



UJI PERFORMA DAN PERBANDINGAN RDBMS MYSQL DAN HIVE-HADOOP

Nurul Azizah¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
nurul.azizah195@gmail.com, henry@nurulfikri.co.id

Abstract

The amount of data that is proliferating today requires fast storage as well, and this is because they need for data is also critical and accessing it also requires fast time. Therefore we need to know a system that supports big data processing or big data promptly. Fast enough. We need to know how the performance of a relational database system and a system that processes data is not relational. The existence of MySQL and Hive-Hadoop can make us find which system we should use to store data and obtain the desired data quickly. This analysis and comparison research wants to know how the performance of the two systems, namely between MySQL and Hive-Hadoop, can access structured data. This research will be conducted by using experimental methods and manipulating the experimental environment with computer virtualization. The conclusion that Hive-Hadoop can be faster to process extensive data compared to MySQL because Hive-Hadoop uses HDFS, where files will distribute in blocks of files to connected computers.

Keywords: *Hive-Hadoop, MySQL, Big Data, HDFS*

Abstrak

Banyaknya data yang berkembang pesat saat ini membutuhkan penyimpanan yang cepat pula, hal ini karena kebutuhan akan data juga sangat penting dan dalam mengakses pun membutuhkan waktu yang cepat pula, oleh karena itu kita perlu mengetahui sistem yang mendukung pemrosesan data besar atau *big data* dengan waktu yang cukup cepat. Kita perlu tau bagaimana performa dari sebuah sistem *database* yang relasional dan sistem yang memproses data secara tidak relasional. Adanya MySQL dan *Hive-Hadoop* bisa membuat kita menemukan sistem mana yang harus kita pakai untuk selanjutnya dalam menyimpan data dan memperoleh data yang diinginkan secara cepat. Penelitian analisa dan perbandingan ini ingin mengetahui bagaimana performa dari kedua sistem yakni antara MySQL dan *Hive-Hadoop* yang keduanya bisa mengakses data secara terstruktur. Penelitian ini akan dilakukan dengan metode percobaan dan memanipulasi lingkungan percobaan dengan virtualisasi komputer. Dengan kesimpulan bahwa *Hive-Hadoop* bisa lebih cepat untuk memproses data besar dibandingkan dengan MySQL, karena *Hive-Hadoop* menggunakan HDFS dimana file akan didistribusikan secara blok-blok file ke dalam komputer yang terhubung.

Kata kunci: *Hive-Hadoop, MySQL, Big Data, HDFS*

1. PENDAHULUAN

Sejak beberapa tahun terakhir ini, perkembangan IT sedang berkembang dengan sangat cepat dan pesat. Dengan perkembangan tersebut, tidak dapat dipungkiri bahwa perkembangan banyaknya data juga sangat berpengaruh dimulai dari memproses data yang besardi suatu kegiatan teknologi. Banyaknya data yang semakin besar dan semakin sulit untuk ditampung pada sistem *database* manapun sehingga sulit untuk dianalisadan diproses dengan cepat. Oleh karena itu, data yang semakin banyak dan besar ini mulai dikenal dengan nama *big data*. Dengan *big data* ini,

proses pencarian, penyimpanan data, dan analisa pada data akan membutuhkan waktu yang cukup lama.

Awalnya banyak sekali organisasi atau perusahaan yang menggunakan RDBMS untuk memproses atau menyimpan data-data yang dimiliki. Namun, seiring perkembangan teknologi informasi yang bertujuan untuk mempermudah kinerja dalam pencarian, penyimpanan data dan lainnya tidak membutuhkan waktu lama lagi untuk melakukan tugasnya. Ada teknologi baru bernama *Hadoop* yang menggunakan teknologi HDFS dan *MapReduce*. Adanya teknologi tersebut maka *Hadoop* memproses data secara

terdistribusi dalam beberapa komputer bahkan ribuan, dengan HDFS dan *MapReduce* maka kinerja *Hadoop* sangat terbantu. Jika makin banyak data yang tersimpan di satu server, maka penyimpanan *storage server* akan semakin penuh. Bila *Hadoop* hanya punya satu *server*, maka kemungkinan *server* tersebut bisa mengalami masalah dan mengakibatkan *storage* tidak dapat diakses kembali.

Kedua teknologi ini sama-sama untuk bisa untuk memproses data namun berbeda cakupan. RDBMS digunakan untuk data yang terstruktur sedangkan *Hadoop* bisa digunakan untuk data yang tidak terstruktur. Dan juga *Hadoop* bukanlah sebuah *database* melainkan sebuah framework yang memungkinkan untuk memproses data berukuran besar. Oleh karena itu, untuk memudahkan dalam pengetahuan dan penggunaan sistem ini, maka penulis membuat analisa performa untuk keduanya sehingga tahu bagaimana performa diantara keduanya dalam memproses data yang besar. Serta tepat dalam menggunakan *software* dan *hardware* yang dipilih dan digunakan.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka ada dua masalah utama dalam pembuatan penelitian ini:

1. Apa sajakah indikator atau parameter yang digunakan untuk menganalisa performa dari teknologi MySQL dan *Hadoop*?
2. Bagaimanakah performa dari kedua teknologi yang dianalisa?

1.2 Batasan Masalah

Adapun untuk batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Analisa performa dari kedua teknologi yang diteliti tidak termasuk keamanan.
2. Penelitian ini dilakukan dalam lingkup percobaan saja bukan di sebuah instansi atau perusahaan.
3. Lingkungan percobaan pada virtualisasi.

1.3 Tujuan dan Manfaat

Secara umum tujuan serta manfaat dari penelitian ini adalah untuk mengetahui bagaimana performa dari *Hadoop* dan MySQL dalam memproses data dalam jumlah besar. Secara khusus, penelitian ini akan menjadi ilmu pengetahuan tambahan bagi peneliti terkait *Hadoop* dan MySQL dan sebagai karya tulis yang bisa digunakan pada lingkungan kampus atau rujukan dalam mengimplementasi.

2. LANDASAN TEORI

2.1 Database

Database adalah kumpulan data (elementer) yang secara logika berkaitan dalam mempresentasikan fenomena/fakta

secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam sistem tertentu [1].

Database menurut Connolly adalah kumpulan data yang secara logis terkait satu sama lain dan digunakan bersama oleh banyak pemakai, serta dirancang untuk memenuhi kebutuhan informasi banyak pemakai dalam suatu organisasi. Artinya *database* merupakan penyimpanan data yang tunggal dan besar yang dapat digunakan secara simultan oleh banyak bagian departemen dan user [2].

2.2 RDBMS

RDBMS (*Relational Database Management System*) merupakan sekumpulan program komputer yang mengendalikan pembuatan, perawatan, dan penggunaan sebuah organisasi *database* dengan menggunakan komputer sebagai *platform*. DBMS memungkinkan pengguna program aplikasi yang berbeda untuk dengan mudah mengakses *database* yang sama. DBMS menggunakan berbagai macam model *database*, seperti *network model* atau *relational model* [3].

2.3 MySQL

MySQL adalah perangkat lunak sistem manajemen basis data yang digunakan untuk menyimpan dan mengolah data [2]. MySQL adalah sebuah server *database SQL multiuser* dan *multi-threaded*. *Database* ini dibuat untuk keperluan sistem *database* yang cukup baik, MySQL untuk sistem operasi Unix bersifat *freeware*, dan terdapat versi *shareware* untuk sistem operasi Windows [3].

2.4 Big Data

Big data adalah data yang melebihi proses kapasitas dari konversi sistem *database* yang ada. Data terlalu besar dan terlalu cepat atau tidak sesuai dengan struktur arsitektur *database* yang ada. Untuk mendapatkan nilai dari data, maka harus memilih jalan alternatif untuk memprosesnya [4].

2.5 HADOOP

Hadoop adalah *platform* perangkat lunak *open source* yang dikelola oleh *Apache Software Foundation*. Ini adalah *platform* yang paling dikenal untuk menyimpan dan mengelola sejumlah besar data secara efisien dan hemat biaya. *Hadoop* merupakan *software open-source* yang dibuat menggunakan Java untuk penyimpanan terdistribusi dan pemrosesan terdistribusi untuk data yang sangat besar yang berada dibawah *Apache Software Foundation*.

Inti dari *Hadoop* ada 2 yakni HDFS dan *MapReduce*. HDFS merupakan file terdistribusi yang tinggi toleransi kegagalannya, maksud dari file terdistribusi sendiri adalah data yang tersimpan tidak berada hanya di satu *hardisk* tetapi disimpan dalam beberapa *hardisk* komputer sedangkan *MapReduce* adalah model pemrograman standar

fungsional dari *Hadoop*, yang digunakan untuk memproses data dalam jumlah besar.

2.6 HIVE

Hive adalah perangkat lunak *data warehouse* berbasis *open source* yang disediakan oleh Apache yang digunakan untuk Query dan mengelola kumpulan data yang didistribusikan secara SQL. *Hive* membantu untuk membuat konsep data dengan cara terstruktur dan menyediakan bahasa *query* seperti SQL yang dikenal sebagai *HiveQL* [5]. *Hive* memiliki *metastore* yang mana sebagai tempat menyimpan data. layanan *metastore* berjalan di JVM yang sama dengan layanan *Hive* yang menggunakan *database* tertanam yakni *derby* yang disimpan di sistem file lokal dalam mode ini.

3. METODE PENELITIAN

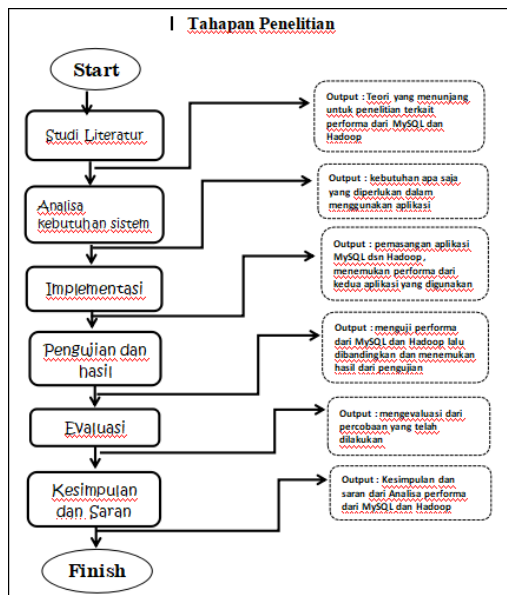
3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah jenis **penelitian percobaan** yang berupaya mengisolasi serta kontrol di masing-masing situasi yang sesuai dengan situasi yang hendak diteliti lalu mengamati pada efek maupun pengaruh saat *conditional sentence* memanipulasi kondisi tersebut.

Peneliti akan melakukan manipulasi atau eksperimen terhadap MySQL dan *Hadoop* sehingga peneliti mengetahui performa keduanya.

3.2 Tahapan penelitian

Berikut adalah alur tahapan penelitian yang akan dilakukan:



Gambar 1. Tahapan Penelitian

Tahapan penelitian yang pertama adalah Studi Literatur mencari teori-teori yang mendukung untuk penelitian, kedua adalah Analisa kebutuhan sistem yakni mencari tahu kebutuhan dari sistem yang akan coba diteliti dan rancangan pengujian sistem yang akan diuji, ketiga adalah

Implementasi yakni melakukan pemasangan aplikasi yang akan digunakan untuk penelitian, keempat adalah pengujian dan hasil dimana pengujian dilakukan dan akan didapati hasil dari rancangan pengujian yang telah dibuat untuk penelitian. Kelima adalah Evaluasi dari percobaan yang sudah dilakukan dalam penelitian, dan yang terakhir adalah kesimpulan dari penelitian yang sudah dilakukan dan saran untuk para peneliti selanjutnya yang akan meneliti sistem yang sama atau sejenisnya.

3.3 Rancangan Penelitian

a. Metode Pengumpulan Data

Digunakan untuk memperoleh informasi dan data sebagai acuan rancangan untuk menyelesaikan masalah yang peneliti ambil. Dalam penelitian ini terdapat beberapa metode yang dilakukan dalam pengumpulan data, yaitu:

1. Studi Pustaka

Langkah pertama yang dilakukan dalam pengumpulan data adalah studi pustaka. Studi pustaka ini akan mengarahkan peneliti untuk mencari data serta informasi melalui dokumen-dokumen berupa dokumen tertulis, foto-foto, gambar maupun dokumen elektronik yang bisa mendukung dalam proses penulisan dan penelitian.

2. Eksperimen

Melakukan eksperimen atau percobaan yang memiliki tujuan untuk menganalisis faktor-faktor yang merupakan performa dari MySQL dan *Hadoop* yang akan digunakan untuk bahan laporan dalam penulisan tugas akhir ini. Melakukan eksperimen atau percobaan yang memiliki tujuan untuk menganalisis faktor-faktor yang merupakan performa dari MySQL dan *Hadoop* yang akan digunakan untuk bahan laporan dalam penulisan tugas akhir ini.

b. Lingkungan Percobaan

Penelitian ini akan dilakukan dalam lingkungan percobaan dengan teknologi virtualisasi.

c. Alat dan Bahan

Untuk melakukan penelitian ini, peneliti menggunakan perangkat:

1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah:

- Model : Toshiba DT01ACA100 (MS20A810)
- Processor : Intel® Core™ i7-7700 CPU @ 3.60GHz × 8
- RAM : 8GB
- Hardisk : 1,0TB

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

- Sistem Operasi Xubuntu 16.04 LTS
- MySQL
- Hadoop
- Hive
- Derby
- JDK 1.8
- SSH

4. ANALISA DAN PERANCANGAN

4.1 Analisa Kebutuhan Software

Software yang akan digunakan pada penelitian ini adalah diantaranya :

1. VirtualBox, karena lingkungan percobaannya adalah virtual, maka diperlukan salah satu aplikasi untuk virtualisasi yakni Virtual Box.
2. OS Linux Xubuntu 16.04, merupakan sistem operasi yang akan digunakan untuk penelitian.
3. MySQL Versi 5.7 , *software database* yang akan dijalankan untuk query *big data*.
4. Hadoop Versi 2.7.2, platform berbasis java untuk mendukung aplikasi yang berjalan pada *bigdata*.
5. Hive Versi 1.2.2, perangkat lunak *datawarehouse* yang digunakan untuk query dan mengolah data yang didistribusikan secara SQL dan Hadoop.
6. Derby Versi 10.13.1.1, *open source* sebagai *metastore* yang tertanam Hive.
7. MapReduce, model komputasi berbasis java pada system terdistribusi dalam rangka mendukung aplikasi *Big Data*.
8. JDK 1.8.0_221, perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode Java ke *bytecode*.
9. SSH, protokol jaringan kriptografi untuk komunikasi data yang aman.

4.2 Analisa Kebutuhan Software

Berdasarkan kebutuhan software yang diperlukan, maka kebutuhan hardware yang digunakan pada penelitian ini adalah sebagai berikut :

- 1) Spesifikasi minimum untuk menggunakan MySQL
 - Processor : Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU.
 - Core : Single (Dual/Quad Core is recommended)
 - RAM : 4 GB (6 GB recommended)
- 2) Spesifikasi minimum untuk menggunakan Hadoop
 - Processor : Intel i3 or above
 - RAM : 4 GB
 - Free Disk Space : 25 GB
- 3) Spesifikasi fisik yang digunakan
 - Processor : Intel® Core™ i7-7700 CPU

- RAM : 8 GB
- Hardisk : 1,0 TB

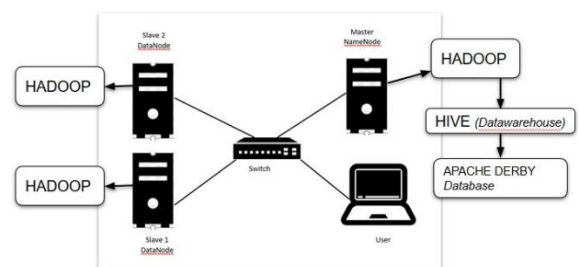
Tabel 1. Spesifikasi Hardware pada Mesin Virtual yang akan digunakan sebagai Penelitian

Node	Processor	RAM	OS
H A D O O P	Master	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
	Slave1	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
	Slave2	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
MySQL Server	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB	Xubuntu 16.04

4.3 Rancangan Sistem

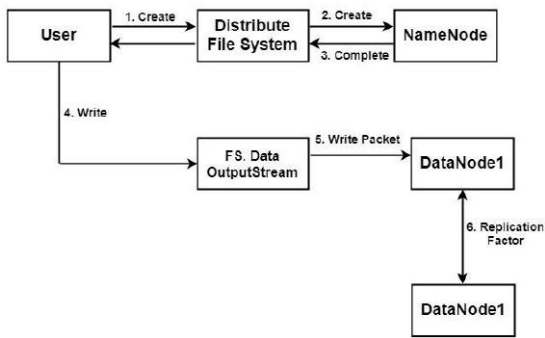
1. Rancangan Arsitektur Sistem Hadoop

Rancangan arsitektur yang akan digunakan untuk sistem Hadoop adalah peneliti akan menggunakan 3 server yang diantaranya 1 master dan 2 slave yang mana pada masing-masing server akan dipasang sebuah Hadoop dan di dalam server master akan dipasang juga sebuah Hive dan Derby sebagai pendukung Hadoop yang akan digunakan untuk memproses data secara SQL.



Gambar 2. Rancangan Arsitektur dari Sistem Hadoop

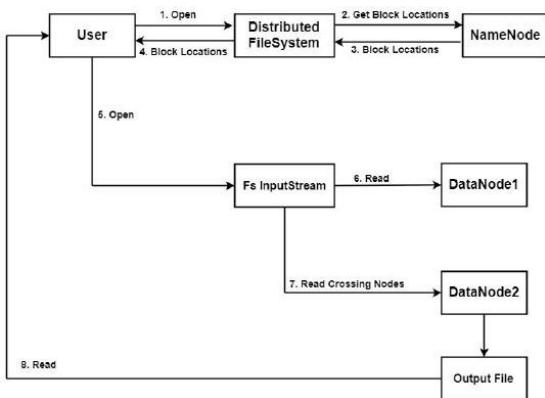
Untuk proses write yang terjadi pada Hadoop akan dijalankan di HDFS dengan skema berikut:



Gambar 3. Proses Write yang Terjadi di HDFS

Pada gambar 3, bisa dijelaskan bahwa jika ada perintah menyimpan oleh *user*, maka komputer *user* akan berkomunikasi dengan *namenode* untuk memberitahu bahwa ada file yang akan disimpan di HDFS serta menanyakan lokasi *datanode* yang bisa diakses untuk menyimpan data yang akan disimpan, setelah mendapat daftar nama dan alamat *datanode* yang tersedia maka komputer *user* akan langsung mentransfer data ke *datanode* yang ada. Data yang ditransfer sudah otomatis dibelah menjadi *block* atau kepingan data yang tersimpan dalam *datanode*. Setelah kepingan data diterima maka *datanode* akan mengirimkan laporan ke *namenode* bahwa data telah diterima dan disimpan.

Sedangkan untuk proses *read* pada HDFS adalah sebagai berikut:

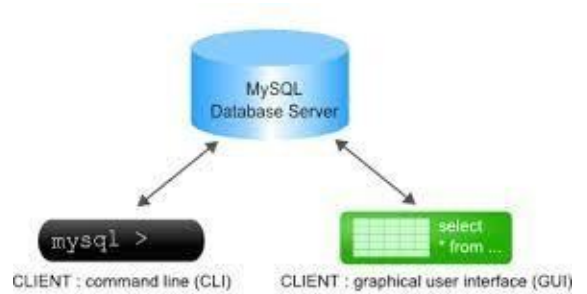


Gambar 4. Proses Read yang terjadi di HDFS

Pada gambar 4, ditunjukkan alur dari proses *read* data atau file di HDFS. Saat *user* memerintahkan eksekusi perintah *read* di komputer *user*, maka akan ada komunikasi dengan *namenode* untuk menanyakan nama dan alamat *datanode* yang harus diakses untuk mendapatkan data yang diinginkan. Maka setelah *user* mendapatkan nama dan alamat *datanode* akan secara langsung mengakses *datanode* yang bersangkutan dan kemudian data akan ditampilkan sesuai perintah dari *user*.

2. Rancangan Arsitektur RDBMS MySQL

Peneliti akan melakukan perancangan arsitektur MySQL dengan menggunakan arsitektur MySQL *Server* yang biasa digunakan oleh para pengguna pada umumnya, yakni peneliti akan menggunakan satu *client* dan satu *server*. Dimana satu *server* ini akan menjadi *node* utama dimana data disimpan dan *client* yang melakukan *read* dan *write*. Seperti yang terlihat pada gambar berikut:



Gambar 5. Rancangan Arsitektur MySQL

Untuk alur kerja dari MySQL adalah akan terjadinya dialog antara *client* dan *server* melalui *daemon* pada MySQL, dimana *daemon* ini adalah sebutan untuk program yang berjalan pada *background* (berjalan tanpa campur tangan manusia). *Daemon* tidak akan aktif bekerja, kecuali jika ada permintaan kepada *daemon* untuk melakukan pekerjaan. Setelah *daemon* MySQL dijalankan, *daemon* tersebut akan memberikan proses dialog dengan *client*. *Client* kemudian diminta untuk melakukan autentifikasi untuk mengolah *database* MySQL.

4.4 Rancangan Pengujian

Pengujian pada *Hadoop* akan menggunakan *Hive* yang merupakan *data warehouse* yang ada di *Hadoop* itu sendiri dan *Derby* sebagai *database* eksternalnya dan untuk MySQL akan diuji secara langsung di MySQL, dimana dua perangkat tersebut memungkinkan untuk melakukan *create* dan *select*, sehingga bisa dibandingkan kinerja keduanya berdasarkan percobaan pada 2 perangkat tersebut. *Query* yang akan digunakan adalah *query* “*select*” dan “*insert*” untuk diujikan pada kedua perangkat.

1. Pengujian *Hadoop*

Perangkat yang akan digunakan dalam pengujian *Hadoop* ini yaitu *Hive* dengan *database* eksternalnya yakni *Derby*. Berikut akan dijabarkan langkah-langkah yang akan dilakukan untuk proses pengukuran pada *Hadoop*.

a. Pengujian Lama Waktu Load Data

Untuk melakukan pengujian lama waktu *load data* adalah sebagai berikut:

1. Memastikan topologi sistem yang diukur.
2. Menentukan *data warehouse* dan *database* yang akan digunakan yaitu *Hive* dan *Derby*.
3. Menentukan data yang akan digunakan untuk

- pengujian.
 - 4. Menentukan *query* yang akan digunakan yaitu *insert*.
 - 5. Menentukan berapa kali pengujian akan dilakukan.
 - 6. Menjalankan *Hive* yang telah terinstal di komputer untuk menguji kinerja *Hadoop*.
 - 7. Melihat *output* yang dihasilkan dari alat ukur yaitu waktu *load data* pada *Hadoop* yang telah ditentukan datanya.
 - 8. Ulangi langkah-langkah di atas untuk penambahan data 2GB-5GB. Dan selanjutnya dicatat waktu pengukuran saat pengujian.
- b. Pengujian Lama Waktu *Select Data*
 Untuk melakukan pengujian lama waktu *select data* adalah sebagai berikut:
1. Memastikan topologi sistem yang diukur.
 2. Menentukan *data warehouse* dan *database* yang akan digunakan yaitu *Hive* dan *Derby*.
 3. Menentukan data yang akan digunakan untuk pengujian.
 4. Menentukan *query* yang akan digunakan yaitu *select*.
 5. Menentukan berapa kali pengujian akan dilakukan.
 6. Menjalankan *Hive* yang telah terinstal di komputer untuk menguji kinerja *Hadoop*.
 7. Melihat *output* yang dihasilkan dari alat ukur yaitu waktu *query data* pada *Hadoop* yang telah ditentukan datanya.
 8. Ulangi langkah-langkah diatas untuk penambahan data 2GB-5GB. Dan selanjutnya dicatat waktu pengukuran saat pengujian.

Tabel 2. Contoh Tabel Hasil Eksekusi *Load* dan *Select Data*

Ukuran data	1	2	3	4	5
	GB	GB	GB	GB	GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1					
Pengujian 2					
Pengujian 3					

2. Pengujian MySQL

Untuk pengujian MySQL rancangan pengujian yang akan digunakan dalam proses pengukuran adalah sebagai berikut:

- a. Pengujian Lama Waktu *Load Data*
1. Memastikan topologi sistem yang akan diukur.
 2. Menentukan *database* yang akan digunakan yaitu MySQL.
 3. Menentukan macam-macam *query* yang akan digunakan, diantaranya *Insert*.
 4. Melakukan pengujian beberapa kali dengan penambahan data untuk menemukan

- perbedaan kecepatan waktu *load data* seiring banyaknya data yang di *insert*.
- 5. Melihat *output* yang dihasilkan dari alat ukur yaitu kecepatan *load data*.

b. Pengujian Lama Waktu *Select Data*

1. Memastikan topologi sisten yang akan diukur.
2. Menentukan *database* yang akan digunakan yaitu MySQL.
3. Menentukan macam-macam *query* yang akan digunakan, diantaranya *select*.
4. Melakukan pengujian beberapa kali dengan penambahan data untuk menemukan perbedaan kecepatan waktu *query* eksekusi seiring banyaknya data yang di *select*.
5. Melihat *output* yang dihasilkan dari alat ukur yaitu kecepatan *query eksekusi*.
6. Setelah itu dibandingkan hasil masing-masing percobaan untuk dianalisa.

Tabel 3. Contoh Tabel Hasil *Load* dan *Select Data* di MySQL

Ukuran data	1	2	3	4	5
	GB	GB	GB	GB	GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1					
Pengujian 2					
Pengujian 3					

5. IMPLEMENTASI DAN ANALISA

5.1 Data Penelitian

Data yang akan digunakan dalam penelitian ini adalah data dengan format CSV yang berukuran 1GB sampai 5GB. Data didapat dari *website* <http://data.gdeltproject.org/events/index.html>. Peneliti menggunakan data yang di dapat dari *website* tersebut dengan ukuran 1GB lalu *dicopy-paste* agar bisa mendapatkan data sampai 5GB. Data ini merupakan data yang terstruktur dalam 1 tabel sehingga menggunakan format CSV agar memudahkan dimasukkan ke dalam format *database* terstruktur. Untuk memudahkan memasukkan data ke dalam tabel sistem SQL, maka data dibuat dengan format delimiter (pembatas) koma, sehingga perintah atau format yang dijalankan bisa disesuaikan dengan format SQL.

5.2 Pengujian

Pada pengujian ini, peneliti sudah memastikan bahwa parameter yang akan digunakan sebagai bahan uji dalam penelitian ini adalah diantaranya

- a. Besarnya data yakni antara 1GB sampai dengan 5GB.
- b. *Query* eksekusi pada *Hive* dan MySQL yakni *Load data* dan *select data*.

Paramater yang digunakan diatas adalah uji coba yang dilakukan oleh peneliti sehingga bisa mengetahui bagaimanakah performa dari MySQL dan Hadoop yang digunakan untuk bahan penelitian.

1. Pengujian Hadoop

Langkah-langkah yang dilakukan untuk menguji Hadoop:

- 1) Setelah memastikan Hadoop, Hive dan Derby terpasang (ada dalam lampiran). Masuk ke sistem hive
\$ bin/hive
- 2) Cek database
hive>show databases;
- 3) Membuat database dan menggunakan database tersebut
hive>create database tugas;
hive>use tugas;
- 4) Create table data
hive>create table data1gb(kode1 int, kode2 int, kode3 char(20), kode4 char(20));
- 5) Melakukan load data ke dalam tabel
hive>load data local inpath '/home/document/data1gb.csv' overwrite into table data1gb;
- 6) Mencatat waktu yang diperoleh dari load data
- 7) Melakukan query select dan mencatat waktu hasil query
hive> select * from data1gb;
- 8) Melakukan langkah 5-7 untuk menguji data 1GB sampai 3 kali sebagai perbandingan.
- 9) Melakukan langkah 4 sampai 8 untuk data 2GB sampai 5GB.

Tabel 5. Pengujian Load Data pada Hadoop

Ukuran data	1 GB	2 GB	3 GB	4 GB	5 GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1	60,642	94,478	107,629	152,722	329,638
Pengujian 2	61,945	106,198	107,334	169,987	329,946
Pengujian 3	62,034	108,041	108,424	171,264	330,103
Rata-rata	61,54	102,9	107,79	257,656	385,229

Hasil keseluruhan pengujian dari select data dapat dilihat dalam tabel berikut:

Tabel 6. Pengujian Select Data pada Hadoop

Ukuran data	1 GB	2 GB	3 GB	4 GB	5 GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1	0 sec, 8map	0 sec, 18 map	0 sec, 26 map	0 sec, 35 map	0 sec, 43 map
Pengujian 2	0 sec, 8 map	0 sec, 17 map	0 sec, 28 map	0 sec, 34 map	0 sec, 43 map
Pengujian 3	0 sec, 8map	0 sec, 17 map	0 sec, 26 map	0 sec, 34 map	0 sec, 44 map
Rata-rata	0 sec, 8 map	0 sec, 17 map	0 sec, 26 map	0 sec, 34 map	0 sec, 43 map

2. Pengujian MySQL

Langkah-langkah pengujian dalam Load data yang dilakukan pada MySQL:

- 1) Pastikan masuk ke dalam MySQL untuk memulaimelakukan pengujian
\$ mysql -u root -p
- 2) Mengecek dan membuat database
mysql > show databases;
mysql > create database tugas;
(keterangan: nama database menggunakan nama sesuai yang diinginkan)
- 3) Menggunakan database yang sudah dibuat
mysql > use tugas;
- 4) Membuat tabel 1GB
mysql > create table data1gb(kode1 int, kode2 int, kode3 int, kode4 int);
- 5) Melakukan load data ke dalam tabel
mysql > load data local infile '/home/document/data1gb.csv' into table data1gb fields terminated by ',' lines terminated by '\n';
[keterangan : load data local infile(pada file penyimpanan local), „home/document/data1gb.csv“ (lokasi datanya), fields terminated by „,“ , lines terminated by „\n“ (baris baru menggunakan enter)]
- 6) Mencatat waktu yang diperoleh dari load data
- 7) Melakukan query select dan mencatat waktu hasil query
mysql> select * from data1;
- 8) Melakukan langkah 5-7 untuk menguji data 1GB sampai 3 kali sebagai perbandingan.
- 9) Melakukan langkah 4 sampai 8 untuk data 2GB sampai 5GB
- 10) Menghitung rata-rata waktu yag didapatkan

Hasil dari pengujian secara keseluruhan pada saat load data dapat dilihat dalam tabel berikut:

Tabel 7. Pengujian *Load Data* pada MySQL

Ukuran data	1	2	3	4	5
Pengujian ke	GB	GB	GB	GB	GB
	(s)	(s)	(s)	(s)	(s)
Pengujian 1	2 hours 29 min 19,09 sec	2 hours 29 min 32,15 sec	2 hours 30 min 43,09 sec	2 hours 32 min 47,59 sec	2 hours 33 min 13,05 sec
Pengujian 2	2 hours 29 min 20,74 sec	2 hours 29 min 32,40 sec	2 hours 30 min 44,19 sec	2 hours 30 min 48,34 sec	2 hours 33 min 13,46 sec
Pengujian 3	2 hours 29 min 21,67 sec	2 hours 29 min 32,42 sec	2 hours 30 min 44,27 sec	2 hours 30 min 48,42 sec	2 hours 33 min 14,04 sec
Rata-rata	2 hours 29 min 20,5 sec	2 hours 29 min 32,32 sec	2 hours 30 min 43,85 sec	2 hours 30 min 48,11 sec	2 hours 33 min 13,51 sec

Hasil pengujian secara keseluruhan pada saat *select data* dapat dilihat dalam tabel berikut:

Tabel 8. Hasil *Select Data* di MySQL

Ukuran data	1	2	3	4	5
Pengujian ke	GB	GB	GB	GB	GB
	(s)	(s)	(s)	(s)	(s)
Pengujian 1	36,20 sec	40,15 sec	44,56 sec	48,51 sec	57,32 sec
Pengujian 2	40,13 sec	48,34 sec	62,39 sec	65,27 sec	71,44 sec
Pengujian 3	44,27 sec	62,41 sec	70,12 sec	73,26 sec	75,17 sec
Rata-rata	40,2 sec	50,3 sec	59,02 sec	62,34 sec	67,97 sec

- Analisa Pengujian *Load Data* MySQL vs *Hadoop*
Pada pengujian yang telah dilakukan pada *load data* di MySQL dan *Hadoop* ternyata semakin banyaknya data semakin banyak waktu yang diperlukan serta setelah dibandingkan waktu yang diperlukan lebih efektif pada saat *load data* di *Hadoop* dibandingkan dengan di MySQL. Berikut adalah tabel perbandingan waktu *load data* antara MySQL dan *Hadoop*:

Tabel 9. Rata-rata Waktu *Load Data* di MySQL dan *Hadoop*

Ukuran Data (GB)	Load data MySQL(s)	Load data Hadoop(s)
1 GB	2 hours 29 min 20,5 sec	61,54 sec
2 GB	2 hours 29 min 32,32 sec	102,9 sec
3 GB	2 hours 30 min 43,85sec	107,79 sec

Ukuran Data (GB)	Load data MySQL(s)	Load data Hadoop(s)
4 GB	2 hours 30 min 48,11 sec	257,656 sec
5 GB	2 hours 33 min 13,51sec	385,229 sec

Berdasarkan tabel 9, waktu yang diperlukan oleh MySQL lebih lama dibandingkan *Hadoop*, dalam hal ini karena MySQL butuh memproses data secara terstruktur dan dalam satu manajemen. Sedangkan untuk *Hadoop*, dalam memproses data dibantu proses dari HDFS yakni file yang dimasukkan akan didistribusikan ke dalam *node-node* lain yang terhubung satu dengan yang lainnya. Sehingga waktu yang diperlukan untuk *load data* jauh lebih cepat dibandingkan dengan MySQL.

- Analisa Pengujian *Select Data* MySQL vs *Hadoop*
Untuk pengujian *select data* di MySQL dan *Hadoop* didapatkan hasil berikut ini:

Tabel 10. Hasil Pengujian *Select Data* di MySQL dan *Hadoop*

Ukuran Data (GB)	Select data MySQL(s)	Select data Hadoop(s)
1 GB	40,2 sec	0 sec, 8 map
2 GB	50,3 sec	0 sec, 17 map
3 GB	59,02 sec	0 sec, 26 map
4 GB	62,34 sec	0 sec, 34 map
5 GB	67,97 sec	0 sec, 43 map

Dari tabel 10, kita bisa simpulkan semakin banyaknya data yang dimasukkan akan semakin lama juga waktu yang dibutuhkan untuk membaca data. MySQL memerlukan waktu untuk membaca satu per satu data yang telah dimasukkan lalu kemudian dipilih untuk dilihat secara keseluruhan. Untuk *Hadoop*, data yang telah dipetakan akan secara cepat didapatkan dan ditampilkan oleh sistem sehingga waktu yang diperlukan sangat cepat dibandingkan dengan MySQL.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan dari pengujian yang telah dilakukan oleh peneliti terkait performa MySQL dan *Hadoop* adalah sebagai berikut:

- Indikator atau parameter yang digunakan pada penelitian ini adalah besarnya data yakni mulai 1GB sampai 5GB, kriteria data tersebut diujikan pada sistem MySQL dan *Hadoop*, dan parameter lainnya adalah kueri eksekusi yang digunakan yakni *load data* dan *select data* karena pada penelitian ini, peneliti

ingin mengetahui performa dari masing-masing sistem dalam bentuk menulis dan membaca data sehingga kueri yang digunakan adalah *load data dan select data*.

2. Performa yang ditampilkan oleh kedua sistem ini sangat signifikan dilihat dari lamanya waktu. Berdasarkan dari parameter yang digunakan oleh peneliti, bahwa ketika *load data* ke dalam MySQL membutuhkan waktu yang sangat lama jika dibandingkan dengan *Hadoop*. Diantara keduanya memiliki karakteristik yang berbeda-beda sehingga jika dibandingkan berdasarkan waktu memang akan berbeda jauh, karena secara umum *Hadoop* merupakan file sistem untuk mendistribusikan data sedangkan MySQL merupakan *database* untuk penyimpanan data.

6.2 Saran

Saran dari peneliti untuk para peneliti yang ingin menguji performa *Hadoop* dan MySQL selanjutnya adalah sebagai berikut:

1. Karena penelitian ini membahas tentang *big data*, maka gunakanlah komputer dengan spesifikasi yang tinggi atau jika ingin melakukan percobaan maka harus dengan RAM minimal 4 GB, *hardisk* menggunakan SSD, *processor* minimal Intel i3.
2. Menggunakan komputer fisik dan bukan komputer virtual untuk uji coba *big data* melalui sistem atau aplikasi apapun, agar hasil yang didapatkan bisa lebih akurat.
3. Pastikan bahwa pada komputer yang akan digunakan hanya ada aplikasi *database* atau yang berkaitan untuk *database*.
4. Hanya gunakan OS yang ringan untuk mempermudah pemrosesan sistem dan data, sebaiknya gunakan OS Linux versi server serta gunakan aplikasi yang terbaru agar masih ada versi yang bisa terkoneksi dengan aplikasi lain.
5. Pastikan hanya menjalankan 1 *service* tanpa ada proses lain yang berjalan ketika *load data* atau *select data*, karena akan sangat mempengaruhi prosesor dalam bekerja.

DAFTAR PUSTAKA

- [1] S. Minarni, "Sistem Informasi *Inventory* Obat pada Rumah Sakit Umum Daerah (RSUD) Padang," *Jurnal Momentum*, p. 105, 2014.
- [2] F. P. H. H. Susanto, "Perbandingan Unjuk Kerja MySQL *Cluster* Versus MySQL Konvensional," Yogyakarta, 2015.
- [3] W. S. D. S. Ida Bagus Teguh Teja Murti, Artist, "Analisa Perbandingan Kinerja DSMS dan RDBMS pada Studi Kasus Transaksi Online," [Art], Institut Teknologi Sepuluh November.
- [4] A.E. T., "Rancangan Aplikasi *Smart City* berbasis *Mobile* untuk meningkatkan Kualitas Layanan Publik Studi Kasus Pemkot Bogor," *Jurnal Teknik Komputer AMIK BSI*, Vol. 3, p.26, 2016.
- [5] K. S. Raste, "*Big Data Analytics-Hadoop Performance Analysis*," in *In Partial Fulfillment of the Requirements for Degree Master of Science in Computer Science*, San Diego, 2014.