

BAB II

LANDASAN TEORI

2.1 PT. Garuda Indonesia (Persero)

Perusahaan yang dijadikan objek penelitian oleh penulis dalam bentuk perseroan terbatas yaitu PT. Garuda Indonesia. Perusahaan ini didirikan pada tahun 1949. PT. Garuda Indonesia berkantor pusat di Jakarta.

2.1.1 Latar Belakang Perusahaan

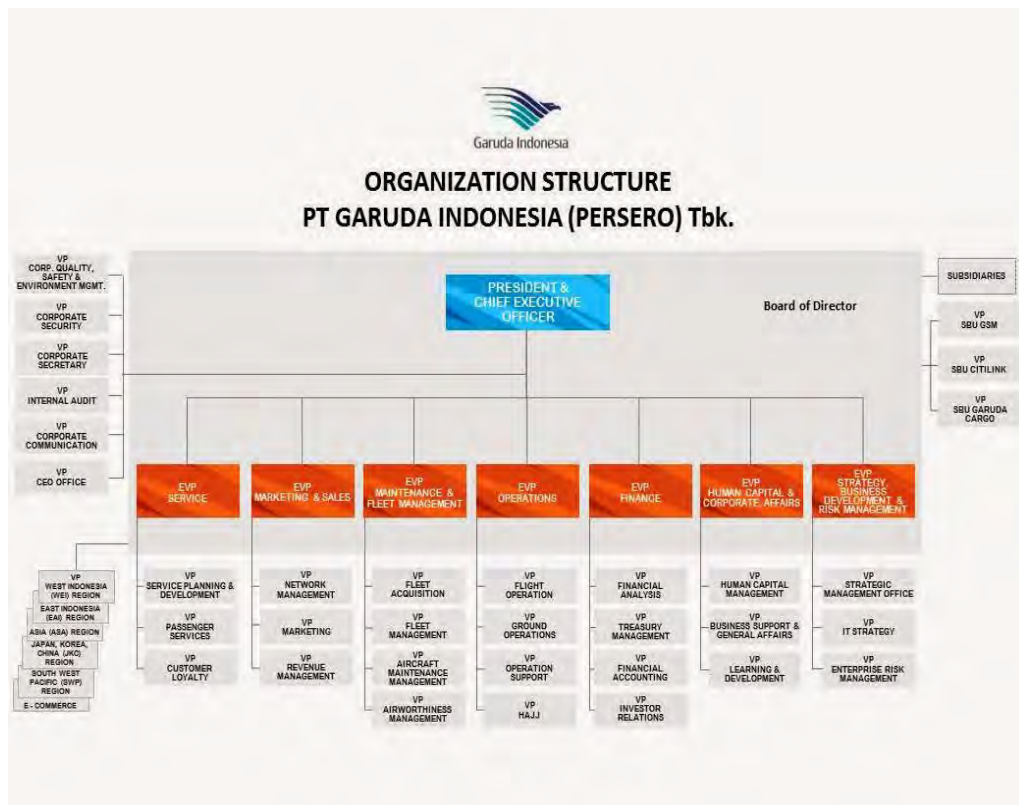
Garuda Indonesia mendapatkan konsesi monopoli penerbangan dari Pemerintah Republik Indonesia pada tahun 1950 dari *Koninklijke Nederlandsch Indie Luchtvaart Maatschappij* (KNILM), perusahaan penerbangan nasional Hindia Belanda. Garuda adalah hasil joint venture antara Pemerintah Indonesia dengan maskapai Belanda *Koninklijke Luchtvaart Maatschappij* (KLM). Pada awalnya, Pemerintah Indonesia memiliki 51% saham dan selama 10 tahun pertama, perusahaan ini dikelola oleh KLM. Pada tahun 1954, KLM menjual sebagian dari sahamnya ke pemerintah Indonesia. Pada mulanya, Garuda memiliki 27 pesawat terbang, staf terdidik, bandara dan jadwal penerbangan, sebagai kelanjutan dari KNILM. Saat ini, Garuda Indonesia telah menerapkan standar internasional sesuai dengan standar ISO 9001 – 2001. (Management, 2007)

2.1.2 Visi dan Misi Perusahaan

Visi PT. Garuda Indonesia adalah menjadi perusahaan penerbangan yang handal dengan menawarkan pelayanan berkualitas kepada masyarakat dunia menggunakan keramahan Indonesia. Misinya adalah sebagai perusahaan penerbangan pembawa bendera bangsa (*flag carrier*) Indonesia yang mempromosikan Indonesia kepada dunia guna menunjang pembangunan ekonomi nasional dengan memberikan pelayanan yang profesional dan pelayanan penerbangan yang menguntungkan.

2.1.3 Struktur Organisasi Perusahaan

PT. Garuda Indonesia (Persero) memiliki batang-batang organisasi atau struktur organisasi yang berfungsi sesuai dengan perannya masing-masing. Berikut bagan struktur PT. Garuda Indonesia (Persero):



Gambar 2.1 Struktur Organisasi PT.Garuda Indonesia

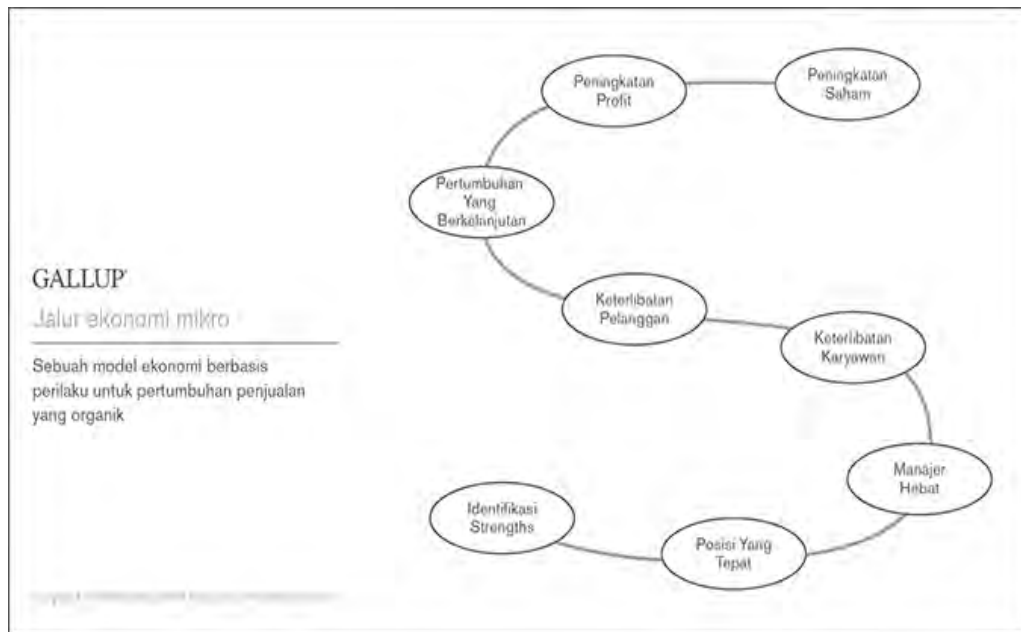
Direktorat Strategi dan Umum (*EVP Strategy & Corp. Affairs*) *Corporate Planning, Human Resources Development, Corporate Communication* dan *Corporate Security* adalah unit-unit yang menjalankan fungsi strategis dan fungsi umum yang dalam kegiatannya bertanggung jawab kepada Direktur Strategi dan Umum.

2.2 Employee Engagement Survey

Employee engagement pertama kali diperkenalkan oleh kelompok peneliti Gallup pada tahun 2004. *Employee engagement* telah diklaim dapat memprediksikan peningkatan produktivitas pada karyawan, profitabilitas, mempertahankan karyawan, kepuasan konsumen serta keberhasilan untuk organisasi, sehingga topik ini menjadi isu yang hangat diperbincangkan dikalangan akademisi dan profesional.

Dalam literatur akademik ada banyak definisi yang menjelaskan arti dari *employee engagement*. Ada yang mendefinisikan *employee engagement* sebagai bentuk keterlibatan individual dan kepuasannya serta sebagai bentuk antusiasme dalam melakukan pekerjaan. Lebih lanjut, *employee engagement* dijelaskan sebagai

sebuah konsep yang dinilai dapat mengatur upaya-upaya karyawan yang sifatnya sukarela, yaitu ketika karyawan memiliki pilihan-pilihan, mereka akan bertindak lebih jauh untuk kepentingan organisasi mereka. Karyawan yang terikat adalah seorang yang terlibat penuh dalam pekerjaannya dan sangat antusias terhadap pekerjaan.



Gambar 2.2 Gallup Jalur Ekonomi Mikro

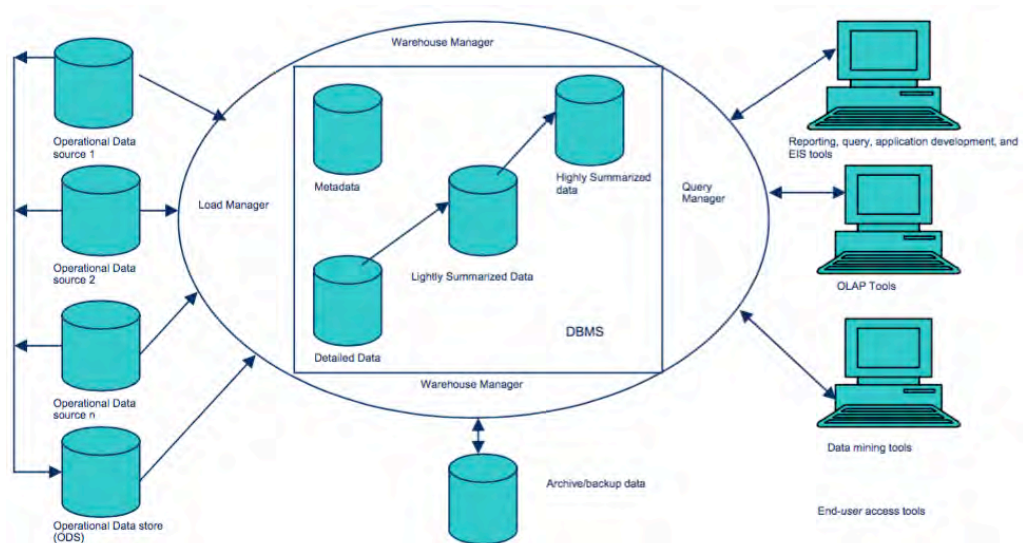
Feeling engaged bukan hanya dapat mendorong perkembangan perusahaan atau organisasi tempat karyawan tersebut bekerja, tetapi juga pertumbuhan ekonomi suatu negara. (Gallup Press, 2007)

2.3 Konsep Data Warehouse

Pengertian data warehouse menurut (Inmon, 2005), “*a data warehouse is a subject oriented, nonvolatile, time variant collection of data in support of management’s decisions* ” atau dapat diartikan “ data warehouse adalah koleksi data yang mempunyai sifat berorientasi subjek, terintegrasi , tidak mengalami perubahan dan mempunyai variasi waktu yang digunakan untuk mendukung proses pengambilan keputusan manajemen”.

Jadi dapat disimpulkan bahwa data warehouse adalah kumpulan data yang telah diringkas dan terintegrasi dari data operasional maupun data external, yang memiliki karakteristik subject-oriented, integrated, nonvolatile dan time variant yang berguna dalam pengambilan keputusan.

2.3.1 Arsitektur Data Warehouse



Gambar 2.3 Arsitektur Data warehouse

Menurut (Connolly & Carolyn E, 2005), komponen utama data warehouse, antara lain :

a) Data Operasional

Data operasional adalah data yang digunakan untuk mendukung proses bisnis sehari-hari.

b) *Operational Data Store* (ODS)

Operational data store adalah tempat penyimpanan data operasional yang bersifat current dan terintegrasi yang digunakan untuk analisis. Atau dengan kata lain, ODS mendukung proses transaksi operasional maupun proses analisis. Dengan adanya ODS maka pembangunan data warehouse menjadi lebih mudah karena ODS dapat menyediakan data yang telah diekstrak dari sumber dan telah dibersihkan sehingga proses pengintegrasian dan restrukturisasi data untuk data warehouse menjadi lebih sederhana.

c) Load Manager

Disebut juga komponen *front end* menangani semua operasi yang berhubungan dengan fungsi *extract data* (menggambil data) dan fungsi *loading data* (menaruh data) ke dalam *data warehouse*.

d) Warehouse Manager

Warehouse manager menangani semua operasi yang berhubungan dengan *management data* dalam data warehouse. Operasi-operasi yang dijalankan oleh *warehouse manager* mencakup:

- Analisis data untuk menjaga konsistensi data.
- Melakukan transformasi dan penggabungan sumber data dari tempat penyimpanan sementara ke dalam tabel-tabel *data warehouse*.
- Melakukan denormalisasi.
- Melakukan agregasi.
- Menyimpan (archive) dan back-up data.

e) Query Manager

Query manager (disebut juga komponen backend) menangani semua operasi yang berhubungan dengan *management* permintaan user (*user queries*). Operasi yang dijalankan oleh *query manager* meliputi kegiatan mengarahkan permintaan ke tabel-tabel data yang tepat dan melakukan penjadwalan eksekusi terhadap permintaan.

f) Detailed Data

Dalam *data warehouse*, area ini adalah tempat penyimpanan semua *detailed data* dalam skema basis data. *Detailed data* dibagi menjadi 2, yaitu *current detail data* (tempat penyimpanan semua *detailed data* yang bersifat *current*) dan *old detailed data* (tempat penyimpanan semua *detailed data* yang bersifat *old*).

g) Lightly and Highly Summerized Data

Area ini adalah tempat penyimpanan sementara data predefinisi yang teringkas secara *light* dan *high* (*predefined lightly and highly summarized*) yang dihasilkan oleh *warehouse manager*. Tujuan dari ringkasan informasi ini adalah untuk mempercepat tanggapan terhadap permintaan *user*. Ringkasan data di-*update* secara berkala seiring dengan bertambahnya data dalam *data warehouse*.

h) Archive / Backup data

Dalam *data warehouse*, area ini digunakan untuk menyimpan *detailed data* dan data yang telah diringkas. Tujuannya adalah untuk penyimpanan

(*archiving*) dan *backup*. Data kemudian ditransfer ke media penyimpanan seperti *magnetic tape* atau *optical disk*.

i) Metadata

Digunakan untuk menyimpan semua definisi metadata (keterangan tentang data) yang digunakan dalam seluruh proses *warehouse*. Metadata digunakan untuk berbagai tujuan, antara lain:

- proses *extracting dan loading*
- proses *warehouse management*
- sebagian proses *query management*

j) End-User Access Tools

End-user access tools adalah *tools* yang memanfaatkan kegunaan dari *data warehouse*. Kegunaan *data warehouse* tersebut, antara lain untuk pembuatan laporan, OLAP, data mining dan proses informasi eksekutif.

2.3.2 Infrastruktur Data Warehouse

Infrastruktur *data warehouse* terdiri dari *software*, *hardware*, pelatihan-pelatihan dan komponen-komponen lainnya yang memberikan dukungan yang dibutuhkan untuk mengimplementasikan arsitektur data warehouse (Poe, 1998). Salah satu instrumen yang mempengaruhi keberhasilan pengembangan data warehouse adalah pengidentifikasian arsitektur mana yang terbaik dan infrastruktur yang dibutuhkan.

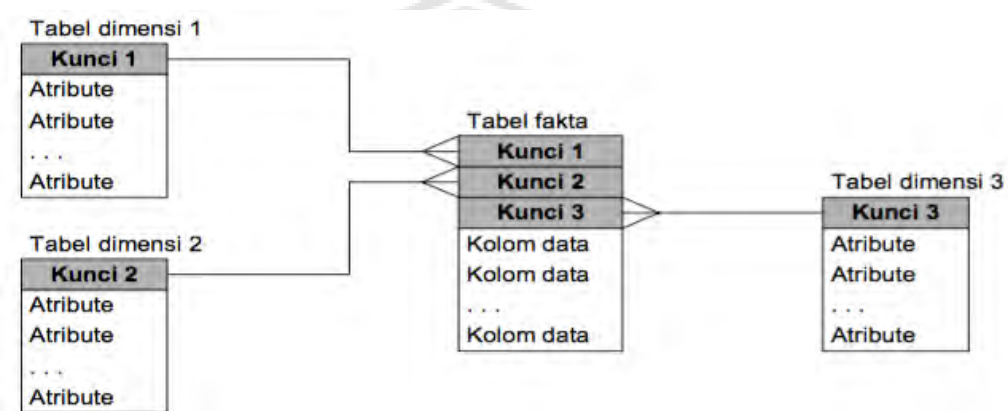
Arsitektur dan infrastruktur sangat erat hubungannya. Arsitektur yang sama mungkin akan membutuhkan infrastruktur yang berbeda, tergantung pada lingkungan perusahaan ataupun organisasi.

2.3.3 Model Dimensional Star Schema

Skema bintang merupakan perancangan yang memiliki struktur sederhana dengan tabel-tabel yang relatif dan penggabungan yang telah diketahui Poe (1998, p191). Skema bintang merupakan suatu rancangan database pada data warehouse yang menggambarkan hubungan yang jelas antara struktur tabel fakta dan tabel dimensi. Skema ini dapat dibaca dengan mudah oleh analis maupun pemakai yang tidak biasa dengan struktur database.

Skema bintang memiliki beberapa keuntungan yang tidak terdapat pada struktur relasional biasa. Berikut merupakan keuntungan dari penggunaan skema bintang:

- Respon data lebih cepat daripada perancangan database operasional.
- Mempermudah dalam hal modifikasi atau pengembangan data warehouse yang terus-menerus.
- End-user dapat menyesuaikan cara berpikir dan menggunakan data.
- Menyederhanakan pemahaman dan penelusuran metadata bagi pemakai dan pengembang.



Gambar 2.4 Model Dimensional *Star Schema*

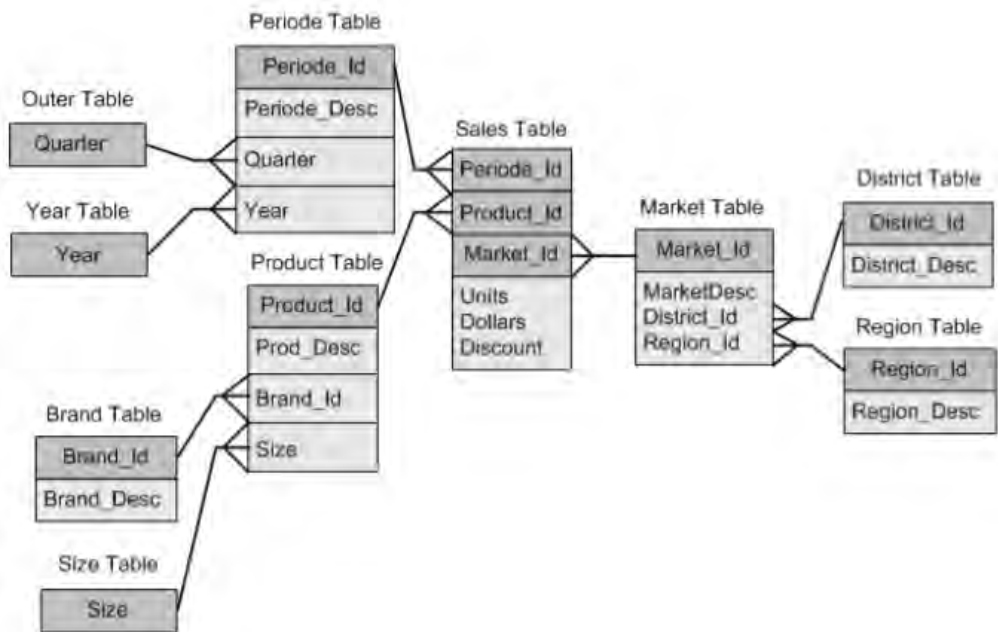
Gambar 2.4 menunjukkan hubungan antara satu (1) tabel fakta dan tiga (3) tabel dimensi. Tabel fakta memiliki *primary key* yang terdiri dari tiga (3) *foreign key*, yaitu kunci ke-1, kunci ke-2, kunci ke-3, yang masing-masing merupakan *primary key* pada tabel dimensi.

2.3.4 Model Dimensional *Snowflake Schema*

Snowflake merupakan variasi lain dari skema bintang dimana tabel dimensi dari skema bintang diorganisasikan menjadi suatu hierarki dengan melakukan normalisasi. Penggunaan tabel dimensi sangatlah menonjol, karena itulah perbedaan dasar dari skema bintang dan skema *snowflake*. Berikut merupakan keuntungan dari penggunaan skema *snowflake* (Poe, 1998):

- Kecepatan dalam pemindahan data dari data *OLAP* ke dalam metadata.

- Sebagai alat pengambil keputusan tingkat tinggi dimana dengan menggunakan skema *snowflake* seperti ini seluruh struktur dapat digunakan



Gambar 2.5 Model Dimensional *Snowflake Schema*

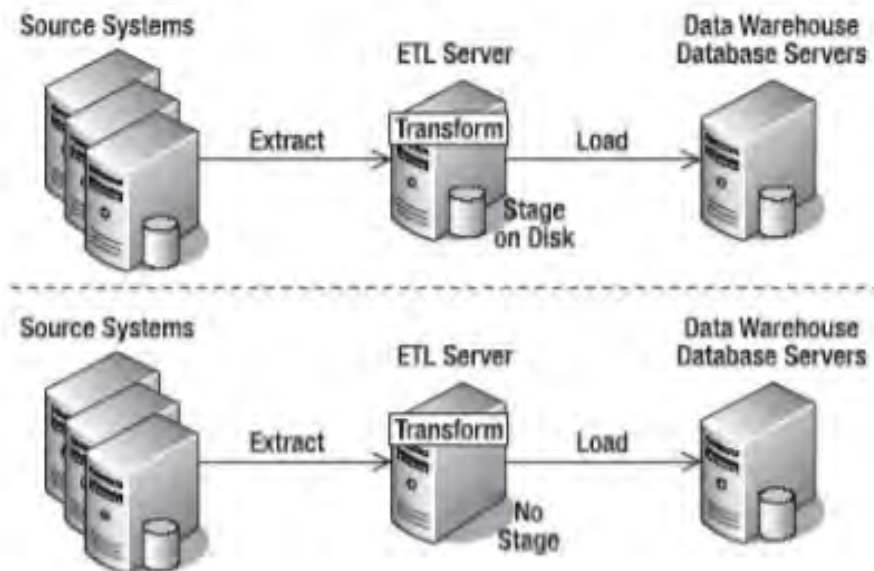
Ciri-ciri skema snowflake adalah sebagai berikut (Poe, 1998):

- Tidak terdapat level pada tabel dimensi.
- Tabel dimensi dinormalisasi dengan dekomposisi pada tabel atribut.
- Setiap dimensi mempunyai satu *key* untuk setiap level pada hierarki dimensi.
- Kunci level terendah menghubungkan tabel dimensi dengan tabel fakta dan tabel atribut level terendah.

2.4 Proses ETL (Extract Transform and Load)

ETL singkatan dari *Extraction, Transformation, and Loading*, adalah proses mengambil dan mengubah data dari sistem sumber dan memasukkannya ke dalam data warehouse (Rainardi, 2008). Dalam proses pengambilan data, data harus bersih agar didapat kualitas data yang baik. Contohnya ada nomor telepon yang invalid, ada kode buku yang tidak eksis lagi, ada beberapa data yang *null*, dan lain sebagainya. Pendekatan tradisional pada proses ETL mengambil data dari data sumber, meletakkan pada *staging area*, dan kemudian mentransform dan meload ke data warehouse. Pada Gambar 2.6 di bawah ini dijelaskan alternative dalam proses

ETL. Proses pertama menggunakan *staging area* dan proses kedua tidak menggunakan *staging area* dalam arti *staging area* tidak disimpan di dalam disk (Rainardi, 2008).



Gambar 2.6 Pendekatan Proses ETL

Memilih tahapan atau proses yang tepat untuk mengkonstruksi *data warehouse* adalah langkah yang kritis dalam pembuatan suatu *data warehouse*. Data pada *data warehouse* harus distandarisasi terlebih dahulu sebelum dimasukkan. Proses yang digunakan dalam memproses data sebelum dimasukkan ke dalam suatu *data warehouse* adalah proses ETL (*extract, transform and load*). Penjelasan dari masing-masing adalah sebagai berikut (Zain, 2008):

a) *Extract*

Proses *extract* adalah proses mengekstrak (*extracting*) dan pengambilan data dari sumber pada sistem untuk di-load ke *data warehouse*. Sumber data dapat diperoleh secara alternatif melalui ODS (*Operational Data Storage*). Data harus dikonstruksi ulang sebelum dimasukkan ke dalam *data warehouse*.

Proses konstruksi ini melibatkan proses:

- *Cleansing data*, yaitu proses pembersihan data kotor. Kotor dalam hal ini adalah data berkualitas rendah seperti ketidak konsistenan penulisan nama, kode id, duplikasi data, data tidak lengkap dan lain-lain.

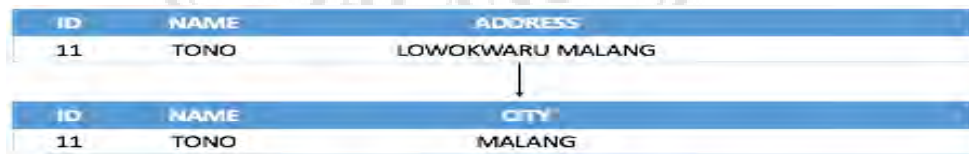
- *Restrukturisasi data* pada *data warehouse* untuk memahami kebutuhan yang ada. Contoh: penambahan atau pengurangan *fields* dan *denormalisasi* data.
- Memastikan sumber data konsisten dengan data yang sudah ada di dalam *data warehouse*.

b) *Transform*

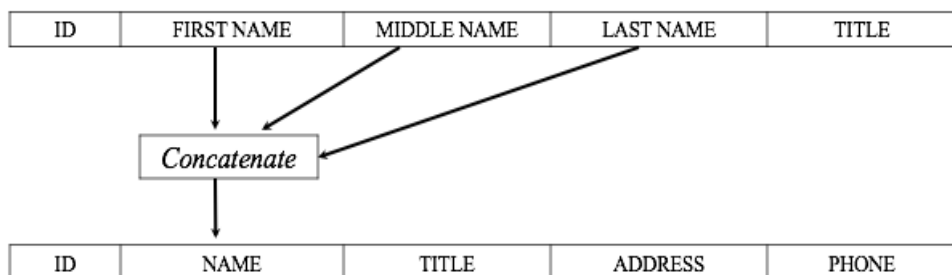
Proses *transform* adalah proses pengubahan data, dimana data yang diperoleh dari proses *extract* (dalam format operasional) menjadi data dalam bentuk *data warehouse*. Proses *transform* ini melibatkan proses:

- *Summarizing the data*, dengan cara pemilihan (*selection*), proyeksi (*projecting*), penggabungan (*joining*), normalisasi (*normalization*), agregasi (*aggregation*) dan *grouping* relasional data menjadi *views* yang lebih nyaman dan berguna bagi pengguna.
- *Packaging the data*, dengan mengkonversi detail data atau *summarized data* menjadi format yang lebih berguna seperti *spread sheets*, *text documents*, *private database*, dan lainnya.

Terdapat dua cara transformasi yaitu dengan menggunakan fungsi *record-level*, dan fungsi *field-level*. Fungsi *record-level* melibatkan proses *Summarizing the data*, sedangkan fungsi *field-level* adalah *single-field* dan *multi-field*. Transformasi *single-field* mengubah satu *field* menjadi *field* yang lain. Berbeda dengan *multi-field* dimana satu data atau lebih diubah menjadi *field* baru.



Gambar 2.7 Contoh Transformasi *Single-Field*.



Gambar 2.8 Contoh Transformasi *Multi-Field*.

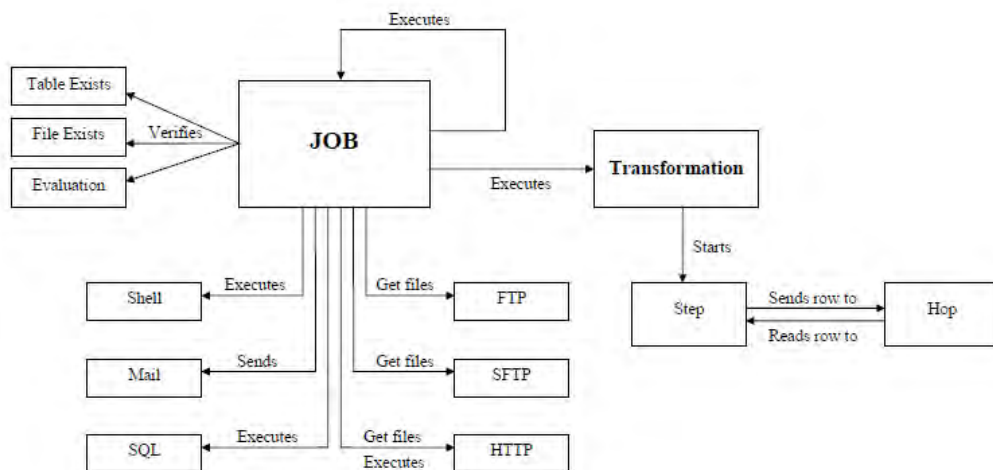
c) *Load*

Proses *load* adalah proses tahapan terakhir dari proses ETL. Pada proses ini akan dilakukan proses pemuatan data dari proses *transform* ke dalam suatu *data warehouse*. Pada proses ini dilakukan juga proses *indexing* untuk memberikan indeks ke masing-masing data untuk mempercepat proses *query*. Terdapat dua mode *loading* ke dalam *data warehouse* yaitu *refresh* dan *update*. Mode *refresh* yaitu proses menuliskan kembali keseluruhan data di dalam *data warehouse* pada suatu interval waktu. Sedangkan untuk mode *update* yaitu suatu proses untuk meng-*update* (tidak menghapus atau menimpa data lain) data yang berubah ke tempat tujuan pada *data warehouse*. Mode *refresh* digunakan pertama kali ketika *data warehouse* berjalan dan hendak dimuat, sedangkan mode *update* umumnya digunakan ketika pemeliharaan data atau ketika *data warehouse* sedang *running*.

2.5 Pentaho Data Integration

Pada tahun 2001, Matt Casters (<http://www.ibridge.be>) seorang programmer dan konsultan *business intelligence* dari Belgia membuat perangkat ETL untuk keperluannya sendiri. Perangkat tersebut diberi nama Kettle yang pada waktu itu merupakan singkatan dari *KDE ETL Environment*, karena pada awalnya perangkat ini ditujukan untuk dapat dijalankan diatas *K Desktop Environment* (KDE) Linux. Di kemudian hari, kepanjangan Kettle berubah menjadi nama berulang, yaitu *Kettle ETL Environment*. ETL pada nama Kettle merupakan singkatan dari *Extraction, Transportation, Transformation, and Loading*. (JRP, 2014)

Beberapa tahun kemudian, Kettle resmi *goes open source*. Karena Kettle memiliki beragam keunggulan dibanding perangkat ETL *open source* lainnya, pada April 2006, kode sumber Kettle resmi diambil alih oleh Pentaho. Oleh Pentaho, Kettle diberi nama *Pentaho Data Integration* (PDI). (JRP, 2014)



Gambar 2.9 Kettle Conceptual Model

Fitur yang terdapat pada *Pentaho Data Integration* adalah sebagai berikut (JRP, 2014):

- Memiliki utilitas grafik yang dapat digunakan merancang *control flow* umum maupun *data flow* (aliran data).
- *Multi platform* - karena dikembangkan di atas java yang dapat berjalan di banyak platform sistem operasi.
- Bersifat *concurrent*, dalam arti row-row data diambil oleh suatu step dan diserahkan ke step lain secara parallel.
- *Scalable* - dapat beradaptasi dengan penambahan kapasitas memori RAM atau pun *storage (scale up)* dan dapat node komputer / *cluster (scale out)*.
- Koleksi step transformation dan job yang cukup banyak.
- *Extensible*, kita dapat membuat step transformation dan job baru dengan sistem plugin.

2.6 OLAP (Online Analytical Processing)

OLAP diperkenalkan oleh E.F Codd yang merupakan bapak *relational databases*. Secara mendasar OLAP adalah suatu metode khusus untuk melakukan analisis terhadap data yang terdapat dalam media penyimpanan data (database) dan kemudian membuat laporannya sesuai dengan permintaan user (Hermawan, 2005).

Online analytical processing (OLAP) adalah sebuah perangkat yang menggambarkan teknologi menggunakan visualisasi multidimensi sejumlah data untuk menyediakan akses yang lebih cepat bagi strategi informasi dengan tujuan

mempercepat analisis. Dalam model data OLAP, informasi di gambarkan secara konseptual seperti kubus (*cube*), yang terdiri atas kategori deskriptif (*dimensions*) dan nilai kuantitatif (*measures*) (Reddy, 2010).

Data multidimensi memiliki atribut tersendiri untuk bisa dikelola dalam OLAP. Terdapat tiga atribut diantaranya adalah (Sulianta, 2011):

- a) Dimensi (*dimension*): adalah suatu atribut yang di tinjau.
- b) Pengukur (*measurement*): besaran yang dapat diukur mengacu pada irisan antara dimensi yang di tinjau.
- c) Kalkulasi (hasil pengukuran): adalah nilai dari *measurement*.

Pada model data kubikal virtual ada beberapa jenis metode pengoperasian diantaranya mencakup (Sulianta, 2011):

- a) Membuat irisan / *Slicing* adalah mengambil atau mengiris satu dimensi dari data virtual kubikal untuk keperluan penyederhanaan informasi atau untuk membuang informasi yang tidak diperlukan dalam analisa.
- b) Membuat banyak irisan / *Dicing* adalah irisan yang dilakukan lebih dari dua dimensi data.
- c) *Drill Up* dan *Drill Down* merupakan teknik analisa untuk men-generalisasi atau menspesifikasi informasi, semakin keatas maka informasi makin ringkas dan semakin kebawah maka informasi semakin rinci.
- d) Rotasi atau *Pivoting* dilakukan dengan memutar atau merotasi data kubikal virtual untuk mendapat sudut pandang berbeda terhadap data yang dianalisa.

Sumber data OLAP bisa dari sumber data *online* atau *offline*. Sumber data *online* maksudnya data yang disimpan pada server basis data dan perangkat lunak OLAP langsung terintegrasi dalam perangkat lunak yang mengelola transaksi basis data. Sedangkan sumber *offline* maksudnya data yang sudah terpisah dari sistem yang berjalan dan dipergunakan untuk keperluan analisa dan statistik untuk kemudian dikustomasi sesuai kebutuhan.