Hadoop-MapReduce Pada YARN Framework

Londa Arrahmando Romadhona[1*], Rani Febrianti[1], Aldi Winata[1], Clarisa Putri Amanda[1], Rosa Julia Erizka[1], Alyatisa[1]
[1] Universitas Sriwijaya, Sistem Komputer

*Korespondensi: londaarrahman@gmail.com

ARTICLE INFO

Article History:
- Received 05 January 2022
- Received in revised form 28 March 2022
- Accepted 14 April 2022
- Available online 30 July 2022

ABSTRAK

Big data, sebagai kumpulan data berskala besar, memiliki karakteristik yang bervariasi, pertumbuhan yang sangat cepat, dan kompleksitas tinggi. Kompleksitas ini terutama terkait dengan data yang tidak terstruktur, memerlukan pemrosesan khusus melalui infrastruktur yang dapat mengelola volume data yang besar. Dalam konteks ini, tugas ini mengulas tantangan utama yang dihadapi oleh big data dan menyajikan kerangka kerja Apache Hadoop pada tingkat pelaksanaan pekerjaan MapReduce yang didasarkan pada YARN (Yet Another Resource Negotiator). Kami mengeksplorasi penerapan konkret dari kerangka kerja ini melalui studi kasus aplikasi wordcount menggunakan bahasa pemrograman Java.

Kata Kunci: YARN, Hadoop, MapReduce, Big Data, Apache

ABSTRACT

Big data is a collection of data on a large scale characterized by varying data types, rapid growth, and complex structures. Complex data, which is often unstructured, requires specialized processing through an infrastructure capable of managing large volumes of data. This task addresses the challenges posed by big data and introduces the Apache Hadoop framework at the execution level of MapReduce based on YARN (Yet Another Resource Negotiator), applying a Java wordcount application.

Keywords: YARN, Hadoop, MapReduce, Big Data, Apache

1. PENDAHULUAN

Belakangan ini, semakin banyak data yang dihasilkan dari berbagai sumber. Ukuran data yang dihasilkan per hari di Internet telah melebihi dua exabyte [1]. Lebih banyak Tweet dibagikan di Twitter dan lebih banyak gambar diposting di Facebook. Beberapa definisi Big Data telah diusulkan dalam literatur. Kebanyakan dari mereka setuju bahwa masalah Big Data memiliki empat karakteristik utama, yang disebut lima V (volume, variasi, variabilitas, kejujuran, dan kecepatan) [2].

Seiring dengan berkembangnya zaman big data merupakan suatu yang menjadi trend dalam dunia informasi. Bisa dibilang big data merupakan kumpulan data yang sangat besar yang di dalamnya mencakup berbagai jenis data. Big Data menjadi kata yang populer seiring dengan bagaimana dapat menyimpan data dalam jumlah yang besar, melakukan proses serta analisa. Sesuatu yang tidak dapat dihindari bagaimana impact dari big data ini dalam kehidupan sehari-hari. Big Data telah memberikan kesempatan atau peluang bisnis bagi banyak perusahaan. Hampir semua industri telah memanfaatkan atau baru melakukan identifikasi tentang pentingnya big data dalam menumbuhkan bisnisnya atau tetap dapat bersaing bahkan menjadi keunggulan dalam berkompetisi [3]. Tingkat adopsi Big Data di Indonesia adalah 20% untuk 2 sampai 3 tahun ke depan [4].

Dari sekian banyak manfaat dan peluang, big data dapat meninggalkan beberapa tantangan diantaranya adalah tantangan teknologi yang dapat menghandle big data ini, tantangan skill atau keahlian orang yang akan mengolah data sehingga data yang tersedia dapat menjadi informasi, insight yang bermanfaat. Dalam dunia akademik, istilah big data mengacu pada aplikasi teknologi informasi untuk menangani masalah data yang sifatnya besar [3].

- **Volume** : ukuran kumpulan data yang biasanya membutuhkan penyimpanan terdistribusi, pertumbuhan volume data seperti itu juga menyebabkan peningkatan kebutuhan akan sistem manajemen basis data baru, sehingga lahirlah Hadoop HDFS yang akan dirinci nanti.
- **Variasi** : mengacu pada fakta bahwa Big Data terdiri dari beberapa jenis data yang berbeda seperti teks, suara, gambar, dan video. Keragaman dapat diukur menggunakan dimensi yang berbeda seperti struktur yang memungkinkan kita untuk membedakan data terstruktur, semi-terstruktur dan tidak terstruktur, atau volume pemrosesan seperti dalam batch versus aliran.
- **Variabilitas** : mengacu pada data yang tidak stabil, yang tidak dapat dengan mudah ditangani, dan sulit dikelola. Menjelaskan data variabel merupakan masalah yang signifikan bagi peneliti [5].
- **Kebenaran** : mengacu pada keandalan, kebisingan dan anomali dalam data.
- **Kecepatan** : mengacu pada kecepatan di mana data baru dihasilkan dari berbagai sumber seperti jejaring sosial dan Internet of Things (IoT).

Masalah Big Data mengarah pada beberapa masalah penelitian seperti : bagaimana merancang lingkungan yang dapat diskalakan, yang memberikan toleransi kesalahan untuk merancang solusi yang efektif. Dalam perspektif inilah kami dalam artikel ini berorientasi pada pengujian kerangka kerja Hadoop-MapReduce.

2. TINJAUAN PUSTAKA

2.1 Big Data

Konsep “Big Data” pertama kali dicetuskan oleh Roger Magoulas di dalam media O’Riley pada tahun 2005. Big Data menjelaskan bahwa data yang ada begitu besar dan banyak sehingga manajemen data tradisional tidak dapat digunakan lagi [6]. *Big Data* merupakan istilah yang menggambarkan suatu *volume* data yang besar, baik yang struktur maupun tidak terstruktur. *Big Data* telah digunakan dalam banyak bisnis. Tidak hanya besar data yang menjadi poin utamanya, tetapi apa yang harus dilakukan organisasi dengan besar data tersebut. *Big Data* dapat dianalisis untuk wawasan yang mengarah pada pengambilan keputusan dan strategi bisnis yang lebih baik [7].

Istilah big data masih terbilang baru dan sering disebut sebagai tindakan pengumpulan dan penyimpanan informasi yang besar untuk analisis. Fenomena big data dimulai pada tahun 2000-an ketika seorang analisis industri Doug Laney menyampaikan konsep big data yang terdiri dari tiga bagian penting , diantaranya:

Volume : Organisasi mengumpulkan dari berbagai sumber, termasuk transaksi bisnis, media sosial dan informasi dari sensor atau mesin. Di masa lalu, aktivitas semacam ini menjadi masalah, namun dengan adanya teknologi baru (seperti Hadoop) bisa meredakan masalah ini [7].

Kecepatan : Aliran data harus ditangani dengan cepat dan tepat bisa melalui hardware maupun software. Teknologi hardware seperti tag RFID, sensor pintar lainnya juga dibutuhkan untuk menangani data yang real-time.

Variasi : Data yang dikumpulkan mempunyai format yang berbeda-beda. Mulai dari yang terstruktur, data numerik dalam database tradisional, data dokumen terstruktur teks, email, video, audio, transaksi keuangan dan lain-lain.

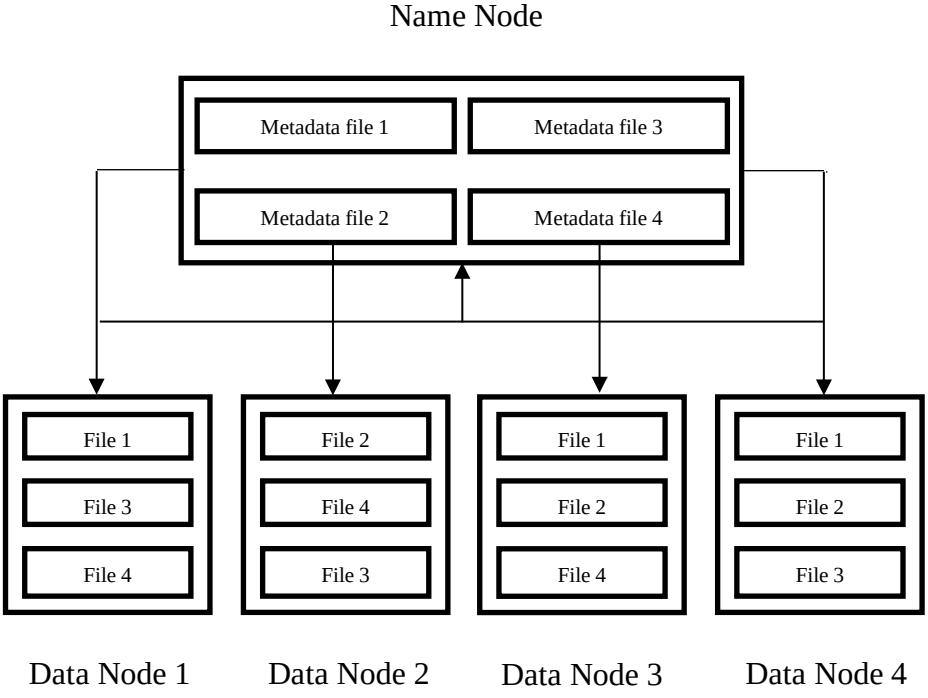
2.2 Hadoop

Big Data mulai jadi trend teknologi saat ini. Salah satu software platform yang bisa digunakan untuk mengelola big data adalah Hadoop. Secara ringkas Hadoop adalah software yang mampu menghubungkan banyak komputer untuk dapat bekerja sama dan saling terhubung untuk menyimpan dan mengelola data dalam satu kesatuan. Hadoop menyimpan dan mengolah big data menggunakan model pemrograman MapReduce. MapReduce adalah model pemrograman rilisan Google yang bisa digunakan untuk memproses data dalam ukuran besar secara terdistribusi dan paralel dalam cluster yang terdiri dari komputer berjumlah ribuan [8].

2.3 Hadoop Distributed File System

Hadoop Distributed File System (HDFS) merupakan sistem penyimpanan terdistribusi, yang melakukan proses pemecahan file besar menjadi bagian-bagian lebih kecil kemudian didistribusikan ke cluster-cluster dari komputer. Cluster ini biasanya terdiri dari banyak node atau komputer atau server. Setiap node di dalam cluster ini harus terinstal Hadoop untuk bisa berfungsi. Sebagai distributed file system, HDFS berguna untuk menangani data berukuran raksasa yang disimpan tersebar dalam clusternya [9].

Sebagai distributed file system, HDFS menyimpan suatu data dengan cara membaginya menjadi potongan-potongan data yang misalnya berukuran 64 MB, dan potongan-potongan data ini kemudian disimpan tersebar dalam komputer-komputer yang membentuk clusternya. Potongan-potongan data tersebut dalam HDFS disebut block, dan ukurannya tidak terpaku harus 64 MB misalnya. Ukuran block dapat diatur sesuai kebutuhan [9].

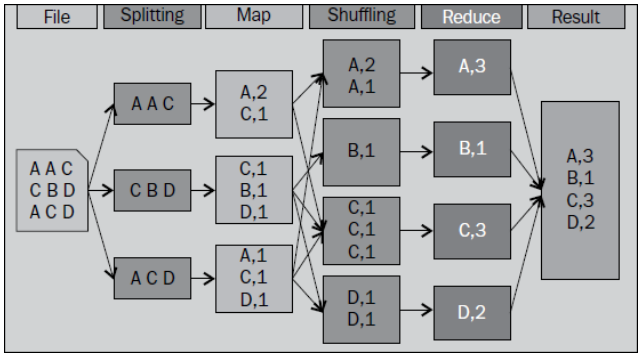


Gambar 1. Penyimpanan Data pada HDFS [9]

HDFS memiliki komponen-komponen utama berupa NameNode dan DataNode. NameNode adalah sebuah komputer yang bertindak sebagai master, sedangkan DataNode adalah komputer-komputer dalam Hadoop Cluster yang bertugas sebagai slaves atau anak buah. NameNode bertanggung jawab menyimpan informasi tentang penempatan block-block data dalam Hadoop Cluster. Ia bertanggung jawab mengorganisir dan mengontrol block-block data yang disimpan tersebar dalam komputer-komputer yang adoop n Hadoop Cluster. Sedangkan DataNode bertugas menyimpan block-block data yang dialamatkan kepadanya, dan secara berkala melaporkan kondisinya kepada NameNode [9].

2.4 MapReduce

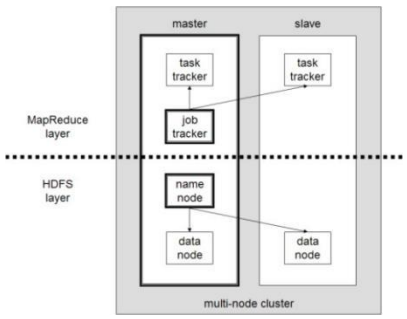
MapReduce [10] adalah sebuah model programming sederhana untuk memproses data. Program yang mengimplementasikan MapReduce dapat berjalan secara paralel sehingga dapat menyelesaikan analisis data dengan ukuran yang sangat besar. MapReduce terdiri dari 2 bagian yaitu fase map dan fase reduce. Kedua fase memiliki sepasang key dan value sebagai input dan output yang tipenya dipilih oleh programmer. Programmer harus menentukan juga isi fungsi mapper dan fungsi reducer. Arsitektur MapReduce pada Hadoop dapat dilihat pada Gambar 1.



Gambar 2. Arsitektur MapReduce pada Hadoop [10]

2.5 Apache Hadoop

Hadoop dibuat oleh Doug Cutting, pembuat Apache Lucene [11]. Apache Hadoop adalah framework yang digunakan untuk pemrosesan data besar yang tersebar di seluruh cluster komputer dengan menggunakan data model pemrograman MapReduce [12]. Ada empat komponen adoop, yaitu MapReduce, Hadoop Utilities, YARN (Yet Another Resource Negotiator) dan HDFS (Hadoop Distributed File System) [13]. Arsitektur dari Hadoop dapat dilihat pada Gambar 2.



Gambar 3. Arsitektur Hadoop Ward [11]

HDFS [14] terdiri dari name node dan data node, sedangkan MapReduce terdiri dari job tracker dan task tracker. Pada saat ini, Hadoop adalah system tercepat untuk mengolah data dalam Terabyte. Hadoop [15] dapat dijalankan dengan 3 macam cara, yaitu Standalone mode, Pseudo – distributed mode dan Fully distributed atau cluster mode.

3. METODOLOGI PENELITIAN

Perkembangan teknologi semakin pesat dan layanan yang disediakan semakin banyak. Terdapat layanan yang berkembang sangat cepat yaitu teknologi *internet*. Dimana dengan *internet* semua perangkat elektronik yang mempunyai alamat *Internet Protocol* maka dapat saling terkoneksi. Layanan *software* dapat digunakan oleh setiap pengguna secara berbayar ataupun gratis. Semakin berkembangnya teknologi pada *software* yangkompleks menyebabkan konsumsi data penyimpanan semakin besar. Oleh karena itu, Segala data yang terdapat pada suatu aplikasi pada *internet* akan berkembang dari data yang kecil kemudian menjadi data yang besar.

Data dalam skala besar yang disebut dengan *Big Data*. *Big Data* adalah kumpulan data yang sangat besar, sangat variatif, sangat cepat pertumbuhannya dan mungkin tidak terstruktur. Proses komputasi yang terjadi pada *big data* dapat berjalan lambat apabila komputer yang digunakan untuk memproses data tersebuttidak memenuhi standar yang dibutuhkan oleh suatu *big data*. Maka, diperlukan suatu algoritma khusus sehingga informasi yang mendalam mudah didapatkan dan dapat membantu pengambilan keputusan yang lebih baik. Solusi untuk *Big Data* yaitu Hadoop. Hadoop adalah *framework open source* di bawah lisensi Apache untuk mensupport aplikasi yang jalan pada *Big Data*. Asal mula Hadoop muncul karena terinspirasi dari makalah tentang *Google MapReduce* dan *Google File System (GFS)* yang ditulis oleh ilmuwan dari Google, *Jeffrey Dean* dan *Sanjay Ghemawat* pada tahun 2003. Penamaan menjadi Hadoop adalah diberikan oleh *Doug Cutting*, yaitu berdasarkan nama dari mainan gajah anaknya.

Hadoop dijalankan pada lingkungan yang menyediakan *storage* dan komputasi secara terdistribusi atau bisa disebut sebagai *Hadoop Distributed File System* (HDFS). *Hadoop* mendistribusikan klaster-klaster dari komputer/*node* menggunakan suatu model pemrograman. Mapreduce adalah paradigma pemrograman yang berjalan di latar belakang Hadoop untuk menyediakan skalabilitas dan mudah solusi pengolahan data. Pengolahan data dapat pada sebuah data yang terstruktur, semi-terstruktur, dan tidak terstruktur.

Dengan mengganti disk yang tidak efisien dengan cache memori *low latency, high-throughput* yang terdistribusi untuk mengoptimalkan proses pengiriman data dari tugas *map* ke tugas *reduce*. Untuk memenuhi kebutuhan tersebut proses komputasi menggunakan fungsi yang terdapat pada Apache Flink.

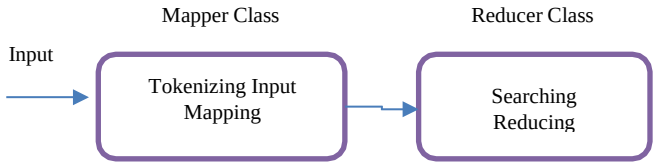
a. Model pemrograman yang digunakan.

Model pemrograman MapReduce digunakan untuk memproses secara efisien kumpulan data yang besar secara paralel *MapReduce* terdiri atas tiga tahap, yaitu tahap *map,* tahap *shuffle*, dan terakhir tahap *reduce*. Untuk tahapan *shuffle* dan *reduce* digabungkan ke dalam satu tahap besaran-nya yaitu tahap *reduce*. Pemrogram dapat menyelesaikan aplikasi MapReduce dengan menerapkan dua fungsi: fungsi Map dan fungsi Reduce. Selanjutnya, kedua dari dua fungsi memiliki <key, value> pair sebagai input dan outputnya.

1) Tahap map, memproses data inputan yang umumnya berupa file yang tersimpan dalam HDFS, inputan tersebut kemudian diubah menjadi tupel yaitu pasangan antara

key dan *value*-nya.

2) Tahap reduce, memproses data inputan dari hasil proses map, yang kemudian dilakukan tahap *shuffle* dan *reduce* yang hasil data set baru-nya disimpan di HDFS kembali. Berikut ini ilustrasi untuk mendapatkan gambaran tentang proses *map* dan *reduce*.



Gambar 4. Mapper & Reducer

Mapper Class mengambil input, *tokenizes* menjadi pasangan kunci nilai, memetakan dan melakukan penyortiran. *Output* dari *Mapper Class* digunakan sebagai input oleh *Reducer Class*, yang kemudian dikelompokkan sesuai kunci nilai yang sama.

Apache flink adalah *platform* yang dapat melakukan suatu komputasi pada *big data*. Flink dapat melakukan *read* atau *write data* dari berbagai macam sistem penyimpanan. Inti dari Flink adalah mesin *streaming dataflow* yang menyediakan distribusi *data*, komunikasi *data*, dan toleransi kesalahan untuk perhitungan terdistribusi melalui aliran *data*. Flink membangun pemrosesan *batch* diatas mesin *streaming*, tersedia untuk iterasi *overlay native*, pengelolaan *memory*, dan pengoptimalan program.

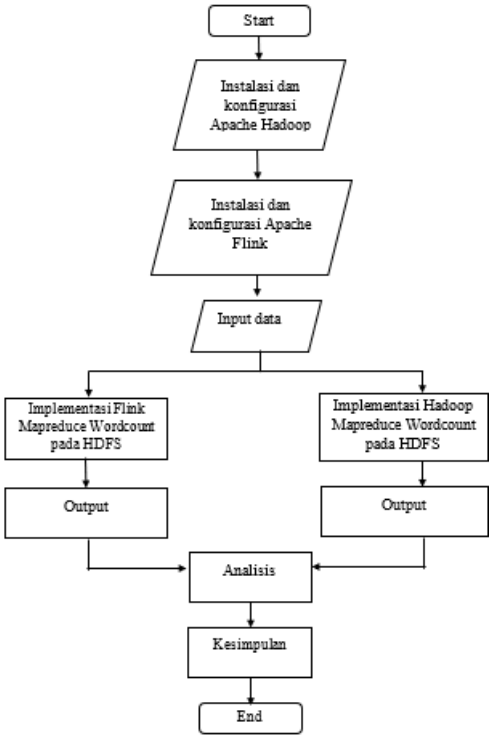
b. Perancangan dan Implementasi Sistem

- Gambaran Umum

  Pada penelitian ini perancangan sistem yang akan dibangun dengan memulai proses instalasi dan konfigurasi HDFS. Dilanjutkan instalasi dan konfigurasi Apache Flink. Kemudian setelah proses konfigurasi dan instalasi selesai maka diperlukan untuk mengatur environment Hadoop pada Apache Flink agar metode Mapreduce dapat dijalankan pada HDFS.

  Setelah Hadoop terinstal, *input data* berupa *file* teks yang kemudian dimasukkan dari *data local* ke HDFS. Terdapat berbagai macam data dengan ukuran *file* yang berbeda. Pengujian dilakukan dimulai dari *data* yang berukuran kecil kemudian bertahap ke *data* yang berukuran besar. *Data* tersebut akan diproses menggunakan program Hadoop Mapreduce pada HDFS.

  Pengujian akan dilakukan dengan dua tahap. Yaitu dengan metode Mapreduce pada Hadoop menggunakan paradigma *disk-based* dan metode Mapreduce pada *flink in-memory batch processing* berbasiskan HDFS. Aplikasi ini berjalan secara lokal atau *standalone*.



Gambar 5. Gambaran Umum Sistem

- Implementasi Metode Mapreduce

  Pada tahap ini akan menjelaskan tentang pengimplementasian program Mapreduce. Program Mapreduce akan dijalankan pada dua *environment* secara terpisah. Analisis akan dilakukan setelah program mapreduce mendapat output dari masing-masing *environment* yang digunakan. Berikut skema implementasi program Mapreduce. Pada Hadoop *environment* program Mapreduce dijalankan pada HDFS. Yang pertama *client* harus lakukan yaitu mendistribusikan data pada HDFS untuk memulai program Mapreduce. Program Mapreduce menggunakan Bahasa java yang berisikan implementasi dari fungsi Mapper dan Reducer.

  Program Mapreduce yang akan dieksekusi diterima oleh Jobtracker. Tugas Jobtracker yaitu mendistribusikan perangkat lunak atau konfigurasi pada pekerja, *job scheduling*, dan memonitor kinerja Mapreduce yang kemudian memberikan informasi kepada *client*. Pada Tasktracker bertugas untuk implementasi Mapreduce dan menghasilkan *output* yang akan disimpan pada *file system*.

c. Pengujian dan Analisis

Pada penelitian ini akan ada beberapa skenario pengujian yang dilakukan seperti dibawah ini:

1. *Input data* berupa file berisi teks yang mempunyai ukuran berbeda, yaitu 1,6 GB dan 2,5GB
2. Pengujian dengan *single node cluster*.
3. Pengujian waktu respon metode Mapreduce Wordcount *disk-based* pada HDFS.
4. Pengujian waktu respon metode Mapreduce Wordcount Apache Flink berbasis HDFS.

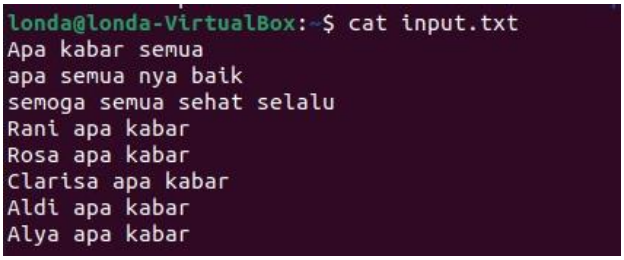
Skenario yang diujikan yaitu waktu respon (*response time*) dan sumber daya (*resource*) yang digunakan pada saat melakukan komputasi dan performa pada komputer.

4. HASIL PENELITIAN

Untuk memproses program, proses yang pertama dilakukan adalah mengaktifkan dfs hingga datanodes, namenodes, dan secondary namenodes dimulai. Kemudian mengaktifkan YARN untuk mengaktifkan management resource.

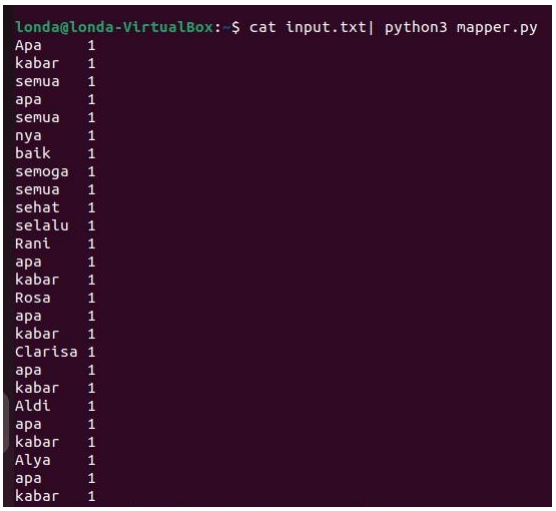
Gambar 6. Proses mengaktifkan YARN

Proses selanjutnya adalah memanggil file input.txt yang akan MapReduce menggunakan framework YARN.



Gambar 7. Memanggil file input

Untuk menghasilkan sebuah output dari program word count, maka dilakukannya proses mapping dan reduce. Data yang di input akan diproses dan dibagi menjadi bagian-bagian yang lebih kecil pada fase mapper. Mapper akan memproses key-value ini menggunakan logika pengkodean untuk menghasilkan sebuah output yang sama. Untuk proses mapping dapat dilihat di bawah.



Gambar 8. Proses Mapping

Untuk proses selanjutnya yaitu proses reduce. Pada proses reduce, akan diperoleh output hasil data baru yang lebih kecil untuk diproses dan disimpan. Berikut contoh gambar dari proses reduce.

Gambar 9. Proses Reduce

Setelah proses pengimplementasi nya berhasil dijalankan, maka akan dihasilkan sebuah output word count dari MapReduce dari file data yang telah dibuat. Berikut hasil dari percobaan.

Gambar 10. Hasil Percobaan

Berdasarkan hasil percobaan yang telah dilakukan, untuk menghasilkan sebuah output maka dilakukan proses mapping dan reduce. Output dari program tersebut ialah word count dari file data input yang dimana kata “apa dan kabar” jumlahnya masing masing adalah 6. Kemudian untuk kata “semua” jumlahnya adalah 3. Dan terakhir adalah kata “Aldi, Alya, Apa, baik, Clarisa, nya, Rani, Rosa, sehat, selalu, dan semoga” adalah 1. Semua kata dalam file input telah di mapping, di reduce dan kemudian disortir menjadi jumlahnya masing-masing.

5. KESIMPULAN

Big Data istilah dimana data memiliki volume yang sangat besar, tersusun baik dari data yang terstruktur maupun yang tidak terstruktur sehingga manajemen data tradisional tidak dapat digunakan lagi. Oleh karena itu, framework *Hadoop* muncul untuk menjadi solusi mengelola kumpulan data dengan melakukan penyimpanan dan pemrosesan data dalam skala besar. *Hadoop* dapat menghubungkan banyak komputer untuk dapat bekerja sama dan saling terhubung untuk menyimpan dan mengelola data dalam satu kesatuan. *Hadoop* menyimpan dan mengolah big data menggunakan banyak modul. Modul-modul tersebut terdiri dari *Hadoop Common, Hadoop Distributed file system (HDFS), Hadoop YARN dan Hadoop MapReduce*.

Pada program yang kami lakukan, kami menerapkan modul YARN untuk melakukan program penerapan *wordcount* yang bertujuan untuk melakukan komputasi jumlah kata dalam sebuah file *plaintext* yang telah disiapkan. YARN framework disebut juga MapReduce 2.0 sebuah kerangka untuk melakukan schedule pekerjaan dan mengelola sumber daya cluster yang membagi Job Tracker dan Task Tracker menjadi Resource Manager, Application Master, Node Manager, dan *Container*. Dimana algoritma *wordcount* sudah disediakan dalam program *Hadoop* yang memiliki dua tahapan yakni *mapping* dimana tiap kata dikeluarkan ditambah jumlah komputasi terkait dan *reducing* untuk melakukan penjumlahan dari tiap jumlah kata yang ada. Hasil dari penelitian program diatas akan menampilkan tiap kata yang ada pada file *plaintext* beserta jumlah banyaknya tiap kata. Dilihat dari hasil output pada terminal tiap kata dipisah dan di urutkan menjadi per tiap satu kata ke bawah yang dilakukan pada tahapan *mapping* dan terdapat nilai angka di sampingnya yang merupakan hasil komputasi *reducing* jumlah kata yang sama atau jumlah berapa kali kata tersebut muncul/ada di dalam file *plaintext* tersebut. Dan juga dengan bantuan command *sort* tiap kata tersebut diurutkan sesuai abjadnya. Sehingga hasil akhir didapatkan setelah proses *mapping* tiap kata, me-*reduce* dengan komputasi jumlah tiap kata yang muncul, kemudian *disortir* sehingga menampilkan kata berurutan abjad.

DAFTAR PUSTAKA

[1] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *Int. J. Inf. Manage.*, vol. 35, no. 2, pp. 137–144, 2015, doi: 10.1016/j.ijinfomgt.2014.10.007.

[2] H. Hashem and D. Ranc, "An integrative modeling of BigData processing," *Int. J. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 1–15, 2015.

[3] T. A. S. Foundation, "MapReduce Tutorial," *Source*, 2008.

[4] A. Aggarwal, *Managing big data integration in the public sector*. 2015. doi: 10.4018/978-1-4666-9649-5.

[5] T. White, *Hadoop: The Definitive Guide*. 2009.

[6] B. Marr, "Why only one of the 5 Vs of big data really matters," *IBM - Big Data Anal. Hub*, 2015.

[7] Y. Permana, "Mengenal Big Data," May 29, 2016. https://codepolitan.com/blog/mengenal-big-data (accessed Nov. 14, 2022).

[8] GamatechnoBlog, "Mengulas Lengkap Tentang Hadoop: Software Pengelolaan Big Data - Blog Gamatechno," Jun. 15, 2017. https://blog.gamatechno.com/software-hadoop-big-data/ (accessed Nov. 14, 2022).

[9] S. Widy, "Hadoop Distributed File System. Pada artikel sebelumnya saya telah… | by Sasongko Widy | SkyshiDigital | Medium," Aug. 21, 2017. https://medium.com/skyshidigital/hadoop-distributed-file-system-c1f5c29e9e6e

(accessed Nov. 14, 2022).

[10] G. Turkington, *Hadoop Beginner ' s Guide*. 2013.

[11] H. S. Bhosale and D. P. Gadekar, "A Review Paper on Big Data and Hadoop," *Int. J. Sci. Res. Publ.*, vol. 4, no. 10, 2014.

[12] S. Pan, "The Performance Comparison of Hadoop and Spark," *Culminating Proj. Comput. Sci. Inf. Technol. 7*, 2016.

[13] S. Humbetov, "Data-intensive computing with map-reduce and Hadoop," in *2012 6th International Conference on Application of Information and Communication Technologies, AICT 2012 - Proceedings*, 2012. doi: 10.1109/ICAICT.2012.6398489.

[14] "Apache Hadoop 2.7.4 – HDFS Architecture." https://hadoop.apache.org/docs/r2.7.4/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html (accessed Nov. 15, 2022).

[15] B. Lublinsky, K. . Smith, and A. Yakubovich, "Professional Hadoop Solutions - Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich - Google Buku." https://books.google.co.id/books?id=od-EAAAAQBAJ&printsec=frontcover&hl=id&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false (accessed Nov. 15, 2022).