

## OPTIMASI PROSES ETL DENGAN METODE HEURISTIK UNTUK MEMBANGUN DATA WAREHOUSE

I Nyoman Adnyana Putra, Rukmi Sari Hartati, Ni Wayan Sri Aryani

Magister Teknik Elektro/Program Pasca Sarjana Universitas Udayana  
Denpasar, Indonesia

[Mangputra27@gmail.com](mailto:Mangputra27@gmail.com), [rukmisari@unud.ac.id](mailto:rukmisari@unud.ac.id), [sriariyani@unud.ac.id](mailto:sriariyani@unud.ac.id)

### Abstrak

ETL (Ekstrak, Transformasi, Load) merupakan suatu proses yang paling penting dalam pembangunan data warehouse. Secara garis besar, fungsi proses ETL adalah untuk mengumpulkan data dari berbagai sumber data yang berbeda-beda menjadi kumpulan banyak data yang terintegrasi. Dengan semakin banyaknya data yang diintegrasikan maka kinerja dari proses ETL juga akan menjadi semakin lama dan semakin berat. Hal ini akan sangat mempengaruhi performa dari kinerja data warehouse secara keseluruhan. Pemerintah Provinsi Bali, dalam melaporkan realisasi pengadaan kepada pimpinan dilakukan setiap tiga bulan dan bersifat manual padahal data dari yang dilaporkan bersumber dari tiga sistem yang sudah diterapkan. Didalam penelitian ini, membangun data warehouse dengan melakukan optimasi pada proses ETL akan menjadi jawaban untuk permasalahan diatas. Optimasi ETL dalam pembangunan data warehouse menggunakan metode Heuristik. Hasil yang didapatkan dari penelitian ini adalah optimasi ETL menggunakan metode heuristik dapat memaksimalkan waktu dan beban memory saat proses ETL. Dari tiga sumber data yang digunakan dengan 17.024 jumlah data yang diolah, selisih waktu antara ETL sebelum optimasi dengan yang di optimasi mencapai 103 menit.

**Kata Kunci : Data Warehouse; ETL; Optimasi;**

### Abstract

ETL (Extract, Transformation, Load) is the most important process in data warehouse development. Broadly speaking, the ETL process function is to collect data from multiple different data sources into a large collection of integrated data. As more data is integrated, the performance of the ETL process will also become longer and heavier. This will greatly affect the performance of the overall data warehouse performance. Bali Provincial Government, in reporting the realization of procurement to the leadership is done every three months and is manual when the data from the reported sourced from three systems that have been applied. In this study, building a data warehouse by optimizing the ETL process will be the answer to the above problems. ETL optimization in data warehouse development using Heuristic method. The result of this research is ETL optimization using heuristic method can maximize memory time and load during ETL process. Of the three data summers used with 17,024 the amount of data processed, the time difference between ETL before optimization and optimization of 103 minutes.

**Keywords—Data Warehouse; ETL; Optimazation;**

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan global teknologi informasi saat ini berperan penting dalam dunia bisnis baik swasta maupun pemerintahan. Dalam dunia pemerintahan, Pemerintah Provinsi Bali yang saat ini masih melakukan proses pelaporan realisasi pengadaan secara manual, dapat memanfaatkan teknologi informasi untuk mendigitalisasi laporan tersebut. Laporan realisasi pengadaan pada Pemerintah Provinsi Bali bersumber dari tiga aplikasi yang sudah diterapkan yakni, SiRUP, SPSE dan SiMPEL. Ketiga aplikasi tersebut berbasis web namun pelaporannya masih bersifat manual. Diperlukan suatu sistem yang dapat mengolah informasi dari ketiga aplikasi tersebut agar sistem pelaporan dapat didigitalisasi sehingga pimpinan dapat melakukan monitoring realisasi pengadaan setiap saat yang nantinya akan berimplikasi pada pengambilan keputusan. Data warehouse merupakan salah satu konsep pengolahan data yang baik dengan sumber data yang dapat berasal dari banyak sistem. Menurut beberapa pakar, sekitar 60%-80% upaya dalam pembangunan data warehouse dihabiskan untuk proses ETL. Proses ETL yang menangani jumlah data yang banyak serta kerja yang rumit, sehingga memerlukan waktu penyelesaian yang lama dan membutuhkan memory yang besar. Dalam penelitian ini akan dilakukan modifikasi untuk mengurangi jumlah baris data yang diproses dengan mengatur ulang transformasi pada proses ETL dengan menggunakan metode heuristik.

### 1.2. Pokok Masalah

Pada jurnal ini membahas bagaimana mengurangi beban ETL pada pembangunan data warehouse dengan melakukan optimasi proses ETL menggunakan metode heuristik.

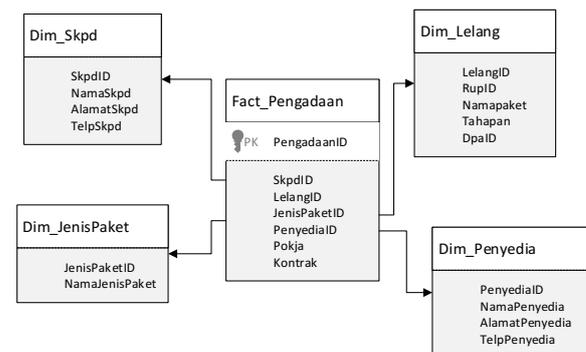
Kemudian menganalisis seberapa baik kinerja heuristik dengan membandingkan sebelum dan setelah dilakukan optimasi proses ETL.

### 1.3. Tujuan

Penelitian ini bertujuan untuk mendesain alur kerja ETL dalam pembangunan data warehouse, kemudian menerapkan metode heuristik dengan melakukan perubahan pada struktur kerja dan kemudian menerapkan aturan-aturan untuk melakukan transformasi dengan operasi-operasi relasi aljabar ke dalam bentuk persamaan. Selanjutnya melakukan perbandingan waktu eksekusi algoritma dan penggunaan memory sebelum dan setelah dilakukan optimasi proses ETL.

## 2. DATA WAREHOUSE

### 2.1. Data Warehouse



**Gambar 1.** Skema Bintang Perancangan Database

Data warehouse merupakan kumpulan data yang mendukung pengambilan keputusan manajemen yang memiliki karakteristik berorientasi objek (*subject oriented*), terintegrasi (*integrated*), memiliki variasi waktu tertentu (*time-variant*) dan tidak dapat berubah (*non-volatile*) [H.W. Inmon (2002, p31)]. Untuk perancangan database dalam pembangunan data warehouse pada penelitian ini menggunakan skema bintang.

Rancangan untuk skema bintang tersebut seperti Gambar 1.

### 2.2 ETL

ETL (Extraction, Transformation, Load) adalah fungsi integrasi data yang melibatkan menggalian data dari sumber luar (sistem operasional) yang kemudian melakukan perubahan agar sesuai dengan kebutuhan bisnis dan terakhir di loading ke data warehouse. ETL merupakan suatu proses awal yang harus dilalui sebelum masuk kedalam data warehouse. Extraction dilakukan untuk membaca data dari sebuah sumber data tertentu dan mengekstrak subset data yang dibutuhkan. Extraction dilakukan secara berkala dan terjadwal, bergantung pada kebutuhan sistem pengolahan data. Transformation adalah suatu proses yang mana data mentah (raw data) hasil extraction dibagi dan diubah sesuai dengan aturan bisnis yang berlaku. Sedangkan Load adalah akhir dari proses ETL dimana yang perlu dilakukan adalah proses penggabungan data atas data yang didapatkan dari hasil transformasi sebelumnya ke dalam data warehouse.

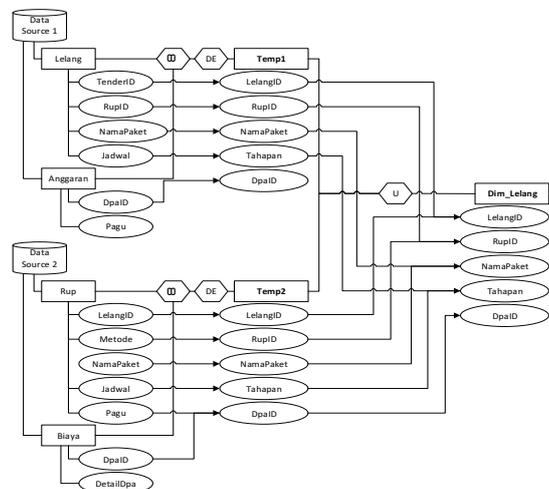
Heuristik (Heuristik Optimization) atau biasa disebut juga dengan *rule based optimization* merupakan optimisasi ETL dengan melakukan perubahan pada mekanisme alur kerja ETL guna mempersingkat waktu proses dan untuk pengolahan datanya menggunakan aturan - aturan heuristik dan dijalankan pada *logical query plan* (rencana query secara logika) yang memiliki urutan operasi - operasi relasional yang biasanya digambarkan sebagai *query tree*.

Optimisasi ETL dapat diproses dengan lebih efisien karena menggunakan aturan - aturan heuristik dan memperoleh sebuah inisial plan dari parser untuk mentransformasikan sebuah query kedalam sebuah bentuk yang sama. Adapun tujuan dari transformasi tersebut adalah :

- Standarisasi, adalah sebuah query yang ditransformasikan ke dalam sebuah bentuk standar tanpa dilakukan optimisasi.
- Simplifikasi, adalah menghilangkan kelebihan pada suatu query.
- Ameliorasi, adalah proses evaluasi bentuk yang melalui penyusunan ekspresi - ekspresi yang sudah dihasilkan dengan baik.

## 3. OPTIMASI HEURISTIK

### 3.1 Teknik Query



**Gambar 2.** Alur Proses optimasi ETL

Banyak aturan - aturan yang digunakan untuk mentransformasikan

suatu operasi - operasi relasi aljabar kedalam suatu bentuk persamaan.

Dibawah ini adalah beberapa aturan-aturan terkait dengan transformasi untuk operasi - operasi dari relasi aljabar.

1. Pengurutan  $\sigma$  : Suatu keadaan konjungtif yang dapat dipisahkan ke sebuah urutan - urutan dari operasi - operasi  $\sigma$  tersendiri :  

$$\sigma c1 \text{ AND } c2 \text{ AND } \dots \text{ AND } cn \text{ }^{(R)} \equiv \sigma c1 (\sigma c2 (\dots (\sigma cn(R)) \dots))$$
2. Perubahan  $\sigma$  : Operasi  $\sigma$  dirubah menjadi:  

$$\sigma c1(\sigma c2(R)) \equiv \sigma c2(\sigma c1(R))$$
3. Pengurutan  $\pi$  : Sebuah urutan dari suatu operasi - operasi  $\pi$ , tetapi diabaikan urutan yang terakhir:  

$$\pi \text{List1}(\pi \text{List2}(\dots (\pi \text{Listn}(R)) \dots)) \equiv \pi \text{List1}(R)$$
4. Mengubah  $\sigma$  dengan  $\pi$ : Jika pada kondisi pilihan  $c$  hanyalah meliputi attribute-attribute dari  $A1, \dots, An$  dalam sebuah daftar proyeksi, maka kedua operasi yang ada dapat dirubah menjadi :  

$$\pi A1, A2, \dots, An(\sigma c (R)) \equiv \sigma c (\pi A1, A2, \dots, An(R))$$
5. Perubahan  $\bowtie$  dan  $\times$  : operasi  $\bowtie$  yang dirubah seperti operasi  $\times$  :  

$$R \bowtie c S \square S \bowtie c R$$

Yang perlu diperhatikan adalah urutan dari attribute yang mungkin berbeda dalam relasi yang dihasilkan dari mekanisme join (atau kedua cartesian product).
6. Melakukan perubahan  $\sigma$  menjadi  $\bowtie$ . Jika semua attribute dalam pilihan berada pada kondisi  $c$  hanyalah meliputi attribute dari satu relasi yang digabung (misal  $R$ ), maka kedua operasi tersebut akan dapat dirubah seperti berikut ini :  

$$\sigma c (R \bowtie S) \equiv (\sigma c(R)) \bowtie S$$
7. Melakukan perubahan  $\pi$  menjadi  $\bowtie$ : Asumsikan bahwa daftar proyeksi adalah  $L = \{A1, \dots, An, B1, \dots, Bm\}$  di mana  $A1, \dots, An$  merupakan attribute dari  $R$  dan  $B1, \dots, Bm$  merupakan attribute dari  $S$ . Jika kondisi gabungan  $c$  hanya terdiri dari

attribute pada  $L$ , maka kedua operasi tersebut dapat dirubah sebagai berikut :

$$\pi L(R \bowtie c S) \equiv a ((\pi A1, \dots, An, An+1, \dots, An+k (R)) \bowtie c (\pi B1, \dots, Bn, Bn+1, \dots, Bn+k (S)))$$

8. Perubahan pada sekumpulan operasi-operasi : Kumpulan dari operasi  $\cup$  dan  $\cap$  adalah sebuah perubahan, tetapi  $L$  bukan merupakan perubahan.
9. Melakukan penggabungan  $\bowtie$ ,  $\times$ ,  $\cup$  dan  $\cap$  : Keempat operasi ini merupakan gabungan dari individu yang ada. Oleh karena itu, apabila  $\theta$  mewakili untuk salah satu dari keempat operasi tersebut maka :  

$$(R \theta S) \square T \theta R \theta (S \theta T)$$
10. Melakukan perubahan  $\sigma$  menjadi sekumpulan operasi - operasi : operasi  $\sigma$  dirubah menjadi  $\cup$ ,  $\cap$ , dan  $-$ . Apabila  $\theta$  mewakili untuk salah satu dari ketiga operasi tersebut, maka :  

$$\pi c (R \theta S) \equiv (\pi c (R)) \theta (\pi c (S))$$
11. Operasi  $\pi$  dirubah menjadi  

$$\cup : \pi L (R \cup S) \equiv (\pi L (R)) \cup (\pi L (S))$$
12. Mengkonversikan suatu urutan  $(\sigma, \times)$  ke dalam  $\bowtie$ : Jika kondisi  $c$  dari sebuah  $\sigma$  yang mengikuti sebuah  $\times$  cocok untuk sebuah kondisi join, maka urutan  $(\sigma, \times)$  dikonversikan ke dalam sebuah  $\bowtie$  sebagai berikut :  

$$\sigma c (R \times S) \equiv (R \bowtie c S).$$

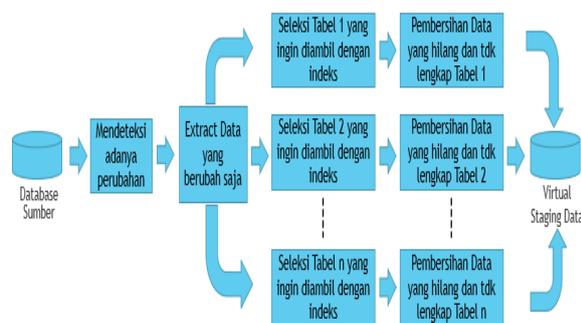
Untuk melakukan aturan transformasi - transformasi tersebut, optimasi ETL perlu mengetahui aturan transformasi mana yang sah dan yang menghasilkan suatu hasil yang sama. Disamping aturan transformasi - transformasi yang sah tersebut, proses optimasi juga perlu mengetahui kapan aturan - aturan transformasi tersebut digunakan untuk suatu query. Setelah menerapkan aturan transformasi - transformasi tersebut, optimasi mengganti kembali operasi - operasi

relasi pada query tree dengan operasi – operasi fisik yang dipakai untuk membuat suatu rencana eksekusi.

### 3.2 Teknik Pengolahan Data

Pembahasan iselanjutnya dalam penelitian ini adalah mengenai alur proses pengolahan idata saat proses extratc, trasform dan loading isebelum dan setelah dilakukan optimasi.

#### A. Optimasi Proses Extract



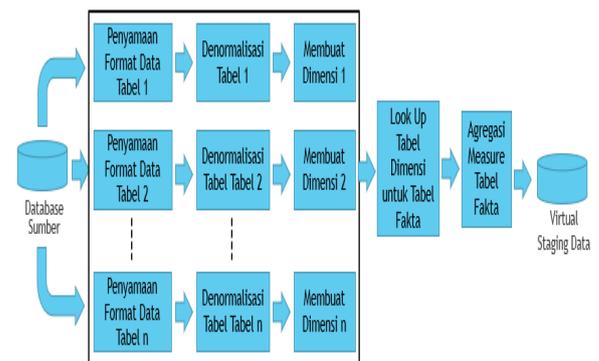
Gambar 3. Optimasi Pengolahan Data Pada Proses Extract

Optimasi dalam suatu proses extract yang menerapkan beberapa teknik guna menangani permasalahan pada proses ETL yang harus tujuannya mengekstraksi seluruh data yang menyebabkan banyak data yang harus diproses. Teknik yang digunakan pada proses ini adalah mendeteksi data yang berubah sehingga tidak mengekstraksi seluruh data sehingga jumlah data yang seharusnya diprosesi menjadi semakin efisien. Untuk dapat mempercepat waktu saat proses ETL, digunakan suatu indeks pada database sumber sehingga proses seleksi data menjadi semakin cepat dan proses ekstraksi data dilakukan secara paralel.

#### B. Optimasi Proses Transform

Optimasi pada proses transform dilakukan penerapan suatu teknik untuk mengatasi permasalahan yang terjadi pada prosesi ETL pada umumnya dengan cara membuat tabel dimensi secara paralel dan menggunakan data stream dalam pemrosesan data. Hal ini

dilakukan dengan pertimbangan bahwa struktur data warehouse yang digunakan menggunakan skema bintang, maka akan ditambahkan suatu proses denormalisasi untuk membuat tabel yang ada. Selain itu diterapkan pula pengoptimalan penulisan aljabar relasional dalam proses denormalisasi dan dalam pembuatan tabel dimensi agar waktu proses menjadi semakin cepat.



Gambar 4. Optimasi Pengolahan Data Pada Proses Transform

#### C. Optimasi Proses Loading



Gambar 5. Optimasi Pengolahan Data Pada Proses Load

Beberapa hal yang menjadi kekuarangan pada prosesi loading pada umumnya adalah memasukkan indeks kedalam data warehouse karena penggunaan Data Manipulation Laguange (DML), dalam melakukan optimasi dilakukan penerapan suatu teknik untuk mengatasi kekuarangan tersebut. Teknik yang dilakukan untuk menutupi kekuarangan tersebut adalah menghapus indeks yang kemudian

dibangun kembali setelah data masuk ke dalam data warehouse dengan menggunakan BULK.

#### 4. PEMBAHASAN

Berdasarkan pada teori literatur sumber yang ada, terdapat dua

perlakuan yang berbeda terhadap ETL pada umumnya. Tabel berikut menampilkan penggunaan waktu dan besar memory yang digunakan saat melakukan proses ETL sebelum dan setelah optimasi proses ETL

NO	Sumber Data	Jumlah data	Extract					
			Waktu (s)			Memory (Kb)		
			NOP	OP	Slh	NOP	OP	Slh
1	Data Sirup	10868	2	2	0	512	512	0
2	Data LPSE	629	2	1,6	0,4	512	512	0
3	Data Simpel	5527	2	2	0	512	512	0
<b>Total</b>		<b>17024</b>	<b>6</b>	<b>5,6</b>	<b>0,4</b>	<b>1536</b>	<b>1536</b>	<b>0</b>

**Tabel 1.** Proses Extraction

NO	Sumber Data	Jumlah data	Transform					
			Waktu (s)			Memory (Kb)		
			NOP	OP	Slh	NOP	OP	Slh
1	Data Sirup	10868	0	0	0	0	0	0
2	Data LPSE	629	0	0	0	0	0	0
3	Data Simpel	5527	0	0	0	0	0	0
<b>Total</b>		<b>17024</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Tabel 2.** Proses Transformation

NO	Sumber Data	Jumlah data	Load					
			Waktu (s)			Memory (Kb)		
			NOP	OP	Slh	NOP	OP	Slh
1	Data Sirup	10868	3274,4	703,8	2570,6	2082,2	668,2	1414
2	Data LPSE	629	893	151,67	741,33	1431	616,5	814,5
3	Data Simpel	5527	3340,4	568,6	2771,8	1932	633,4	1298,6
<b>Total</b>		<b>17024</b>	<b>7507,8</b>	<b>1424,07</b>	<b>6083,73</b>	<b>5445,2</b>	<b>1918</b>	<b>3527,1</b>

**Tabel 3.** Proses Load

Tabel diatas menunjukkan waktu eksekusi dan penggunaan memory untuk ketiga sumber data pada saat proses ETL. Dari tabel diatas menunjukkan proses ETL berjalan dengan baik dengan menunjukkan waktu dan penggunaan memory yang lebih kecil atau sama. Pada tabel transformasi tidak terdapat pergerakan sama sekali dikarenakan format dan struktur data dari ketiga

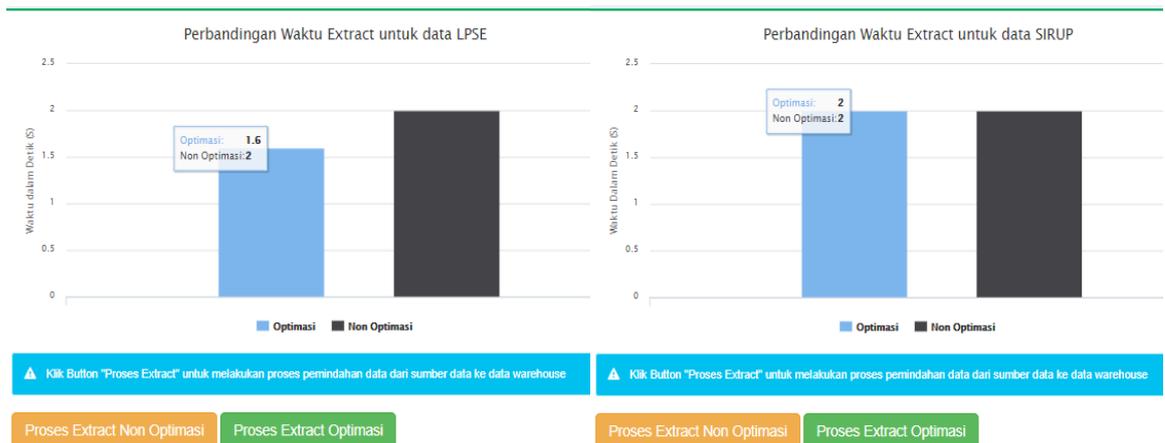
sumber data yang diolah tidak terdapat perbedaan. Hal ini dikarenakan sumber data tersebut sama-sama terkait dengan pengadaan barang/jasa pemerintah. Berikutnya adalah tampilan grafik dari tabel diatas.

#### 4.1 Perbandingan Penggunaan Waktu

Pada grafik dibawah, terdapat selisih waktu saat proses pada extract untuk data LPSE sebelum dan setelah

optimasi. Namun, pada data SiRUP waktu sebelum dan setelah optimasi

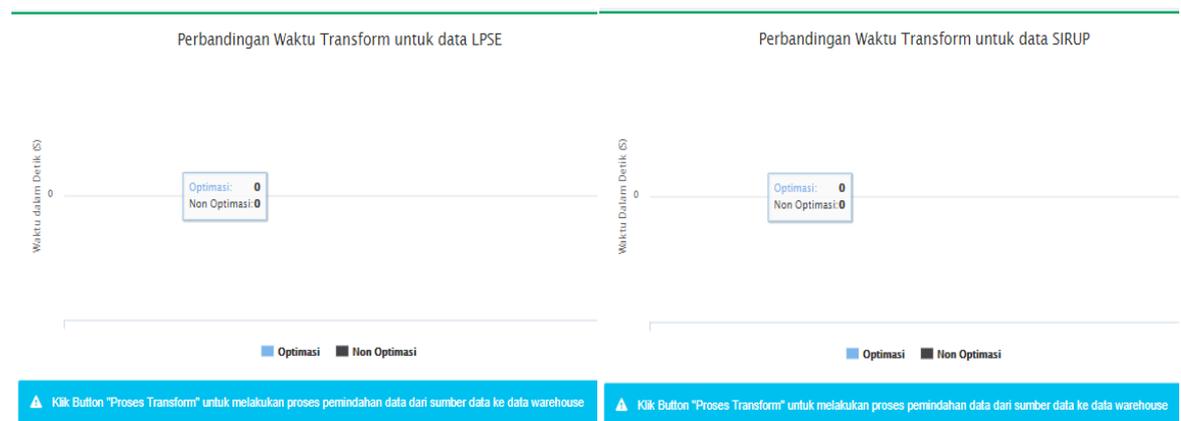
membutuhkan waktu yang sama, demikian juga saat proses data SiMPEL.



**Gambar 3.** Grafik Analisis Waktu Extract

Seperti yang telah dijelaskan sebelumnya, untuk proses transform tidak terjadi pengolahan data karena ketiga sumber data yang diolah memiliki struktur dan format data yang sama. Hal

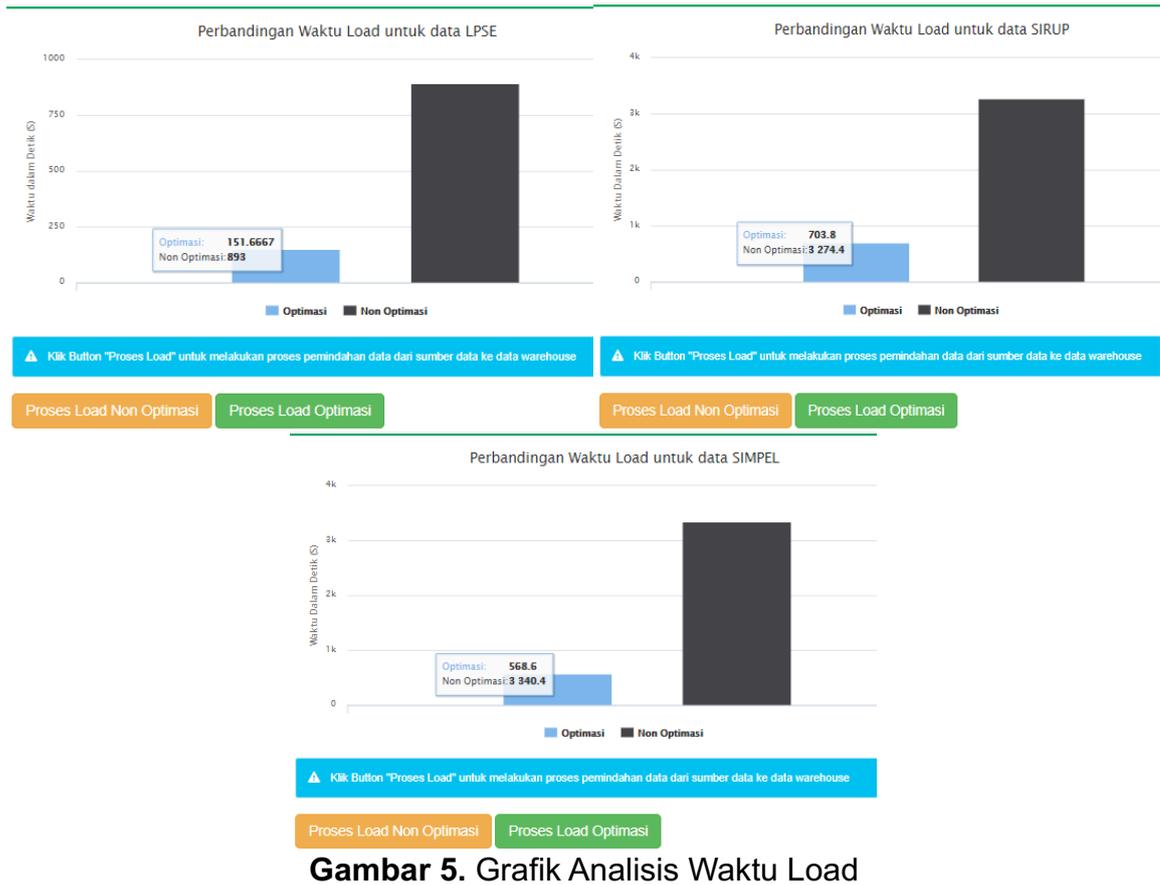
tersebut yang menyebabkan tidak membutuhkan waktu dan memory. Untuk grafik analisis waktu transform dapat dilihat pada Gambar 4.



**Gambar 4.** Grafik Analisis Waktu Transform

Pada Gambar 5. grafik analisis waktu load, menunjukkan perbandingan waktu saat proses load dari ketiga sumber data yang ada, terdapat selisih perbedaan waktu yang cukup signifikan antara waktu sebelum dilakukan

optimasi dengan waktu setelah dilakukan optimasi. Hal ini menunjukkan bahwa proses optimasi berjalan dengan baik karena mendapatkan efisiensi yang cukup signifikan.

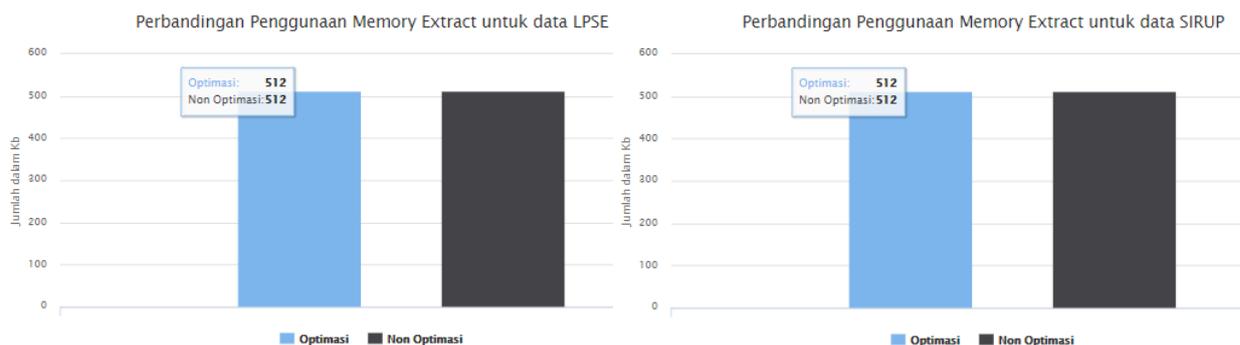


**Gambar 5.** Grafik Analisis Waktu Load

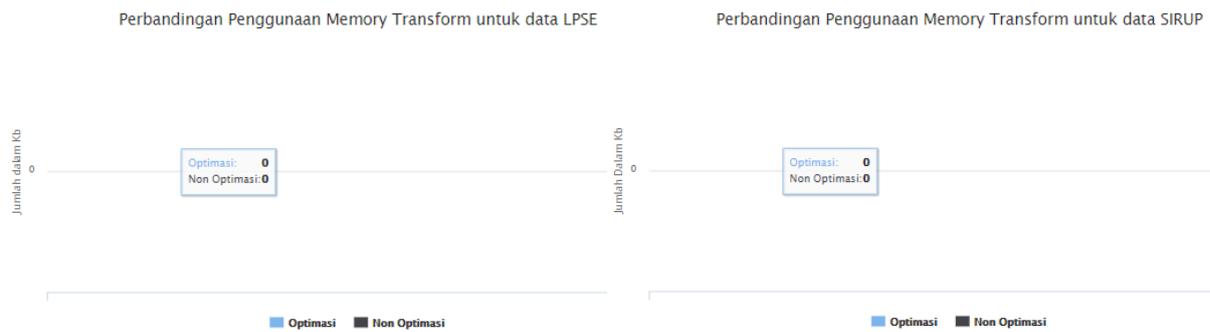
#### 4.2 Perbandingan Penggunaan Memory

Penggunaan memory pada proses extract untuk ketiga sumber data membutuhkan besaran yang sama antara sebelum dilakukan optimasi menggunakan metode heuristik dengan setelah dilakukan optimasi menggunakan metode yang sama. Grafik perbandingan penggunaan memory saat proses extract dapat dilihat

pada Gambar 6. Begitu halnya dengan penggunaan memory saat proses transform, namun saat transform tidak membutuhkan memory karena tidak ada pengolahan data dikarenakan struktur dan format data yang diolah memiliki kesamaan. Grafik perbandingan penggunaan memory saat proses transform dapat dilihat pada Gambar 7. dibawah ini.



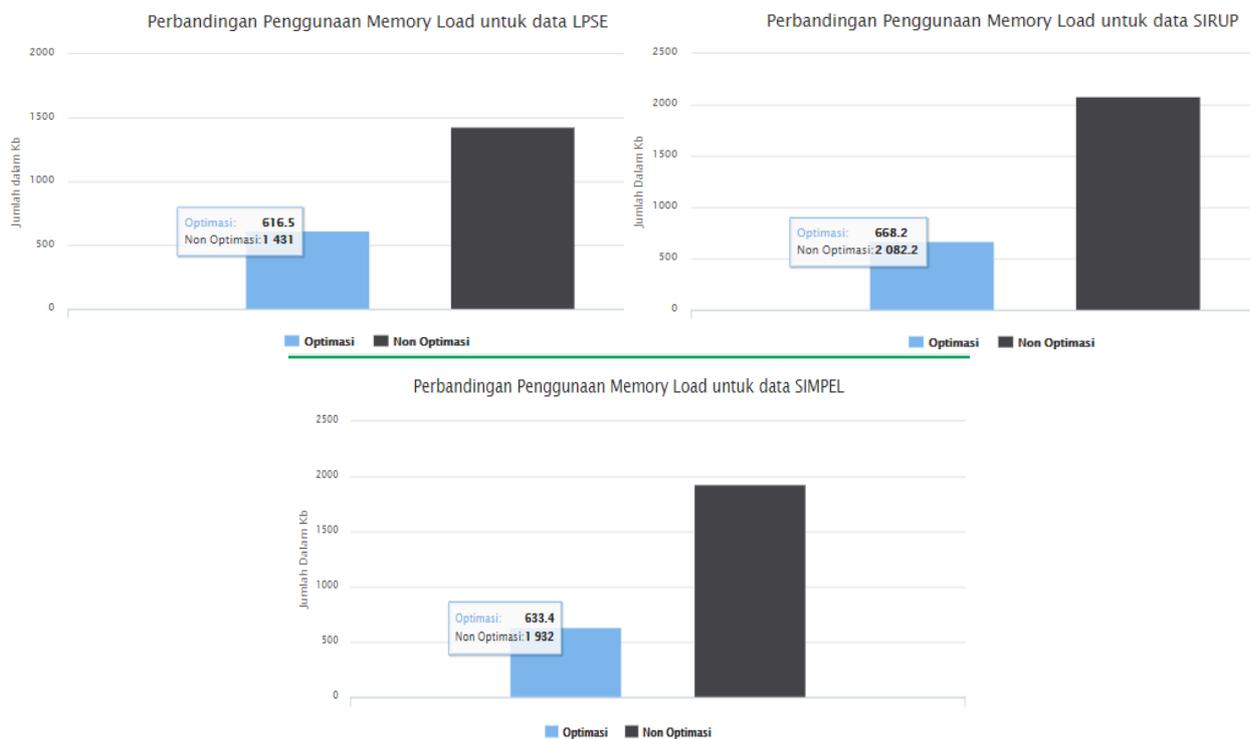
**Gambar 6.** Grafik Analisis Memory Extract



**Gambar 7.** Grafik Analisis Memory Transform

Jika penggunaan memory saat extract dan transform nilainya sama, tidak begitu halnya saat proses load. Saat proses load, penggunaan memory antara sebelum dan setelah optimasi nilainya cukup signifikan. Rata-rata

perbandingan penggunaan memory untuk tiga sumber yang digunakan dalam pembangunan data warehouse adalah 35,98%. Grafik perbandingan penggunaan memory saat proses load dapat dilihat pada Gambar 8.



**Gambar 8.** Grafik Analisis Memory Load

**5. KESIMPULAN**

Kesimpulan yang dapat penulis peroleh dari hasil pengujian ini adalah dengan menerapkan metode optimasi heuristik pada proses ETL dalam pembangunan data warehouse, didapatkan efisiensi terhadap waktu dan

memory dibandingkan dengan tanpa optimasi. Total efisiensi yang didapatkan dalam optimasi ETL dari ketiga sumber data yang digunakan mencapai 103 menit dari 17.024 jumlah data yang diolah.

## 6. DAFTAR PUSTAKA

- [1] Vishal Gour; Dr. S. S. Sarangdevot, Govind Singh Tanwar, Anand Sharma, (2010). Improve Performance of Extract, Transform and Load (ETL) in Data Warehouse. *International Journal on Computer Science and Engineering Vol. 02, No. 03*.
- [2] Kommineni Sivaganesh, P Srinivasu, Dr Suresh Chandra Satapa, (2012). Optimization of ETL Work Flow in Data Warehouse. *International Journal on Computer Science and Engineering (IJCSE)*.
- [3] Rudy Adipranata, Felicia Soedjianto, Wahyudi Tjondro. *Perbandingan Algoritma Exhaustive, Algoritma Genetika Dan Algoritma Jaringan Syaraf Tiruan Hopfield Untuk Pencarian Rute Terpendek*. Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra.
- [4] Slamet Sudaryant. (2007). Optimasi Query Database Menggunakan Teknik Heuristic. *Techno.Com, Vol. 7 No. 2*.
- [5] Alkis Simitsis, Panos Vassiliadis dan Timos Sellis. *Optimizing ETL Processes in Data Warehouses*.