Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 **Link : https://dx.doi.org/10.29408/jit.v4i2.3565**

Analisa Performa Klastering Data Besar pada Hadoop

Hadian Mandala Putra[1*], Taufik Akbar[2], Ahwan Ahmadi[3], Muhammad Iman Darmawan[4]
[1,2,3]Program Studi Teknik Komputer, Universitas Hamzanwadi
[4]Program Studi Teknik Lingkungan, Universitas Hamzanwadi
hadian_mandala@hamzanwadi.ac.id

Abstrak

Data Besar adalah suatu kumpulan data dengan ukuran besar dan komplek, terdiri dari berbagai tipe data serta diperoleh dari berbagai macam sumber, berkembang pesar dalam waktu yang singkat. Beberapa masalah yang akan muncul ketika mengolah data besar antara lain terkait dengan penyimpanan dan pengaksesan dari data besar yang terdiri dari berbagai tipe data dengan kompleksitas yang tinggi yang tidak mampu ditangani oleh model relasional. Salah satu teknologi yang mampu mengatasi masalah penyimpanan dan pengaksesan data besar yaitu *Hadoop*. *Hadoop* adalah teknologi yang mampu menyimpan dna memproses data besar dengan cara mendistribusikan data besar ke dalam beberapa partisi data (blok-blok data). Masalah timbul apabila suatu proses analisis memerlukan seluruh data yang tersebar menjadi satu entitas data, misalnya pada proses klatser data (*clustering*). Salah satu alternatif penyelesaiannya adalah dengan melakukan analisis secara paralel dan tersebar, kemudian melakukan analisis secara terpusat dari hasil analisis tersebar. Penelitian ini mengkaji dan menganalisis metode *K-Medoids* Mapreduce dan algoritma K-Medoids dan K-Modes sebagai pembanding algoritma. Dataset yang digunakan adalah dataset tentang mobil yang terdiri dari 3.5 juta baris data dengan ukuran 400MB yang disimpan secara terdistribusi pada teknologi penyimpanan *Hadoop*. *Hadoop* memiliki fitur *mapreduce,* terdirii dari 2 fungsi yaitu *map* dan *reduce.* Fungsi map melakukan seleksi untuk mengambil pasangan *key, value* dan mengembalikan nilai berupa koleksi pasangan *key, value*, selanjutnya fungsi *reduce* akan menggabungkan keseluruhan pasangan *key,value* dari beberapa fungsi map. Hasil evaluasi kualitas cluster diuji dengan menggunakan metrik pengujian *Silhouette Coefficient.* Algoritma *K-Medoids MapReduce* untuk dataset mobill memberikan nilai silhouette sebesar 0.99 dengan jumlah 2 cluster.

Kata kunci: Data Besar, *Hadoop, Mapreduce, Clustering*

Abstract

Big Data is a collection of data with a large and complex size, consisting of various data types and obtained from various sources, overgrowing quickly. Some of the problems that will arise when processing big data, among others, are related to the storage and access of big data, which consists of various types of data with high complexity that are not able to be handled by the relational model. One technology that can solve the problem of storing and accessing big data is Hadoop. Hadoop is a technology that can store and process big data by distributing big data into several data partitions (data blocks). Problems arise when an analysis process requires all data spread out into one data entity, for example, in the data clustering process. One alternative solution is to do a parallel and scattered analysis, then perform a centralized analysis of the results of the scattered analysis. This study examines and analyzes two methods, namely K-Medoids Mapreduce and K-Modes without Mapreduce. The dataset used is a dataset about cars consisting of 3.5 million rows of data with 400MB distributed in a Hadoop Cluster (consisting of more than one engine). Hadoop has a MapReduce feature, consisting of 2 functions, namely map and reduce. The map function performs a selection to retrieve a key, value pairs, and returns a value in the form of a collection of key value pairs, and then the reduce function combines all key value pairs from several map functions. The results of the cluster quality evaluation are tested using the Silhouette Coefficient testing metric. The K-Medoids MapReduce algorithm for the car dataset gives a silhouette value of 0.99 with a total of 2 clusters.

Keywords: Big Data, Hadoop, Mapreduce, Clustering

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

1. Pendahuluan

Data merupakan suatu catatan berupa kumpulan fakta yang terjadi di dunia nyata yang diperoleh melalui suatu pengamatan atau melalui pencarian ke beberapa sumber tertentu. Data dapat disimpan untuk memudahkan seseorang dalam membuat, mengolah, memproses, memanipulasi, dan menganalisis data yang telah ada [1]. Dalam penyimpanan dan pengolahan, terdapat metode dan teknik yang digunakan untuk mempermudah penyimpanan dan pengolahan data, yaitu dengan model relasional yang menyimpan data dalam bentuk baris dan kolom.

Perkembangan data yang terus berkembang dengan berbagai keragaman tipe data, menjadikan model relasional tidak lagi mampu menangani keberagaman jenis data yang ada. Pertumbuhan data yang pesat secara terus menerus dikenal dengan istilah data besar (*big data*) [2]. Data Besar terdiri dari volume yang besar, variasi tipe data, data dihasilkan dalam waktu yang cepat, dan termasuk perubahan jumlah ukuran data yang terus berkembang. Mapreduce adalah salah satu metodologi pada big data yang bekerja secara efisien untuk membagi dan mempartisi data, selanjutnya dapat digunakan untuk mengambil keputusan yang tepat dari hasil analisis teknik *clustering*. Metode *clustering* akan secara efisien mengelola sistem basis data skala besar, tujuan utama klister adalah mengelompokkan data yang serupa, yang dapat diproses dengan mudah dan untuk kebutuhan organisir data [3].

Banyak teknologi serta algoritma yang dapat digunakan untuk menyimpan, memproses dan menganalisis data besar. Salah satu teknologi yang berkembang berdasarkan karakteristik data besar adalah *Hadoop*. Hadoop mendukung penggunaan cluster untuk penyimpanan setiap data dengan harga yang murah, dan setiap *node* penyimpanan dapat berdiri sendiri. Dalam data besar, data besar akan diproses berdasarkan mapreduce, data pertama kali akan dikumpulkan, disimpan dalam sistem terdistribusi seperti *HDFS* (*Hadoop Distributed File System*). Ketika data ingin dianalisis menjadi satu entitas yang utuh, maka data akan dibaca pada *node* penyimpanan untuk menentukan data yang akan diakses dan kebutuhan penggunan node yang diperlukan untuk analisis data [4].

Clustering menjadi solusi yang dapat diandalkan ketika membutuhkan analisis dari keselurahan data yang terdistribusi menjadi satu kesatuan entitas. Penelitian ini menggunakan *clustering* sebagai solusi dari metode untuk menangani masalah data besar. Metode *clustering* yang digunakan adalah K-Medoids dengan Mapreduce sebagai baseline penelitian yang akan dibandingkan dengan K-Medoids tanpa Mapreduce. Keadaan yang berbeda diterapkan pada penelitian ini untuk melihat performa dari masing-masing algoritma, yaitu data disimpan

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

pada HDFS, dan disimpan secara lokal diluar HDFS.

2. Tinjauan Pustaka

2.1. Penelitian Terkait

Beberapa penelitian terkait dengan *clustering* pada data besar sebagian besar berfokus pada performa dan skalabilitas dari algoritma yang digunakan. Banyak dari algoritma yang digunakan dalam menangani data besar tidak efisien sehingga tidak memenuhi untuk kebutuhan analisis data. Menurut [5], dalam penelitiannya terkait penyimpanan data tenaga listrik pada negara China menggunakan teknologi HDFS dan mampu menangani perkembangan data yang berkembang sangat besar secara efisien ke dalam beberapa cluster. [6] dalam penelitiannya memaparkan penggunaan raspberry pi dalam penanganan data besar pabrik pintar dalam menghasilkan, menyimpan, dan menganalisa data besar menggunakan teknologi HDFS mampu menangani data yang tercipta secara efisien. [7] meneliti tentang clustering terkait data imunisasi pada puskesmas dengan membandingkan 2 algoritma, yaitu K-Means menghasilkan nilai silhouette -0.018 dan Fuzzy C-Means dengan nilai silhouette 0.129. Untuk kasus cluster pada penelitian [7] nilai silhouette yang rendah dapat dipengaruhi algoritma yang tidak sesuai terhadap dataset.

Penelitian terkait *clustering* tidak selalu mampu memetakan *cluster* data dengan bagus, terlebih dalam menangani data dengan ukuran besar. Oleh karena itu diperlukan teknologi dan metode yang sesuai yang terhadap data yang akan diolah dan dianalisis.

2.2. Landasan Teori

1. Data Besar

Data besar memiliki anuran yang sangat besar dan kompleks yang beragam, dengan demikian perlu dibangun cluster yang mengumpulkan, memfilter, serta mengirimkan data terstruktur atau tidak terstruktur dalam jumlah besar secara tepat waktu. Data besar diperoleh dari berbagai sumber. Pada tahun 2011, Gartner sebuah perusahaan penasihat dan penelitian teknologi informasi, mendefinisikan data besar sebagai tantangan pertumbuhan data yang berfokus pada ukuran data, variasi data, serta kecepatan data dihasilkan[6][8].

2. *Hadoop*

Hadoop adalah salah satu teknologi yang memungkinkan untuk mengumpulkan dan menjalankan data besar secara terdistribusi dalam pengaturan jaringan komputer dengan model pemrograman yang sederhana. Keseluruhan proses dapat meningkat dari mesin tunggal hingga ribuan mesin tergantung pada jumlah yang disimpan. *Hadoop* sendiri (*master-node* dan *slaves-node*) dapat berdiri sendiri pada

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

satu mesin atau dapat berjalan secara simultan pada beberapa mesin (*master-node* terpisah dengan *slaves-node*) untuk menyimpan dan mengolah data [9].

Hadoop terdiri dari 2 modul utama, yaitu *Hadoop Distribution File System* (*HDFS*) dan MapReduce. *HDFS* berfungsi sebagai tempat penyimpanan data, dimana HDFS terdiri dari 2 bagian yaitu *name-node* dan *data-node. name-node* ada pada satu mesin yang bertugas menyimpan *metadata* dari data yang disimpan dan bertugas memantau kesehatan *data-node*. Hadoop memastikan ketersediaan data aktual tersedia dalam *data-node*, melalui *name-node* yang melakukan replikasi data yang disimpan ke dalam beberapa *data-node* yang ada, sehingga ketika satu *node* mati, maka data tetap ada pada *data-node* yang lain. Sedangkan *data-node* sendiri bertugas menyimpan data aktual. Jika *name-node* hanya terdiri dari satu bagian dari mesin, maka *data-node* terdiri dari beberapa bagian yang dijumpai dalam beberapa mesin yang terhubung dalam jaringan[10].

MapReduce adalah pemrograman parallel yang ada pada *Hadoop* yang berguna untuk memproses data besar secara tepat waktu diantara masing-masing *node* yang ada. *MapReduce* terdiri dari beberapa fase dalam menyelesaikan suatu komputasi secara parallel, yaitu proses *map*, *shuffles*, *sort* dan *reduce.* Setiap pekerjaan yang dieksekusi akan dipetakan oleh *name-node* ke data-node dan selanjutnya dieksekusi oleh *mapreduce. Name-node* akan menginformasikan lokasi data pada *data-node* yang akan dieksekusi ke proses *map* yang ditangani oleh *map worker*. Pada fase map, masukan yang diberikan berupa pasangan *key/value* (k, v) dan menghasilkan urutan pasangan *key/value* yang lain sebagai keluaran tergantung pada algoritma yang akan diterapkan. Fase *shuffle* dimulai setelah fase map selesai memberikan pasangan *key/value* dari masing-masing *datanode* yang terdistribusi secara parallel, kemudian fase *sort* mengurutkan pasangan key,value dengan hasil yang sama dari keseluruhan *data-node*. Terakhir fase *reduce* akan mengurangi jumlah pasangan *key/value* yang sama sebagai satu *cluster* dengan ojek lainnya yang memiliki kemiripan jarak dengan pasangan *key/value* sesuai dengan algoritma yang digunakan [11].

3. *Clustering*

Analisa *clustering* adalah metode dalam penelitian untuk mengelompokkan objek data ke dalam satu kelompok yang memiliki kemiripan. *Clustering* adalah metode pembelajaran yang tak diawasi dan merupakan teknik yang umum digunakan untuk analisis data secara statistical dalam berbagai bidang, termasuk *machine learning, data mining, pattern recognition, image analysis bioinformatics*, dan *marketing*. Tujuan dari *clustering* adalah mengelompokkan data

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

dengan yang memiliki kemiripan ke dalam satu cluster [12].

Salah satu permasalahan dalam pengolahan data besar adalah kebutuhan analisis data terkait yang tersebar dalam penyimpanan dalam hadoop sebagai satu entitas. Dalam penelitian ini, metode clustering menggunakan algoritma *K-Medoids MapReduce* diterapkan pada hadoop untuk menganalisa data beserta performa dari cluster yang dihasilkan dari algoritma yang diusulkan. *Clustering* cocok digunakan untuk menangani analisis data besar yang tersimpan pada hadoop, terutama dengan menggunakan fitur *mapreduce* efisiensi pengolahan data dapat lebih baik karena dibagi secara paralel pada proses eksekusi algoritmanya.

Penelitian ini membandingkan performa dan kualitas *cluster* yang dihasilkan dari penerapan algoritma *K-Medoids mapredude* dengan *K-Medoids* dan *K-Modes* tanpa *mapreduce*.

4. Algortima *K-Medoids*

Algoritma *K-Medoids* merupakan metode *clustering* yang mengelompokkan sekumpulan n objek menjadi sejumlah k *cluster*. Algoritma ini menggunakan objek terpilih sebagai medoids diantara keseluruhan objek pada sebuah *cluster. Cluster* dibangun dengan menghitung kedekatan yang dimiliki antara medoids dengan objek non-medoids. Dalam membangun *cluster*, algoritma ini menggunakan sejumlah k sebagai pusat *cluster* diawal proses *clustering*. Untuk setiap objek yang dekat dengan pusat (medoids) akan dikelompokkan dalam satu cluster yang sama, selanjutnya secara acak menentukan medoids baru dari masing-masing cluster yang telah ditentukan sebelumnya. Jarak antar objek i dan c dihitung dengan menggunakan *dissimilarity measurement function*. Analisis ini mencoba meminimumkan ketidaksamaan setiap objek dalam satu *cluster* dengan meminimumkan nilai *absolute error* [13][14]. Nilai dari *absolute error* dirumuskan sebagai berikut:

$$E = \sum_{c=1}^{k} \sum_{i=1}^{n_c} |p_{ic} - O_c| \qquad (3.1)$$

dimana:

E = *absolute error*

n_c = jumlah objek dalam *cluster* ke-c

p_{ic} = objek *non-medoids* i dalam *cluster* ke-c

O_c = *medoids* di *cluster* ke-c

i = 1, 2, 3, …,n_c

c = 1, 2, 3, …, k

Menurut [13], langkah penetuan *cluster* dengan algortima *K-Medoids* adalah sebagai berikut:

a. Memilih k objek untuk menjadi Oc yang merupakan *medoids* di cluster ke-c
b. Menghitung kemiripan antara objek Oc dan *non-medoids* dengan menggunakan jarak *Euclidean*
c. Menempatkan objek *non-medoids* ke dalam kelompok yang paling dekat dengan Oc
d. Memilih Oc baru dari objek *non-medoids* sebagai pengganti Oc awal

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

e. Menghitung kemiripan antara objek non-medoids dengan Oc baru menggunakan jarak Euclidean
f. Menempatkan objek *non-medoids* ke dalam kelompok yang paling mirip dengan Oc baru
g. Menghitung nilai *absolute error* sebelum dan sesudah pertukaran Oc dengan Oc baru, jika $E_{baru} < E_{awal}$ maka tukar Oc awal dengan Oc baru dan sebaliknya
h. Mengulangi langkah d sampai g hingga semua objek *non-medoids* dikelompokkan dalam cluster dan tidak terjadi perubahan pada Oc.

5. Indeks *Silhouette*

Indeks *silhouette* merupakan metode yang digunakan untuk melihat kuantitas dari sebuah *cluster*. Indeks *silhouette* mengevaluasi objek secara visual baik yang berada dalam *cluster* maupun berada diluar *cluster.* Nilai silhouette mempunyai rentang dari -1 sampai 1, yang artinya jika kualitas sebuah *cluster* baik maka nilai silhouettenya akan mendekati positif 1, begitupula sebaliknya, jika kualitas *cluster* buruk maka nilai *silhouette* yang dihasilkan akan mendekati negatif 1. Rentang nilai silhouette dirumuskan sebagai berikut [15]:

$$s_i = \frac{b_i - a_i}{\max[b_i, a_i]} \qquad (3.2)$$

dimana:

s_i = nilai *silhouette* objek ke-$i, i = 1, 2, \dots, n$

a_i = jarak rata-rata antara objek ke-i dengan lainnya dalam satu cluster

b_i = jarak rata-rata minimum objek ke-i dengan objek lainnya di masing-masing *cluster*

Interpretasi dapat dilakukan dengan menghitung hasil indeks *silhouette* terhadap jumlah cluster yang dihasilkan sehingga dapat memberikan analisa dan gambaran yang logis terkait hasil *cluster*.

3. Metode Penelitian

Penelitian ini menggunakan dataset yang diperoleh dari Kaggle, berupa dataset mobil[1] dengan ukuran 400MB terdiri dari 16 atribut dengan jumlah 3.5 juta baris data. Dataset diolah dengan membandingkan 2 perlakuan pengolahan analisis, yaitu data disimpan dan diolah pada *HDFS* dan diluar *HDFS* yang selanjutnya akan diproses dengan masing-masing algoritma yang disebutkan sebelumnya. Untuk evaluasi performa dan kualitas *cluster,* pengujian dengan menggunakan metrik uji *silhouette score*.

Desain Eksperimen

Penelitian ini menggunakan konfigurasi 1 mesin yang sudah diinstal dengan Hadoop 2.8.5 pada ubuntu 18.04.3, dengan sistem operasi 64 bit, ram 8 GB dan penyimpanan SSD 240 GB.

[1] www.kaggle.com/mirosval/personal-cars-classified

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

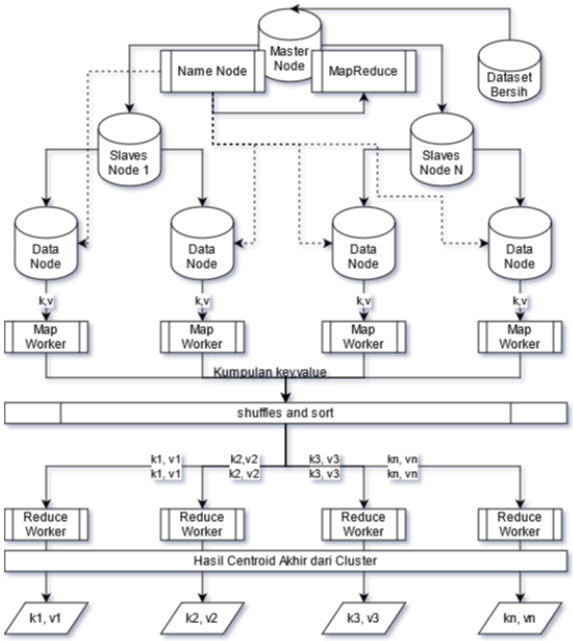
DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

Eksperimen dieksekusi dengan menggunakan bahasa pemrograman python.

Dataset iklan mobil yang telah disebutkan sebelumnya diterapkan pra proses sebelum mulai pengimplementasian algoritma *clustering.* Dalam pra proses dataset, dilakukan proses perbaikan terhadap nilai yang hilang dan data yang tidak memiliki nilai (*nan*), selanjutnya mengecek apakah *dataset* mengandung pencilan atau tidak, dalam hal ini pencilan yang ada akan dibersihkan sehingga bebas dari pencilan. Tahap akhir dari pra proses dataset adalah melakukan normalisasi terhadap dataset.

Eksperimen dilakukan dengan cara memasukkan data yang sudah melewati tahap pra proses ke HDFS dan mengeksekusi algoritma *K-Medoids MapReduce* pada sebuah mesin, serta dibandingkan dengan *K-Medoids* dan *K-Modes* yang mana algoritma ini akan diterapkan pada dataset iklan mobil yang sudah dipra proses sebelumnya dan disimpan juga diluar penyimpanan HDFS.

Pada penelitian ini, jumlah k maksimal dibatasi pada 10 *cluster* dari setiap perlakuan eksperimen. Penelitian dilakukan dengan beberapa kali percobaan untuk memastikan hasil *cluster* yang didapatkan bagus dan melihat performa algoritma dari segi waktu eksekusi. Setelah algoritma sukses dieksekusi selanjutnya diuji kualitas *cluster* yang dihasilkan dengan menggunakan metrik uji indeks *silhouette*. Gambar 1 menujukkan langkah-langkah yang dilakukan dalam penelitian ini.



Gambar 1. Langkah Eksekusi Algoritma *K-Medoids* dengan *MapReduce*

Pada gambar 1 dataset bersih dimasukkan ke dalam *master-node* HDFS yang dibaca dan distribusikan oleh *name-node* ke *data-node* yang aktif. Setelah disebarkan ke beberapa *data-node* yang aktif, selanjutnya algoritma *K-Medoids* dieksekusi dan *name-node* akan mencari dataset yang akan dieksekusi dan memerintahkan *mapreduce* agar membagi tugas eksekusi ke dalam beberapa map dan reduce sampai algoritma selesai dieksekusi dan memberikan output jumlah *cluster*.

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

4. Hasil dan Pembahasan

Hasil dari penelitian ini ditunjukkan pada Tabel 1. Dataset iklan mobil dieksekusi dengan menggunakan algoritma *K-Medoids MapReduce, K-Medoids*, dan *K-Modes*. Hasil cluster yang didapatkan dievaluasi dengan menggunakan uji *silhouette* dan dievaluasi juga performa waktu eksekusi dari seluruh algoritma.

Tabel 1. Hasil Eksperimen

Jumlah *Cluster*	*K-Medoids Map Reduce*	*K-Medoids*	*K-Modes*
2	0.99	0.61	0.26
3	0.15	0.66	-0.16
4	0.12	0.69	-0.43
5	0.56	0.62	-0.44
6	0.30	0.62	-
7	0.63	0.59	-
8	0.46	0.59	-
9	0.16	0.56	-
10	0.11	0.56	-

Tabel 1 menunjukkan hasil cluster yang dihasilkan dengan menggunakan algoritma *K-Medoids MapReduce* memberikan skor *silhouette* sebesar 0.99 dengan jumlah 2 cluster yang menandakan bahwa kualitas cluster yang dihasilkan sangat bagus, dengan performa waktu eksekusi adalah 2533.36 detik. Algoritma *K-Medoids* dan *K-Modes* dengan data iklan mobil yang dieksekusi tidak pada *framework hadoop* menghasilkan nilai *silhouette* dan jumlah cluster berturut-turut adalah 0.69 dengan 4 *cluster* dan 0.26 dengan 2 *cluster.* Adapun untuk performa waktu eksekusi yang dihasilkan dari algoritma *K-Medoids* dan *K-Modes* adalah 8012.94 detik dan K-Modes gagal menyelesaikan proses *clustering* dan terhenti pada jumlah 4 *cluster* yang mana hasilnya juga sudah menunjukkan nilai negatif yang artinya kualitas cluster yang dihasilkan buruk. Dalam eksperimen ini, digunakan 10% data dari keseluruhan total data sebagai sampel dalam menguji algoritma dan mensiasati keterbatasan mesin yang digunakan.

5. Kesimpulan

Berdasarkan ekperimen yang dilakukan, teknologi hadoop mampu menjadi salah satu solusi dalam penanganan penyimpanan dan pengelolaan data besar untuk kebutuhan analisis dilihat dari hasil kualitas *cluster* dan performa waktu yang diperlukan untuk mengeksekusi dataset. Dalam eksperimen yang dilakukan data yang dieksekusi hanya 10%, sebenarnya dengan menggunakan penyimpanan terdistribusi *hadoop*, data bisa dieksekusi 100% karena data disimpan dan diproses secara paralel dan terdisitribusi pada beberapa mesin, akan tetapi karena dalam hal ini mesin yang digunakan hanya satu dan untuk menangani keterbatasan itu maka hanya digunakan 10% *dataset.* Percobaan yang dilakukan tanpa mengandalkan bantuan framework *hadoop* kemungkinan akan mengalami kegagalan dalam proses *clustering* yang mana hal ini bergantung pada algoritma yang diterapkan. Dengan menggunakan beberapa mesin, algoritma yang diterapkan

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 Link : https://dx.doi.org/10.29408/jit.v4i2.3565

mampu mengeksekusi keseluruhan data dan menigkatkan kinerja waktu eksekusi, dilihat dari waktu rata-rata dari beberapa kali uji eksperimen dari masing-masing algoritma, *K-Medoids MapReduce* memberikan waktu paling signifikan dibandingkan dengan 2 algoritma lainnya.

Berdasarkan eksperimen yang sudah dilakukan sebelumnya, nilai *silhouette* yang lebih dari besar dari 0.5 menunjukkan kualitas *cluster* yang baik yang diperoleh oleh algoritma K-Medoids MapReduce dan K-Medoids. Dalam hal ini walaupun jumlah cluster yang dihasilkan berbeda, akan tetapi jika dilihat pada Tabel 1 setelah beberapa *cluster* terbentuk, terdapat hasil yang diperoleh hamper sama yaitu berada pada jumlah 6 dan 7 *cluster*, hal ini memungkinkan terjadinya jumlah *cluster* yang berbeda karena algoritma yang digunakan berbeda terutama dalam *MapReduce* walaupun sama-sama menggunakan K-Medoids sebagai dasar dari algoritma.

Penelitian ke depannya dapat ditingkatkan dalam beberapa hal termasuk penggunaan jumlah mesin yang lebih banyak dalam mengeksekusi algoritma yang bekerja berdasarkan *mapreduce* serta dapat dibandingkan dengan pemanfaatan sumber seperti penggunaan GPU dengan mengimplementasikan TensorFlow dan Keras sebagai backend untuk proses clustering sebagai pembanding.

6. Daftar Pustaka

[1] R. Elmasri and S. B. Navathe, “Fundamentals of Database Systems 4th edition,” *Database*, 2003.

[2] Y. Hajjaji and I. R. Farah, “Performance investigation of selected NoSQL databases for massive remote sensing image data storage,” in *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Mar. 2018.

[3] S. Dhanasekaran, R. Sundarrajan, B. S. Murugan, S. Kalaivani, and V. Vasudevan, “Enhanced Map Reduce Techniques for Big Data Analytics based on K-Means Clustering,” *IEEE Int. Conf. Intell. Tech. Control. Optim. Signal Process. INCOS 2019*, pp. 1–5, 2019.

[4] L. Wang, E. Zou, C. Zeng, X. Xi, and Y. Lu, “Research and Implementation of Big Data Clustering Based on Spark,” *Shuju Caiji Yu Chuli/Journal Data Acquis. Process.*, vol. 33, no. 6, pp. 1077–1085, 2018.

[5] H. Liu, F. Huang, H. Li, W. Liu, and T. Wang, “A Big Data Framework for Electric Power Data Quality Assessment,” in *2017 14th Web Information Systems and Applications Conference (WISA)*, 2017..

[6] C. S. Kim and S. B. Son, “A Study on Big Data Cluster in Smart Factory using Raspberry-Pi,” *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*.

[7] P. Ramadar Noor Saputra and A. Chusyairi, “Perbandingan Metode Clustering dalam Pengelompokan Data Puskesmas,” *J. Rekayasa Sist. dan Teknol. Inf.*, vol. 4, no. 6, pp. 1077–1084, 2020.

[8] C. Verma and R. Pandey, “Big Data representation for grade analysis through Hadoop framework,” *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 312–315, 2016..

[9] C. Kaushal and D. Koundal, “Recent trends in big data using hadoop,” *Int. J. Informatics Commun. Technol.*, vol. 8, no.

Infotek : Jurnal Informatika dan Teknologi
Vol. 4 No. 2, Juli 2021
Hal. 174-183

e-ISSN 2614-8773

DOI : 10.29408/jit.v4i2.3565 **Link : https://dx.doi.org/10.29408/jit.v4i2.3565**

1, p. 39, 2019.

[10] M. B. Masadeh, M. S. Azmi, and S. S. S. Ahmad, "Available techniques in hadoop small file issue," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 2097–2101, 2020.

[11] D. C. Vinutha and G. T. Raju, "An accurate and efficient scheduler for hadoop mapreduce framework," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 3, pp. 1132–1142, 2018.

[12] N. A. ERILLI, "Comparison of fuzzy clustering methods in economic freedom ranking in Asia-Pacific," *J. Perspekt. Pembiayaan dan Pembang. Drh.*, vol. 7, no. 2, pp. 157–168, 2019.

[13] E. Setyowati, A. Rusgiyono, and M. A. Mukid, "Analisis Pengelompokan Daerah Menggunakan Metode Non-Hierarchical Partitioning K-Medoids Dari Hasil Komoditas Pertanian Tanaman Pangan," *J. Gaussian*, vol. 4, no. 4, pp. 825–836, 2015.

[14] Z. Mustofa and I. S. Suasana, "Algoritma Clustering K-Medoids Pada E-Government Bidang Information And Communication," *J. Teknol. dan Komun.*, vol. 9, pp. 1–10, 2018.

[15] E. Okta, N. Satyahadewi, and N. N. Debataraja, "Penerapan Metode K-Medoids Pada Pengelompokan," vol. 08, no. 4, pp. 813–820, 2019.