JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

ANALISIS KINERJA ALGORITMA DELAY SCHEDULING PADA HADOOP TERHADAP KARAKTERISTIK RESPONS TIME UNTUK PENGIRIMAN 2 JOB YANG BERBEDA

WAHYUDDIN SAPUTRA[1], K.TONE[2] , ALVIANUS DENGEN[3]
Jurusan Teknik Informatika[1,2],
Fakultas Sains dan Teknologi, Universitas Islam Negeri Alauddin Makassar
Jurusan Teknik Elektro[3]
Universitas Teknologi Sulawesi
Email : wahyuddin.saputra@uin-alauddin.ac.id[1] , ktone57@yahoo.co.id[2]
alvianusdengen@utsmakassar.ac.id[3]

ABSTRAK

Perkembangan teknologi sistem informasi sangat pesat yang berbanding lurus dengan pertumbuhan yang sangat besar. Data tersebut sudah sangat sulit untuk dikoleksi, disimpan, dikelola maupun dianalisa, dibutuhkan infrastruktur dan teknologi komputer yang disebut *Parallel computing.* Parallel computing adalah penggunaan beberapa komputer yang saling terhubung untuk mengolah data dalam ukuran yang besar dengan menggunakan Hadoop. merupakan *platform* untuk mengolah data yang berukuran besar *(big data*) secara terdistribusi dan dapat berjalan diatas cluster. Pada penelitian ini menggunakan metode FIFO dan *Delay Scheduling* pada *Hadoop* sebagai *job scheduler*. Kedua algoritma nantinya dibandingkan dan diimplementasikan pada berbagai jenis karakteristik *job* dengan menggunakan parameter *respons time* sebagai acuan perhitungan performansi sistem terhadap karakteristik tugas dan jumlah tugas yang dijalankan pada masing - masing *job scheduler*.

Kata Kunci : Analisis, *Hadoop*, *Delay Scheduling*.

I.PENDAHULUAN

Seiring dengan berkembangnya teknologi yang ada, maka ukuran data yang diolah juga semakin besar. Data yang semakin besar tersebut sudah sangat sulit untuk dikoleksi, disimpan, dikelola maupun dianalisa dengan menggunakan sistem *database* biasa dikarenakan ukurannya yang terus bertambah.

Data yang memiliki ukuran yang semakin besar tentunya membutuhkan tempat penyimpanan yang sangat besar dan sistem pengelolaan yang tepat agar

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

mudah dalam mengolahnya. Ada beberapa platform yang dapat digunakan untuk menyimpan dan mengolah data yang berukuran besar (*big data*) antara lain 1010data, *Actian*, *Amazon Web Services* (AWS), *Cloudera*, IBM *SmartCloud*, *Rackspace*, dan lain-lain. Untuk dapat menyimpan dan mengolah data yang berukuran besar (*big data*) secara baik dan cepat dibutuhkan teknologi komputer yang khusus yang disebut *high performance computer* atau super computer, tetapi untuk membangun suatu sistem super computer tersebut membutuhkan biaya yang tidak murah. Untuk mengatasi masalah ini, maka *platforms* yang digunakan untuk menyimpan dan mengolah big data menggunakan sebuah sistem yang disebut *parallel computing*. *Parallel computing* adalah penggunaan beberapa komputer yang saling terhubung untuk mengolah data dalam ukuran yang besar. Salah satu *platform* yang masih sering digunakan sampai saat ini untuk mengolah data yang berukuran besar *(big data*) secara terdistribusi dan dapat berjalan diatas cluster adalah *Hadoop*.

Hadoop merupakan *framework software* berbasis java dan open-source yang berfungsi untuk mengolah data yang besar secara terdistribusi dan berjalan diatas cluster yang terdiri atas beberapa komputer yang saling terhubung. *Hadoop* mempunyai kelebihan dari segi ekonomi karena tidak berbayar dan dapat diimplementasikan pada perangkat keras dengan spesifikasi yang tidak terlalu tinggi. arsitektur *Hadoop* terdiri dari dua layer yaitu layer *MapReduce* dan *layer Hadoop Distributed File System* (HDFS). *MapReduce* merupakan *framework* dari aplikasi yang terdistribusi sedangkan *Hadoop Distributed File System* (HDFS) merupakan data yang terdistribusi.

II. METODE PENELITIAN

Hadoop merupakan *framework* software berbasis Java dan open source yang berfungsi untuk mengolah data yang memiliki ukuran yang besar secara terdistribusi dan berjalan diatas cluster yang terdiri dari beberapa komputer yang saling terhubung (parallel computing). Berdasarkan *Hadoop* dapat mengolah data dalam jumlah yang sangat besar hingga petabyte (1 petabyte = 10245 bytes) dan dijalankan di atas ratusan bahkan ribuan komputer. *Hadoop* dibuat oleh Doug

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Cutting yang pada asalnya *Hadoop* ini adalah sub project dari Nutch yang digunakan untuk *search engine*. *Hadoop* bersifat open source dan berada di bawah bendera Apache *Software Foundation*.

1. **Arsitektur *Hadoop***

Hadoop terdiri dari *common Hadoop* yang berguna dalam menyediakan akses ke dalam file *system* yang didukung oleh *Hadoop*. *Common Hadoop* ini berisi paket yang diperlukan oleh JAR file, skrip yang dibutuhkan untuk memulai *Hadoop* dan dokumentasi pekerjaan yang telah dilakukan oleh *Hadoop*.

Bedasarkan inti dari *Hadoop* adalah terdiri dari:

1. *Hadoop* Distributed File *System* (HDFS) →Untuk data yang terdistribusi.
2. *MapReduce* → *Framework* untuk aplikasi dan programing yang

Sebuah *cluster* kecil pada *Hadoop* dapat terdiri dari satu *master* node dan beberapa *slave* node. *Master* node ini terdiri dari *NameNode* dan *JobTracker*, sedangkan *slave* node terdiri dari *DataNode* dan *TaskTracker*. *Hadoop* membutuhkan JRE 1.6 atau JRE dengan versi yang lebih tinggi. Dalam menjalankan dan menghentikan sistem pada *Hadoop* dibutuhkan ssh yang harus dibentuk antar *node* pada sebuah *cluster*.

2. HDFS

Hadoop Distributed File System (HDFS) merupakan file *system* berbasis Java yang terdistribusi pada *Hadoop*. Sebagai file *system* terdistribusi, HDFS berguna untuk menangani data dalam jumlah besar yang disimpan dan tersebar didalam banyak komputer yang berhubungan yang biasa disebut dengan *cluster*. File *system* terdistribusi pada *Hadoop* dapat diartikan sebagai file *system* yang menyimpan data tidak dalam satu Hard Disk Drive (HDD) atau media penyimpanan lainnya, tetapi data dipecah-pecah (file dipecah dalam bentuk block dengan ukuran 64 MB – bisa dikonfigurasi besarnya) dan disimpan tersebar dalam suatu cluster yang terdiri dari beberapa computer.HDFS menyimpan suatu data dengan cara membelahnya menjadi potongan potongan data yang berukuran 64 MB(default) dan potongan-potongan data tersebut kemudian disimpan tersebar dalam setiap node

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

yang membentuk clusternya. Potongan-potongan data tersebut didalam HDFS disebut block.Ukuranblock pada setiap file tidak terpaku harus 64 MB, dimana ukuran block tersebut dapat disesuaikan dengan keinginan user. Meskipun data yang ada disimpan secara tersebar ke beberapa node, namun dari kacamata user, data tersebut tetap terlihat seperti halnya kita mengakses file pada satu komputer. File yang secara fisik tersebar dalam banyak komputer dapat diperlakukan layaknya memperlakukan file dalam satu computer.

3. *MapReduce*

MapReduce adalah sebuah *framework* untuk aplikasi dan programming yang diperkenalkan oleh Google dan digunakan untuk melakukan suatu pekerjaan dari komputasi terdistribusi yang dijalankan pada sebuah cluster. *MapReduce* ini terdiri dari konsep fungsi map dan reduce yang biasa digunakan pada *functional programming*. Salah satu program yang menggunakan konsep *MapReduce* yang telah disediakan oleh *Hadoop* adalah *Wordcount*. *Wordcount* merupakan program yang bertujuan untuk menghitung kata pada file plaintext. Proses *MapReduce* pada *Wordcount* ini dibagi menjadi 2 tahap yaitu proses mapping dan reducing.

4. ***FIFO Scheduling***

Algoritma FIFO merupakan *default* yang digunakan oleh *Hadoop* pada proses penjadwalan. Algoritma ini mengatasi permasalahan antrian pada *job* dengan cara menjalakan sebuah *job* yang datang untuk pertama kali. Algoritma FIFO tidak menangani adanya skala prioritas dan perhitungan *long jobs* atau *short jobs*. Sehingga mengakibatkan penggunaan algoritma FIFO kurang efektif. Pada implementasi algoritma FIFO dalam server berskala besar, algoritma FIFO dapat menurunkan performansi dari sisi server terutama pada sistem yang mempunyai layanan *sharing data* pada *multiple-user.*

5. ***Delay Scheduling***

Algoritma *Delay Scheduling* mempunyai struktur yang berbeda dibandingkan dengan penjadwalan pada konfigurasi *default* dari sebuah *Hadoop* sistem. Pada konfigurasi *default* dari sebuah *Hadoop* adalah memakai algoritma *FIFO* sebagai teknik utama penjadwalan.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Delay Scheduling tidak menggunakan mekanisme *FIFO* yang memindahankan data pada *virtual hard disk* yang ada pada *Hadoop* dan memerlukan sinkronisasi terhadap data yang dipakai. Sehingga beberapa perpindahan *Resource* ini membuat *job* mengalami *fail* dan dilakukan pengulangan *job* tidak digunakan karena pada awal *job* dibuat, *Delay Scheduling* sudah membagi *Resource* data terhadap *pool* yang sesuai dengan *Resource* data yang ada pada *virtual hard disk Hadoop* karena konfigurasi *pool* merupakan karakterisktik dari *Fair Scheduler* yang dimodifikasi dengan algoritma *Delay Scheduling*. Selain itu *Delay Scheduler* akan menggunakan metode menunda jalannya *jobs* selanjutnya untuk memperbaiki data lokalitas sebelumnya. Sehingga dapat meminimalkan *Response Time* dan memaksimalkan *Job Throughput*.

6. Skenario Pengujian

Tabel 1. Skenario Pengujian

Jenis Job	FIFO dan Delay scheduling					
Wordcount & Grep	Skenario 1	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 jobs
Wordcount & RandomTextwriter	Skenario 2	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 jobs
Grep & RandomTextWriter	Skenario 3	5 *jobs*	10 *jobs*	20 *jobs*	50 *jobs*	50 jobs

Tabel di atas merupakan skenario pengujian yang akan dilakukan dimana pada proses pengiriman job menggunakan tiga jenis *job* yaitu *job wordcoun*, *job grep, job randomtextwriter*, dengan menggunakan kombinasi 2 jenis *job* yaitu antara *job wordcount* dengan *job grep, job wordcount* dengan *job randomtextwriter*,dan *job grep* dengan *job randomtextwriter*.

Semua skenario terdiri dari lima *virtual* user yang jumlah *job*-nya diberikan tidak harus sama setiap user karena jumlah user tidak mempengaruhi performansi scheduling melainkan jumlah *job*. Adapun *resource data* yang digunakan sebagai berikut:

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Table 2. R*esource* data

Data	Sumber	Size
1	facebook-names-withcount.txt	2,3 GB
2	facebook-names-unique.txt	1,5 GB
3	wikipedia-wordlist-sraveau-20090325.txt	700 MB
4	facebook-f.last.txt	155 MB
5	facebook-firstnames-withcount.txt	70 MB

7. Teknik Analisis Data

Semua skenario terdiri dari lima *virtual* user yang jumlah *job*-nya diberikan tidak harus sama setiap user karena jumlah user tidak mempengaruhi performansi scheduling melainkan jumlah *job*. Hal ini dilakukan untuk dapat meneliti antara antrian yang memiliki *job* berjenis sama dan antrian yang memiliki jenis *job* yang berbeda yang penempatan jenis *job* dilakukan secara acak. Adapun perhitungan parameter pengujian sebagai berikut:

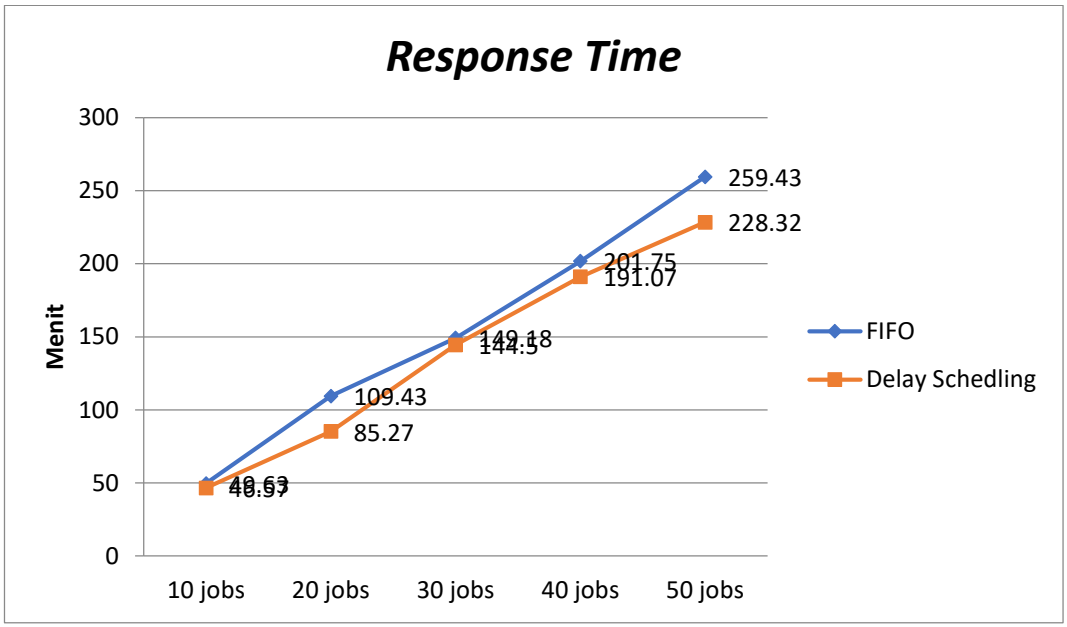
Pada parameter ini akan diukur waktu yang digunakan oleh *server* untuk menyelesaikan *job* yang diberikan pada satu antrian, satuan yang dipakai adalah menit.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

III.HASIL DAN PEMBAHASAN

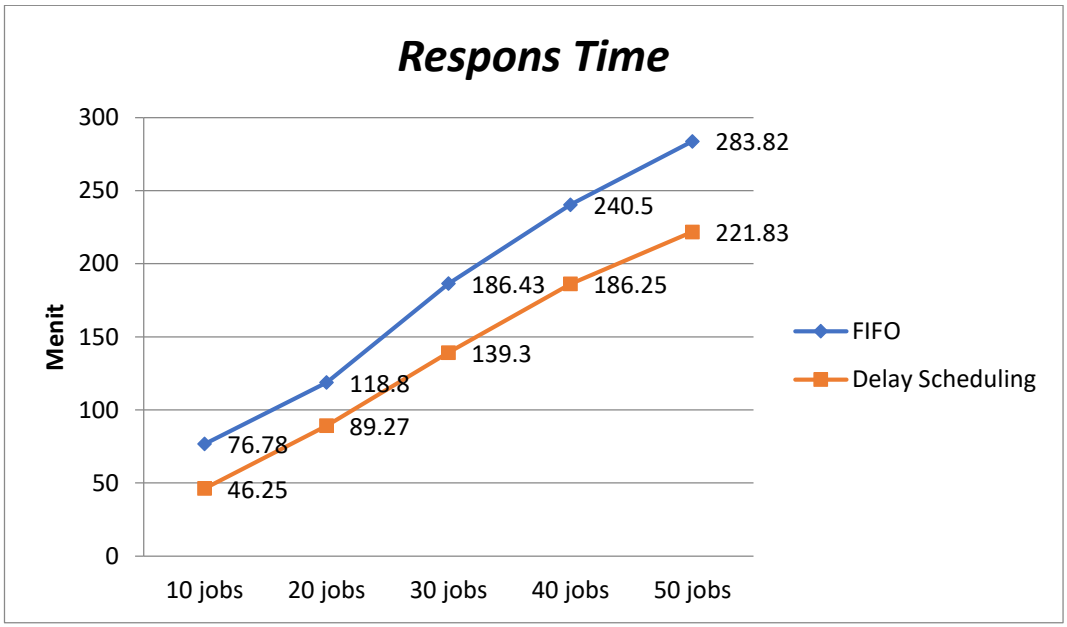
A.Analisis Yang Telah Dilakukan

a. Pengiriman Job Wordcount & Grep



Grafik 1. *Respons Time* Skenario 1

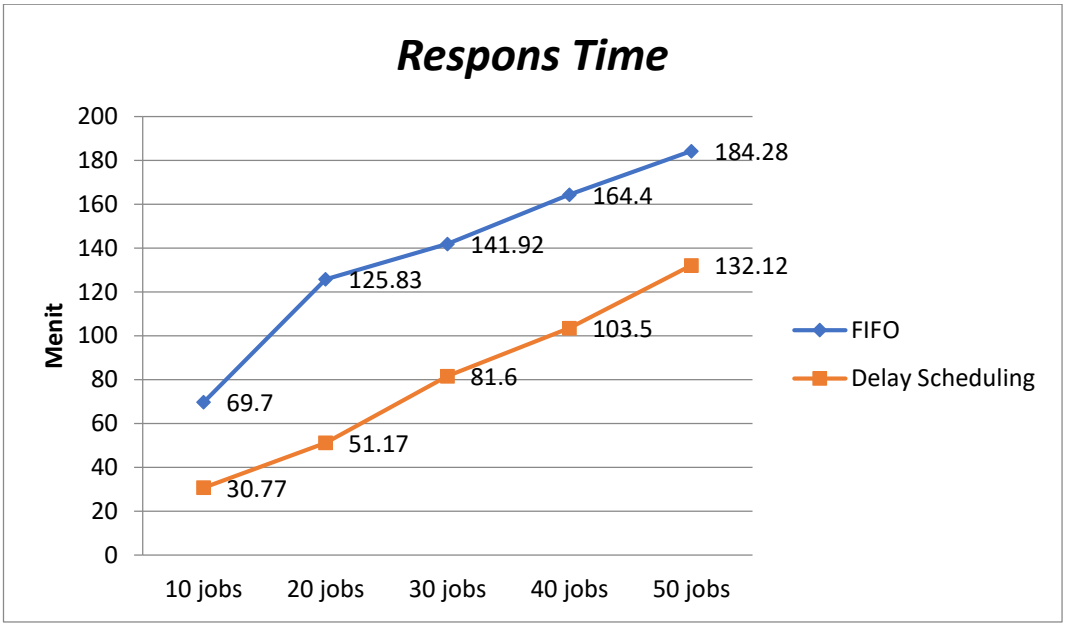
b. Pengiriman Job Wordcount & RandomTextwriter



Grafik 2. *Respons Time Skenario 2*

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

c. Pengiriman Job Grep & RandomTextWriter



Grafik 3. *Respons Time* Skenario 3

B.Pembahasan

Pada grafik 1 parameter *Response Time* pada algoritma *Delay Scheduling* memiliki ciri menjadi semakin lama jika jumlah *job* bertambah. Akan tetapi karakteristik algoritma *Delay Scheduling* yaitu memiliki waktu tambah dalam penundaan *job* untuk memperkecil nilai *fail* pada *job* tetap berlaku. Selain itu karakteristik dari jenis *job* grep yang memiliki jumlah *job* dua yaitu *grep search* dan *grep-sort*. Dimana grep-sort akan masuk antrian *job*s setelah *grep search* selesai dieksekusi dan membutuhkan waktu tambah dalam proses *grep sort* masuk kedalam antrian. Sehingga dalam penggunaan algoritma *Delay Scheduling* ,jumlah *job*s mengurangi waktu pada parameter *Response Time*. Hal ini diperkuat pada grafik 12 pada penggunaan 50 job mengalami selisih 31 menit 11 detik dari FIFO.

Pada grafik 2 parameter Response Time pada algoritma Delay Scheduling memiliki ciri menjadi semakin sedikit jika dibandingkan dengan FIFO. Hal ini

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

dapat terjadi karena karakteristik algoritma Delay Scheduling yaitu memiliki waktu tambah dalam penundaan job untuk memperkecil nilai fail pada job. Sehingga dalam penggunaan algoritma Delay Scheduling, penambahan jobs yang dilakukan akan mengakibatkan waktu eksekusi yang bertambah dari kedua algoritma, walaupun demikian respons time dari algoritma delay scheduling lebih sedikit dibanding algoritma FIFO untuk pengiriman job sebesar 50 terlihat pada delay scheduling lebih cepat 61 menit 99 detik dari FIFO. Hal ini diperkuat dengan grafik 15 pada algoritma Delay Scheduling pada semua jumlah jobs menunjukkan nilai yang lebih kecil dari FIFO.

Pada grafik grafik 3 parameter *Response Time* pada algoritma *FIFO* memiliki ciri menjadi semakin lama jika jumlah *job* bertambah. Hal ini dapat terjadi karena proses kerja dari FIFO yang menyelesaikan job berdasarkan waktu pengiriman sehingga job yang kedua akan dijalankan apabila job pertama telah selesai dikerjakan selain itu karakteristik dari jenis *job* grep yang memiliki jumlah *job* dua yaitu *grep search* dan *grep-sort*. Dimana *grep-sort* akan masuk antrian *job*s setelah grep search selesai dieksekusi dan membutuhkan waktu tambah dalam proses grep sort masuk kedalam antrian walau demikian penggunaan algoritma *delai shcduling* menunjukkan nilai *respons time* yang lebih kecil dari pada algoritma FIFO. Hal ini diperkuat pada jumlah *job* 50 *job*s waktu *Response Time* lebih cepat 52 menit 16 detik dengan menggunakan algoritma *Delay Scheduling* untuk penggunaan dua jenis *job* yang berbeda.

IV.KESIMPULAN

Pada pengukuran *respons time* untuk kedua algoritma menunjukkan penambahan jumlah job akan mengakibatkna nilai *respons time* bertambah pada masing-masing algoritma meski demikian algoritma *delay scheduling* menghasilkan *respons time* yang lebih sedikit daripada algoritma FIFO sehingga dapat meningkatkan nila *throughput* pada algoritma *Delay Scheduling.*

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI

Volume 5 Nomor. 1, April 2020
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

DAFTAR PUSTAKA

Arslan Engin, Shekhar mrigank, and kosar Tevfik .2014.Locality and Network-Aware Reduce Task Scheduling forData-Intensive Applications.DataCloud 2014.

Apache TM *Hadoop* @ homepage. http://*Hadoop*.apache.org/. Diakses 17 Oktober 2013.

Chuck Lam. (2011). *Hadoop* In Action. Stamford: Mainning Publications Co.

Colin White. (2012, January). *MapReduce* and the Data Scientist. BI Research.

Dima May. (2012). *Hadoop* Distributed File *System* (HDFS) Overview. coreservlets.com.

Magang Industri. (2013). Definisi Cloud Computing. Meruvian.org Cloud Computing.

Priagung Khusumanegara. 2014. *Analisis Performa Kecepatan MapReduce pada Hadoop Menggunakan TCP Packet Flow Analysis.* Universitas Indonesia

Rasooli Aysan, Douglas G. Down.(2011). Guidelines for Selecting *Hadoop* Schedulers based on *System* Heterogeneity, Hamilton.

Thirumala Rao. B, Reddy. Dr. L.S.S.2011.*Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environment.*International Jurnal of Cumputer Application (0975-8887) volume 34- No.9, November 2011

Wang Yiantian, Rao Ruonan, and Wang Yingling. 2014. *A Round Robin with Multiple Feedback Job Sceduler in Hadoop.*978-1-4799-3 /114. 2014 IEEE.