

Analisis Metode Load Data Ke Dalam Database Oracle 21c: Studi Pada Data Central Mapper Kemenkeu

M. Iqbal Anshori*¹, Aditio Pra Utama*², Samidi³

^{1,2,3}Computer Information Systems, Universitas Budi Luhur, Indonesia

Email: ¹2311601807@student.budiluhur.ac.id, ²2311601864@student.budiluhur.ac.id,
³samidi@budiluhur.ac.id

Abstrak

Dalam era informasi saat ini, pengelolaan dan analisis data menjadi aspek krusial yang mendukung pengambilan keputusan yang efektif. Kementerian Keuangan (Kemenkeu) memiliki kebutuhan untuk mengelola data yang akurat dan terstruktur guna mendukung berbagai fungsi keuangan dan administrasi. Aplikasi Central Mapper digunakan untuk sinkronisasi data penyaluran yang terhubung dengan registri penerima manfaat, memainkan peran penting dalam mengelola integrasi data bantuan sosial, memfasilitasi konfirmasi, verifikasi, dan validasi data dari kementerian/lembaga (K/L), mencocokkan data yang diterima dari pemilik program K/L dengan data dari lembaga penyalur, serta menjadi sumber data untuk dasbor. Penelitian ini berfokus pada perbandingan tiga metode pemuatan data umum ke dalam database Oracle: SQL*Loader, External Table, dan Pentaho Data Integration. Penelitian ini bertujuan untuk membandingkan metode-metode tersebut dari segi kecepatan, penggunaan resource, dan akurasi data, dengan menggunakan data Central Mapper Kemenkeu sebagai studi kasus. Tujuan akhirnya adalah memberikan rekomendasi metode terbaik untuk memuat data ke dalam database Oracle, sehingga dapat meningkatkan efisiensi proses bisnis dan pengambilan keputusan. Penelitian melibatkan persiapan lingkungan pengujian, persiapan data uji, implementasi masing-masing metode pemuatan, dan pengukuran waktu eksekusi, penggunaan sumber daya CPU dan memori, serta akurasi data. Hasil penelitian menunjukkan bahwa SQL*Loader dengan opsi "direct" dan "unrecoverable" menawarkan performa terbaik untuk volume data besar. Metode External Table dan Pentaho Data Integration juga menunjukkan performa baik untuk ukuran data yang lebih kecil. Penelitian ini mengidentifikasi kelebihan dan kekurangan masing-masing metode, memberikan wawasan berharga bagi Kemenkeu dan institusi serupa dalam memilih metode pemuatan data yang paling sesuai dengan kebutuhan operasional mereka.

Kata kunci: *basis data oracle, external table, load data, optimasi database, pentaho data integration, performansi basis data*

Analysis of Data Loading Methods for the Oracle 21c Database: A Study on the Central Mapper Data of Kemenkeu

Abstract

In today's information era, data management and analysis are crucial aspects supporting effective decision-making. The Ministry of Finance (Kemenkeu) requires accurate and structured data management to support various financial and administrative functions. The Central Mapper application, used for synchronizing disbursement data connected with beneficiary registries, plays a vital role in managing the integration of social assistance data, facilitating data confirmation, verification, and validation from ministries/agencies (K/L), matching data received from K/L program owners with data from distribution institutions, and serving as a data source for dashboards. This study focuses on comparing three common data loading methods into Oracle databases: SQL*Loader, External Table, and Pentaho Data Integration. The research aims to compare these methods in terms of speed, resource usage, and data accuracy, using Kemenkeu's Central Mapper data as a case study. The goal is to provide recommendations on the best method for loading data into Oracle databases, thus enhancing business processes and decision-making efficiency. The study involves setting up a testing environment, preparing test data, implementing each loading method, and measuring execution time, CPU and memory resource usage, and data accuracy. Results indicate that SQL*Loader with the "direct" and "unrecoverable" options offers the best performance for large data volumes. External Table and Pentaho Data Integration methods also show good performance for smaller data sizes. This research highlights the strengths and weaknesses of each method, providing valuable insights for Kemenkeu and similar institutions in selecting the most appropriate data loading method for their operational needs.

Keywords: *database optimization, database performance, external table, load data, oracle database, pentaho data integration*

1. PENDAHULUAN

Dalam era informasi saat ini, pengelolaan dan analisis data menjadi salah satu aspek krusial dalam mendukung pengambilan keputusan yang efektif. Kementerian Keuangan (Kemenkeu) memiliki kebutuhan besar dalam mengelola data yang terstruktur dan akurat untuk mendukung berbagai fungsi keuangan dan administrasi. Seiring dengan meningkatnya volume data yang dikelola, tantangan utama yang dihadapi adalah efisiensi dalam proses pemuatan data ke dalam database, terutama dalam lingkungan dengan data dalam jumlah besar dan kompleksitas tinggi.

Aplikasi Central Mapper (Pemeta Data Pusat) merupakan Aplikasi yang digunakan oleh Kemenkeu untuk sinkronisasi data penyaluran bantuan sosial yang terkoneksi dengan data penerima manfaat (*beneficiary registry*) dan pemberian penandaan bagi penerima manfaat bantuan sosial/bantuan pemerintah. Aplikasi ini berperan untuk mengelola data integrasi bansos; memfasilitasi konfirmasi, verifikasi, dan validasi data penerima manfaat dari K/L, *matching data* yang diterima dari K/L pemilik program bansos dengan data dari Lembaga Penyalur, serta sumber data untuk *dashboard*. Setiap periode pencairan, sistem menerima data dalam jumlah besar yang perlu dimuat ke dalam Oracle 21c Database secara efisien dan akurat.

Data aplikasi central mapper bersumber dari Kementerian Negara / Lembaga pemilik program bantuan yang mengirimkan data penerima bantuan/manfaat setiap periode pencairan bantuan/bulanan. Salah satu tantangan yang dihadapi dalam aplikasi central mapper adalah proses load data yang efisien ke dalam sistem database, khususnya Oracle Database yang digunakan sebagai basis data utama.

Proses load data atau memuat data ke dalam database adalah langkah awal yang penting dalam manajemen data. Proses ini dapat mempengaruhi kecepatan akses, akurasi informasi, serta penggunaan sumber daya sistem seperti CPU dan memori [17]. Proses Load data ke target akhir, ke/dari basis data, atau *datawarehouse*, merupakan proses yang memakan waktu. Proses ini dapat dilakukan dari file sumber yang terstruktur maupun dari basis data lainnya[1].

1.1. Research Gap

Meskipun terdapat berbagai metode pemuatan data ke dalam database, penelitian sebelumnya masih terbatas dalam membandingkan efektivitas metode-metode tersebut dalam konteks Oracle 21c Database. Beberapa penelitian telah membahas metode pemuatan data ke dalam data warehouse atau big data systems, tetapi belum ada studi yang secara spesifik meneliti implementasi dan perbandingan metode data loading dalam sistem operasional dengan volume data besar, seperti yang dihadapi oleh Central Mapper Kemenkeu [18], [19].

Dalam konteks ini, terdapat tiga metode utama yang umum digunakan untuk pemuatan data ke dalam Oracle Database, yaitu:

1. SQL*Loader – Metode berbasis batch yang digunakan untuk memuat data dari file eksternal ke dalam tabel Oracle secara efisien.
2. External Table – Memungkinkan akses langsung ke file eksternal sebagai tabel virtual dalam Oracle, sehingga mengurangi waktu pemuatan.
3. Pentaho Data Integration (PDI) – Alat ETL (Extract, Transform, Load) pihak ketiga yang menawarkan fitur otomatisasi dalam pemrosesan data.

Sebagian besar penelitian yang telah dilakukan lebih berfokus pada optimasi ETL secara umum, tetapi belum banyak penelitian yang membahas perbandingan langsung antara SQL*Loader, External Table, dan Pentaho Data Integration dalam lingkungan Oracle 21c [20], [21]. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan melakukan analisis komparatif terhadap ketiga metode pemuatan data ke dalam Oracle 21c Database, khususnya dalam konteks aplikasi Central Mapper Kemenkeu.

1.2. Rumusan Masalah

Fokus bahasan pada penelitian ini meliputi:

1. Metode manakah yang paling optimal dalam hal kecepatan pemuatan data pada Oracle 21c?
2. Bagaimana perbandingan penggunaan CPU dan memori dari masing-masing metode?
3. Apakah ada kompromi antara kecepatan dan akurasi data dalam memilih metode pemuatan data?

1.3. Kontribusi Penelitian

Kontribusi utama pada penelitian ini adalah:

1. Analisis komparatif tiga metode *data loading* yang digunakan dalam lingkungan *Oracle 21c Database* dengan skala data besar.
2. Evaluasi empiris terhadap kecepatan, konsumsi sumber daya (CPU dan memori), serta akurasi data yang dimuat.
3. Rekomendasi berbasis data untuk pemilihan metode pemuatan data yang paling sesuai dalam konteks aplikasi operasional pemerintah.

Studi kasus dilakukan pada Data Central Mapper Kemenkeu untuk mendapatkan gambaran nyata mengenai performa masing-masing metode dalam situasi yang relevan dengan kebutuhan operasional Kemenkeu. Penelitian ini akan menganalisis beberapa aspek penting seperti kecepatan, penggunaan resource, dan akurasi data yang dimuat.

Dengan memahami kelebihan dan kekurangan masing-masing metode, diharapkan penelitian ini dapat memberikan rekomendasi yang tepat bagi Kemenkeu dan instansi serupa maupun pihak lain yang memiliki kepentingan yang sama dalam memilih metode terbaik untuk memuat data ke dalam database Oracle, sehingga dapat mendukung proses bisnis dan pengambilan keputusan dengan lebih cepat dan lebih baik.

2. METODE PENELITIAN

Metodologi ini menjelaskan langkah-langkah yang diambil untuk membandingkan tiga metode loading data ke dalam database Oracle dengan menggunakan External Table, SQL*Loader, dan program pihak ketiga (*third-party*) Pentaho Data Integration. Studi kasus dilakukan pada data Central Mapper Kementerian Keuangan (Kemenkeu).

2.1. Studi Literatur

SQL*Loader merupakan utilitas yang disediakan oleh Oracle untuk memuat data dari file eksternal ke dalam tabel database Oracle. SQL*Loader merupakan alat fleksibel yang dapat menangani beberapa input file dan memuat ke dalam beberapa tabel dengan berbagai tipe data[2]. SQL*Loader terkenal dengan kecepatan dan efisiensinya dalam menangani volume data yang besar. SQL*Loader mempunyai 2 metode load data yaitu *conventional path* dan *direct path load*. *Conventional path load* menggunakan insert statement dan data buffer untuk load data ke database sedangkan *direct path load* menggunakan API *direct path* untuk meneruskan data kedalam server load[3]. Untuk mempercepat metode load *direct path* dapat dilakukan dengan meminimalisir penggunaan log dan dapat dilakukan salah satunya dengan menambahkan parameter “*unrecoverable*” pada perintah loadnya[3].

External Table adalah fitur Oracle yang memungkinkan akses data eksternal seolah-olah data tersebut berada dalam database Oracle[2]. External table merupakan fitur pelengkap SQL*Loader untuk menjadikan data external menjadi seperti data internal database[3]. Metode ini menawarkan fleksibilitas dalam membaca data dari file tanpa perlu memuatnya secara langsung ke dalam database. External table harus ditempatkan pada server yang sama dengan database sehingga database engine dapat akses langsung ke file sumber external table[3].

Selain itu, terdapat pula program third party yang dirancang khusus untuk memuat data ke dalam database dengan berbagai fitur tambahan dan optimasi yang tidak selalu tersedia dalam alat bawaan Oracle. Dalam penelitian kali ini program *third party* yang digunakan adalah Pentaho Data Integration.

Pentaho merupakan salah satu perusahaan yang fokus pada produk dan solusi terkait *business intelligence*. Salah satu aplikasi dari Pentaho yang digunakan untuk mengelola data dan melakukan proses *extract, transform, dan load* (ETL) data adalah Pentaho Data Integration. Aplikasi ini dapat digunakan untuk migrasi data, membersihkan data, memuat dari file ke basis data atau sebaliknya dalam volume yang besar[4][5].

2.2. Penelitian Terdahulu

Penelitian sebelumnya pernah dilakukan oleh Martina Durneková dan Michal Kvet [2] yang membahas tentang “Data Import and Export Methods”. Pada penelitian tersebut dilakukan metode ekspor dan import data ke dan dari database oracle menggunakan fitur oracle datapump dan SQL*Loader. Penelitian tersebut membandingkan durasi yang dibutuhkan untuk melakukan ekspor dan import data dengan kesimpulan terkait import data yaitu oracle datapump merupakan proses yang lebih cepat bila dibandingkan dengan fitur SQL*Loader standar.

Penelitian selanjutnya dilakukan oleh Pedro Martins, Filipe Sá, Cristina Wanzeller dan Maryam Abbasi [1]. Penelitian tersebut berjudul “A performance study on different data load methods in relational databases” yang membandingkan metode load data ke dalam database yaitu *Java JDBC batch, datapump, SQL*Loader, SQL*Loader Direct, , Insert Into Select* dan menghasilkan kesimpulan penggunaan SQL*Loader dengan

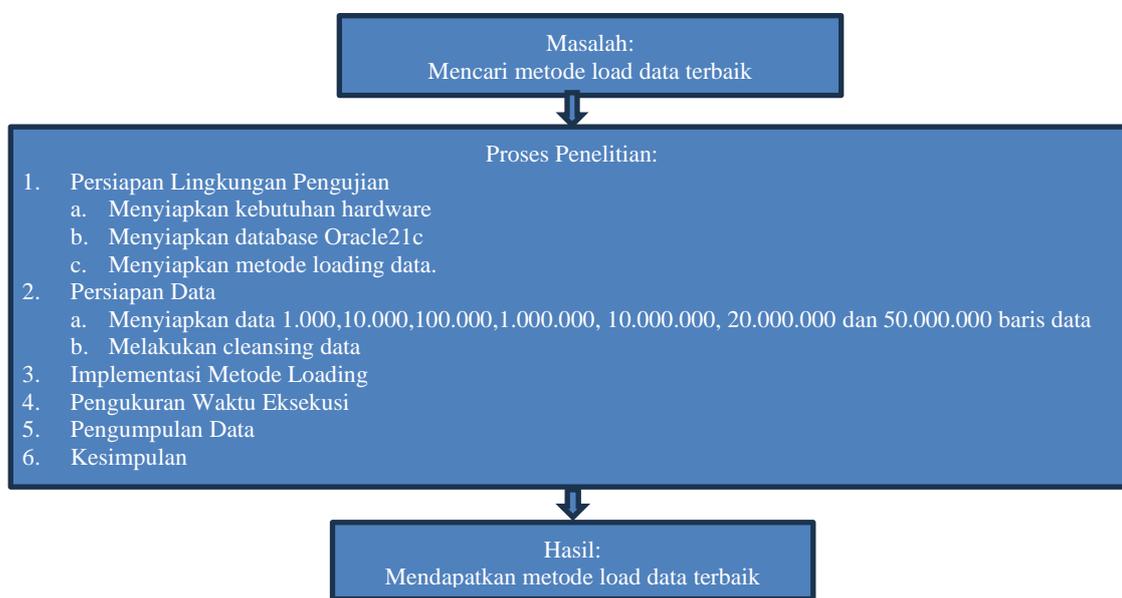
parameter “direct” dan “unrecoverable” menjadi metode yang paling cepat dan optimal bila dibandingkan dengan metode datapump dan JDBC import pada database oracle.

Prabin R. Sahoo and Chetan Phalak [6] dalam penelitiannya membahas “*High Speed Data Loading for Large Sized RDBMS Tables*”. Penelitian tersebut membandingkan load data dengan single loader dan parallel loader dengan kombinasi partisi table yang menghasilkan kesimpulan penggunaan fitur SQL*Loader dan dilakukan dengan parallel dapat mencapai target untuk load data dalam jumlah besar ke dalam database Oracle.

Pada penelitian sebelumnya telah dicoba beberapa metode load data ke database dengan menggunakan beberapa metode load data diantaranya datapump, SQL*Loader, java jdbc batch dan insert into select yang memberikan hasil yang berbeda-beda. Pada penelitian kali ini peneliti akan membandingkan metode load data sqllldr, external table dan pentaho data integration dengan menambahkan opsi parallel dan membandingkan durasi penyelesaiannya atas semua metode load data. Pada penelitian ini peneliti juga menambahkan Pentaho Data Integration yang merupakan salah satu alat pengolahan dan ETL data yang mumpuni untuk dibandingkan dengan fitur database oracle[7].

2.3. Kerangka Konsep Penelitian

Pada penelitian ini peneliti menggunakan Kerangka Konsep Penelitian sebagai berikut :



Gambar 1. Kerangka Konsep Penelitian

2.4. Metode Penelitian

Penelitian ini menggunakan desain eksperimental kuantitatif untuk membandingkan performa ketiga metode dalam hal kecepatan, penggunaan resource, dan akurasi.

Populasi dan sampel pengujian ini merupakan data yang dimiliki oleh Central Mapper Kemenkeu dengan sampel data tertentu yang dipilih secara acak dari Central Mapper dengan variasi ukuran baris data yang mewakili besaran data yang biasa diproses pada aplikasi central mapper untuk menguji performa dari ketiga metode load data.

Data penerima bantuan/manfaat dari Kementerian Negara / Lembaga pemilik program bantuan secara umum bervariasi namun paling tidak memuat data NIK, nama, lokasi, jenis bantuan, tanggal transaksi dan nominal transaksi. Pada penelitian ini peneliti membatasi data yang diproses sesuai tabel 1 yang meliputi data mandatori dan data opsional.

Tabel 1. Struktur tabel uji coba pada database oracle

No	Nama Kolom	Tipe Data	Keterangan
1.	NIK	Varchar2	Mandatory
2.	NAMA	Varchar2	Mandatory
3.	KODE_LOKASI	Varchar2	Mandatory
4.	PROV_NAME	Varchar2	Opsional
5.	KAB_NAME	Varchar2	Opsional
6.	KEC_NAME	Varchar2	Opsional
7.	KEL_NAME	Varchar2	Opsional
8.	TANGGAL_TRANSAKSI	Date	Mandatory
9.	JENIS_BANTUAN	Varchar2	Mandatory
10.	NOMINAL_TRANSAKSI	Number	Mandatory
11.	FLAG_BAYAR	Varchar2	Opsional

Variabel Penelitian pada semua metode loading data (External Table, SQL*Loader, Pentaho Data Integration) diamati dengan beberapa variabel yaitu Waktu loading, penggunaan sumber daya (CPU, memori), dan akurasi data setelah loading.

Instrumen yang diperlukan pada penelitian ini meliputi :

- a. Perangkat Keras: Laptop dengan spesifikasi Operating System Windows 11 Professional, processor 8 core dan memory 16 GB.
- b. Perangkat Lunak: Oracle Database versi 21c Express Edition, program SQL*Loader, dan program *third-party* Pentaho Data Integration Community Edition untuk loading data serta Windows Powershell untuk memonitor penggunaan resource.
- c. Data: Dataset dari Central Mapper Kemenkeu dengan jumlah data 1.000, 10.000, 100.000, 1.000.000, 10.000.000, 20.000.000 dan 50.000.000 baris data.

Prosedur Penelitian yang akan dilakukan pada penelitian ini meliputi langkah-langkah sebagai berikut :

1. Persiapan lingkungan pengujian
 - a. Melakukan instalasi Oracle Database 21c Express Edition.
 - b. Melakukan konfigurasi user dan tablespace serta tabel untuk menampung hasil load data.
 - c. Menyiapkan script Powershell yang akan digunakan untuk memonitor penggunaan resource pada saat proses load data.
 - d. Melakukan instalasi Pentaho Data Integration Community Edition.
 - e. Mempersiapkan konfigurasi baik di sisi sistem operasi maupun database terkait SQL*Loader, external table dan pentaho data integration untuk proses loading data.
2. Persiapan Data:
 - a. Melakukan ekstraksi dataset dari Central Mapper Kemenkeu dengan jumlah data sesuai variasi baris data yaitu 1000, 10.000, 100.000, 1.000.000, 10.000.000, 20.000.000 dan 50.000.000 baris data
 - b. Melakukan pembersihan dan cleansing data.
3. Pelaksanaan Pengujian:

Pengujian dilakukan dengan 2 skenario sebagai berikut :

 - a. Pengujian proses load data tanpa menggunakan parallel.
 - b. Pengujian dengan menggunakan parallel.

Untuk masing-masing skenario dilakukan pengujian dengan tahapan sebagai berikut :

 - a. Pada SQL*Loader dilakukan Penulisan dan eksekusi skrip SQL*Loader terhadap semua sampel baris data. Pada metode SQL*Loader ini dilakukan 2 jenis pengujian terhadap 2 konfigurasi yang berbeda yaitu konfigurasi standar dan konfigurasi kustom/advance dengan menambahkan parameter "*direct*" dan "*unrecoverable*" yang dapat mempercepat proses load data.
 - b. Pada External Table dilakukan konfigurasi dan implementasi load data menggunakan External Table. Selanjutnya dilakukan pemuatan tabel dengan mengambil data dari external tabel untuk dimasukkan ke dalam tabel oracle untuk semua sampel baris data.
 - c. Dilakukan Konfigurasi dan penggunaan Pentaho Data Integration untuk proses load data sesuai sampel baris data.
4. Pengukuran dan Pengumpulan Data:
 - a. Waktu load data dilakukan dengan mengukur waktu yang dibutuhkan masing-masing metode untuk menyelesaikan proses loading pada masing-masing sampel data.
 - b. Melakukan pengumpulan penggunaan resource CPU dan Memori pada saat pelaksanaan load data.
 - c. Akurasi Data dilakukan dengan Memeriksa konsistensi dan akurasi data setelah proses load data selesai.

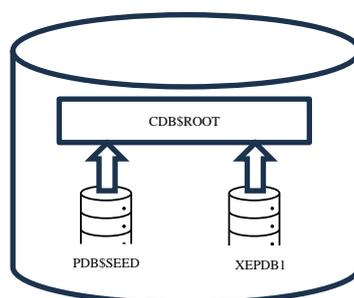
- d. Melakukan pengumpulan data hasil penelitian dan melakukan analisis statistik untuk membandingkan performa ketiga metode.

Dengan metodologi ini, penelitian akan memberikan gambaran yang jelas tentang keunggulan dan kelemahan masing-masing metode load data ke dalam database Oracle, serta memberikan rekomendasi metode terbaik berdasarkan kinerja yang diukur sehingga bisa dijadikan pembanding dengan mekanisme proses load data yang saat ini berjalan.

3. HASIL DAN PEMBAHASAN

3.1. Persiapan Lingkungan Pengujian

Persiapan lingkungan pengujian meliputi instalasi Database Oracle versi 21c Express Edition yang dalam bundle instalasinya sudah termasuk *tools* SQL*Loader dengan program sqlldr dan juga external table yang secara default juga menggunakan fitur sqlldr. Proses instalasi Database oracle menggunakan topologi Container Database (CDB) yang merupakan default topologi instalasi Oracle Database 21 Express Edition.



Gambar 2. Topologi Database Oracle 21c XE

Pada gambar 2 dijelaskan topologi database Oracle 21 Express Edition yang terdiri dari 2 Pluggable Database (PDB) yaitu PDB SEED yang merupakan template PDB dan juga XEPDB1 yang merupakan PDB default yang nantinya merupakan tempat dilakukan pengujian.

Setelah proses instalasi database berhasil dilakukan, selanjutnya dilakukan proses konfigurasi pada database dengan membuat tablespace untuk menampung data pengujian serta membuat user database baru yang nantinya akan digunakan untuk melakukan pengujian.

Script untuk pembuatan tablespace adalah sebagai berikut:

```
CREATE BIGFILE TABLESPACE MANDAT_BIGTBS DATAFILE 'C:\...\XE\XEPDB1\MANDAT.DBF' SIZE 10G;
```

Sedangkan script untuk membuat user yang digunakan untuk pengujian adalah sebagai berikut:

```
CREATE USER mandat IDENTIFIED BY
DEFAULT TABLESPACE MANDAT_BIGTBS
TEMPORARY TABLESPACE TEMP
PROFILE "DEFAULT"
QUOTA UNLIMITED ON MANDAT_BIGTBS;
```

```
GRANT "CONNECT" TO mandat;
GRANT "RESOURCE" TO mandat;
GRANT CREATE SESSION TO mandat;
GRANT CREATE TABLE TO mandat;
GRANT CREATE ANY DIRECTORY TO mandat;
```

Setelah dilakukan pembuatan tablespace dan user pada database oracle selanjutnya dibuat tabel-tabel baru untuk menampung data yang akan dimasukkan. Tabel-tabel tersebut dipisahkan untuk masing-masing baris data yang akan di-load dengan nama tabel dan peruntukannya sebagaimana dijelaskan pada tabel 2.

Tabel 2. Nama tabel uji coba pada database oracle

No	Nama Tabel	Peruntukan
1.	MANDAT_DATA_1K	Data 1.000 baris
2.	MANDAT_DATA_10K	Data 10.000 baris
3.	MANDAT_DATA_100K	Data 100.000 baris
4.	MANDAT_DATA_1M	Data 1.000.000 baris
5.	MANDAT_DATA_10M	Data 10.000.000 baris

6.	MANDAT_DATA_20M	Data 20.000.000 baris
7.	MANDAT_DATA_50M	Data 50.000.000 baris

Sampel script pembentukan tabel sebagaimana dimaksud pada tabel 2 tersebut untuk nama tabel MANDAT_DATA_1K adalah sebagai berikut:

```
CREATE TABLE MANDAT.MANDAT_DATA_1K
(
  NIK          VARCHAR2(20 BYTE),
  NAMA         VARCHAR2(180 BYTE),
  KODE_LOKASI  VARCHAR2(10 BYTE),
  PROP_NAME    VARCHAR2(150 BYTE),
  KAB_NAME     VARCHAR2(150 BYTE),
  KEC_NAME     VARCHAR2(150 BYTE),
  KEL_NAME     VARCHAR2(150 BYTE),
  TANGGAL_TRANSAKSI VARCHAR2(10 BYTE),
  JENIS_BANTUAN VARCHAR2(255 BYTE),
  NOMINAL_TRANSAKSI NUMBER,
  FLAG_BAYAR   VARCHAR2(10 BYTE)
)
TABLESPACE MANDAT_BIGTBS;
```

Selanjutnya dilakukan instalasi Pentaho Data Integration Community Edition versi 9.4 dengan menggunakan file installer. Pada penelitian kali ini menggunakan community edition dikarenakan bersifat open source dan fitur load data ke dalam database oracle sudah tersedia pada versi ini sehingga sudah bisa digunakan untuk melakukan pengujian [8].

Pada metode SQL*Loader disiapkan konfigurasi *control file* yang dibutuhkan sebagai konfigurasi untuk proses load data ke dalam database oracle menggunakan SQL*Loader. Pada *control file* SQL*Loader ini digunakan 2 versi yaitu versi default yang masih menggunakan *convensional path* dan versi kustom yang sudah menggunakan parameter “*direct*” dan “*unrecoverable*” untuk mempercepat proses load data. Sampel isi *control file* kustom adalah sebagai berikut:

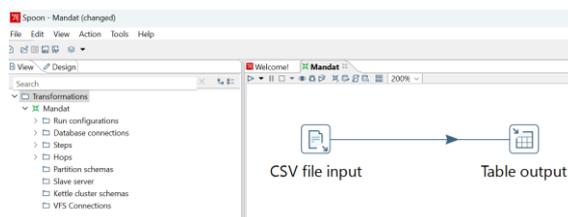
```
OPTIONS (skip=1, direct=true, silent=(feedback))
UNRECOVERABLE
LOAD DATA
INFILE 'C:\...\DATA\6_file_centralmapper_20m.csv'
APPEND
INTO TABLE MANDAT_DATA_20M
FIELDS TERMINATED BY '|'
(NIK,NAMA,KODE_LOKASI,PROP_NAME,KAB_NAME, KEC_NAME,KEL_NAME,TANGGAL_TRANSAKSI DATE "yyyy-mm-dd", JENIS_BANTUAN,NOMINAL_TRANSAKSI, FLAG_BAYAR)
```

Pada metode external tabel juga dipersiapkan script untuk membuat external tabel sehingga data sudah bisa dibaca dari internal database dan siap untuk dilakukan pengujian load/penarikan data ke dalam database. Durasi proses load data external table ini dihitung dari kecepatan query untuk load data dari external table ke dalam database Oracle [9]. Contoh script pembentukan external tabel untuk jumlah baris data 10 juta record adalah sebagai berikut:

```
CREATE TABLE mandat_external_10m (
  NIK          VARCHAR2 (20 Byte),
  NAMA         VARCHAR2 (180 Byte),
  KODE_LOKASI  VARCHAR2 (10 Byte),
  PROP_NAME    VARCHAR2 (150 Byte),
  KAB_NAME     VARCHAR2 (150 Byte),
  KEC_NAME     VARCHAR2 (150 Byte),
  KEL_NAME     VARCHAR2 (150 Byte),
  TANGGAL_TRANSAKSI VARCHAR2 (10 Byte),
  JENIS_BANTUAN VARCHAR2 (255 Byte),
  NOMINAL_TRANSAKSI NUMBER,
  FLAG_BAYAR   VARCHAR2 (10 Byte)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY MANDAT
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    SKIP 1
    FIELDS TERMINATED BY '|'
    MISSING FIELD VALUES ARE NULL
  )
)
```

```
LOCATION('5_file_centralmapper_10m.csv')
)
REJECT LIMIT UNLIMITED
;
```

Pada metode Pentaho Data Integration hal pertama yang dilakukan adalah konfigurasi koneksi ke database oracle kemudian menyiapkan transformation file yang digunakan untuk melakukan load data ke dalam database dimana proses transformation terdiri dari “CSV file input” yang digunakan untuk membaca text file yang sudah disiapkan untuk pengujian dan diteruskan ke “tabel output” yang langsung diarahkan ke dalam tabel yang ada di dalam database pengujian. Sampel transformation file dimaksud sebagaimana gambar 3 yang menggambarkan objek-objek yang digunakan pada Pentaho Data Integration.



Gambar 3. Persiapan tools Pentaho Data Integration

Penggunaan fitur pada Pentaho Data Integration dibuat paling sederhana dikarenakan data sudah disiapkan dan dibersihkan dengan baik sehingga dapat langsung dilakukan load data ke dalam database. Konfigurasi Pentaho menggunakan konfigurasi standar aplikasi yang sudah menggunakan batch proses dalam proses load datanya.

Persiapan juga dilakukan pada Windows Powershell yang akan digunakan untuk menangkap penggunaan resource CPU dan memory. Proses monitor penggunaan resource tersebut dilakukan dengan melihat penggunaan resource yang digunakan dengan interval setiap 2 detik yang berjalan selama proses load data berlangsung. Script yang digunakan untuk mrmonitor penggunaan resource melalui Windows Powershell adalah sebagai berikut:

```
while ($true) {
    # Mendapatkan penggunaan CPU
    $cpu = Get-Counter '\Processor(_Total)\% Processor Time'
    $cpuUsage = $cpu.CounterSamples.CookedValue

    # Mendapatkan penggunaan memori
    $mem = Get-Counter '\Memory\Available MBytes'
    $totalMem = (Get-CimInstance -ClassName Win32_ComputerSystem).TotalPhysicalMemory / 1MB
    $usedMem = $totalMem - $mem.CounterSamples.CookedValue
    $memUsage = ($usedMem / $totalMem) * 100

    # Menampilkan penggunaan CPU dan memori dengan waktu saat ini
    $currentTime = Get-Date -Format "HH:mm:ss"
    Write-Host ("[$currentTime] CPU Usage: {0:N2}%" -f $cpuUsage)
    Write-Host ("[$currentTime] Memory Usage: {0:N2}%" -f $memUsage)

    # Menunggu 1 detik sebelum loop berikutnya
    Start-Sleep -Seconds 0
}
```

Lingkungan pengujian yang telah disiapkan telah merepresentasikan lingkungan sebenarnya. Proses load data saat ini menggunakan SQL*Loader standar untuk data yang besar dan aplikasi third party untuk data yang relatif kecil. Dengan pengujian ini diharapkan penelitian ini dapat memberikan masukan untuk lebih mengoptimalkan proses load data yang saat ini berjalan baik dengan penambahan fitur/parameter tertentu maupun dengan penggantian metode load data yang digunakan.

Proses load data dengan menggunakan SQL*Loader sangat cocok digunakan untuk proses yang sumber datanya standar dan frekuensi proses load datanya besar sehingga dapat dilakukan otomasi proses load datanya. Sedangkan untuk data yang relatif kecil dan format yang kadang berubah (misalnya penyampaian data dengan menggunakan Ms. Excel) proses load data lebih cocok dilakukan dengan program *third party* seperti Pentaho Data Integration, Talend, Informatica, Toad for Oracle[10].

3.2. Persiapan Data

Proses persiapan data dilakukan dengan membentuk flat file dalam ekstensi csv sejumlah 6 file yang berisi data penerima bantuan central mapper dengan variasi data disesuaikan dengan jumlah data yaitu sejumlah 1.000, 10.000, 100.000, 1.000.000, 10.000.000, 20.000.000 dan 50.000.000 baris data. Pemilihan jumlah data ini diharapkan dapat mewakili kecepatan dan kehandalan proses load data oleh metode yang akan dibandingkan dan mewakili besaran transaksi yang dilakukan pada aplikasi central mapper[11].

NIK	NAMA	KODE_LOKASI	PROP_NAME	KAB_NAME	KEC_NAME	KEL_NAME	TANGGAL_TRANSAKSI	JENIS_BANTUAN	NOMINAL_TRANSAKSI	FLAG_BAYAR
1107*****0001	RIZWAN MUHAMMAD JAMIL	1107182015	ACEH	KAB. PIDIE	Simpang Tiga	Nien	2023-11-20	FBI JK	40000	1
1107*****0003	ZAHARA RAMADHANI	1107182015	ACEH	KAB. PIDIE	Simpang Tiga	Nien	2023-11-20	FBI JK	40000	1
1107*****0003	HAJJAH	1107182001	ACEH	KAB. PIDIE	Simpang Tiga	Raya Paya	2023-11-20	FBI JK	40000	1
1107*****0001	MUHAMMAD RIZAL	1107182033	ACEH	KAB. PIDIE	Simpang Tiga	Seuk Ceukok	2023-11-20	FBI JK	40000	1
1107*****0001	FAUZIYAH	1107182036	ACEH	KAB. PIDIE	Simpang Tiga	Kampung Blang	2023-11-20	FBI JK	40000	1
1107*****0001	FARHAN	1107182036	ACEH	KAB. PIDIE	Simpang Tiga	Kampung Blang	2023-11-20	FBI JK	40000	1
1107*****0001	MIRNA	1107182036	ACEH	KAB. PIDIE	Simpang Tiga	Kampung Blang	2023-11-20	FBI JK	40000	1
1107*****0002	BLQIS	1107182029	ACEH	KAB. PIDIE	Simpang Tiga	Peukan Tuha	2023-11-20	FBI JK	40000	1
1107*****0001	ZULKARNAINI	1107182034	ACEH	KAB. PIDIE	Simpang Tiga	Pulo Tu	2023-11-20	FBI JK	40000	1
1107*****0001	MURNIATI ABDULLAH	1107182047	ACEH	KAB. PIDIE	Simpang Tiga	Mesjid Tungoe	2023-11-20	FBI JK	40000	1
1107*****0003	SYAMSIMAR	1107182050	ACEH	KAB. PIDIE	Simpang Tiga	Cot Jaja	2023-11-20	FBI JK	40000	1
1107*****0001	NURHAYATI	1107182029	ACEH	KAB. PIDIE	Simpang Tiga	Peukan Tuha	2023-11-20	FBI JK	40000	1

Gambar 4. Persiapan data uji

Setelah file data uji disiapkan sebagaimana contoh pada gambar 4, selanjutnya dilakukan pembersihan/cleansing data sehingga file data uji sudah tidak mengandung karakter tidak standar maupun delimiter yang tidak seragam. Hal ini dimaksudkan untuk mengeliminasi faktor-faktor lain di luar proses load data sehingga perbandingan metode load data ini dapat lebih akurat.

3.3. Pelaksanaan Pengujian dan Pembahasan

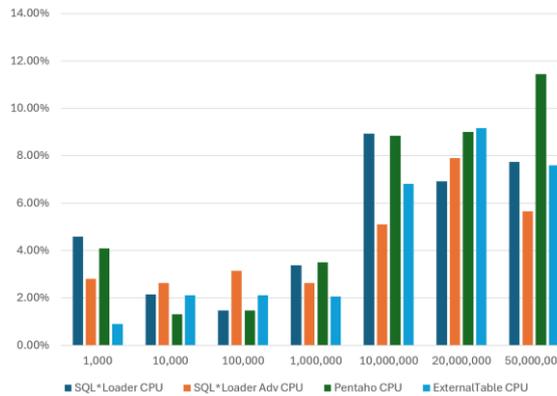
Pengujian skenario pertama dilakukan dengan melakukan proses load data menggunakan SQL*Loader, external table dan Pentaho data integration untuk semua file data uji. Proses load data dilakukan oleh masing-masing metode pengujian untuk semua data dengan menggunakan tabel baru untuk masing-masing data uji. Setelah selesai dilakukan pengujian dengan salah satu tool maka dilakukan proses delete tabel dan restart database sebelum pengujian dengan metode selanjutnya untuk menghilangkan log/cache yang mungkin terjadi.

Hasil pengujian skenario pertama dapat dilihat pada tabel 3 dimana pada proses load data 1 ribu, 10 ribu dan 100 ribu baris data semua metode menghasilkan durasi load data dibawah 1 detik dengan proses tercepat dicapai oleh Pentaho dan External tabel yang sedikit lebih unggul daripada SQL*Loader.

Tabel 3. Hasil pengujian skenario 1

Jumlah Baris Data	SQL*Loader	SQL*Loader Adv	Pentaho	ExternalTable
1 ribu	00:00.14	00:00.15	00:00.01	00:00.07
10 ribu	00:00.22	00:00.21	00:00.10	00:00.07
100 ribu	00:00.70	00:00.27	00:00.60	00:00.24
1 juta	00:05.58	00:00.97	00:04.90	00:02.07
10 juta	00:51.64	00:07.53	00:51.60	00:20.84
20 juta	01:41.44	00:14.93	01:54.00	00:44.81
50 juta	04:28.53	00:44.17	04:57.00	01:53.25

Selanjutnya untuk proses load data 1 juta sampai dengan 50 juta didominasi oleh SQL Loader kustom/advance dengan keunggulan yang sangat jauh dibandingkan dengan metode yang lain dan kemudian diikuti oleh External table dan selanjutnya ada SQL Loader standar dan Pentaho yang memiliki waktu load data yang kurang lebih sama.



Gambar 5. Penggunaan CPU pada skenario 1

Akurasi data terkait hasil pengujian skenario pertama menunjukkan bahwa semua metode load data mempunyai akurasi sempurna dengan tidak ditemukannya error load data pada semua metode load data untuk semua proses file uji.

Data penggunaan resource CPU pada skenario pertama dilakukan dengan mengambil penggunaan CPU tertinggi selama durasi pengujian masing-masing metode dengan hasil penggunaan CPU sebagaimana dijelaskan pada gambar 5. Secara umum Penggunaan CPU untuk semua metode load data relatif aman dan tidak terlalu besar dimana rata-rata metode load data menggunakan CPU dibawah 10%. Pada proses load data yang besar dengan jumlah baris data 10,20 dan 50 juta baris data, penggunaan CPU Pentaho relatif lebih besar dari metode yang lain dan SQL*Loader kustom/advance memiliki rata-rata penggunaan CPU yang paling kecil.

Penggunaan resource memori pada skenario pertama relatif konsisten untuk semua metode load data sebagaimana ditampilkan pada gambar 6 dengan rata-rata penggunaan sebesar 60% dengan pengecualian metode Pentaho yang penggunaan memorinya selalu lebih besar daripada metode yang lain. Penggunaan memori yang hampir seragam dan cukup besar ini didominasi oleh memori yang digunakan oleh database oracle.



Gambar 6. Penggunaan memori pada skenario 1

Selanjutnya dilakukan Pengujian skenario kedua dengan langkah-langkah yang sama persis dengan skenario pertama. Skenario kedua dijalankan dengan perbedaan parameter paralel pada setiap metode load data. Untuk metode SQL*Loader standar/default tidak mempunyai opsi paralel sehingga tidak ikut dalam pengujian skenario kedua. Paralel pada metode Pentaho dan SQL*Loader kustom/advance menggunakan opsi paralel bawaan tanpa menyebutkan derajat tingkat paralelnya dikarenakan opsi paralel pada kedua metode tersebut hanya sebatas true/false sedangkan pada metode external tabel paralel diset sebanyak 8 paralel sesuai dengan jumlah core processor yang dimiliki oleh perangkat pengujian.

Pada hasil pengujian skenario kedua dapat dilihat sebagaimana ditampilkan pada tabel 4 masih memiliki pola yang sama dengan hasil skenario pertama dimana SQL*Loader kustom/advance masih unggul jauh dibandingkan dengan metode yang lain diikuti dengan external tabel dan Pentaho menjadi metode yang paling lambat sesuai hasil pengujian ini.

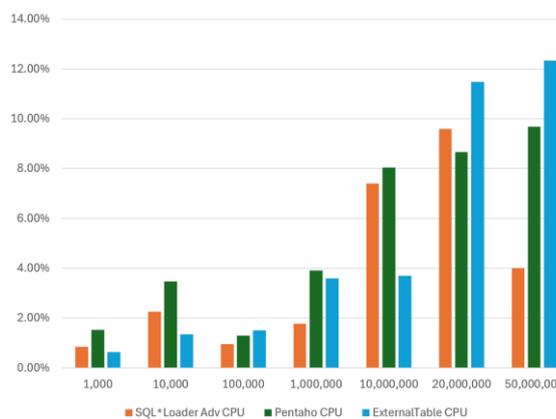
Tabel 4. Hasil pengujian skenario 2

Jumlah Baris Data	SQL*Loader Adv.	Pentaho	ExternalTable
1 ribu	00:00.19	00:00.01	00:00.06
10 ribu	00:00.19	00:00.10	00:00.07
100 ribu	00:00.27	00:00.50	00:00.25

1 juta	00:00.96	00:04.80	00:02.02
10 juta	00:07.45	00:49.20	00:21.05
20 juta	00:14.25	01:52.00	00:54.11
50 juta	00:36.99	04:50.00	02:13.39

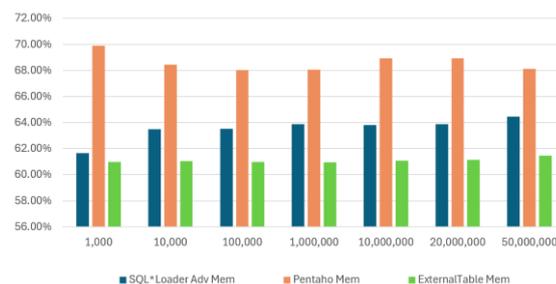
Pada jumlah baris data 1,10 dan 100 ribu durasi load data masih dibawah 1 detik sedangkan untuk baris data 1,10,20 dan 50 juta ternyata tidak mengalami peningkatan performa yang signifikan ditandai dengan pengurangan durasi waktu load data yang tidak terlalu mencolok untuk metode SQL*Loader kustom/advance dan Pentaho sedangkan untuk metode external tabel malah lebih lama durasi waktunya meskipun pada skenario kedua ini sudah ditambahkan opsi paralel pada metode load datanya.

Penggunaan resource CPU pada skenario kedua sedikit meningkat bila dibandingkan dengan skenario pertama. Akan tetapi secara umum penggunaannya masih aman dan tidak menyebabkan lonjakan penggunaan CPU meskipun diterapkan paralel pada metode load datanya. Metode external tabel yang paling kelihatan meningkat penggunaannya CPU nya pada skenario kedua ini sebagaimana digambarkan dalam gambar 7.



Gambar 7. Penggunaan CPU pada skenario 2

Penggunaan memori pada skenario kedua ini meningkat bila dibandingkan dengan skenario pertama. Pada skenario pertama penggunaan memori rata-rata berada pada 60% kecuali metode pentaho yang sedikit lebih tinggi. Penggunaan memori pada skenario kedua sebagaimana ditampilkan pada gambar 8 menunjukkan peningkatan terutama pada metode pentaho yang penggunaan memorinya berada pada kisaran 68-70% dan metode SQL*Loader kustom/advance pada kisaran 62-64% sedangkan metode external table sedikit meningkat di atas 60%.



Gambar 8. Penggunaan memori pada skenario 2

Pada pengujian kedua ini secara akurasi data juga menunjukkan hasil yang sempurna dengan tidak adanya error yang ditemukan pada semua skenario load data. Hal ini menunjukkan semua metode load data dapat memproses data uji yang disiapkan dengan sempurna tanpa adanya error load data.

Berdasarkan hasil pengujian pada skenario pertama dan kedua ditemukan hasil pengujian bahwa SQL*Loader menunjukkan performa terbaik dalam hal durasi dan penggunaan resource pada saat pengujian. Hal ini sejalan dengan penelitian sebelumnya oleh Pedro Martins, Filipe Sá, Cristina Wanzeller dan Maryam Abbasi [1]. Akan tetapi hasil pengujian ini berbeda dengan penelitian oleh Prabin R. Sahoo and Chetan Phalak [6] dimana pada penelitian ini opsi paralel pada SQL*Loader tidak memberikan dampak signifikan pada durasi load data serta berbeda juga dengan penelitian Trumstedt, K[12] karena pada penelitian ini SQL*Loader lebih cepat dan optimal dibandingkan dengan External Table.

4. DISKUSI

Pengujian pada penelitian ini dilakukan dengan membandingkan tiga metode load data: SQL*Loader, External Table, dan Pentaho Data Integration untuk melakukan load data ke dalam database [13]. Pengujian dilakukan dengan menghilangkan log/cache sebelum setiap metode dengan melakukan restart database Oracle pada setiap pengujian untuk memastikan hasil yang valid dan konsisten.

Pada hasil pengujian baik skenario pertama maupun skenario kedua dapat dikelompokkan menjadi 2 bagian yaitu untuk baris data kecil dengan jumlah data 1 ribu hingga 100 ribu baris data semua metode berhasil melakukan load data dalam waktu kurang dari 1 detik dan untuk Pentaho dan External Table sedikit lebih cepat daripada SQL*Loader. Akan tetapi hasil pengujian pada kelompok data kecil masih bisa dilanjutkan pengujian selanjutnya mengingat durasi semua proses load data hanya di bawah 1 detik. Sedangkan untuk kelompok baris data besar dengan jumlah data 1 juta hingga 50 juta SQL*Loader kustom/advance menunjukkan performa terbaik, terutama pada jumlah data yang lebih besar kemudian External Table berada di posisi kedua dan SQL*Loader standar dan Pentaho memiliki waktu load data yang kurang lebih sama, tetapi lebih lambat dibandingkan dua metode lainnya.

Secara umum, penggunaan resource untuk semua metode load data ini masih relatif aman. Penggunaan CPU untuk semua metode juga relatif rendah, rata-rata penggunaan masih di bawah 10% dengan Pentaho sedikit lebih tinggi untuk skenario tanpa opsi paralel dan external table lebih tinggi untuk skenario dengan opsi paralel. Penggunaan memori untuk semua metode hampir sama dengan pengecualian metode pentaho yang menggunakan memori lebih tinggi baik dengan opsi paralel maupun opsi non paralel.

Berdasarkan hasil pengujian yang telah dilakukan ditemukan metode yang paling efektif untuk melakukan load data yaitu dengan menggunakan Oracle SQL*Loader dengan tambahan opsi direct dan unrecoverable. Penambahan opsi direct memungkinkan proses load dilakukan langsung ke server load dan opsi unrecoverable menjadikan proses load tidak membuat log sehingga dapat mempercepat proses load data[14]. Penggunaan metode SQL*Loader dengan tambahan opsi direct dan unrecoverable tersebut untuk saat ini belum memungkinkan untuk diterapkan pada semua data dikarenakan masih terdapat data yang dikirim dalam format Ms. Excel dan belum standar sehingga diharapkan kepada pengelola aplikasi central mapper dapat berkomunikasi lebih lanjut dengan pihak-pihak terkait untuk dapat membuat standar baku data yang dibutuhkan dan penyedia data dapat menyediakan data dengan standar yang telah disepakati.

5. KESIMPULAN

Berdasarkan hasil penelitian yang sudah dilakukan pada studi aplikasi central mapper dapat disimpulkan bahwa SQL*Loader kustom/advance dengan penggunaan opsi parameter direct dan unrecoverable menunjukkan performa terbaik dalam hal durasi load data untuk jumlah data yang besar sedangkan Pentaho dan External Table menunjukkan hasil yang baik untuk jumlah data yang lebih kecil, tetapi performa menurun pada data yang lebih besar meskipun dengan opsi paralel.

Terkait dengan penggunaan resource selama proses load data semua metode menunjukkan penggunaan CPU yang relatif rendah, dengan SQL*Loader kustom/advance paling efisien. Penggunaan memori konsisten di sekitar 60%, dengan Pentaho menunjukkan penggunaan tertinggi. Selanjutnya penggunaan opsi paralel yang dilakukan pada skenario kedua, dapat disimpulkan bahwa opsi paralel tidak menunjukkan peningkatan performa signifikan untuk SQL*Loader kustom/advance dan Pentaho sedangkan pada External Table malah menunjukkan penurunan performa dengan opsi paralel.

Proses Load data pada aplikasi central mapper selanjutnya dapat menggunakan SQL*Loader kustom/advance dengan penggunaan opsi parameter direct dan unrecoverable karena menunjukkan performa terbaik pada data yang besar dan tidak terlalu jauh tertinggal pada data dengan jumlah kecil serta sesuai dengan karakteristik proses load data pada aplikasi central mapper yang membutuhkan kecepatan dalam proses load datanya.

Penelitian ini memiliki beberapa keterbatasan, termasuk lingkungan pengujian yang berbeda dari kondisi sistem yang berjalan saat ini. Faktor eksternal seperti beban jaringan dan beban kerja bersamaan juga tidak diperhitungkan meskipun dalam proses pengujian sudah diminimalisir dengan menonaktifkan jaringan komputer dan penutupan program-program lain selain yang digunakan untuk pengujian. Penelitian ini hanya fokus pada durasi load data dan penggunaan resource dasar tanpa mempertimbangkan pengaruh penuh dari operasi yang berulang atau penggunaan di DBMS selain Oracle.

Rekomendasi untuk penelitian selanjutnya yaitu pengujian pada lingkungan sistem yang berbeda serta lebih memperdalam penelitian terkait opsi paralel yang pada penelitian saat ini kurang berpengaruh terhadap hasil pengujian. Selain itu dapat ditambahkan juga terkait pengujian dengan data dan tabel yang lebih kompleks misalnya tabel yang memiliki primary key, trigger, index dengan jumlah tertentu serta kemampuan menangani error yang terjadi serta analisis penggunaan resource yang lebih mendalam dengan mengukur faktor eksternal[15][16]. Penggunaan alat dan tools monitoring resource yang lebih kompleks juga menjadi rekomendasi selanjutnya karena pada penelitian ini hanya menggunakan tools internal sistem operasi Windows.

DAFTAR PUSTAKA

- [1] P. Martins, F. Sá, C. Wanzeller, and M. Abbasi, "A performance study on different data load methods in relational databases," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 2019, pp. 1–7. doi: 10.23919/CISTI.2019.8760615.
- [2] M. Durneková and M. Kvet, "Data Import and Export Methods," in *Conference of Open Innovations Association, FRUCT*, 2021, pp. 423–428.
- [3] Oracle, "Oracle Database Utilities." Accessed: Jul. 25, 2024. [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/21/sutil/index.html>
- [4] M. A. S. Panatagama, "Migrasi Basis Data Non-Relasional MongoDB ke MySQL Menggunakan Pentaho Data Integration," *AUTOMATA*, vol. 1, no. 1, 2020.
- [5] A. R. Iskandar, A. Junaidi, and A. Herman, "Extract, Transform, Load sebagai upaya Pembangunan Data Warehouse," *Journal of Informatics and Communication Technology (JICT)*, vol. 1, no. 1, pp. 25–35, 2019.
- [6] P. R. Sahoo and C. Phalak, "High Speed Data Loading for Large Sized RDBMS Tables," in *Signal Processing and Information Technology: First International Joint Conference, SPIT 2011 and IPC 2011, Amsterdam, The Netherlands, December 1-2, 2011, Revised Selected Papers 1*, 2012, pp. 282–286.
- [7] A. D. Barahama and R. Wardani, "Utilization Extract, Transform, Load for Developing Data Warehouse in Education Using Pentaho Data Integration," in *Journal of Physics: Conference Series*, 2021, p. 12030.
- [8] A. M. I. Florea, V. Diaconita, and R. BOLOGA, "Data integration approaches using ETL.," *Database Systems Journal*, vol. 6, no. 3, 2015.
- [9] Y. Y. Putra, O. Purwaningrum, and R. H. Winata, "Perbandingan Performa Respon Waktu Kueri MySQL, PostgreSQL, dan MongoDB," *Jurnal Sistem Informasi dan Bisnis Cerdas*, vol. 15, no. 1, pp. 39–48, 2022.
- [10] P. Dhanda and N. Sharma, "Extract Transform Load Data with ETL Tools.," *International journal of advanced research in computer science*, vol. 7, no. 3, 2016.
- [11] W. S. Ginanjar and M. Nishom, "An Efficient Method for Speeding up Large-Scale Data Transfer Process to Database: A Case Study," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.
- [12] K. Trumstedt, "Evaluation of methods for loading and processing of measurement data in Oracle," 2016.
- [13] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retrieval operations using a web/android application to explore load balancing—Sharding in MongoDB and its advantages," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 325–330.
- [14] M. I. Porter and I. L. Coat, "Evaluation of methods for rapidly inserting data into an Oracle relational database," 2003.
- [15] J. Otrzasek, W. Mueller, and K. Koszela, "Methodology of comparing the performance of SQL Insert operations in selected RDBMS," *Journal of Research and Applications in Agricultural Engineering*, vol. 57, no. 2, pp. 134–137, 2012.
- [16] A. B. Winnetou, S. A. Wicaksono, and A. Pinandito, "Analisis Peningkatan Performa Proses ETL (Extract, Transform, Dan Loading) Pada Data Warehouse Dengan Menerapkan Delta Extraction Menggunakan Historical Table," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 4, pp. 1366–1371, 2018.
- [17] X. Zhou, Y. Li, and K. Wang, "Performance Optimization in Data Retrieval: A Comprehensive Study," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 567–580, 2020.
- [18] J. Smith, A. Brown, and R. Miller, "Data Loading Techniques for Large-Scale Data Warehouses," *International Journal of Data Science*, vol. 28, no. 3, pp. 210–225, 2019.
- [19] L. Chen and X. Li, "Optimized ETL Strategies for Big Data Systems: A Comparative Study," *IEEE Transactions on Big Data*, vol. 7, no. 2, pp. 340–355, 2021.
- [20] S. Gupta and M. Sharma, "A Comparative Study on ETL Processes for Data Warehousing," *International Journal of Information Systems*, vol. 19, no. 1, pp. 112–130, 2022.
- [21] R. Patel and D. Johnson, "Efficient Data Integration Techniques for Enterprise Databases," *Journal of Database Management*, vol. 32, no. 2, pp. 85–102, 2020.