel dimana terdapat sebuah tabel fakta dengan n-dimensi berada di tengah-tengah yang berfungsi sebagai penghubung antara tabel-tabel dimensi yang ada di sekitarnya.



Gambar 3.4 Skema Kepingan Salju (*Snowflake*)

Tabel fakta merupakan tabel yang menyimpan data pengukuran, metrik, atau fakta yang terkait dengan operasi bisnis. Dalam skema bintang, tabel ini

berada di tengah dan dikelilingi oleh tabel-tabel dimensi. Tabel fakta memiliki kunci yang berasal dari kunci utama (*primary key*) tabel-tabel dimensi yang terhubung dengan tabel fakta. Dari gambar skema terdapat 5 tabel dimensi yaitu: tabel *Time*, tabel *Location*, tabel *Item*, tabel *Category* dan tabel *Branch*. Pada skema *snowflake* ini dimungkinkan tabel dimensi juga memiliki kunci asing dari tabel dimensi lain seperti pada gambar 3.4, tabel *Item* memiliki 1 kunci asing dari tabel *Category*.

Sedangkan tabel dimensi merupakan tabel yang berisi deskripsi objek dalam tabel fakta dan memberikan informasi tentang dimensi seperti nilai, karakteristik, dan kunci. Kunci utama (*primary key*) dari tabel dimensi nantinya akan digunakan untuk menautkan ke tabel fakta dan menjadi kunci asing (*foreign key*), yaitu *item_key*, *time_id*, dan *branch_id*.

Adapun kelebihan model skema *snowflake* adalah:

- a. Ukuran data lebih kecil di dalam tempat penyimpanan
- b. Lebih mudah dilakukan maintenance dan update
- c. Proses query lebih cepat pada saat proses *ETL*

Sedangkan kekurangan model skema *snowflake* adalah:

- a. Cenderung lebih sulit dipahami karena kompleksitasnya
- b. Sulit mencari data yang dibutuhkan karena melihat strukturnya yang kompleks dan bercabang-cabang