OPEN ACCESS
J. OF ICT, VOL. 1, NO. 1, PP.001-009, MAY 2019

ISSN : 123456 (ONLINE)
DOI: XXXXXX

Journal of Informatics and Information Society

Extract, Transform, Load sebagai upaya Pembangunan Data Warehouse

Ade Rahmat Iskandar, Apri Junaidi, Asep Herman
[1)]Program studi Teknik Telekomunikasi, Akademi Telkom Jakarta
[2)] Program studi Teknik Informatika, ITTelkom Purwokerto
[3)]Universitas Kebangsaan Republik Indonesia
ader@akademiktelkom.ac.id[1)],
apri@ittelkom-pwt.ac.id[2)],
asepherman.apn@universitaskebangsaan.ac.id[3)]

Abstract

Paper ini dibuat untuk memberikan gambaran secara general dalam proses transformasi Ekstract, Transform, dan Load (ETL) sebagai data masukan untuk multidimensional modeling data mart dan data warehouse. Artikel ini dibuat dengan mengimplementasikan database dari Online Transaction Processing (OLTP) kedalam database Online Analytical processing (OLAP). Pada penelitian ini digunakan database classicmodels yang bersifat open source dari Mysql.Metode yang dilakukan dalam penelitian ini adalah, dengan melakukan proses Extract, Transform danLoad (ETL) pada data classic models yang dilakukan dengan cara melakukan ketiga proses tersebut (ETL) dari database OLTP kedalam database OLAP.Luaran dari penelitian ini adalah terbntuknya fact oder berisi data dari semua data dimension yang dbiuat untuk data classic model menggunakan perngkat lunak Pentaho Data Intergarion (Kettle) dan database management system MySQL

Keywords: *ETL, OLAP, Datawarehosue, MySQL*

I. INTRODUCTION

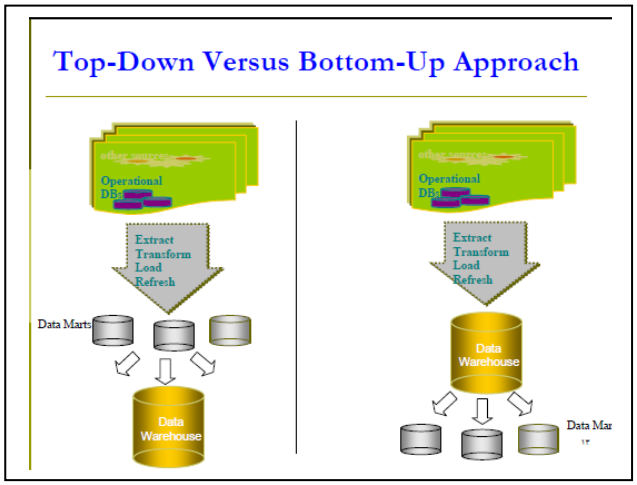
Dalam kurun 1970an penerapan teknologi database sudah sangat familiar digunakan di berbagai instansi baik pemerintahan maupun swasta. Teknologi ini dianggap sangat berperan dalam mengelola data perusahaan secara sistematis dalam pengelolaan sistem yang mudah untuk di insert, update dan delete. Beberapa 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 datau 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 Informsi Akademik di suatu universitas secara realtime yang meningtegrasikan data dari mulai presensi baik yang diinput manual, atau pun menggunakan RFID, web based mapupun mobile sampai pengelolaan data para sivitas akademik secara updatable merupakan salah satu contoh implementasi dari database OLTP.

II. LITERATURE REVIEW

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

Istilah data mart merupakan versi skala yang lebih keci dari dataarehouse. Data mart merupakan small warehousing yang dirancang untuk keperluan tingkat departemen[2].
Berikut adalah merupakan artistektur pembangunan data warehouse yang terdiri dati data mart yang bersifat botom up :



Gambar 1 Peracangan Data Warehouse[2]

- Keuntungan Top-Down Approach
  Pada peracangan top-down approach suatu perusahaan bisa melihat data secara menyeluruh, arsitektur terpusat (tidak ada pembagian antar data marts), data bersifat terpusat, pengedalian 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 mart 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 akir-akhir ini untuk mengelola data eKTP di Indonesia yang terdiri lebih dari 262 juta penduduk di indonesia[3].

Pengelolaan data penduduk yang teritegrasi merupakan salah satu upaya untuk memudahkan dalam mengelola tiap personel individu yang ada di Indonesia, dengan konsep eKTP yang melakukan perakaman data untuk setiap penduduk di Indonesia diharapkan semua jenis data yang berhubungan dengan data personal suatu individu dapat terekam dari satu data dasar yaitu dari eKTP, misalnya pengajuan pembuatan surat ijin mengemudi, pengajuan pembuatan asuransi Badan Pengelolaan Jasa Kesehatan (BPJS), pengajuan pembuatan rekening di perbankan, tes masuk sekolah mulai dari tingkat sekolah dasar sampai dengan perguruan tinggi dan proses-proses lainnya yang relevan dapat dikelola dari satu imput data transaksi eKTP tersebut yang sudah disimpan dalam mega server data kependudukan.

Selanjutnya apa keterkaitan pengelolaan data besar dari transaksional tersebut dengan proses Extracting, Tansforming dan Loading atau lebih populer dikenal dengan nama ETL? Pada dasarnya pengelolaan database sudah trend sejak awal tahun 1960an silam, pengeloaan database biasanya bersifat transaksional untuk mengelola data dengan engan konsep create, update dan delete atau dikenal dengan istilah (CRUD) beberapa tipe sistem informasi dengan konsep transaksion processing system (TPS), sistem informasi manajemen (SIM) biasanya lebih sering dalam mengelola data dengan konsep transaksional, makanya seringkali pengelolaan database transaksional dikenal dengan istilah online transaksional processing (OLTP). Pada buku ini akan dipaparkan lebih mengacu pada pembahasan dari generasi database berikutnya yaitu data warehouse, dimana proses extracting, transforrming dan loading (ETL) terdapat pada saat transfromasi dari sumber-sumber data eksternal seperti dari database management system (DBMS) MysQL, SQl Server atau DBMS lainnya, flat files, xml atau pun sumber data lainnya kedalam database dengan konsep multidimensional modelling atau dikenal dengan istilah lain online analitical processing (OLAP).

Pada paper ini akan dibahan proses ETL menggunakan software open source Pentaho Data Integration versi 5.0, pembaca dapat menggunakan versi terbaru sesuai yang dikehendaki. 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 untu terhubung dengan beragam sumber data, dan membuat data tersebut tersedia bagi proses-proses selanjutnya, misalnya (Read file Excell, Mengambil data dari database, Mebaca file dari XML, dan lain-lain)

- Transform
  Bagian ini mengacu pada fungsi apa saja yang berfungsi untu 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 untu mengisi data ke target. Misalanya hasil dari proses sebelumnya disimpan kedalam database OLAP dan hasil dari proses sebelumnya disimpan kedalam file Excell.

Dalam pembangunan data integration diperlukan tools untuk mengintegrasikan data tersebut. Beberapa tools untuk ETL ada yang berifat license mapupun open source. Beberapa softwae atu tools tersebut adalah Oracle Data Integration, Pentaho Data Integration (Kettle), Microsoft SQL Server Integration Services (SSIS) dan Microsoft SQL Server Integration Services (SSIS).

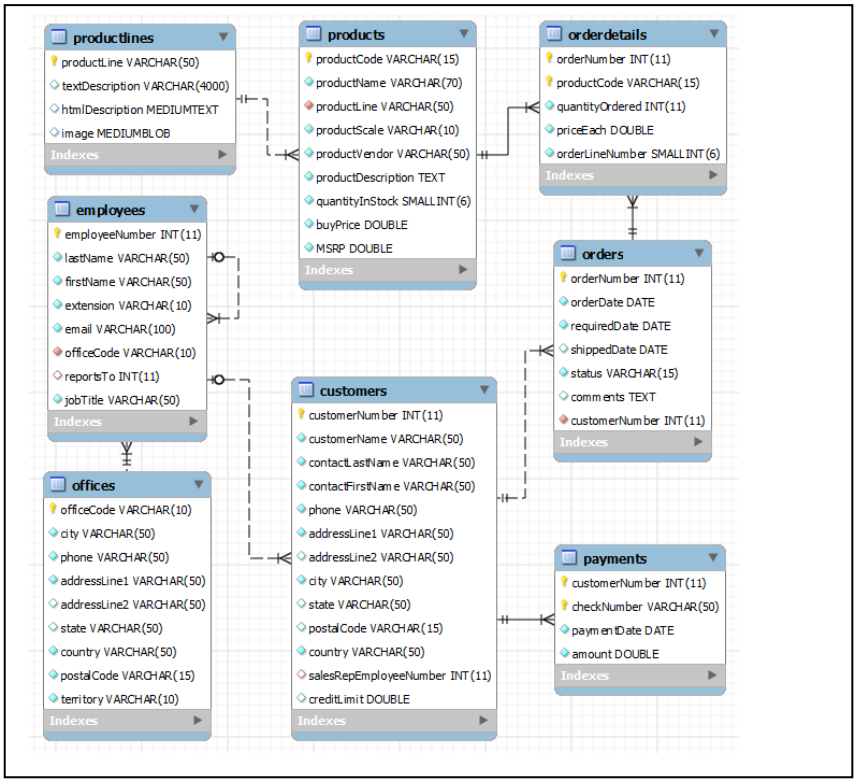
III. RESEARCH METHOD

Metode penelitian yang dilakukan adalah, pertama melakukan collecting data dari data classic model yang bersifat open source dari MySQL, selanjutnya dilakukan beberapa tahapan Extract dan trasnform dari database OLTP yang sudah dirancang dan melakukan load data kedalam database OLAP. Proses transofrmasi dilakukan dengan menggunakan tools untuk mengolah data integration menggunakan Pentaho Data Integration dan RDBMS MySQL.

IV. HASIL DAN PEMBAHASAN

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

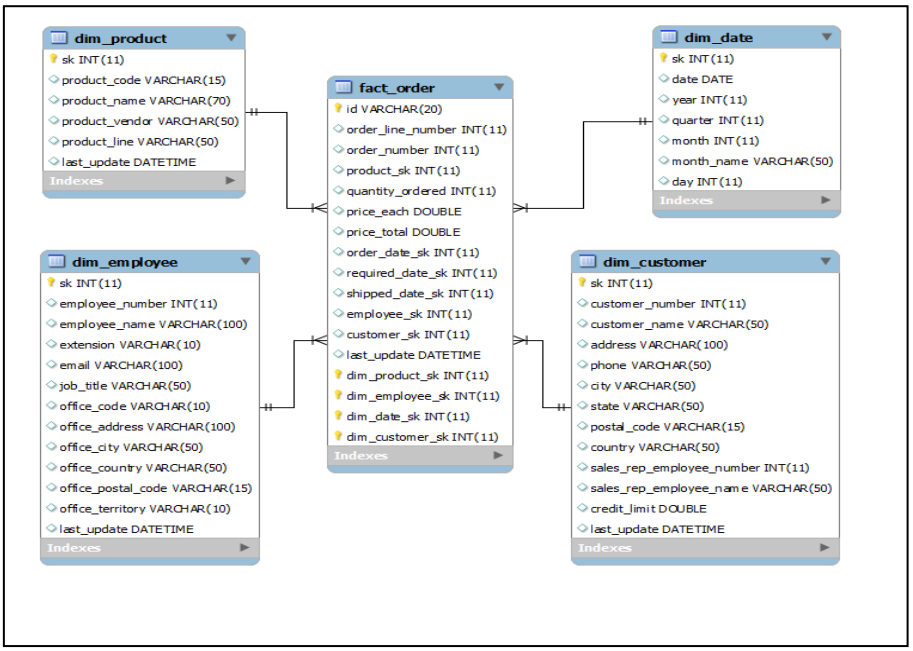
1. Pembangunan skema database OLTP
   Berikut ini adalah model skema rancangan database dari database OLTP yang sudah dibuat mengacu pada transformasi data integration Datawarehouse dengan Pentaho[3]



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

2. Pembagunan Database OLAP

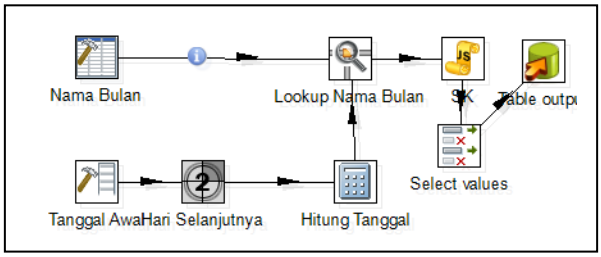
Online Analitycal Processing (OLAP) merupakan database yang akan dibuat berdasarkan data dan skema dari dari OLTP yang sudah dibuat. Pada paper ini Data Mart Sekama dirancang menggunakan Star Schema, berikut adalah hasil rancangan mengacu pada skema OLAP Classic model.



Dalam pembangunan database OLAP ini, dibuat dengan cara mentransformasikan tabel-tabel pada database OLTP kedalam 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 drown waktu yang lebih detail (year, qaurter, month, week dan day), langkah terakhir adalah membuat tabel fact yang merupakan representasi utama dalam pembangunan data mart atau data warekhouse, pada penelitian ini dibuat fact_order yaitu tabel fakta untuk proses transaksi order.

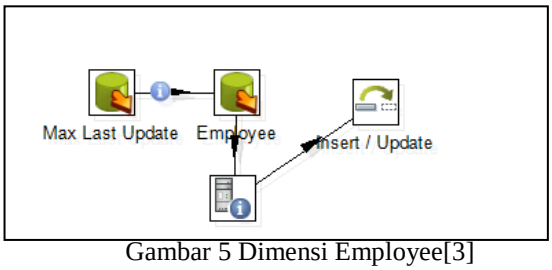
1. Rancangan Dimensi Date

Dimensi date merupakan tabel dimensi yang dibuat untuk membuat cube atau laporan pada dashboard berdasarkan leveling watku yang diharapkan (tahun, quartal, bulan, minggu dan hari), berikut adalah racangan dimensi date pada penelitian ini :



Gambar 4 Dimensi Tanggal[3]

1. Rancangan dimensi Employee
   Dimensi Employee merupakan transformasi tabel Employee dan Office pada database OLTP kedalam dimensi Employee pada database OLAP. Berikut adalah rancangan dimensi Employee:



Gambar 5 Dimensi Employee[3]

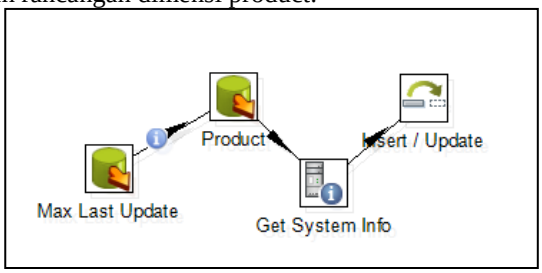
Berikut ini adalah perintah SQL yang digunakan untuk mentransformasikan data dari tabel Emplolyee dan Office kedalam dimensi Employee :

```
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 terebut ditransfromasikan 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 sema field-field yang bersesuaian).

1. Rancangan dimensi Product

Dimensi Product merupakan transformasi tabel product pada database OLTP kedalam dimensi product pada database OLAP. Berikut adalah rancangan dimensi product:

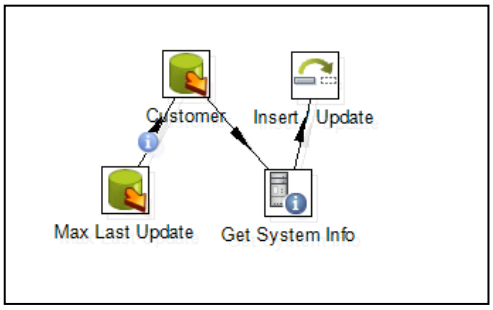


Gambar 6 Dimensi Product[3]

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

Pada perintah SQL terebut ditransfromasikan 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 sema field-field lain yang bersesuaian).

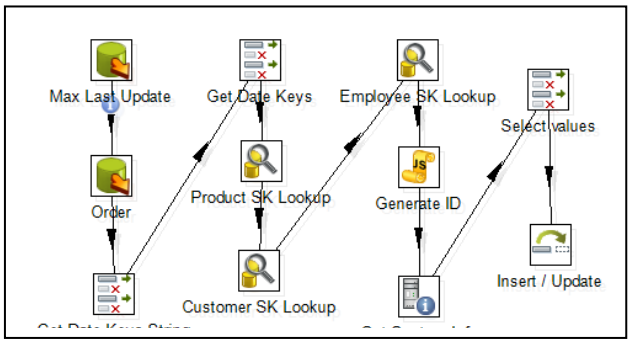
1. Rancangan dimensi Customer
   Dimensi customer merupakan transformasi tabel customer pada database OLTP kedalam dimensi customer pada database OLAP. Berikut adalah rancangan dimensi product:



Gambar 7 Dimensi Customer[3]

```
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 > ?
```

1. Rancangan tabel Fatc Order
   Pada penelitan ini, digunakan salat 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 adalah fact order dari classis models[3].



Gambar 8 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 source code dari kmponen table input Order :

```
 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 source code tersebut, bisa dianalisis bahwa terdapt korelasi dan imelemtnasi 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.

V. KESIMPULAN

Kesimpulan dari penelitan ini adalah, berbagai varians data dapat ditrasnformasikan dari database OLTP kedalam database OLAP. output yang diperoleh dari hasil transformasi OLTP kedalam database OLAP diperoleh data berikut :

id	order_number	order_line_number	product_sk	quantity_ordered	price_each	price_total	order_date_sk	required_date_sk
10100S18_1749	10100	3	23	30	136	4080	20030106	20030113
10100S18_2248	10100	2	27	50	55.09	2754.5	20030106	20030113
10100S18_4409	10100	4	50	22	75.46	1660.12	20030106	20030113
10100S24_3969	10100	1	80	49	35.29	1729.21	20030106	20030113
10101S18_2325	10101	4	29	25	108.06	2701.5	20030109	20030118
10101S18_2795	10101	1	33	26	167.06	4343.56	20030109	20030118
10101S24_1937	10101	3	61	45	32.53	1463.8500000000001	20030109	20030118
10101S24_2022	10101	2	64	46	44.35	2040.1000000000001	20030109	20030118
10102S18_1342	10102	2	19	39	95.55	3726.45	20030110	20030118
10102S18_1367	10102	1	20	41	43.13	1768.3300000000002	20030110	20030118
10103S10_1949	10103	11	2	26	214.3	5571.8	20030129	20030207
10103S10_4962	10103	4	6	42	119.67	5026.14	20030129	20030207
10103S12_1666	10103	8	9	27	121.64	3284.28	20030129	20030207
10103S18_1097	10103	10	17	35	94.5	3307.5	20030129	20030207
10103S18 2432	10103	2	30	22	58.34	1283.48	20030129	20030207

Database OLAP, merupakan data hasil proses Extract, transform dan Load (ETL) berupa data dari file sumber database MySQL. Data historical yang diperoleh adalah sebgai berikut :

order_date_sk	required_date_sk	shipped_date_sk	employee_sk	customer_sk	last_update
20030106	20030113	20030110	10	86	2018-07-27 20:33:3
20030106	20030113	20030110	10	86	2018-07-27 20:33:3
20030106	20030113	20030110	10	86	2018-07-27 20:33:3
20030106	20030113	20030110	10	86	2018-07-27 20:33:3
20030109	20030118	20030111	17	8	2018-07-27 20:33:3
20030109	20030118	20030111	17	8	2018-07-27 20:33:3
20030109	20030118	20030111	17	8	2018-07-27 20:33:3
20030109	20030118	20030111	17	8	2018-07-27 20:33:3
20030110	20030118	20030114	11	28	2018-07-27 20:33:3
20030110	20030118	20030114	11	28	2018-07-27 20:33:3
20030129	20030207	20030202	17	5	2018-07-27 20:33:3
20030129	20030207	20030202	17	5	2018-07-27 20:33:3
20030129	20030207	20030202	17	5	2018-07-27 20:33:3
20030129	20030207	20030202	17	5	2018-07-27 20:33:3

Data historical sangat diperlukan dalam manajemen data menggunakan Data mart atau data warehouse, pada penelitian ini data diiplementasikan untuk 10000 record data.

REFERENCES

[1] Iskandar Ade, Ichsan ”Receomendation for Implementting DataWarehouse for Higher Education in Indonesia”, Proc. IEEXplore, CTSM 2016.

[2] Qasem Al-Radaideh, Data warehouse and the building block, 2012, Yarmouk Univeristy. Avaliable at Databooks.katadata.co.id/Accesed Februari 2018.

[3] Perancangan Data warehouse menggunakan Pentaho Data Integration, 2016_