

---

## ANALISIS FAKTOR OPTIMALISASI PROSES ETL PADA DATA WAREHOUSE SEBAGAI PENDUKUNG PENGAMBILAN KEPUTUSAN MANAGEMENT DENGAN BUSINESS INTELLIGENCE

MUHAMMAD HILMAN  
DJAMALUDIN

Program Studi Teknik Informatika  
Fakultas Teknik, Universitas Islam Syekh Yusuf  
Jalan Maulana Yusuf No.10, Kota Tangerang, Banten 15118  
Email: mhilman@unis.ac.id, djamaludin@unis.ac.id

**Abstract.** The company's need to make decisions appropriately and quickly on the basis of market facts is becoming more critical in order to survive. It is supported by Business Intelligence (BI) tactical in the company. The application of tactical BI as a support for tactical decision making requires some analysis of the current BI system such as data that becomes the material of decision-making must be current data and have a fast query process. The analysis used can be with ETL process and fast query process to support tactical decision-making environment because ETL (Extract Transform Loading) in developing data warehouse process is a process that takes the longest time. The success of the ETL process is strongly influenced by the quality of existing data on OLTP databases. This research aims to find the optimization of ETL process with data warehouse development method. The research method used in the form of inductive study of ETL in a data warehouse. The conclusion is those optimization techniques that have significant impacts can be used to implement Business Intelligence such as the application of parallel transformation, pipelining data, and dimensional data with bulk and parallel data load. For large amounts of data, then the results provided will also have greater impact and more optimal. By applying this optimization, it is expected to support faster query processing and more optimal ETL processes.

*Keywords: Business Intelligence, Data warehouse, ETL, Query, Optimalization*

**Abstrak.** Kebutuhan perusahaan untuk mengambil keputusan secara tepat dan cepat berdasarkan fakta pasar menjadi lebih kritis agar dapat bertahan. Hal ini didukung dengan *Business Intelligence* (BI) taktis di perusahaan. Penerapan BI taktis sebagai pendukung pengambilan keputusan taktis membutuhkan beberapa analisis dari sistem BI yang ada saat ini seperti data yang menjadi bahan pengambilan keputusan haruslah data terkini dan memiliki proses *query* yang cepat. Analisis yang digunakan dapat dengan proses ETL dan proses *query* yang cepat untuk mendukung lingkungan pengambilan keputusan taktis karena ETL (*Extract Transform Loading*) pada proses *develop data warehouse* merupakan suatu proses yang memakan waktu paling lama. Kesuksesan proses ETL sangat dipengaruhi oleh kualitas data yang ada pada database OLTP. Penelitian ini bertujuan untuk mencari optimalisasi proses ETL dengan metode pengembangan *data warehouse*. Metode yang digunakan adalah dengan kajian induktif ETL pada *data warehouse*. Simpulannya ialah teknik-teknik optimasi yang berdampak signifikan bisa dipakai untuk mengimplementasikan *Business Intelligence* seperti penerapan transformasi paralel, data pipelining, data dimensi dengan bulk dan load data secara paralel. Untuk jumlah data yang besar, maka hasil yang diberikan juga akan berdampak lebih besar dan lebih optimal. Dengan menerapkan optimalisasi ini, diharapkan menunjang *query processing* yang lebih cepat dan proses ETL yang lebih optimal.

*Keywords: Business Intelligence, Data warehouse, ETL, Query, Optimalisasi*

## PENDAHULUAN

Kesuksesan dalam membangun *data warehouse* untuk dilakukan analisis dalam mengambil keputusan-keputusan yang tepat dan cepat, sangat dipengaruhi oleh kesuksesan dalam proses ETL dari database OLTP ke database *data warehouse*. ETL (*Extract Transform Loading*) merupakan suatu proses yang memakan waktu paling lama. Kesuksesan proses ETL sangat dipengaruhi oleh kualitas data yang ada pada database OLTP.

Hal diatas bisa didukung oleh adanya *Business Intelligence* (BI) taktis di perusahaan. BI saat ini dianalogikan sebagai *supply chain* informasi. Data didistribusikan dari database operasional secara periodik dengan melewati tahapan ekstraksi, pembersihan, integrasi, dan transformasi yang kemudian dimasukkan ke dalam sebuah *data warehouse*. Selanjutnya, data yang ada di *data warehouse* akan dianalisis dengan cara *di-query* oleh aplikasi analisis seperti OLAP atau *Data Mining*.

Peningkatan performansi *query* dibutuhkan dalam lingkungan pendukung pengambilan keputusan taktis. Keputusan taktis dibuat berkali-kali dalam sehari dan memiliki relevansi yang tinggi berhubungan dengan waktu. Tidak seperti keputusan strategis yang memiliki waktu hidup yang lama seperti bulan atau tahun, keputusan taktis memiliki waktu hidup sekitar menit atau bahkan detik seperti yang dikatakan sebelumnya yaitu *real-time*. Setiap menit keterlambatan yang terjadi dalam keputusan juga dapat menyebabkan keterlambatan proses yang lain yang berarti kerugian yang dihasilkan mungkin bertambah.

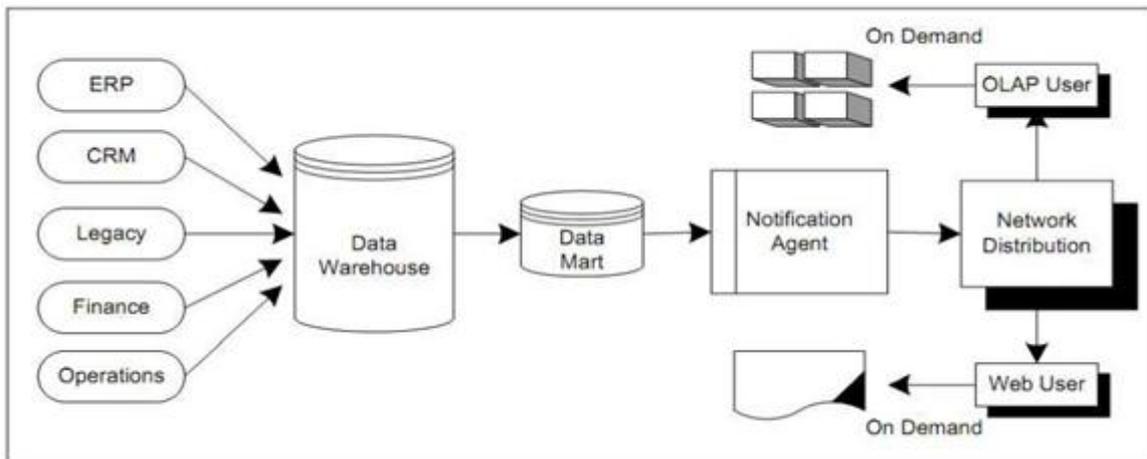
Dari penjelasan di atas, dibutuhkan suatu BI yang berbeda dari yang biasa digunakan. BI yang digunakan haruslah dapat mendukung kesegaran data dan performansi pengambilan informasi yang cepat. Oleh karena itu, solusi untuk permasalahan tersebut yaitu penerapan optimalisasi kedua proses dapat mengimplementasikan *Business Intelligence* taktis di perusahaan. Dengan menerapkan optimalisasi ini, diharapkan menunjang *query processing* yang lebih cepat dan proses ETL yang lebih optimal.

### ***Business Intelligence (BI)***

Menurut Turban et al. (2007), *Business Intelligence* sebuah istilah umum yang meliputi alat, arsitektur, basis data, *data warehouse*, manajemen kinerja, metodologi, dan sebagainya, yang semuanya terintegrasi ke dalam paket perangkat lunak terpadu.

Sedangkan menurut Moss & Hoberman (2004), proses, teknologi, dan peralatan yang diperlukan untuk mengubah data menjadi informasi, informasi menjadi pengetahuan dan pengetahuan menjadi rencana yang mendorong tindakan bisnis yang menguntungkan. BI meliputi *data warehouse*, alat-alat analisis bisnis dan konten/manajemen pengetahuan

Sehingga dapat dikatakan bahwa *Business Intelligence (BI)* adalah kumpulan dari *data warehouse*, *data mining*, analisis data, pelaporan, dan visualisasi data untuk mengumpulkan, mengintegrasikan, membersihkan, dan menggali informasi perusahaan untuk pengambilan keputusan. Data didistribusikan dari database operasional secara periodik dengan melewati tahapan ekstraksi, pembersihan, integrasi, dan transformasi yang kemudian dimasukkan ke dalam sebuah *data warehouse*. Selanjutnya, data yang ada di *data warehouse* akan dianalisis dengan cara *di-query* oleh aplikasi analisis seperti OLAP atau *Data Mining*.



Gambar 1 Arsitektur *Business Intelligence*

Salah satu ciri khas arsitektur BI terletak pada *data warehouse*. Data diekstraksi dari sistem operasional dan didistribusikan dengan teknologi yang ada. Data yang spesifik yang dibutuhkan untuk BI akan di-*download* menuju *data mart* yang digunakan oleh pihak perencanaan dan pihak eksekutif. Output dari BI berupa hasil analisis dari OLAP ataupun teknologi analisis data lainnya. Output dari BI bisa berupa berbagai macam *form* seperti laporan ataupun respon dari permintaan data yang spesifik.

#### **Data Warehouse**

Menurut Ponniah (2010) *Data warehouse* merupakan paradigma baru dilingkungan pengambilan keputusan strategik. *Data warehouse* bukan suatu produk tetapi suatu lingkungan dimana pengguna dapat menemukan informasi strategik.

*Data warehouse* mengandung beberapa elemen penting antara lain (Mallach, 2000):

1. Sumber data yang digunakan oleh *data warehouse*, *database* transaksional dan sumber data eksternal.
2. Proses ETL (*Extraction, Transformation, Loading*) dari sumber data ke *database data warehouse*.
3. Membuat suatu ringkasan atau *summary* terhadap *data warehouse* misalkan dengan menggunakan fungsi agregat.
4. Metadata.

Metadata mengacu data tentang data. Metadata menguraikan struktur dan beberapa arti tentang data, dengan demikian mendukung penggunaan efektif atau tidak efektif dari data.

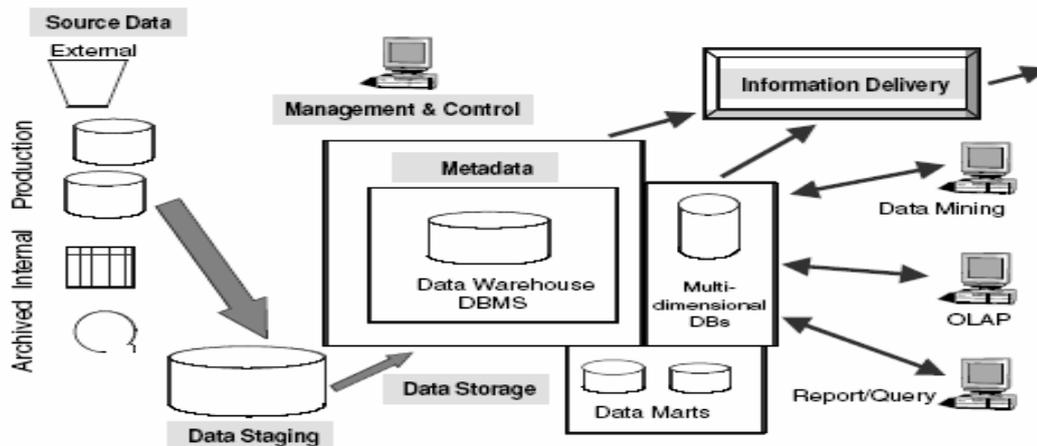
5. *Database data warehouse*.

*Database* ini berisi data yang detail dan ringkasan data dari data yang ada di dalam *data warehouse*. Karena *data warehouse* tidak digunakan dalam proses transaksi individu, maka databasenya tidak perlu diorganisasikan untuk akses transaksi dan untuk pengambilan data, melainkan dioptimisasikan untuk pola akses yang berbeda di dalam analisis.

6. *Query Tools* yaitu dengan OLAP (*Online Analytical Processing*) dan data mining. Tool untuk *query* ini meliputi antarmuka pengguna akhir dalam mengajukan pertanyaan kepada database, dimana proses ini disebut sebagai *On-line Analytical Processing* (OLAP). Tool ini juga terdiri dari tool otomatis yang menemukan pola-pola di dalam data, yang sering disebut sebagai data mining. *Data warehouse* harus memiliki salah satu dari kedua tipe ini atau malah keduanya.

7. Pengguna (*User*).

Pengguna yang memanfaatkan *data warehouse* tersebut.



Gambar 2. Arsitektur *Data warehouse*

Sumber: Poniah, 2001, h. 29.

**ETL (*Extraction, Transformation, Load*)**

Menurut Rainardi (2008) ETL (*Extraction, Transformation, Load*) merupakan proses mengambil dan mengirim data dari data sumber ke *data warehouse*. 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 me-load ke *data warehouse*.

Kualitas data merupakan hal terpenting yang harus diperhatikan dalam membangun *data warehouse*, karena kualitas data mempengaruhi proses ETL. Pada proses ETL jika pada data terjadi suatu *noise* maka proses ETL akan gagal. Kualitas data dapat dilihat dari beberapa parameter, yaitu (melisadata.com, 2010) :

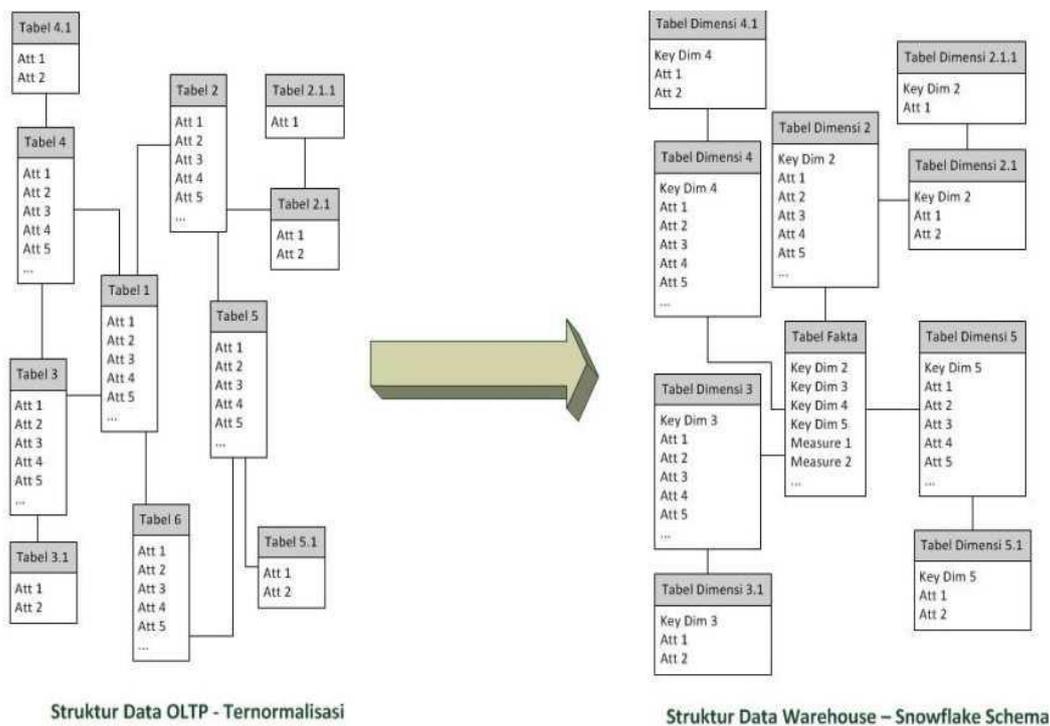
1. Akurat (*accurate*)  
Ketika melihat record alamat konsumen, maka alamat harus mengandung kota, kode pos. Jika konsumen memiliki suatu bisnis maka alamat konsumen juga berisikan alamat atau lokasi dari bisnisnya.
2. Tepat waktu (*Up to date*)  
Selalu memberikan informasi terbaru jika terjadi proses perubahan.
3. Lengkap (*complete*)  
Setiap data harus berisikan informasi penting, misalkan untuk proses surat-menyurat. Misalkan nama apartemen, no apartemen, jalan, kode pos, dan jika dibutuhkan denah alamatnya atau rutenya.
4. Tidak ada redundansi (*not redudant*)  
Misalkan hanya ada satu record per contact untuk setiap alamat dalam surat menyurat.
5. Standar (*standardized*)  
Setiap *record* harus standar dalam pemberian nama, proses pembacaan, dan singkatan.

## HASIL DAN PEMBAHASAN

### Optimalisasi Skema *Data Warehouse*

Proses pengoptimalan struktur *data warehouse* saat ini kurang digunakan. Padahal dengan digunakannya teknik pengoptimalan pada struktur *data warehouse*, dapat diperoleh seperti kemudahan perawatan data dan meningkatnya waktu proses untuk *query*.

Pada umumnya, struktur data OLTP merupakan struktur data yang ternormalisasi. Hal ini disebabkan oleh adanya kebutuhan akan data yang tidak redundan dan perawatan data yang lebih mudah. Saat struktur data OLTP berupa struktur data yang dinormalisasi, maka akan jauh lebih mudah untuk membuat struktur *data warehouse* yang dinormalisasi juga yaitu dengan menggunakan skema snowflake seperti pada Gambar 3.

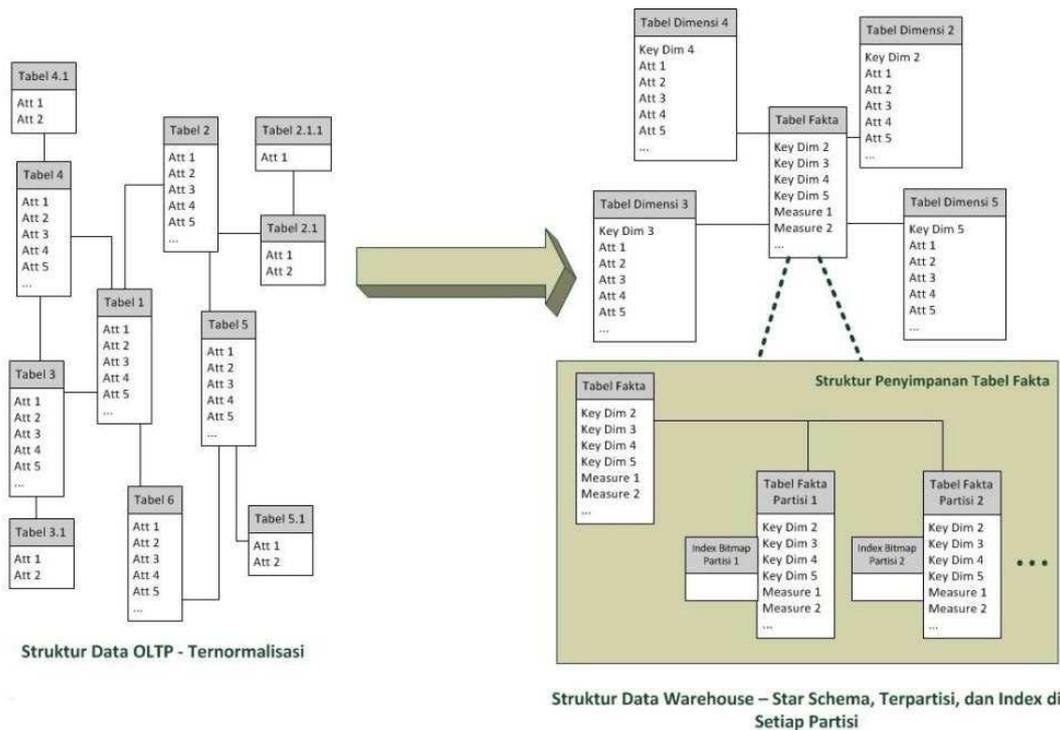


Gambar 3 Struktur *Data warehouse* Sebelum Dioptimalkan

Skema *snowflake* memiliki kelemahan-kelemahan utama yaitu lambatnya waktu *query* yang dibutuhkan karena banyaknya *join* yang harus dilakukan, maka diterapkan teknik-teknik pengoptimalan yaitu mendenormalisasikan skema menjadi skema star dan menerapkan penggunaan indeks untuk mengoptimalkan proses *query*.

Perawatan data akan lebih sulit yang untuk skema bintang akan menggunakan partisi penyimpanan tabel fakta. Dengan partisi penyimpanan tabel fakta, perawatan data akan jauh lebih mudah karena hanya merawat partisi-partisi yang berubah saja, bukan data keseluruhan. Selain itu, tersedianya partisi penyimpanan tabel fakta menyediakan fasilitas untuk bisa melakukan pemrosesan *query* di setiap partisi secara paralel.

Berdasarkan analisis diatas, diterapkanlah tiga teknik pengoptimalan untuk struktur *data warehouse* yaitu denormalisasi skema, penggunaan indeks, dan partisi penyimpanan tabel fakta. Penerapan teknik ini bisa dilihat di struktur *data warehouse* yang telah dioptimalisasi pada Gambar 4.



Gambar 4 Struktur *Data warehouse* yang Telah Dioptimalkan

**Optimalisasi Skema ETL (*Extract, Transform, Load*)**

Setelah mengekstraksi data dari berbagai macam sistem operasional, data harus disiapkan untuk disimpan ke dalam *data warehouse*. Data-data hasil ekstraksi yang berasal dari beberapa sumber berlainan harus diubah, dikonversi, dan membuatnya siap dalam satu format yang sesuai untuk disimpan dan digunakan bagi keperluan *query* dan analisis. Tahap pembersihan ini dikenal juga dengan istilah *Extraction, Transformation, and Loading* (ETL). Tahap pembersihan ini berlangsung di sebuah *staging area*. *Data staging* menyediakan sebuah tempat dengan satu set fungsi untuk membersihkan, mengubah, menggabungkan, mengkonversi, mencegah duplikasi data, dan menyiapkan data sumber untuk penyimpanan dan penggunaan dalam *data warehouse*.

Selanjutnya akan dijelaskan mengenai proses pengoptimalan pada proses ETL. Pengoptimalan pada proses ETL ditujukan untuk mempercepat waktu proses ETL secara keseluruhan. Penjelasan bagian ini akan dibagi per proses utama ETL yaitu proses ekstraksi, proses transformasi, dan proses *load*.

**A. Optimalisasi Proses Ekstraksi**

Pada proses ekstraksi, yang biasa dilakukan adalah mengekstraksi keseluruhan data, menseleksi data yang dibutuhkan, lalu membersihkan data yang hilang dan tidak lengkap. Setelah itu, data dimasukkan dengan menggunakan *data manipulation language* yaitu sintaks SQL INSERT. Proses ini dapat dilihat pada Gambar 5.



Gambar 5 Proses Ekstraksi yang Belum Dioptimalkan

Kekurangan pada proses ini adalah kurang optimal karena harus mengekstraksi seluruh data. Mengekstraksi seluruh data menyebabkan banyaknya data yang harus diproses pada proses ETL. Hal ini tidak menjadi masalah saat proses ETL pertama saat membuat *data warehouse*. Akan

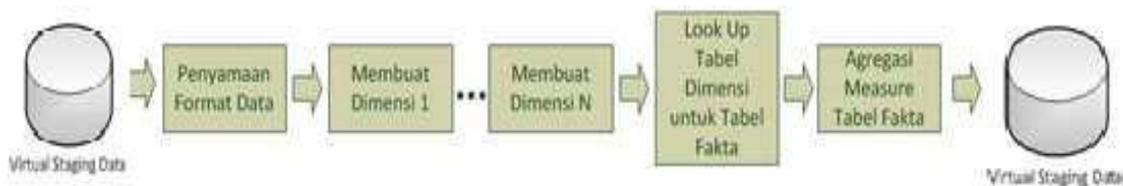
tetapi menjadi kekurangan yang signifikan saat proses ETL selanjutnya karena terdapat redundansi data yang diperoleh dari sumber. Oleh karena itu, diterapkanlah beberapa teknik pengoptimalan untuk menangani permasalahan tersebut. Teknik yang digunakan antara lain adalah mendeteksi load yang berubah agar tidak mengekstraksi seluruh data sehingga jumlah data yang harus diproses semakin efisien. Untuk mempercepat waktu ETL digunakan indeks pada database sumber agar proses seleksi data semakin cepat dan mengekstraksi data secara paralel. Keseluruhan proses optimalisasi dapat dilihat pada Gambar 6.



Gambar 6 Proses Ekstraksi yang Sudah Dioptimalkan

#### B. Optimalisasi Proses Transformasi

Proses transformasi dimulai dari penyamaan format data. Tidak semua sumber memiliki bentuk atau struktur data yang sama. Oleh karena itu, dibutuhkannya transformasi data untuk menstandarisasi domain data dan ukurannya.



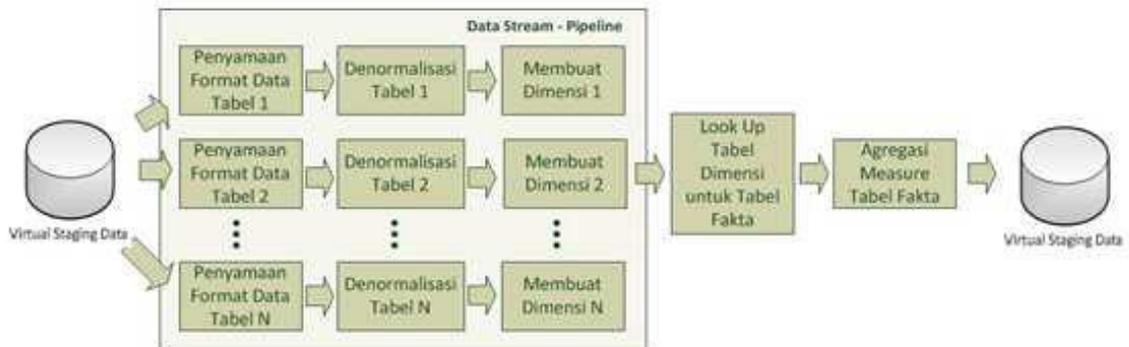
Gambar 7. Proses Transformasi yang Belum Dioptimalkan

Dengan asumsi bahwa struktur data sumber berupa struktur data yang dinormalisasi dan struktur *data warehouse* yang dipilih adalah skema *snowflake*, maka proses yang dilakukan selanjutnya adalah membuat tabel dimensi untuk *data warehouse*. Pembuatan tabel dimensi ini dapat dilakukan langsung tanpa preproses karena struktur data OLTP sudah berupa data yang dinormalisasi. Proses dilakukan secara sekuensial untuk setiap dimensi yang dibuat dan data yang diproses berupa batch. Selanjutnya proses transformasi dilanjutkan dengan membuat tabel fakta yaitu mencari kunci untuk setiap dimensi dan mengisi measure tabel fakta dengan agregasi-agregasi atribut dimensi. Hal ini tergambar pada Gambar 7.

Pada proses ini, terdapat beberapa kelemahan seperti pembuatan dimensi secara sekuensial serta pemrosesan data berupa batch. Pembuatan dimensi secara sekuensial adalah proses yang tidak optimal melihat bahwa antar dimensi tidak memiliki keterhubungan satu sama lain sehingga bisa proses ini bisa dilakukan secara paralel. Sedangkan pemrosesan data berupa batch memiliki kekurangan yaitu membutuhkan kapasitas memori yang besar dan menyebabkan delay waktu yang cukup lama.

Oleh karena itu diterapkan teknik-teknik untuk mengatasi hal tersebut yaitu pembuatan tabel dimensi secara paralel dan menggunakan data stream untuk pemrosesan data. Dengan

menimbang bahwa struktur *data warehouse* yang digunakan adalah skema star, maka akan ditambahkan proses denormalisasi untuk pembuatan tabel yang ada. Selain itu, diterapkan pula pengoptimalan penulisan aljabar relasional dalam proses denormalisasi dan pembuatan tabel dimensi agar waktu proses semakin cepat. Penerapan teknik-teknik pengoptimalan pada proses transformasi dapat dilihat pada Gambar 8.



Gambar 8: Proses Transformasi yang Sudah Dioptimalkan

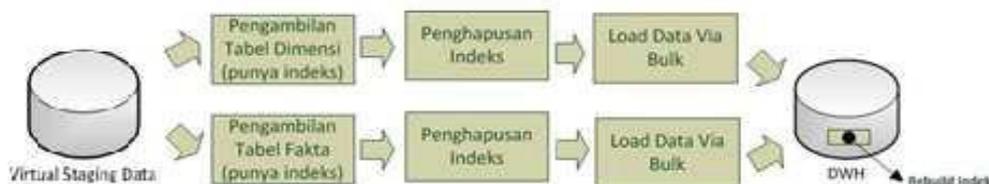
### C. Optimalisasi Proses Load

Pada proses load, dilakukan proses memasukkan data ke dalam *data warehouse*. Data yang dimasukkan adalah data yang telah ditransformasikan agar siap di-*query* dalam bentuk struktur *data warehouse*. Data dimasukkan beserta dengan indeksnya lalu memasukkannya menggunakan DML. Hal ini dapat dilihat pada Gambar 9.



Gambar 9. Proses Load Data yang Belum Dioptimalkan

Memasukkan data menggunakan DML juga memiliki kekurangan untuk lingkungan *data warehouse*. Hal ini disebabkan oleh proses memasukkan data menggunakan DML membutuhkan proses pencatatan pada suatu log untuk setiap row yang dimasukkan. Hal ini akan menjadi kekurangan yang signifikan saat data yang dimasukkan berjumlah jutaan. Permasalahan timbul saat kita memasukkan data beserta dengan indeksnya. Saat data yang berindeks dimasukkan ke dalam *data warehouse*, maka akan terjadi overhead proses yaitu mengupdate indeks untuk setiap row data yang dimasukkan. Salah satu penanganan untuk masalah ini adalah menghapus indeks lalu dibangun kembali setelah data masuk ke dalam *data warehouse*. Berdasarkan analisis tersebut, diterapkan beberapa teknik pengoptimalan seperti penghapusan indeks dan memasukkan data dengan bulk seperti yang digambarkan pada Gambar 10.



Gambar 10. Proses Load Data yang Sudah Dioptimalkan

## Pengujian

Tujuan dari dilakukannya pengujian skema *data warehouse* dan skema ETL adalah untuk bisa melihat apakah hasil implementasi dari teknik optimalisasi skema *data warehouse* dan teknik optimalisasi skema ETL berdampak cukup signifikan untuk menghasilkan *query processing* dan proses ETL yang lebih cepat dan lebih optimal.

Pengujian dilakukan di platform komputer dengan spesifikasi sebagai berikut,

Operating System : Windows 7 Professional 32-bit (6.1, Build 7600)  
Processor : Intel(R) Core(TM) 2 Duo CPU T5750 2.00GHz (2 CPUs), ~2.0GHz  
Memory : 3072MB RAM

Pengujian dilakukan di dua lingkungan yang berbeda yaitu lingkungan dengan jumlah data kecil dan lingkungan dengan jumlah data besar. Lingkungan data kecil memiliki data dengan besar berkisar antara 2 – 16.000 rows. Sedangkan lingkungan data besar memiliki data dengan jumlah berkisar 109 – 150.000 rows.

### Pengujian Skema *Data warehouse*

Pada pengujian waktu *query*, dilakukan pengujian untuk setiap skema *data warehouse* yang dibuat. Skema *data warehouse* yang dibuat berjumlah 3 yaitu skema *data warehouse* utama dan skema *data warehouse* lain sebagai pembanding. Skema *data warehouse* yang menjadi skema utama adalah skema data jenis bintang yang sudah menerapkan optimalisasi di dalamnya. Sedangkan skema *data warehouse* pembanding adalah skema data jenis snowflake dan skema jenis bintang yang tidak teroptimalisasi.

Dalam pengujian ini, dieksekusi 5 *query* yang sekiranya akan menjadi *query* yang sering digunakan dalam pengambilan keputusan untuk bisnis DVD rental. Kelima *query* tersebut adalah antara lain:

1. Menghitung jumlah pembayaran rental untuk setiap cabang toko
2. Menghitung jumlah peminjaman DVD berdasarkan kota dari setiap *customer* peminjam
3. Menghitung rata-rata durasi peminjaman setiap negara dari cabang toko
4. Menghitung rata-rata durasi peminjaman untuk setiap kota dari customer peminjam
5. Menghitung jumlah peminjaman per bulan

Hasil pengujian skema *data warehouse* terbagi menjadi dua hasil pengujian untuk setiap lingkungan pengujian yang berbeda. Hasil untuk kedua lingkungan pengujian skema *data warehouse* dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Skema *Data warehouse*

<i>Query</i>	Data Kecil			Data Besar		
	Skema <i>Snowflake</i>	Skema Bintang	Skema Bintang Optimal	Skema <i>Snowflake</i>	Skema Bintang	Skema Bintang Optimal
1	96 ms	130 ms	125 ms	375 ms	306 ms	265 ms
2	62 ms	70 ms	78 ms	530 ms	425 ms	312 ms
3	78 ms	94 ms	94 ms	218 ms	200 ms	187 ms
4	78 ms	79 ms	74 ms	296 ms	270 ms	234 ms
5	46 ms	48 ms	50 ms	203 ms	210 ms	203 ms

### Hasil Pengujian Skema ETL

Pada pengujian skema ETL, dilakukan pengujian untuk setiap skema ETL yang dibuat. Skema ETL yang dibuat berjumlah 3 yaitu satu skema *ETL* utama dan dua skema *ETL* lain sebagai pembanding. Skema ETL yang menjadi skema utama adalah skema ETL untuk skema data jenis bintang yang sudah menerapkan optimalisasi di dalamnya. Sedangkan skema ETL pembanding antara lain adalah skema ETL untuk skema data jenis snowflake dan skema ETL untuk skema data jenis bintang yang belum teroptimalisasi.

Pada bagian ini, akan dijelaskan mengenai rangkuman dari hasil pengujian untuk kedua lingkungan pengujian. Rangkuman ini berupa tabel yang memperlihatkan apakah teknik optimasi yang diterapkan memberikan dampak yang signifikan atau tidak kepada optimasi. Sebuah teknik optimasi dianggap memberikan dampak yang signifikan saat waktu proses berkurang sebanyak 30% dan waktu kecepatan pemrosesan per barisnya meningkat. Hasil pengujian ini dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Skema ETL

Jenis Proses	Teknik Optimasi	Dampak yang Diberikan
Ekstraksi	Pengambilan Data yang Dibutuhkan Saja dengan seleksi atribut (bukan select *)	Tidak Signifikan
	Ekstraksi secara paralel	Tidak signifikan
Tranformasi	Transformasi secara Paralel	Signifikan
	Penulisan ulang aljabar transformasi (Membuat perubahan pada urutan subproses agar efisien)	Tidak Signifikan
	Data <i>Pipelining</i> saat Transformasi	Signifikan
Load	Memasukkan data dimensi dengan Bulk	Signifikan
	Drop Indeks Basis Data Target Sebelum Proses Load Dimensi dilakukan	Tidak signifikan
	Load Data Secara Paralel	Signifikan

## PENUTUP

### Simpulan

Berdasarkan apa yang sudah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

Penerapan skema bintang, pengindeksan, dan partisi memberikan dampak yang cukup signifikan untuk proses optimalisasi untuk lingkungan pengujian data besar. Teknik optimalisasi seperti penghapusan indeks basis data target sebelum proses load, ekstraksi secara paralel, penulisan ulang aljabar relasional, dan pengambilan data yang berubah tidak memberikan dampak yang signifikan sehingga tidak terlalu cocok untuk diterapkan sebagai teknik optimalisasi pada lingkungan *Business Intelligence* taktis. Untuk bisa mengikuti lingkungan *Bisnis Intelligence Taktis*, maka teknik-teknik optimasi yang memberikan dampak yang signifikan bisa dipakai seperti penerapan transformasi paralel, data *pipelining*, load secara paralel. Tidak semua penerapan teknik optimasi memberikan dampak yang signifikan pada semua kasus. Sebagai contoh, penerapan transformasi paralel hanya memberikan dampak yang signifikan untuk proses konversi format dan pencarian key tetapi tidak

untuk proses agregasi. Walau pun banyak teknik optimasi yang tidak memperlihatkan dampaknya pada lingkungan pengujian data kecil, tetapi banyak dari teknik optimasi tersebut yang memperlihatkan perbedaan yang cukup signifikan pada lingkungan data besar. Jika teknik optimasi ini diterapkan di jumlah data yang lebih besar dari lingkungan pengujian saat ini, maka hasil yang diberikan juga akan lebih berdampak besar dan lebih optimal.

#### **DAFTAR PUSTAKA**

- Turban, E, Sharda, R, Aronson, J & King, D. 2007, *Decision Support Systems and Intelligent Systems (Sistem Pendukung Keputusan dan Sistem Cerdas)* Jilid 1. Andi Offset: Yogyakarta.
- Moss, L & Hoberman, S. 2004. *The Importance of Data Modeling as a Foundation for Business Insight*. Teradata
- Mallach, Efrem G. 2000. *Decision Support and Data Warehouse Systems*. Singapore: Irwin McGraw Hill.
- Melisa Data, 2010, Scalable Data Quality [online],  
<http://www.melissadata.com/dqt/whitepaper/scalable-data-quality-whitepaper.pdf> tanggal akses 13 Februari 2018.
- Ponniah, Paulraj. 2001. *Data Warehouse Fundamentals: a Comprehensive Guide for IT professional*. New York: John Wiley & Sons.
- Rainardi, Vincent, 2008, *Building a Data Warehouse with Examples in SQL Server*. Apress: New York.
- Dayal, U., Castellanos, M., Simitsis A., & Wilkinson, K. (2009). *Data Integration Flows for Business Intelligence*. EDBT 09.
- Han, Jiawei K., Michelin. (2006). *Data Mining: Concepts and Techniques, 2nd Edition*. Morgan Kaufmann Publisher.
- Kimball, & Ralph. (2004). *The Data warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data/Ralph Kimball, Joe Caserta*. Wiley Publishing Inc. Indianapolis.
- Tang Jun, dkk. (2009). *The Research & Application of ETL tool in Business Intelligence Project. 2009. International Forum on Information Technology and Applications*. IEEE.