Analisis Pengolahan *Text File* pada Hadoop Cluster dengan Memperhatikan Kapasitas *Random Access Memory (RAM)*

Irvan Nur Aziz[1], Fitriyani[2], Kemas Rahmat Saleh W[3]
Fakultas Informatika, School of Computing, Universitas Telkom
Jalan Telekomunikasi No.1, Dayeuh Kolot, Bandung 40257
Van_ziz@ymail.com [1], fitriyani.y@gmail.com [2], bagindok3m45@gmail.com [3]

Abstrak

Implementasi Hadoop cluster untuk pengolahan data secara terdistribusi dalam skala besar sudah menjadi tren saat ini. Hadirnya hadoop cluster sangat membantu dalam bidang pengolahan data, banyak perusahaan yang mengimplementasikan hadoop cluster seperti facebook, yahoo, dan amazon. Hal ini didasari oleh kelebihan hadoop yang dapat memiliki performansi tinggi dengan menggunakan hardware sederhana.

Tujuan dari penelitian ini adalah mengimplementasikan hadoop cluster dengan menggunakan benchmark wordcount sebagai tools untuk mengetahui tingkat performansi dari jenis *file text* dengan memperhatikan kapasitas *Random Access Memory (RAM).*

Waktu ujicoba yang dihasilkan dari jenis-jenis *text file* tersebut menunjukan urutan tingkat performansi terbaik dimulai dari jenis text file csv, txt, xlsx, pdf dan yang terakhir adalah jenis *file* doc. Waktu peningkatan performansi dari semua jenis *file* tidak mengalami peningkatan yang sama dengan peningkatan kapasitas RAM, pada saat kapasitas RAM ditingkatkan menjadi 100% hasil percobaan menunjukan performansi dari jenis file doc mengalami peningkatan sebesar 4,58%, *file* pdf sebesar 7,57%, *file* csv sebesar 8,87%, file xlsx sebesar 8,35% dan *file* txt sebesar 12,82%.

Kata Kunci : Cluster, Hadoop, MapReduce, HDFS, RAM, Bandwidth

Abstract

Nowadays, Implementation of Hadoop clusters for distributed processing of data on a large scale has become a trend. The presence of cluster hadoop very helpful in the field of data processing, some companies are implementing hadoop cluster such as facebook, yahoo and amazon. It is based on the hadoop excess which can have high performance by using simple hardware.

The aim of this research is to implement hadoop cluster by using Wordcount benchmarks as tools to determine the level of performance of this text file type regard to Random Access Memory (RAM) capacity. Time trials resulting from the types of the text file showing the best performance level sequence starting from the csv, txt, xlsx, pdf, and doc. The performance enhancement time of all kinds of text files are not proportional with the capacity of RAM, when the RAM capacity increased to 100% of the performance results of the experiment showed doc file increased by 4.58%, pdf file increased by 7.57%, csv file increased by 8.87%, xlsx file increased by 8,35% and txt file increased by 12.82%.

Keywords : Cluster, Hadoop, MapReduce, HDFS, RAM, Bandwidth

1. Pendahuluan

Perkembangan teknologi terus berkembang pesat dari tahun ke tahun hal ini karena permintaan dan kebutuhan masyarakat yang semakin banyak, mulai dari penelitian, pekerjaan, hingga hiburan. Salah satu sektor teknologi yang terkena dampak dari perkembangan yang sangat pesat ini adalah sektor data elektronik, ukuran data-data elektronik yang dimiliki oleh perusahaan sebelumnya hanya *gigabyte* hingga *terabyte* namun saat ini sudah mengalami peningkatan yang sangat signifikan, data yang dimiliki perusahaan sudah mencapai ukuran *petabyte*. Dengan data sebesar itu maka diperlukan metode pengolahan data yang optimal dan tempat penyimpanan yang juga sangat besar.

Adakalanya aplikasi yang dibuat membutuhkan komputer dengan sumber daya yang tinggi sebagai lingkungan implementasi dan biasanya harga untuk komputer dengan sumber daya yang tinggi tidaklah murah sedangkan untuk komputer dengan spesifikasi yang tidak terlalu tinggi akan kurang *reliable* dalam menangani data yang begitu besar (Venner, 2009) ([4]).
Untuk melakukan komputasi dengan data yang sangat besar, Google memberikan suatu metode yang dinamakan *MapReduce*. *MapReduce* melakukan komputasi dengan membagi beban komputasi dan diproses secara paralel atau bersama-sama (Dean, 2004) [4].

Terinspirasi oleh adanya *Google File System* (GFS) yang dikembangkan oleh Google yang digunakan untuk mengolah data mentah dengan jumlah yang sangat besar maka apache membuat *framework* berbasis java yang diberi nama Hadoop. Hadoop diciptakan oleh Doug Cutting dan Mike Cafarella pada tahun 2005.
Hadoop adalah sebuah *framework* berbasis java. Hadoop bekerja secara terdistribusi dengan 2 buah proses utama yaitu *MapReduce* dan *Hadoop Distributed File System* (HDFS). Hadoop memiliki kelebihan dapat secara cepat dan optimal didalam mengolah data yang sangat besar dengan kualitas *hardware* yang standar.

Random access memory (RAM) adalah memori tempat penyimpanan sementara pada komputer saat komputer dijalankan. RAM berfungsi untuk mempercepat pemprosesan data pada komputer, semakin besar ukuran RAM pada komputer maka pemprosesan data akan semakin cepat.
Text file merupakan dokumen yang biasanya digunakan untuk media pengolahan kata pada perangkat komputer. Terdapat berbagai macam jenis file yang dapat dibedakan berdasarkan formatnya.

Maka pada tugas akhir ini akan dilakukan analisis pengolahan *text file* pada hadoop cluster dengan memperhatikan kapasitas (RAM).

2. Landasan Teori

2.1 *Cluster Computer*

Cluster Computer merupakan kumpulan atau gabungan dari dua buah komputer atau lebih yang digabungkan menjadi satu bagian melalui sebuah jaringan berupa interkoneksi atau *Local Area Network (LAN)*. Secara fungsional gabungan dari komputer menjadi satu bagian namun secara fisik komputer-komputer terpisah satu dengan lainya. Pemanfaatan *cluster* biasanya untuk mendukung sebuah pekerjaan yang membutuhkan sumberdaya komputer dengan spesifikasi tinggi.

2.2 *Hadoop Cluster*

Hadoop merupakan salah satu *framework* berbasis java milik apache yang diciptakan oleh Doug Cutting dan Mike Cafarella pada tahun 2005. Hadoop berfungsi untuk mengolah data skala besar (*Big Data*) dengan kecepatan berkali-kali lipat dibandingkan dengan metode konvensional, tidak hanya data dengan ukuran *Gigabyte* dan *Terabyte* yang dapat diolah, namun data dengan ukuran *Petabyte* dapat diolah oleh Hadoop.

Hadoop bekerja dengan prinsip membagi-bagi skala data berukuran besar menjadi beberapa bagian kecil dan kemudian memproses data-data potongan kecil tersebut secara paralel.

Hadoop dapat bekerja pada sebuah komputer atau lebih (*cluster*). *cluster* adalah 2 buah komputer atau lebih yang saling terhubung

melalui sebuah jaringan. Maka Hadoop sering dikenal dengan istilah Hadoop cluster karena proses kerjanya pada sebuah *cluster*.
Hadoop memiliki 2 proses utama:

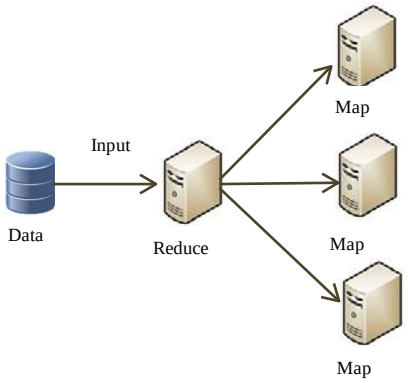
1. *MapReduce*
2. *Hadoop Distributed File System (HDFS)*

2.3 *MapReduce*

MapReduce pertama kali dikenalkan oleh Jeffrey Dean dan SanjayGhemawat dari Google,Inc. *MapReduce* adalah model pemograman terdistribusi yang digunakan untuk melakukan pengolaha data digunakan pengolahan data besar (Ghemawat, 2004). Mapreduce membagi input data menjadi beberapa potongan data, masing-masing ditugaskan sebagai map task yang dapat memproses data secara pararel.*MapReduce* didalam prosesnya dibantu oleh *Jobtracker* dan *Tasktracker* sehingga proses dapat berjalan dengan baik. *MapReduce* memiliki 2 tahapan utama, yaitu *Map* dan *Reduce.*

2.3.1 *Map Procedure*

Berikut ini merupakan gambaran arsitektur dari prosedur *Map* pada proses *MapReduce.*



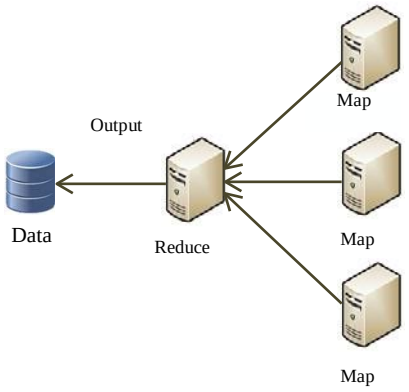
Gambar 2.1 *Map Architecture*

Pada proses ini *jobtracker* menerima atau membaca data masukan dalam bentuk pasangan *key/value* yang kemudian dipecah-pecah dengan ukuran tertentu, setelah dipecah-pecah data dibagikan kepada *tasktracker* untuk di proses dan disimpan didalam tempat-tempat penyimpanan yang tersedia secara terdistribusi. *Output* dari proses *map* ini adalah pasangan *key/value.*

Map memiliki identitas *value* dan *key* yang disebut dengan pasangan *intermediate*, yang berguna sebagai alamat ketika dalam proses *reduce.*

2.3.2 *Reduce Procedure*

Berikut ini merupakan gambaran arsitektur dari prosedur *Map* pada proses *MapReduce.*



Gambar 2.2 *Reduce Architecture*

Pada proses ini *jobtracker* menerima permintaan data kemudian memanggil para *tasktracker* (penyimpan data yang diminta) sesuai dengan pasangan *intermediate* <*value,key*>. setelah itu data-data dari *tasktracker* dikumpulkan menjadi satu secara terdistribusi untuk memberi jawaban dari data yang diminta.

2.4 *Hadoop Distributed File System (HDFS)*

Hadoop Distributed File System (HDFS) adalah *file* sistem terdistribusi yang berasal dari Hadoop. *File* sistem terdistribusi adalah *file* sistem yang bekerja dengan cara menyimpan data dengan membagi-bagi data kedalam ukuran tertentu dan menempatkan pada tempat penyimpanan yang berbeda di dalam sebuah *cluster*. Potongan-potongan data yang dibagi menjadi beberapa bagian disebut dengan HDFS blok. HDFS memiliki 2 buah komponen utama, yaitu *NameNode* dan *DataNode*.

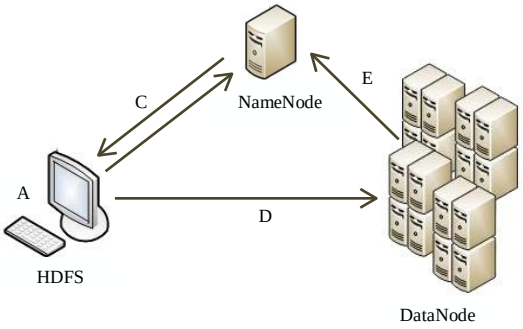
NameNode adalah sebuah komputer yang berperan sebagai kepala pada *cluster*, sedangkan *DataNode* merupakan komputer-komputer yang berperan sebagai anak buah pada *cluster*. Keberhasilan proses distribusi *file* sistem pada HDFS ini ditentukan oleh kinerja dari 2 buah komponen utama diatas.

Sebagai kepala, *NameNode* bertugas untuk mengatur penempatan data-data yang masuk untuk ditempatkan pada blok-blok yang tersedia pada *cluster* dan bertanggung jawab atas data-data tersebut. Sedangkan *DataNode* bertugas untuk menjaga blok-blok data yang sudah terisi data dan melaporkan kondisinya secara berkala kepada *NameNode*, kondisi ini disebut dengan *Heartbeat*.

HDFS memiliki 2 buah prosedur, yaitu menyimpan data dan membaca data.

2.4.1 Prosedur Menyimpan Data

Untuk prosedur menyimpan data harus ada sebuah komputer *client* yang terhubung dengan sebuah Hadoop cluster.



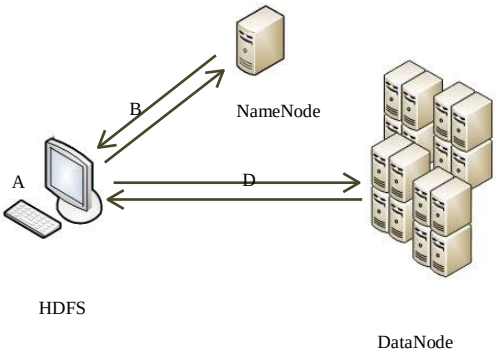
Gambar 2.6 Arsitektur Menyimpan Data

Langkah-Langkah prosedur menyimpan data sebagai berikut:

A. *User* memasukan perintah masukan pada komputer *client*
B. Komputer *client* berkomunikasi dengan *NameNode* memberitahu bahwa ada data yang akan disimpan dan menanyakan lokasi blok-blok tempat menyimpan data.
C. Komputer *client* mendapat jawaban dari *NameNode* berupa lokasi blok-blok untuk penyimpanan data.
D. Komputer *client* langsung berkomunikasi dengan *DataNode* untuk memasukan data-data pada lokasi blok-blok yang sudah diatur *NameNode*. Data sudah otomatis terbelah-belah sesuai dengan ukuran yang di *setting* sehingga dapat langsung menempati blok-blok yang sudah ditentukan.
E. *DataNode* memberikan laporan kepada *NameNode* bahwa data-data telah masuk dan menempati blok-blok yang sudah ditentukan oleh *NameNode*.

2.4.2 Prosedur Membaca Data

Untuk prosedur membaca data harus ada sebuah komputer *client* yang terhubung dengan sebuah Hadoop cluster.



Gambar 2.7 Arsitektur Membaca Data

Langkah-Langkah prosedur membaca data sebagai berikut:

A. *User* memasukan perintah untuk mengambil data pada komputer *client*.
B. Komputer *client* berkomunikasi dengan *NameNode* untuk menanyakan alamat *DataNode* penyimpan data yang diinginkan.
C. Komputer *client* mendapat jawaban dari *NameNode* berupa lokasi blok-blok tempat penyimpanan data yang inginkan.
D. Komputer *client* secara langsung berhubungan dengan *DataNode* untuk mengakses lokasi blok-blok tempat penyimpanan data yang diinginkan.
E. *DataNode* akan memberikan data yang diinginkan dan data secara otomatis akan ditampilkan pada layar komputer *client.*

2.5 Cloudera

Cloudera merupakan pelopor dari berdirinya sebuah hadoop distribusi didalam dunia teknologi, hadoop distribusi itu sendiri merupakan sebuah *software* yang menambahkan *tools-tools* serta konfigurasi khusus pada hadoop. Cloudera didirikan oleh orang-orang yang memiliki kontribusi didalam terciptanya hadoop apache, saat ini cloudera memiliki beberapa versi dari mulai versi gratis sampai versi berbayar.

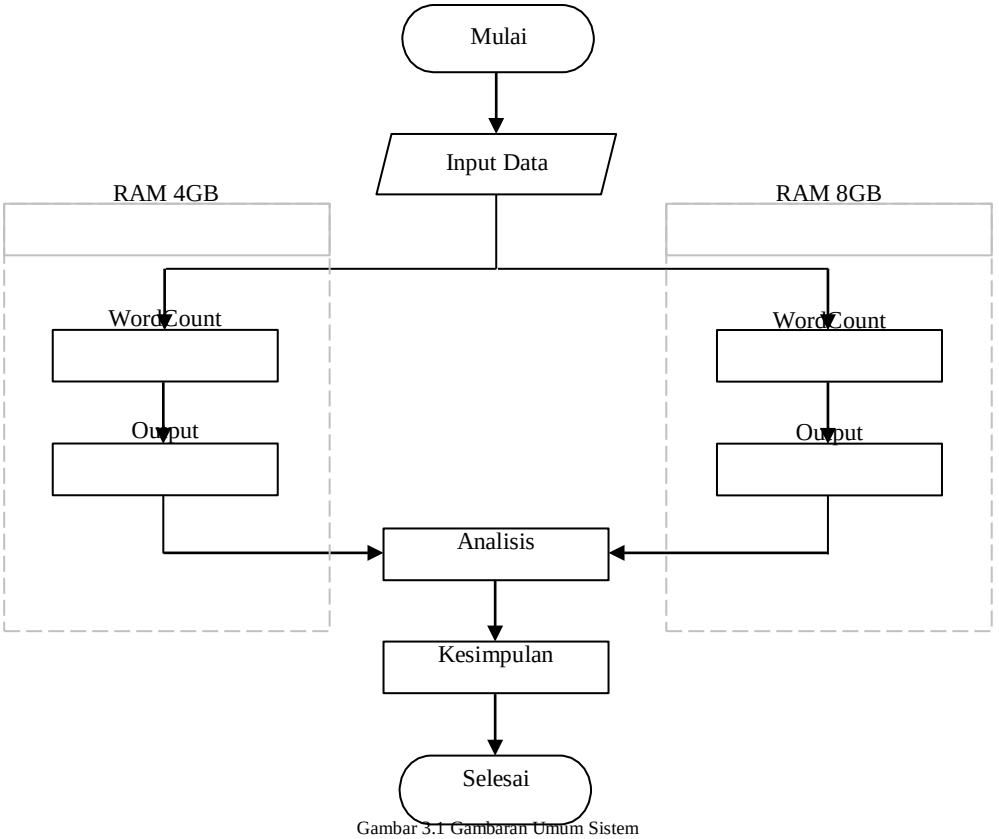
2.6 Wordcount

Wordcount merupakan salah satu *benchmark tools* yang dimiliki oleh hadoop yang berfungsi untuk melakukan perhitungan jumlah kata-kata yang sama dalam sebuah dokumen dengan skala besar.

3. Perancangan Sistem

Berikut ini merupakan gambaran umum sistem:

Tahap selanjutnya adalah instalasi hadoop dengan menggunakan Cloudera Manager, jika tahap instalasi hadoop sudah selesai maka proses selanjutnya adalah pengolahan data menggunakan *fitur* dari hadoop yaitu *wordcount* dengan menggunakan sebuah dokumen sebagai inputan. Setelah *wordcoun*t dilakukan didapatkan hasil berupa sebuah data yang selanjutnya akan dianalisis.



Gambar 3.1 Gambaran Umum Sistem

Perancangan sistem dimulai dengan pemilihan *hardware* dengan spesifikasi tertentu sesuai dengan kebutuhan *cluster*, spesifikasi pada setiap komputer didalam *cluster* harus memiliki spesifikasi yang sama agar *cluster* dapat berjalan dengan baik.

Tahap selanjutnya adalah proses *networking*, *networking* dilakukan pada komputer-komputer yang akan digabungkan menjadi sebuah *cluster* agar setiap komputer dapat terhubung kedalam sebuah jaringan dan dapat saling berkomunikasi satu dengan lainnya.

4. Pengujian Sistem dan Analisis

Pengujian merupakan tahap uji coba terhadap sistem yang sudah dibangun, pengujian dibutuhkan untuk mengetahui apakah sistem telah berjalan dengan lancar dan siap untuk digunakan. Dan analisis diperlukan untuk mendapatkan kesimpulan dari hasil ujicoba yang dilakukan pada sistem yang sudah melewati tahapan pengujian, didalam tugas akhir ini analisis difokuskan pada pengaruh RAM, sehingga yang menjadi parameter utama yang diperhatikan didalam analisis adalah RAM.

Pada tahapan pengujian sistem dan analisis ini akan menjelaskan bagaimana skenario pengujian dilakukan sampai dengan pengambilan hasil pengujian.

4.1 Pembagian Data

Pada proses pembagian data, data pada setiap *file* dibagi menjadi 5 bagian yang berbeda- beda ukuran (100MB, 200MB, 300MB, 400MB, 500MB) dan disesuaikan untuk kepentingan selanjutnya yaitu sekenario pengujian sistem. Berikut ini merupakan tabel pembagian data.

Pembagian Data					
No	Word	Pdf	Csv	Xlsx	Txt
1	100 MB	100 MB	100 MB	100 MB	100 MB
2	200 MB	200 MB	200 MB	200 MB	200 MB
3	300 MB	300 MB	300 MB	300 MB	300 MB
4	400 MB	400 MB	400 MB	400 MB	400 MB
5	500 MB	500 MB	500 MB	500 MB	500 MB

Tabel 4.1 Pembagian Data

4.2 Skenario Pengujian

Skenario pengujian merupakan gambaran bagaimana cara pengujian sistem akan dilakukan. Dalam tugas akhir ini pengujian dilakukan berdasarkan parameter utama yaitu kapasitas RAM, sekenario pengujian pada sistem dibagi menjadi 4 buah tahapan. Tahap pertama menggunakan RAM 2GB, tahap kedua menggunakan RAM 4GB, tahap ketiga menggunakan RAM 8GB dan terakhir menggunakan RAM 16 GB.

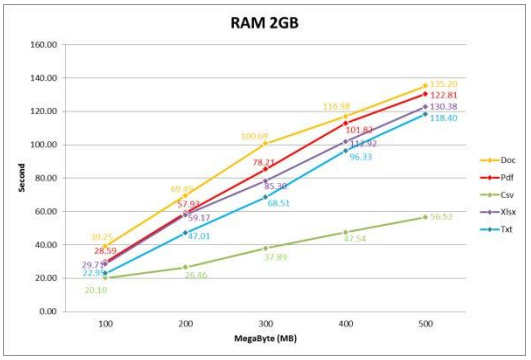
4.3 Analisis Hasil Percobaan

Berikut ini merupakan hasil dari tahap pengujian dengan menggunakan kapasitas RAM sebesar 2 GB.

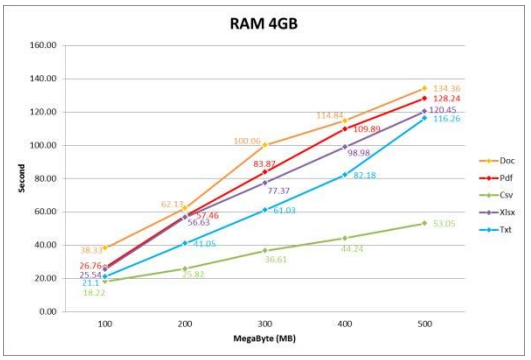
4.3.1 Hasil Percobaan Berdasarkan Kapasitas RAM

Pada seluruh hasil Percobaan file doc menjadi *file* dengan waktu eksekusi paling lambat, hal ini dikarenakan karakteristik *file* doc yang dapat mengandung semua tipe data sedangkan jenis *file* pdf memiliki waktu eksekusi lebih rendah dari doc karena *file* pdf merupakan *file* yang memiliki kompresi khusus sehingga jenis *file* ini tidak bisa di ubah-ubah seperti jenis *file* doc. Selanjutnya jenis *file* xlsx memiliki waktu eksekusi lebih cepat dari pdf dikarenakan *file* exel hanya mengandung matriks tidak seperti pdf yang dapat mengandung teks, huruf, citra dan grafik vektor dua dimensi. Selanjutnya jenis *file* txt memiliki waktu eksekusi lebih cepat dari xlsx karena karakteristik *file* txt yang hanya dapat mengandung tipe data string (array 1 dimensi), sedangkan *file* xlsx mengandung matrix (array multidimensi).

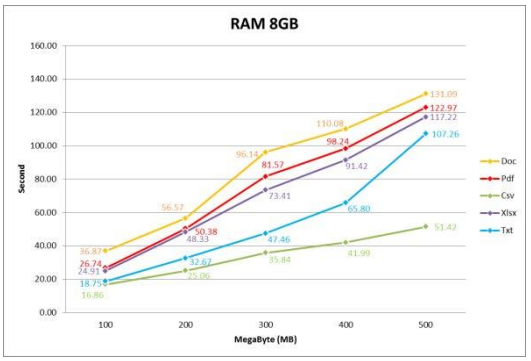
Jenis *file* yang memiliki waktu eksekusi tercepat adalah *file* csv, hal ini dikarenakan karakteristik dari *file* csv yang sederhana tidak memiliki kompresi khusus dan setiap karakter didalamnya hanya dipisahkan oleh koma (,) dan titik koma (;).



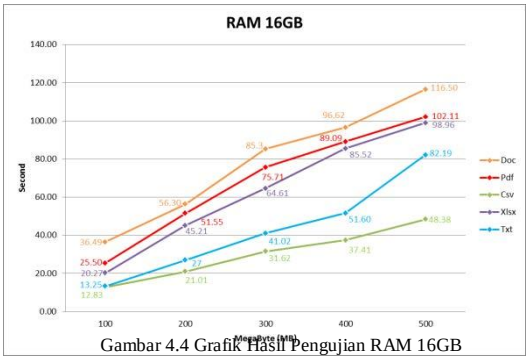
Gambar 4.1 Grafik Hasil Pengujian RAM 2GB



Gambar 4.2 Grafik Hasil Pengujian RAM 4GB



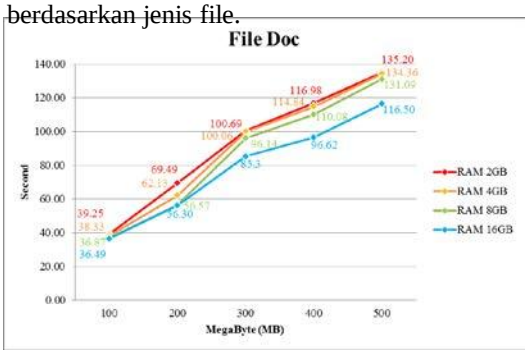
Gambar 4.3 Grafik Hasil Pengujian RAM 8GB



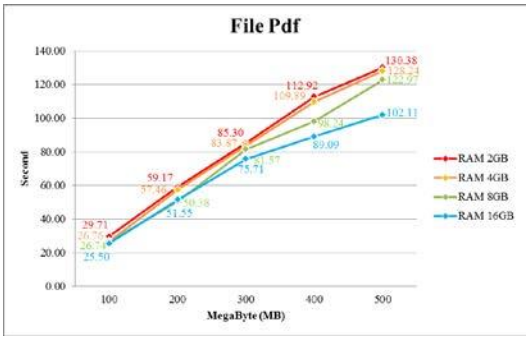
Gambar 4.4 Grafik Hasil Pengujian RAM 16GB

4.3.2 Hasil Percobaan Berdasarkan Jenis *File*.

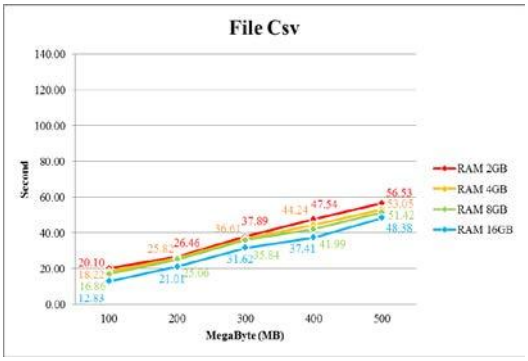
Berikut ini merupakan grafik hasil pengujian berdasarkan jenis file.



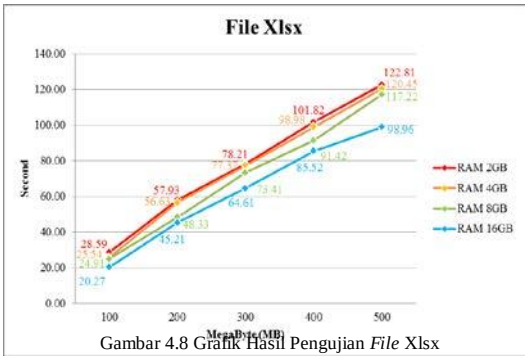
Gambar 4.5 Grafik Hasil Pengujian *File* Doc



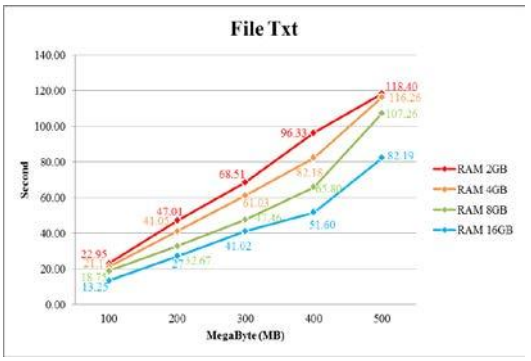
Gambar 4.6 Grafik Hasil Pengujian *File* Pdf



Gambar 4.7 Grafik Hasil Pengujian *File* Csv



Gambar 4.8 Grafik Hasil Pengujian *File* Xlsx

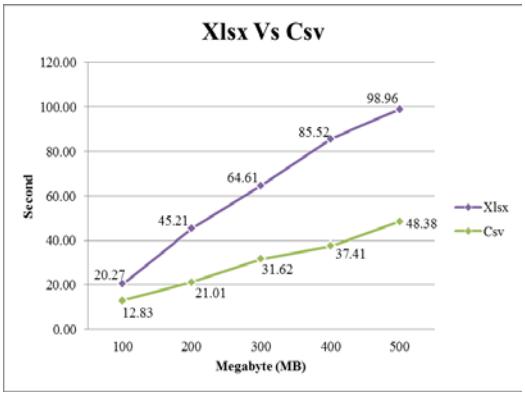


Gambar 4.9 Grafik Hasil Pengujian *File* Txt

Berdasarkan proses pengujian yang dilakukan didapatkan selisih waktu eksekusi antara kapasitas RAM 2GB, 4GB, 8GB dan 16GB, dari setiap peningkatan kapasitas RAM terjadi peningkatan waktu eksekusi, namun peningkatan yang terjadi berbeda-beda. Rata-rata peningkatan performansi ketika kapasitas RAM ditingkatkan 100% pada jenis *file* doc adalah sebesar 4,58%, *file* pdf adalah sebesar 7,57%, *file* csv adalah sebesar 8,87%, file xlsx adalah sebesar 8,35% dan *file* txt adalah sebesar 12,82%. Persentase ini masih jauh jika dibandingkan dengan peningkatan kapasitas RAM yang mencapai 100%.

4.4 *File* Csv Vs *File* Xlsx

Diantara 5 buah jenis *file* yang dilakukan percobaan terdapat 2 buah *file* yang berasal dari *software* yang sama namun berbeda jenis ekstensi yaitu, *file* Csv dan *file* Xlsx. Meskipun kedua jenis *file* ini memiliki isi dan ukuran data yang sama namun memiliki perbedaan waktu yang sangat signifikan.



Gambar 4.10 Grafik Hasil Pengujian Xlsx vs Csv

pada saat proses percobaan, *file* csv memiliki waktu eksekusi yang lebih cepat dibandingkan *file* xlsx, hal ini dikarenakan oleh karakter *file* csv yang menggunakan koma (,) dan titik koma (;) sebagai pemisah antar elemen sehingga mempermudah saat dilakukan pemrosesan pada data. Tidak seperti jenis yang terdiri dari berbagai macam kompleksitas fungsi.

Karena kesederhanaan karakter ini maka jenis *file* csv memiliki tingkat kompabilitas yang tinggi, hal ini telah dibuktikan dengan *file* csv memiliki waktu eksekusi yang lebih cepat dari *file* xlsx dan menjadikan *file* csv sebagai format standar dalam pengolahan data.

5. Penutup

5.1 Kesimpulan

Berdasarkan hasil analisis terhadap percobaan yang dilakukan pada sistem, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Hadoop Cluster telah berhasil diimplementasikan dengan menggunakan cloudera manager sebagai hadoop distribusinya.
2. Peningkatan 2 kalilipat kapasitas RAM pada hadoop cluster tidak membuat performansi meningkat menjadi 2 kalilipat, ketika kapasitas RAM ditingkatkan menjadi 100 % hasil percobaan menunjukan performansi dari jenis file doc mengalami peningkatan sebesar 4,58%, *file* pdf sebesar 7,57%, *file* csv sebesar 8,87%, file xlsx sebesar 8,35% dan *file* txt sebesar 12,82% hal ini dapat disebabkan oleh kompleksitas content yang berbeda-beda disetiap file.
3. Jenis *file* csv merupakan jenis *file* terbaik dari segi waktu eksekusi yang dapat diolah oleh hadoop cluster karena memiliki waktu eksekusi paling rendah diantara jenis *file* lainya.
4. Meskipun berasal dari *software* yang sama namun jenis *file* csv memiliki kualitas yang lebih baik dibandingkan dengan jenis file xlsx jika dilihat dari waktu eksekusi pada proses pengolahan data menggunakan hadoop cluster.

5.2 Saran

Pengembangan lebih lanjut yang dapat dilakukan terhadap tugas akhir ini adalah sebagai berikut :

1. Kapasitas RAM dan jumlah *slave node* dapat ditingkatkan pada pembangunan sistem selanjutnya agar mendapatkan hasil penelitian yang lebih optimal.
2. Pengembangan terkait pengaruh RAM dapat lebih dikembangkan dengan menggunakan *tools benchmark* yang lainya, seperti TeraSort, TestDFSIO dll.
3. Proses uji coba pada sistem dapat dikembangkan dengan menggunakan jenis *file* lainya, seperti gambar, video dan suara.

Daftar Pustaka

[1] Apache Hadoop. (2011). Retrieved October 30,2014, from Apache Software Foundation.: http://Hadoop.apache.org/

[2] B.He.W.Fang, Q.Luo, N.Govindaraju, and T.Wang. *Mars: a MapReduce framework on graphic processors.* ACM 2008.

[3] D.Borthakur. The Hadoop Distributed File System: Architecture and Design. The Apache Software Foundation, 2007.

[4] Gusti Dading Zainul: Mapreduce Distributed Programming Using Hadoop Framework, 2012. Informatics Engineering of Institut Teknologi Surabaya of Indonesia.

[5] Huang, S., & Huang, J. 2009. The HiBench Benchmark Suite: Characterization of the Mapreduce -Based Data Analysis. Intel China Software Center, Shanghai,P.R. China

[6] jiang,Dawei, Chin Ooi, Beng, dkk. 2009. The Performance of Mapreduce : An In-

Depth Study. School of Computing National University of Singapore

[7] M.Rafique, B.Rose, A.Butt, and D.Nikolopoulos. Supporting mapreduce on large-scale asymmetric multi-core clusters. *SIGOPS Oper. Syst. Rev.*, 43(2):25–34, 2009.

[8] http://data.gov.uk/dataset/road-accidents-safety-data.
Tanggal Akses 5 Juni 2015

[9] https://ianspace.wordpress.com/2011/02/22/jenis-%E2%80%93-jenis-file-dokumen/
Tanggal Akses 1 juli 2015

[10] http://pandusolusi.com/hadoop-adalah.htm
Tanggal Akses 21 Juni 2015

[11] https://azerdark.wordpress.com/2009/03/23/csv-comma-separated-value/
Tanggal akses 3 Agustus 2015