Knowledge Module pada Oracle Data Integrator (ODI)

Yuafanda Kholfi Hartono
yuafanda@yahoo.com
http://ilmukomputer.org/author/yofanda/

Knowledge Modules (KMs) adalah komponen dalam Oracle Data Integrator (ODI) yang dapat melakukan transformasi data secara berulang-ulang serta proses ETL (*extract, transform, and load*) pada teknologi yang berbeda-beda. Kekuatan KMs terletak pada daya pakai ulang dan fleksibilitasnya. Misalnya, Anda dapat mengembangkan dan menerapkan strategi pemuatan (loading) untuk tabel fact, dan kemudian, dengan klik mouse, menerapkan strategi loading untuk semua table fact Anda. Jika Anda menyesuaikan strategi ETL anda karena perubahan kebutuhan bisnis dan memodifikasi KM terkait, semua interface yang menggunakan KM akan ikut mengalami perubahan. Anda dapat memadupadankan bahasa pemrograman yang berbeda, jenis, dan gaya (RDBMS SQL asli, bahasa scripting seperti Jython atau JavaScript, atau bahkan Java). Inilah yang membuat KMs sangat fleksibel. Pendekatan inovatif ini disebut Desain deklaratif Oracle Data Integrator dan bisa sampai 10 kali lebih cepat daripada pendekatan tradisional ETL.

Pendahuluan

Knowledge Modules (KMs) adalah komponen dalam Oracle Data Integrator (ODI) yang dapat melakukan transformasi data secara berulang-ulang serta proses ETL (*extract, transform, and load*) pada teknologi yang berbeda-beda.

Oracle Data Integrator adalah *middleware* yang dikeluarkan oleh Oracle Corp. yang berfungsi dalam proses integrasi platform data yang komprehensif. [3] Oracle Data Integrator terintegrasi dengan seluruh teknologi Oracle, termasuk Oracle Database, Exadata Database Machine, Exalogic, Big data Appliance, WebLogic Server, Business Intelligence, dan Aplikasi Oracle. ODI adalah platform integrasi data yang strategis untuk Oracle. [4]

Staging area alam hal data warehouse, merupakan tempat penyimpanan perantara antara source informasi (source) dan data warehouse (DW) atau Data mart (DM). Staging area biasanya bersifat sementara, dan isinya bisa dihapus setelah DW / DM telah berhasil dimuat. [5]

Service Oriented Architecture (SOA) adalah struktur yang mendasari dan mendukung komunikasi antara services. SOA mendefinisikan bagaimana dua entitas komputasi, seperti program, berinteraksi sedemikian rupa untuk memungkinkan satu entitas melakukan unit proses atas nama entitas lain. Interaksi services didefinisikan menggunakan bahasa deskripsi. Setiap interaksi berdiri sendiri dan mudah untuk digabungkan, sehingga setiap interaksi adalah independen dari interaksi lain. [6]

Repository adalah tempat sentral di mana sebuah agregasi data disimpan dan dipelihara dengan cara yang terorganisir, biasanya dalam sebuah media penyimpanan komputer. Istilah ini berasal dari bahasa Latin repositorium, kapal atau ruang di mana hal-hal tertentu dapat ditempatkan, dan itu bisa berarti tempat di mana hal-hal dikumpulkan. [7]

Isi

Knowledge Modules (KMs) adalah komponen dalam Oracle Data Integrator (ODI) yang dapat melakukan transformasi data secara berulang-ulang serta proses ETL (*extract, transform, and load*) pada teknologi yang berbeda-beda. Salah satu contoh dari fungsi KMs adalah mengekstrak data melalui data capture dari Oracle Database 10g dan memuat data tersebut menjadi partitioned fact table dalam Oracle Database 11g, atau menciptakan ekstraksi data berbasis timestamp dari database SQL Server Microsoft dan memuat data ini dalam sebuah Teradata enterprise data warehouse. [2]

KMs adalah kode template. Setiap KM didedikasikan untuk tugas individu dalam keseluruhan proses integrasi Data. Kode KMs muncul di dekat form yang akan diexecute kecuali bahwa kode tersebut termasuk metode substitusi Oracle Data Integrator (ODI) yang memungkinkan untuk digunakan secara umum oleh banyak job integrasi yang berbeda. Kode yang akan dihasilkan dan dieksekusi berasal dari aturan deklaratif dan metadata yang didefinisikan dalam modul Designer ODI. [1]

KMs dibagi dalam 6 kategori yang berbeda seperti yang dirangkum dalam tabel di bawah ini:

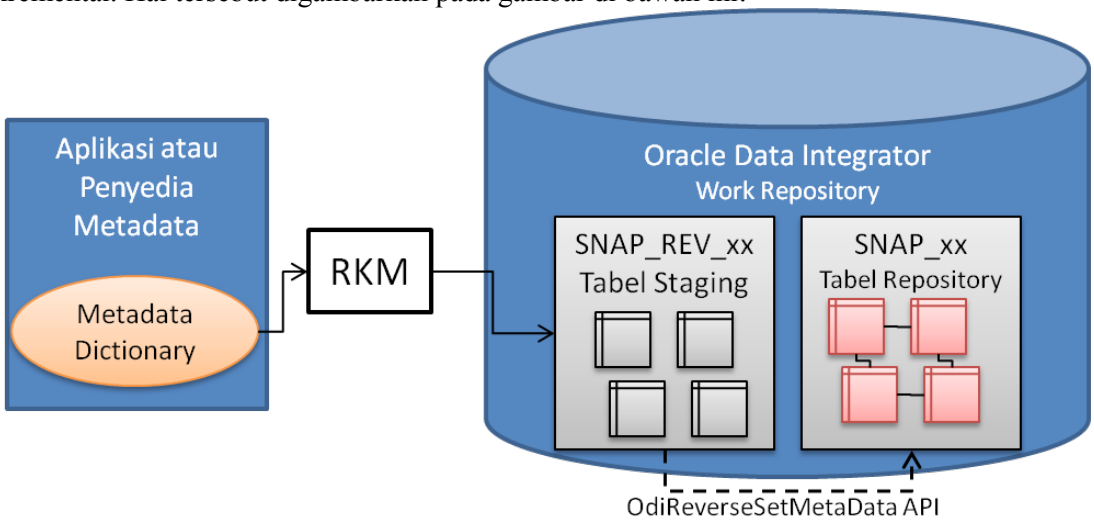
Knowledge Module	Uraian	Digunakan saat
Reverse-engineering KM	Mengambil metadata ke work repository Oracle Data Integrator	Digunakan dalam model untuk melakukan reverse-engineering yang telah disesuaikan (di customized)
Check KM	Cek konsistensi data terhadap constraint	* Digunakan dalam model, sub model dan datastore untuk integritas data audit *Digunakan dalam interface untuk kontrol alur atau kontrol statis
Loading KM	Memuat data yang heterogen ke staging area	Digunakan dalam interface dengan source data yang heterogen
Integration KM	Mengintegrasikan data dari staging area ke target	Digunakan di interface
Journalizing KM	Membuat kerangka objek Data Capture dalam daerah staging area	Digunakan dalam model, sub model dan datastore untuk membuat, memulai dan menghentikan jurnal dan untuk mendaftarkan pelanggan

Service KM	Menghasilkan manipulasi data pada web service	Digunakan dalam model dan datastores

Bagian berikut ini menjelaskan setiap jenis Knowledge Module.

Reverse-engineering Knowledge Modules (RKM)

Peran utama RKM adalah untuk melakukan reverse engineering yang dapat disesuaikan untuk sebuah model. RKM yang bertanggung jawab menghubungkan ke aplikasi atau penyedia metadata kemudian mengubah dan menulis metadata yang dihasilkan tersebut ke dalam repositori Oracle Data Integrator. Metadata ditulis sementara ke tabel SNP_REV_xx.
RKM kemudian memanggil Oracle Data Integrator API untuk membaca dari tabel ini dan menulis ke tabel metadata Oracle Data Integrator di work repository dalam mode update inkremental. Hal tersebut digambarkan pada gambar di bawah ini:



Gambar 1 : Reverse-engineering Knowledge Module (RKM)

Sebuah RKM secara umum mengikuti langkah-langkah:

1. Membersihkan tabel SNP_REV_xx dari eksekusi sebelumnya dengan menggunakan perintah OdiReverseResetTable

2. Mengambil sub model, datastore, kolom, unique key, foreign key dan kondisi dari
   penyedia metadata untuk tabel SNP_REV_SUB_MODEL, SNP_REV_TABLE, SNP_REV_COL, SNP_REV_KEY, SNP_REV_KEY_COL, SNP_REV_JOIN, SNP_REV_JOIN_COL, SNP_REV_COND.

3. Update model dalam work repository dengan memanggil API OdiReverseSetMetaData

Check Knowledge Modules (CKM)

CKM bertugas memeriksa setiap record dari kumpulan data konsisten dengan pengaturan constraint yang telah ditetapkan. CKM digunakan untuk menjaga integritas data dan berpartisipasi inisiatif dalam kualitas data secara keseluruhan. CKM dapat digunakan dalam 2

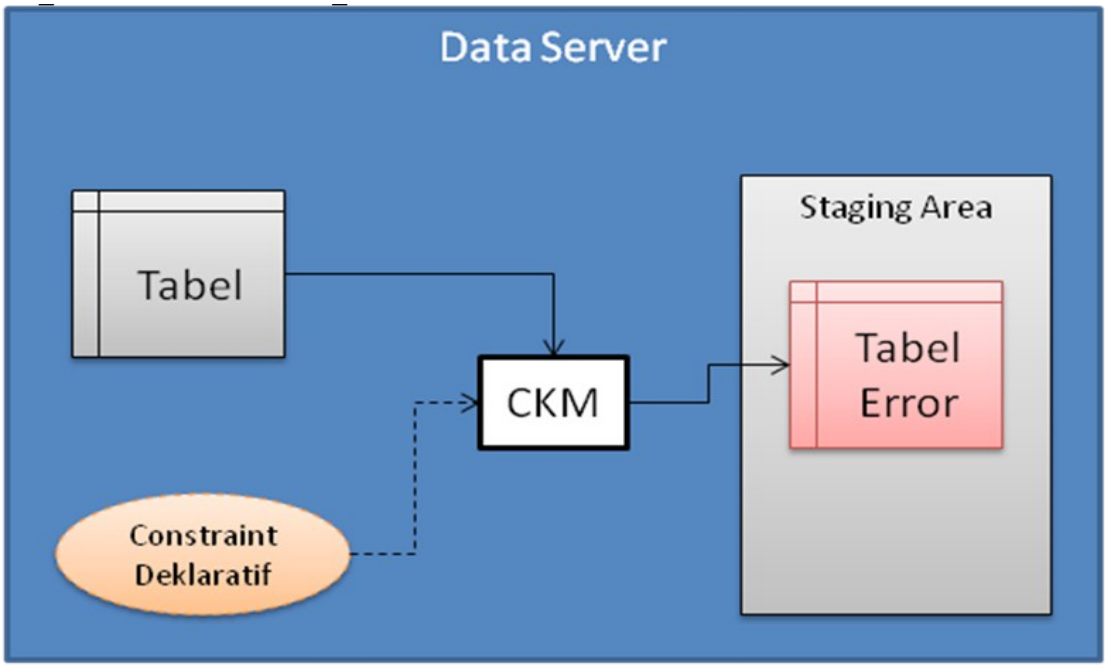
cara:

- Untuk memeriksa konsistensi existing data. Hal ini dapat dilakukan pada datastore apapun atau dalam interface, dengan menetapkan pilihan STATIC_CONTROL ke mode "Yes". Dalam kasus pertama, data yang diperiksa adalah data yang saat ini ada dalam datastore. Dalam kasus kedua, data dalam datastore target diperiksa setelah data tersebut dimuat.

- Untuk memeriksa konsistensi data yang masuk sebelum memuat record ke datastore target. Hal ini dilakukan dengan menggunakan opsi FLOW_CONTROL. Dalam kasus pertama, CKM mensimulasikan constraint datastore target pada alur yang dihasilkan sebelum menuliskannya ke target.

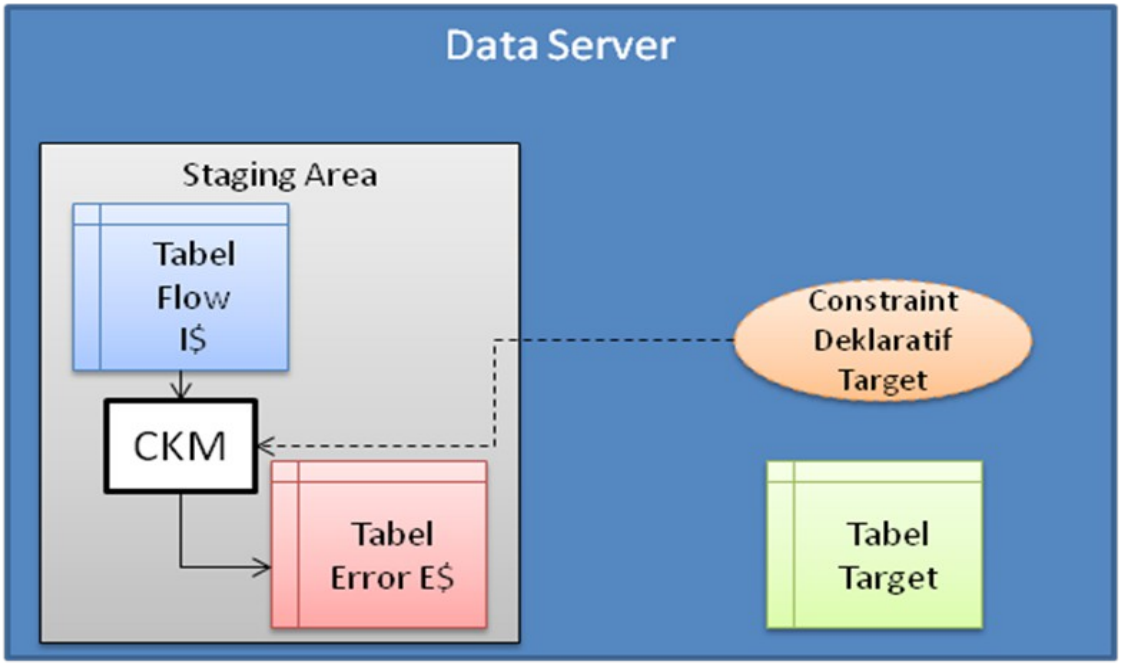
Singkatnya: CKM dapat memeriksa baik tabel yang ada atau tabel sementara "I$" yang dibuat oleh suatu IKM.

CKM menerima satu set constraint dan nama tabel untuk diperiksa. Kemudian CKM tersebut menciptakan sebuah tabel error "E$" yang menulis semua record yang ditolak. CKM juga dapat menghapus record yang salah dengan memeriksa hasil set.
Gambar berikut akan menunjukkan bagaimana CKM yang beroperasi dalam dua mode STATIC_CONTROL dan FLOW_CONTROL.



Gambar 2 : Check Knowledge Module (STATIC_CONTROL)

Dalam mode STATIC_CONTROL, CKM membaca constraint dari sebuah tabel dan memeriksa constraint tersebut terhadap Data dari tabel. Record yang tidak sesuai dengan constraint ditulis ke tabel error "E$" dalam staging area.



Gambar 3 : Check Knowledge Module (FLOW_CONTROL)

Dalam mode FLOW_CONTROL, CKM membaca constraint dari Interface tabel target. Ia memeriksa constraint terhadap data yang terdapat dalam flow tabel "I$" dari staging area. Record yang melanggar constraint ditulis dalam tabel staging area "E$".

Dalam kedua kasus tersebut, CKM biasanya melakukan kegiatan sebagai berikut:

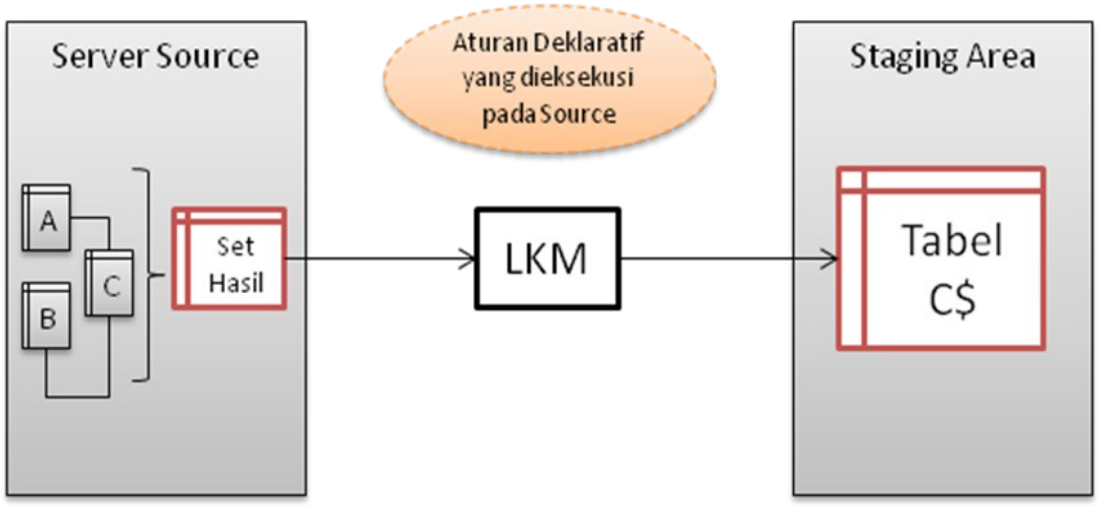
1. Membuat tabel error "E$" pada staging area. Tabel error harus berisi kolom yang sama dengan datastore serta kolom tambahan untuk melacak pesan kesalahan, asal pemeriksaan, tanggal pemeriksaan dan lain-lain

2. Mengisolasi catatan yang salah dalam tabel "E$" untuk setiap primary key, alternative key, foreign key, Kondisi, kolom wajib yang perlu diperiksa.

3. Jika diperlukan, menghapus data yang salah dari tabel yang telah diperiksa.

Loading Knowledge Modules (LKM)

Sebuah LKM bertanggung jawab untuk memuat source data dari remote server ke staging area. KMs ini digunakan oleh interface ketika beberapa datastore source tidak berada pada server data yang sama sebagai staging area. LKM menerapkan aturan deklaratif yang perlu dijalankan pada server source dan mengambil sebuah hasil tunggal yang ditetapkan dan menyimpannya dalam table "C$" di staging area, seperti yang digambarkan di bawah ini.



Gambar 4 : Loading Knowledge Module (LKM)

1. LKM menciptakan tabel sementara "C$" di staging area. Tabel ini akan menahan sementara record-record dimuat dari server source.

2. LKM memperoleh satu set record sebelum-ditransformasikan dari server source dengan menjalankan transformasi yang sesuai pada sourcenya. Biasanya, hal ini dilakukan oleh query tunggal SELECT SQL ketika server source adalah sebuah RDBMS. Ketika source tidak memiliki kapasitas SQL (seperti flat file atau aplikasi), LKM tersebut hanya membaca source data dengan metode yang tepat (membaca file atau mengeksekusi API).

3. LKM ini memuat record ke dalam tabel "C$" dari staging area.

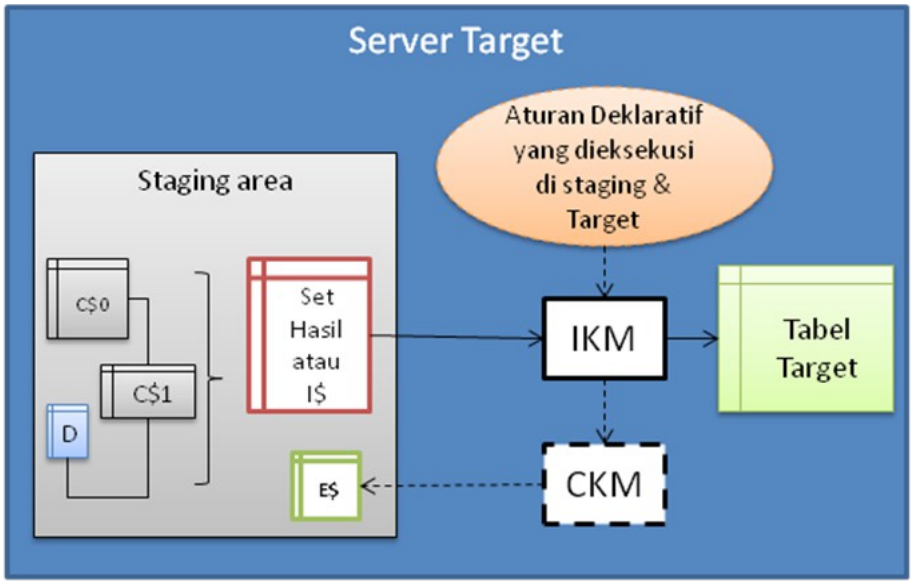
Sebuah interface mungkin memerlukan beberapa LKM ketika menggunakan datastore dari source yang berbeda. Ketika source semua datastores berada pada server data yang sama sebagai staging area, LKM tidak diperlukan.

Integration Knowledge Modules (IKM)

IKM bertugas menuliskan data final, dan mentransformasikannya ke tabel target. Setiap interface menggunakan satu IKM. Ketika IKM dimulai, diasumsikan bahwa semua fase loading untuk remote server sudah melakukan tugas-tugas mereka. Ini berarti bahwa semua source data remote set telah dimuat oleh LKM ke tabel sementara "C$" di staging area, atau datastore source yang berada pada server data yang sama seperti staging area. Oleh karena itu, IKM hanya perlu menjalankan transformasi "Staging dan Target", melakukan proses join dan filter pada tabel "C$", dan tabel yang terletak pada server data yang sama dengan staging area. Hasil set biasanya diproses oleh IKM dan ditulis ke dalam tabel temporary "I$" sebelum dilakukan loading ke target.

Record perubahan status terakhir dapat ditulis dalam beberapa cara tergantung pada IKM yang dipilih dalam Interface Anda. Mereka mungkin hanya ditambahkan ke target, atau dibandingkan untuk update inkremental atau mengubah dimensi secara bertahap. Ada 2 jenis IKM: IKM yang dapat digunakan ketika kondisi staging area sama dengan server yang berfungsi sebagai datastore target, dan mereka yang dapat digunakan ketika staging area dan server yang berfungsi sebagai datastore target tidak sama. Hal tersebut diilustrasikan pada

gambar di bawah ini:



Gambar 5 : Integration Knowledge Module (staging area pada target)

Ketika staging area berada pada server target, IKM biasanya melakukan langkah-langkah berikut:

1. IKM mengeksekusi SELECT statement set-oriented tunggal untuk menjalankan staging area dan aturan target deklaratif pada semua tabel "C$" dan tabel lokal (seperti D pada gambar). Proses ini akan menghasilkan result set.

2. "append" IKM simple akan langsung menulis hasil set ini ke dalam tabel target. IKM yang lebih kompleks akan menciptakan tabel "I$" untuk menyimpan hasil yang telah ditetapkan.

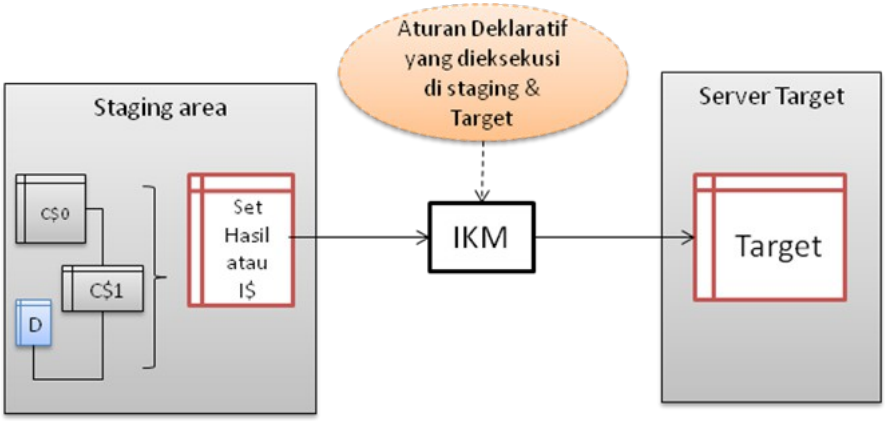
3. Jika aliran data yang perlu diperiksa terhadap constraint dalam target, IKM akan memanggil CKM untuk mengisolasi record yang salah dan membersihkan tabel "I$".

4. IKM menulis record dari tabel "I$" untuk target mengikuti strategi yang telah didefinisikan sebelumnya(incremental update, perubahan dimensi secara bertahap, dll).

5. IKM melakukan proses drop tabel sementara "I$".

6. Secara opsional, IKM dapat memanggil CKM kembali untuk memeriksa konsistensi dari datastore target.

Jenis KMs ini tidak memanipulasi data di luar server target. Pengolahan data lebih berorientasi untuk efisiensi maksimum ketika melakukan proses pada volume besar.



Gambar 6 : Integration Knowledge Module (staging area berbeda dengan target)

Ketika staging area berbeda dari server target, seperti yang ditunjukkan pada Gambar 6, IKM akan melakukan langkah-langkah berikut:

1. IKM mengeksekusi SELECT statement set-oriented tunggal untuk menjalankan staging area dan aturan target deklaratif pada semua tabel "C$" dan tabel lokal (seperti D pada gambar). Proses ini akan menghasilkan result set.

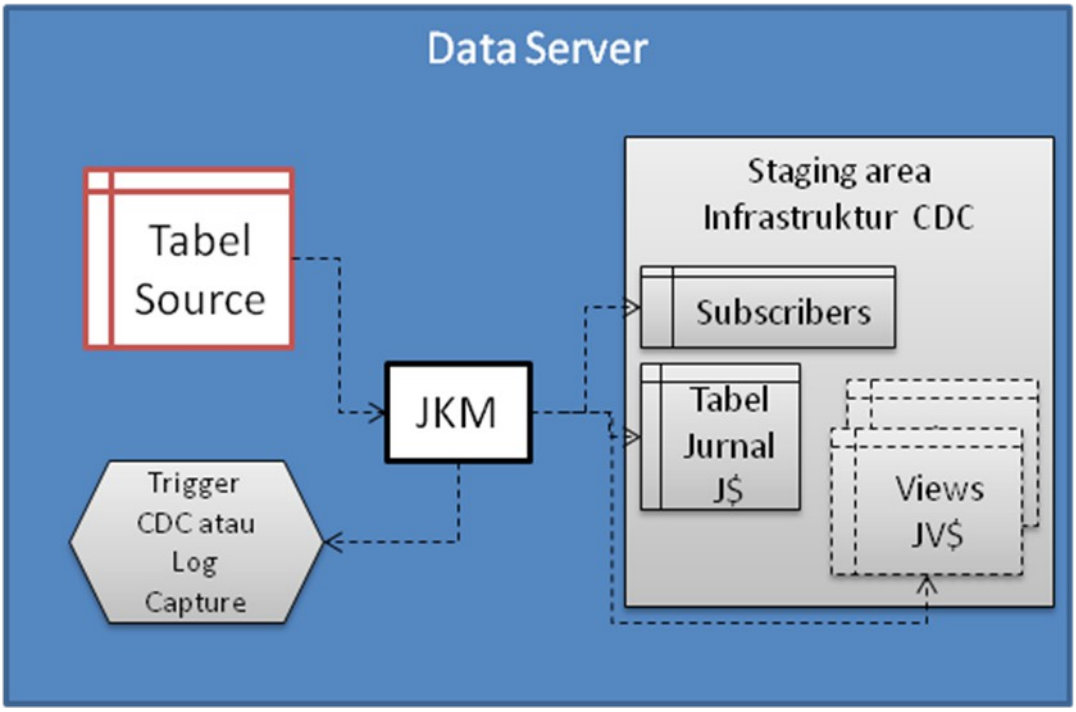
2. IKM memuat result set ke datastore target, mengikuti strategi yang ditetapkan (append atau inkremental update).

Arsitektur ini memiliki keterbatasan tertentu, seperti:

- Sebuah CKM tidak dapat digunakan untuk melakukan audit integritas data pada saat data sedang diproses.
- Data harus diekstrak dari staging area sebelum dimuat ke target, yang mungkin mengakibatkan masalah performance server.

Journalizing Knowledge Modules (JKM)

JKMs membuat infrastruktur untuk menangkap Perubahan pada Data model, sub model atau sebuah datastore. JKMs tidak digunakan dalam interface, melainkan dalam model untuk menentukan bagaimana infrastruktur CDC diinisialisasi. Infrastruktur KMs ini terdiri dari error tabel, tabel perubahan, view pada tabel tersebut dan satu atau lebih trigger atau program log capture seperti yang digambarkan di bawah ini.



Gambar 7 : Journalizing Knowledge Module (JKM)

Service Knowledge Modules (SKM)

SKMs bertanggung jawab membuat dan menggunakan manipulasi data Web service untuk infrastruktur Service Oriented Arsitektur (SOA) Anda. SKMs ditetapkan pada Model. Mereka mendefinisikan operasi yang berbeda untuk menghasilkan untuk setiap web service datastore. Tidak seperti KMs lainnya, SKMs lakukan tidak menghasilkan dan mengeksekusi kode melainkan menyebarkan file arsip Web Services. SKMs dirancang untuk menghasilkan kode Java menggunakan framework Oracle Data Integrator, untuk Web Services. Kode ini kemudian disusun dan akhirnya dideploy di kontainer server aplikasi.

Penutup

Oracle Data Integrator dapat memuat berbagai standar KMs secara bersamaan. Anda dapat langsung menggunakan KMs dalam interface ETL Anda (setara dengan pemetaan Oracle Warehouse Builder) atau menyesuaikannya dengan kebutuhan spesifik Anda. Oracle Data Integrator menerapkan enam jenis KMs. Masing-masing mencakup satu tahap dalam proses transformasi dari source ke target. Tiga jenis yang paling penting dari knowledge module adalah Integration Knowledge Modules (IKM), Loading Knowledge Modules (LKM), dan Check Knowledge Modules (CKM).

Referensi

[1] Oracle Corp., *"Oracle Data Integrator – Knowledge Modules Developer's Guide"*, 2006
http://www.oracle.com/technetwork/testcontent/oracledi-km-development-129055.pdf

[2] Uli Bethke , "*Developing a Knowledge Module in Oracle Data Integrator*", 2009
http://www.oracle.com/technetwork/articles/bethke-odi-090881.html

[3] Oracle Corp., *"Oracle Data Integrator"*, 2012
http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html

[4] ODI Product Management, "*Oracle Data Integrator 11g New Features Overview*", 2012
http://www.oracle.com/technetwork/middleware/data-integrator/overview/odi-11g-new-features-overview-1622677.pdf

[5] Wikipedia, *"Staging (Data)"*, 2012
http://en.wikipedia.org/wiki/Staging_%28data%29

[6] Margaret Rouse, "*Service Oriented Architecture*", 2008
http://searchsoa.techtarget.com/definition/service-oriented-architecture

[7] Margaret Rouse, "*Repository*", 2005
http://searchoracle.techtarget.com/definition/repository

Biografi Penulis

Yuafanda Kholfi Hartono. Menyelesaikan S1 di Universitas Indonesia. Sebelumnya menamatkan pendidikan di Sekolah Tinggi Akuntansi Negara, minat dan ketertarikan pada bidang IT membuat pada akhirnya ditempatkan pada Direktorat Jenderal Bea dan Cukai Kementerian Keuangan sebagai Database Administrator (DBA).