

Rancang Bangun Online Analytical Processing (OLAP) Classic Model Data

‘Literature Review’

Ade Rahmat Iskandar, Ilfiantri Intias
Jurusan Telekomunikasi,
Akademi Telkom Jakarta
Jl. Daan Mogot Km 11, Jakarta Barat 11710
ader@akademitelkom.unjani.ac.id

Abstrak—Data warehouse menjadi sistem informasi yang sangat penting di era digital ini. Sistem ini memiliki fungsi untuk mengelola data yang bersifat subjektif dan secara umum untuk mengelola historical data yang dapat digunakan untuk analisis data, data mining dan big data. Paper ini dibuat dengan mengimplementasikan database dari Online Transaction Processing (OLTP) kedalam database Online Analytical Processing (OLAP). Pada penelitian ini digunakan data set Classic Model, tool yang digunakan dalam perancangan Data Warehouse yaitu menggunakan tool open source Pentaho Data Integration. Metode yang dilakukan dalam penelitian ini adalah, dengan melakukan perancangan model Data Warehouse star scheme, melakukan tahapan Extract, Transform dan Load (ETL) pada data set classic model dan mentransformasikannya kedalam Online Analytical Processing. Luaran dari penelitian ini adalah terbentuknya rancang model Data Warehouse classic model.

Kata kunci—Data Warehouse, ETL, OLAP, Classic Model

I. PENDAHULUAN

Dalam kurun 1970an penerapan teknologi database sudah sangat familiar digunakan di berbagai instansi baik pemerintahan maupun swasta. Teknologi ini dianggap sangat berperan dalam mengelola data perusahaan secara sistematis dalam pengelolaan sistem yang mudah untuk di insert, update dan delete. Beberapa vendor DBMS sangat familiar bagi kalangan pengelola data, baik yang bersifat relational (RDMBS) seperti Oracle, SQL Server, IBM Db2, MySQL maupun beberapa Dynamyc Database yang beberapa tahun terakhir sangat trend terutama dalam pengelolaan data dinamis dan Big Data seperti MongoDB, Casandra, Hadoof, dan lain-lain.

Database transaksional atau dalam istilah lain Database OLTP (Online Transactional Processing) lebih trend digunakan pada beberapa pengelolaan data transaksional dimana data dapat di update secara realtime. Pembangunan sistem Informasi Akademik di suatu universitas secara realtime yang mengintegrasikan data dari mulai presensi baik yang diinput manual, atau pun menggunakan RFID, web based mapupun mobile sampai pengelolaan data para sivitas akademik secara

updateable merupakan salah satu contoh implementasi dari database OLTP.

Pada penelitian terdahulu sudah dibuktikan bahwa perkembangan pengelolaan data yang bersifat historical menggunakan Data warehouse sudah dbanyak dilakukan baik di perusahaan skalam multinasional maupun dalam isitansi RistekDikti yang mengelola data akademik yang sangat besar untuk semua perguruan tinggi yang ada di Indonesia. Penelitian tersebut diantaranya adalah mengenai kesiapan implemetansi Sistem Pangkalan Data Pendidikan Tinggi yang menggunakan Oracle Data warehouse yang sudah dibuktikan bahwa *readiness* sebelum implementasi pada tahun 2016 dianggap baik[1].

Istilah data mart merupakan versi skala yang lebih kecil dari dataarehouse. Data mart merupakan small warehousing yang dirancang untuk keperluan tingkat departemen[2].

Berikut adalah merupakan arsitektur pembangunan data warehouse yang terdiri dari data mart yang bersifat bottom up :



Gambar 1. Arsitektur Data Warehouse Top Down

- Keuntungan Top-Down Approach

Pada perancangan top-down approach suatu perusahaan bisa melihat data secara menyeluruh, arsitektur terpusat (tidak ada pembagian antar data marts), data bersifat terpusat, pengendalian

dan aturan terpusat, dapat memberikan hasil yang cepat jika diimplementasikan menggunakan iterasi.

- Keuntungan Bottom-Up Approach

Lebih cepat dan lebih mudah diimplementasikan, return on investment (ROI) lebih mudah diperoleh, sedikit resiko untuk terjadi kegagalan, dapat dibuat penjadwalan untuk data yang penting, memudahkan tim untuk belajar dan berkembang[2]

Pengelolaan data yang besar saat ini sangat diperlukan oleh beberapa perusahaan skala internasional, baik di instansi-instansi pemerintahan maupun di beberapa perusahaan-perusahaan yang mengelola data besar seperti Google, Facebook, LinkedIn dan lain-lain, pengelolaan data tersebut tidak berupa pengelolaan data transaksi tetapi sudah mengacu pada proses untuk pengelolaan data historical yang akan digunakan sebagai upaya untuk menganalisis data secara lebih jauh oleh perusahaan atau instansi tersebut, misalnya pengelolaan data akhir-akhir ini untuk mengelola data eKTP di Indonesia yang terdiri lebih dari 262 juta penduduk di indonesia[3].

Datawarehouse adalah alat penyimpanan data yang dirancang untuk memfasilitasi dan mempercepat analisis dan peningkatan data yang divalidasi dihasilkan dari berbagai sumber. Ada dua pendekatan utama untuk manajemen data dalam suatu datawarehouse: model dinormalisasi dan model dimensi yang diusulkan oleh Kimball & Ross[5].

Ralph dan Kimball menggambarkan berdasarkan model data relasional. Model yang dinormalisasi menghargai bentuk normal untuk menghindari redundansi informasi, tetapi, hal tersebut akan meningkatkan jumlah tabel, mengarah ke beberapa proses penggabungan selama permintaan, dan tidak memfasilitasi interpretasi hasil dengan melihat skema itu sendiri. Model dimensi dirancang untuk memfasilitasi data analisis dan interpretasi, dengan mengorbankan sejumlah redundansi data, ruang penyimpanan yang lebih besar, dan memerlukan waktu pemrosesan awal yang lebih tinggi.

Query dari datawarehouse berdasarkan model dimensional jauh lebih cepat karena tabel sudah ada sebagian yang didenormalisasi dan dalam beberapa kasus dikumpulkan untuk beberapa analisis. Oleh karena itu database tidak harus digabungkan dengan semua tabel yang termasuk dalam skema relasional yang dinormalisasi. Lebih jauh, model dimensi merekomendasikan untuk membiarkan hanya satu derajat pemisahan antara tabel (gabungan tunggal), dengan pengecualian terbatas[6].

Datawarehouse memungkinkan informasi disusun untuk memfasilitasi akuisisi pengetahuan dan menyederhanakan pemrosesan data dari satu domain ke domain lainnya (Adamson, 2010). Semakin banyak jumlah data meningkat, semakin besar kebutuhan untuk struktur umum dan komparabilitas antara sumber data yang berbeda menjadi penting untuk melakukan upaya yang ketat dalam menganalisis data tersebut dalam kerangka waktu yang lama. faktanya, struktur datawarehouse berdasarkan pemodelan dimensi memungkinkan ilmuwan dan masyarakat umum untuk memilih dataset yang bersifat kustom untuk melakukan eksplorasi mereka sendiri (Manoochehri, 2013).

Buliung & Morency (2010) menjelaskan bahwa visualisasi merupakan latihan eksplorasi yang dapat mendorong proposisi hipotesis penelitian baru dan, jika dibingkai dengan baik, dapat mencegah salah tafsir oleh pengguna awam. Kejelasan,

presisi dan efisiensi merupakan kriteria paling penting untuk menilai objek visualisasi dan memfasilitasi objek dan memahami fenomena yang lebih kompleks. Lebih khusus lagi, geovisualisasi (MacEachren & Kraak, 2001) mengintegrasikan dimensi spasial dengan visualisasi dan merupakan alat penting untuk eksplorasi dan analisis data spasial. Seluruh kelompok penelitian dan asosiasi spesialis visualisasi (ARUP; Citylab; van der Wijk) mempelajari dan mengusulkan objek visualisasi untuk mengklarifikasi masalah yang berkaitan dengan transportasi.

Menurut Pack (2010), dengan membuat data yang lebih mudah diakses melalui objek visualisasi yang efisien dapat meyakinkan lembaga yang bertanggung jawab dalam manajemen transportasi untuk membiayai program khusus untuk pembuatan alat visualisasi. Permasalahannya adalah, dalam menciptakan alat seperti itu membutuhkan banyak waktu dan sumber daya. Selain itu, fitur-fitur baru yang ditawarkan disertai dengan peningkatan *user experience*, yang pada gilirannya meningkatkan kebutuhan sumber daya dan waktu yang lebih lama. Oleh karena itu, untuk membatasi kosntrain tersebut, diperlukan otomatisasi tingkat tertentu dalam pembuatan objek visualisasi. Setelah data disimpan di Data Warehouse, banyak objek visualisasi dapat dibangun.

Pada penelitian ini dibahas proses Extract, Transform dan Load yang merupakan proses atau tahapan dalam Online Analytical Processing (OLAP) atau Data warehouse menggunakan tool open source Pentaho Data Integration. Pentaho Data Intergration (PDI) tidak hanya digunakan sebagai tools untuk ETL tetapi dapat digunakan untuk migrasi data dalam aplikasi-aplikasi database kedalam flat files, data cleansing dan lain-lain. Pentaho Data Integration memiliki feature grafis, drag dan drop design environment. Pada kenyataannya perkembangan aplikasi dengan konsep *enterprise resource planning* (ERP) sangat trend dalam beberapa tahun terakhir ini, hal ini yang membuat vendor-vendor ternama membuat aplikasi yang lingkupnya bukan hanya dapat mengintegrasikan proses ETL tersebut tetapi lebih jauh secara sistematis dapat digunakan dalam pengelolaan data dengan konsep big data.

ETL atau kependekan dari extract, transform dan load. Dalam pengertian sederhana ETL adalah sekumpulan proses untuk mengambil dan memproses data dari satu atau banyak sumber menjadi sumber baru, misalkan mengolah data OLTP menjadi OLAP. Proses-proses ETL adalah :

- Extract

Semua proses yang diperlukan untuk terhubung dengan beragam sumber data, dan membuat data tersebut tersedia bagi proses-proses selanjutnya, misalnya (Read file Excell, Mengambil data dari database, Membaca file dari XML, dan lain-lain)

- Transform

Bagian ini mengacu pada fungsi apa saja yang berfungsi untuk mengubah data yang masuk menjadi data yang dikehendaki. Beberapa fungsi, diantaranya Pemindahan data, Perhitungan modifikasi isi, tipe atau struktur data

- Load

Semua proses yang diperlukan untuk mengisi data ke target. Misalnya hasil dari proses sebelumnya disimpan ke dalam database OLAP dan hasil dari proses sebelumnya disimpan ke dalam file Excell.

Dalam pembangunan data integration diperlukan tools untuk mengintegrasikan data tersebut. Terdapat beberapa tools ETL yang berifat lisensi maupun open source. Beberapa software atau tools tersebut adalah Oracle Data Integration, Pentaho Data Integration (Kettle), Microsoft SQL Server Integration Services (SSIS) dan Microsoft SQL Server Integration Services (SSIS).

II. METODE

Metode penelitian yang dilakukan adalah Eksperimental Data Intergration dengan tahapan-tahapan sebagai berikut :

1. Pengoleksian data set classic model, data diperoleh dalam bentuk database MySQL.
2. Membuat rancangan database menggunakan tool RDBMS MySQL, membuat alter tabel dan triger yang diperlukan pada saat pembangunan data warehouse.
3. Merancang Skema Star untuk data warehouse Classic Model.
4. Melakukan transformasi Extract, transform dan Load dari database OLTP ke dalam database OLAP menggunakan tool Pentaho Data Intergration.

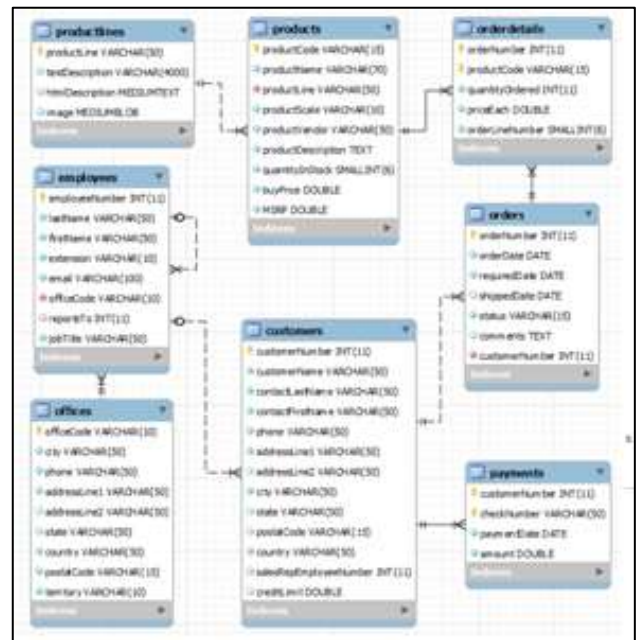
III. HASIL DAN DISKUSI

Pada bagian hasil dan pembahasan ini, dijabarkan proses sistematis dalam pembangunan Data warehouse atau Data mart order, sebagai upaya pembangunan departmental Data Warehouse untuk kasus Fact Order.

1) Pembangunan skema database OLTP

Berikut ini adalah model skema rancangan database dari database OLTP yang sudah dibuat mengacu pada transformasi data integration Datawarehouse dengan Pentaho[3]

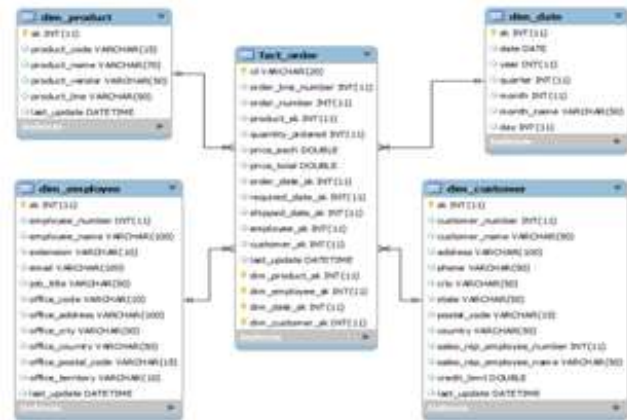
Database OLTP terdiri dari beberapa tabel yang saling berelasi yaitu employee, office, customer, product, productLines, payment, orders dan ordersDetail. Pembangunan database OLTP dibuat seperti halnya pembangunan database biasa, setiap tabel terdiri dari suatu primary key, dan atribut atau field independent lainnya, suatu tabel dapat berelasi dengan tabel lain dengan cara dibuat suatu *foreign key* dalam tabel tersebut yang mengacu pada tabel yang dituju menggunakan nama dan tipe data serta *constraints* yang sama untuk variabel kunci tersebut.



Gambar 2. Arsitektur Data Warehouse Top Down

2) Pembangunan Database OLAP

Online Analytical Processing (OLAP) merupakan database yang akan dibuat berdasarkan data dan skema dari dari OLTP yang sudah dibuat. Pada paper ini Data Mart Sekema dirancang menggunakan Star Schema, berikut adalah hasil rancangan mengacu pada skema OLAP Classic model.



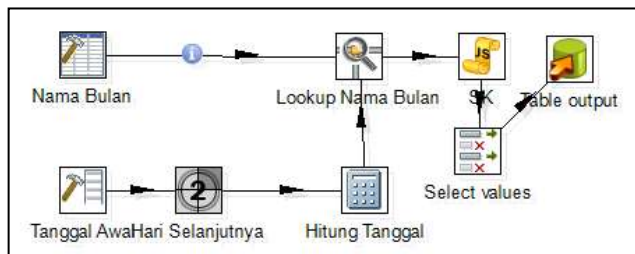
Gambar 3. Star Schema OLAP Classic Model[3]

Dalam pembangunan database OLAP ini, dibuat dengan cara mentransformasikan tabel-tabel pada database OLTP ke dalam database OLAP. Dimensi Employee dibuat dengan cara menggabungkan field-field pada tabel employee dan office, dimensi product, dibuat dengan cara menggabungkan field-field pada tabel products dan productLines, dimensi customer dibuat dari hasil transformasi tabel customer, dimensi date merupakan dimensi tambahan yang biasanya diperlukan dalam pembangunan Data Mart ataupun Data Warehouse.

Dimensi date ini diperlukan untuk membuat *cube* berdasarkan *drill down* waktu yang lebih detail (year, quarter, month, week dan day), langkah terakhir adalah membuat tabel fact yang merupakan representasi utama dalam pembangunan data mart atau data warehouse, pada penelitian ini dibuat fact_order yaitu tabel fakta untuk proses transaksi order.

3) Rancangan Dimensi Date

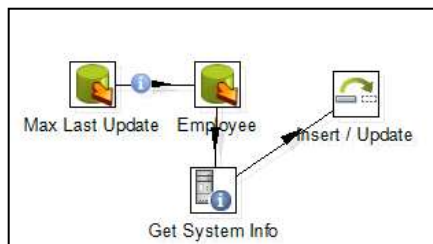
Dimensi date merupakan tabel dimensi yang dibuat untuk membuat *cube* atau laporan pada dashboard berdasarkan leveling waktu yang diharapkan (tahun, kuartal, bulan, minggu dan hari), berikut adalah rancangan job untuk dimensi date pada menggunakan tool Pentaho Data Integrarion :



Gambar 4. Dimensi Tanggal[3]

4) Rancangan dimensi Employee

Dimensi Employee merupakan transformasi tabel Employee dan Office pada database OLTP kedalam dimensi Employee pada database OLAP. berikut adalah rancangan job untuk dimensi Employee pada menggunakan *tool* Pentaho Data Integrarion:



Gambar 5. Dimensi Employee[3]

Berikut ini adalah potongan perintah SQL yang digunakan untuk mentransformasikan data dari tabel Employee dan Office kedalam dimensi Employee :

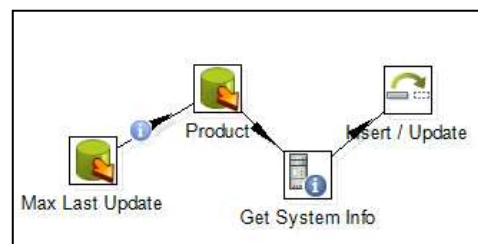
```
SELECT
  e.employeeNumber AS employee_number
  ,CONCAT(COALESCE(e.firstName,''), CASE WHEN
  (ISNULL(e.lastName)) THEN '' ELSE ' ' END,
  COALESCE(e.lastName,'')) AS employee_name
  ,e.extension AS extension
  ,e.email AS email
  ,jobTitle AS job_title
  ,o.officeCode AS office_code
  ,CONCAT(COALESCE(o.addressLine1,''), CASE
  WHEN (ISNULL(o.addressLine2)) THEN '' ELSE ' '
  END, COALESCE(o.addressLine2,'')) AS
  office_address
```

```
,o.city AS office_city
,o.country AS office_country
,o.postalCode AS office_postal_code
,o.territory AS office_territory
FROM employees e
LEFT JOIN offices o ON o.officeCode =
e.officeCode
WHERE e.updated > ?
```

Pada perintah SQL tersebut ditransformasikan data field dari tabel employee dan office kedalam dimensi employee (employeeNumber merupakan salah satu field dari tabel employee yang ditransformasikan kedalam dimensi employee untuk field employee_number, begitu pula untuk semua field-field yang bersesuaian).

5) Rancangan dimensi Product

Dimensi Product merupakan transformasi tabel product pada database OLTP kedalam dimensi product pada database OLAP. berikut adalah rancangan job untuk dimensi product pada menggunakan tool Pentaho Data Integrarion:



6) Dimensi Product[3]

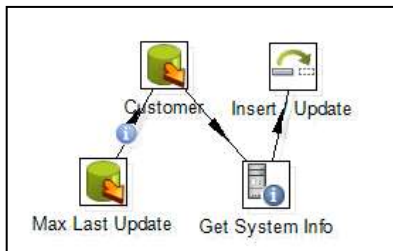
Berikut ini adalah potongan perintah SQL yang digunakan untuk mentransformasikan data dari tabel product dan Office kedalam dimensi Employee :

```
SELECT
  productCode AS product_code
  , productName AS product_name
  , productLine AS product_line
  , productVendor AS product_vendor
FROM products
WHERE updated > ?
```

Pada perintah SQL tersebut ditransformasikan data field dari tabel product kedalam dimensi product (productCode merupakan salah satu field dari tabel product yang ditransformasikan kedalam dimensi product untuk field product_code, begitu pula untuk semua field-field lain yang bersesuaian).

7) Rancangan dimensi Customer

Dimensi customer merupakan transformasi tabel customer pada database OLTP kedalam dimensi customer pada database OLAP. berikut adalah rancangan job untuk dimensi customer pada menggunakan tool Pentaho Data Integrarion:



Gambar 6. Dimensi Customer[3]

Berikut ini adalah potongan perintah SQL yang digunakan untuk mentransformasikan data dari tabel product dan Office kedalam dimensi Employee :

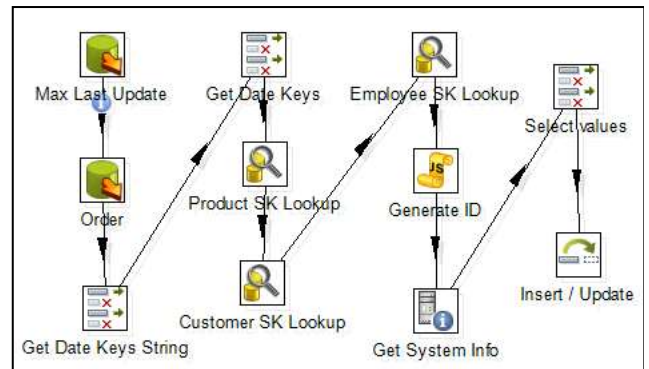
```

SELECT
  c.customerNumber AS customer_number
  ,c.customerName AS customer_name
  ,c.phone AS phone
  ,CONCAT(COALESCE(c.addressLine1, ''), CASE
WHEN (ISNULL(c.addressLine2)) THEN ' ' ELSE ' '
END, COALESCE(c.addressLine2, '')) AS address
  ,c.city AS city
  ,c.state AS state
  ,c.postalCode AS postal_code
  ,c.country AS country
  ,c.salesRepEmployeeNumber AS
sales_rep_employee_number
  ,CONCAT(COALESCE(e.firstName, ''), CASE WHEN
(ISNULL(e.lastName)) THEN ' ' ELSE ' ' END,
COALESCE(e.lastName, '')) AS
sales_rep_employee_name
  ,c.creditLimit AS credit_limit
FROM customers c
LEFT JOIN employees e ON
c.salesRepEmployeeNumber = e.employeeNumber
WHERE c.updated > ?

```

8) Rancangan tabel Fact Order

Pada penelitian ini, digunakan salah satu tools untuk proses data intergarion yaitu menggunakan Pentaho Data Intergration. Tahapan-tahapan yang dilakukan dalam penelitian ini adalah membangun file-file transformasi untuk semua tabel dimensi yang sudah dibuat (dimensi date, dimensi customer, dimensi employee) dan terakhir adalah membuat fact table (yang berisi dari gabungan dimension tables yang sudah dibuat). Berikut ini merupakan job fact order dari classis models yang dirancang menggunakan tool Pentaho Data Integration [3].



Gambar 7. Fact Order Classis Model

Tabel fakta untuk dimensi order dan orderdetails atau fact order diperoleh dari rancangan hop antar dimensi yang digabungkan. Pada komponenen Max Last Update dideklarasikan data historical yang diharapkan dalam pembangunan data mart atau data warahouse. Berikut adalah potongan kode SQL dari kmponen table input Order :

```

SELECT
  od.orderLineNumber AS order_line_number
  ,od.orderNumber AS order_number
  ,od.productCode AS product_code
  ,od.quantityOrdered AS quantity_ordered
  ,od.priceEach AS price_each
  ,od.quantityOrdered * od.priceEach AS
price total
  ,o.orderDate AS order_date
  ,o.requiredDate AS required_date
  ,o.shippedDate AS shipped_date
  ,e.employeeNumber AS employee_number
  ,o.customerNumber AS customer_number
FROM orderdetails od
LEFT JOIN orders o ON o.orderNumber =
od.orderNumber
LEFT JOIN customers c ON c.customerNumber =
o.customerNumber
LEFT JOIN employees e ON e.employeeNumber =
c.salesRepEmployeeNumber
WHERE o.orderDate > ?
ORDER BY od.orderNumber, od.orderLineNumber

```

Dari potongan kode SQL tersebut, bisa dianalisis bahwa terdapat korelasi dan implementasi data dari tabel OLTP kedalam dimensi product (OLAP). Pada perintah SQL tersebut terdapat statements OrderLineNumbere yang diperoleh dari tabel OrderDetails as (order_line_number untuk field yang sama pada tabel dimensi fact order) , begitu juga untuk semua implementasi field lainnya yang bersesuaian dari database OLTP kedalam database OLAP.

9) Eklporasi Pengujian Data set Clasic Model

Pada bagian akhir, adalah melakukan experimental data integration. Dalam penelitian ini digunakan perangkat keras Laptop Core i3 dengan RAM 2 GB. Data set yang diujikan kurang adalah 10000 record. dalam Data Warehouse data dapat dikategorikan data besar jika eklsporasi dari size GB to TB. Hasil eksperimental data intergarion untuk dataset data diatas

500.000 record tidak dapat diimplementasikan menggunakan perangkat Laptop yang memiliki RAM 2GB.

IV. KESIMPULAN

Kesimpulan dari penelitian ini adalah dapat diintegrasikan data dari database OLTP classic model kedalam database OLAP dengan dataset adalah 10000 record. Berikut adalah, luaran hasil eksplorasi data set kedalam data warehouse berupa data historical fact_order:

order_date_sk	required_date_sk	shipped_date_sk	employee_sk	customer_sk	last_update
20030106	20030113	20030110	10	86	2018-07-27 20:33:34
20030106	20030113	20030110	10	86	2018-07-27 20:33:34
20030106	20030113	20030110	10	86	2018-07-27 20:33:34
20030106	20030113	20030110	10	86	2018-07-27 20:33:34
20030109	20030118	20030111	17	8	2018-07-27 20:33:34
20030109	20030118	20030111	17	8	2018-07-27 20:33:34
20030109	20030118	20030111	17	8	2018-07-27 20:33:34
20030109	20030118	20030111	17	8	2018-07-27 20:33:34
20030110	20030118	20030114	11	28	2018-07-27 20:33:34
20030110	20030118	20030114	11	28	2018-07-27 20:33:34
20030129	20030207	20030202	17	5	2018-07-27 20:33:34
20030129	20030207	20030202	17	5	2018-07-27 20:33:34
20030129	20030207	20030202	17	5	2018-07-27 20:33:34

Gambar 8. Fact Order Classis Model

Data historical sangat diperlukan dalam manajemen data menggunakan Data mart atau data warehouse.

UCAPAN TERIMAKASIH

Terimakasih kepada Akademi Telkom Jakarta atas pendanaan yang diberikan untuk Seminar Nasional Informatika Univeristas Jenderal Ahmad Yani 2019.

DAFTAR PUSTAKA

- [1] Iskandar Ade, Ichsan "Receomendation for Implementting DataWarehouse for Higher Education in Indonesia", Proc. IEEExplore, CTSM 2016.
- [2] Qasem Al-Radaideh, Data warehouse and the building block, 2012, Yarmouk Univeristy.
- [3] Avaliable at Databooks.katadata.co.id/Accessed Februari 2018.
- [4] Mulyana JRP,"Pentaho: Solusi Open Source untuk Membangun Data Warehouse", Andi Publisher, 2016.
- [5] Ralph Kimblall et al,"Kimball's Data Warehouse ToolKit Classics", Welly, 2014
- [6] Bourbonais et al,"A robust datawarehouse as a requirement to the increasing quantity and complexity of travel survey data", Transportation Research Procedia 32 (2018) 436-447, Elsevier, 2018