BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dengan besarnya kebutuhan data yang digunakan pada saat ini maka terjadi peningkatan suatu data yang akan terus bertambah. Pada tiap tahunnya kebutuhan data elektronik akan terus meningkat hingga 40%. Dan diperkirakan pada tahun 2020 kebutuhan data bisa hampir mencapai 45 ZB (Oracle,2012). Besarnya suatu data inilah disebut dengan istilah *Big Data. Big data* merupakan sekumpulan data berskala besar sehingga sangat sulit untuk diproses (Edd Dumbill,2012). Pada Big Data dikenal istilah 3V yaitu, *Volume, Variety,* dan *Velocity. Volume* disini mengacu pada jumlah ukuran data, *Variety* yaitu, jenis atau tipe data yang akan dikelola kompleksitasnya, sedangan pada *Velocity* yaitu, kecepatan pada pemrosesan data (Azzeddine, 2015). Semakin besar ukuran data yang diolah justru tidak berjalan seimbang dengan bagaimana kecepatan data itu diolah dan diproses. Dengan kebutuhan data yang bertambah besar dimana sistem teknologi komputer secara *konvensional* tidak bisa menangani lagi, hal tersebut dapat diatasi dengan menggunakan Apache Hadoop.

Hadoop merupakan sebuah *framework* yang digunakan untuk pengolahan data dengan skala besar dan tersimpan dalam sekelompok komputer yang saling terhubung dalam suatu jaringan secara terdistribusi dan berjalan diatas *cluster*. Pada Hadoop terdapat tiga komponen utama didalamnya yaitu, *MapReduce,* Hadoop *Distributed File* System (HDFS) dan *Yet Another Resource Negotiator* (YARN). MapReduce merupakan model pemrograman yang digunakan untuk melakukan pemrosesan data dengan skala besar. HDFS merupakan sistem *file* terdistribusi yang digunakan sebagai tempat penyimpanan dan pengolahan data berukuran besar. YARN digunakan untuk mengatur sumber daya komputasi dalam *cluster* dan *scheduling* Hadoop. Untuk melakukan pengolahan data yang besar diperlukan sebuah pemrosesan terhadap *job*. Jumlah *job* yang harus dieksekusi harus lebih besar dari jumlah mesin yang tersedia. Sehingga kondisi ini harus membutuhkan sebuah antrian. Untuk memilih salah satu *job* yang harus dieksekusi dalam sebuah antrian maka diperlukan scheduling.

Pada penelitian sebelumnya, Alfian Dzulfikar K. melakukan penelitian tentang analisis algoritme FIFO dan *Capacity Scheduling* pada Hadoop versi 1. Dengan jenis *job* yaitu, *job wordcount, job grep,* dan *job randomtextwritter* dan parameter pengujian yaitu, *Job Fail Rate, Latency, dan response time* (Alfian,2015). Penelitian kedua dilakukan oleh Komaratih Dian P. tentang analisis algoritme FIFO dan *Delay Scheduling* pada Hadoop versi 1 dengan jenis *job* yaitu, *job wordcount, job grep,* dan *job randomtextwritter* dan parameter pengujian yaitu, *Job Fail Rate, Latency, dan response time* (Komaratih,2015). Penelitian ketiga dilakukan oleh Tri Retno P. tentang analisis penggabungan *Fair Share Scheduling*, dan *Delay Improve*

Fair Share Scheduling pada Hadoop versi 1. Dengan jenis *job* yaitu, *job wordcount, job grep,* dan *job randomtextwritter* dan parameter pengujian yaitu, *Job Fail Rate, Latency, dan response time* (Tri Retno,2016). Perbedaan antara penelitian ini dengan penelitian sebelumnya yang digunakan sebagai referensi adalah pengunaan versi Hadoop, algoritme *job scheduling*, penggunaan jenis *job*, jumlah *job,* dan skenario pengujian yang dilakukan juga berbeda.

Penelitian ini akan membandingan dua algoritme penjadwalan yaitu, *Fair Share Scheduling* dan *Capacity Scheduling*. *Fair Share Scheduling* adalah metode alokasi pembagian resource pada antrian secara adil untuk seluruh *job* yang masuk pada antrian. *Capacity Scheduling* mendukung antrian secara hirarki yaitu, dapat membagi resource yang tersedia pada *cluster* ke beberapa antrian dan terdapat fitur prediksi *job* sehingga dapat meminimalkan nilai *failed* pada *job* yang akan dijalankan (Tom White, 2015). Dengan melakukan perbandingan tersebut bertujuan untuk mengetahui kinerja terbaik diantara kedua algoritme penjadawalan tersebut pada pengiriman *job* pada *cluster* Hadoop berdasarkan pada parameter pengujian yang telah digunakan.

Pengujian ini dilakukan dengan 3 skenario pengujian yang berbeda berdasarkan pada variabel bebas dan variabel tetap. Untuk skenario pengujian pertama menggunakan variabel bebas yaitu, variasi ukuran data dan variabel tetap yaitu, jumlah *job* dan jenis *job*. Skenario pengujian kedua menggunakan variabel bebas yaitu, variasi jumlah *job* dan variabel tetap yaitu, ukuran data dan jenis *job*. Skenario pengujian ketiga menggunakan variabel bebas yaitu, variasi jenis *job* dan variabel tetap yaitu, ukuran data dan jumlah *job*. Pada pengujian ini akan menggunakan parameter pembanding yaitu, *Job Fail Rate, Latency, dan Throughput* sebagai acuan perhitungan kinerja sistem untuk mendapatkan hasil berupa grafik nilai yang dilakukan untuk perbandingan kinerja antara kedua algoritme tersebut.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan pada sebelumnya, maka permasalahan dalam penelitian ini dapat diidentifikasikan sebagai berikut:

1. Bagaimana perancangan lingkungan dan skenario pengujian terhadap perbandingan kinerja algoritme *Fair Share Scheduling* dengan *Capacity Scheduling* pada pengiriman *job ?*
2. Bagaimana hasil analisis dari pengujian kinerja algoritme *scheduling* yang lebih baik pada saat melakukan pengiriman *job* berdasarkan nilai grafik pada parameter pengujian *Job Fail Rate, Latency, dan Throughput?*

1.3 Tujuan

Pada tujuan penelitian ini digunakan untuk menjawab permasalahan dalam penelitian ini sesuai hasil yang ingin dicapai sebagai berikut.

1. Dapat melakukan perancangan lingkungan Hadoop multi*node* dengan menggunakan 1 *node* komputer *master* dan 5 *node* komputer *slave* untuk melakukan pengujian terhadap algoritme *Fair Share Scheduling* dan *Capacity Scheduling.*
2. Dapat melakukan 3 skenario pengujian dengan perbedaan pada variasi jumlah ukuran data, variasi jumlah *job*, dan variasi jenis *job* pada algoritme *Fair Share Scheduling* dan *Capacity Scheduling.*
3. Untuk mengetahui grafik nilai pada parameter *Job Fail Rate, Latency, dan Throughput* setelah melakukan pengujian terhadap algoritme *Fair Share Scheduling* dan *Capacity Scheduling*.
4. Dapat menjelaskan hasil proses analisis dari percobaan yang telah dilakukan dengan menemukan solusi kinerja terbaik antara algoritme *Fair Share Scheduling* dan *Capacity Scheduling*.

1.4 Manfaat

Manfaat yang diperoleh dari penelian ini adalah sebagai berikut :

Bagi penulis :

1. Sebagai media untuk melakukan implementasi terhadap penerapan ilmu selama mengikuti matakuliah yang diperoleh dari Teknik Informatika Universitas Brawijaya.
2. Mendapatkan pemahaman tentang penerapan sistem Hadoop terhadap implementasi dari algoritme *Fair Share Scheduling* dengan *Capacity Scheduling*.

Bagi pembaca :

Memberikan referensi terhadap pemilihan algoritme *Fair Share Scheduling* dengan *Capacity Scheduling* berdasarkan dengan hasil kinerja terbaik dengan menggunakan parameter *Job Fail Rate, Latency,* dan *Throughput* sebagai parameter pembandingnya.

1.5 Batasan Masalah

Batasan-batasan masalah yang terdapat dalam penelitian ini adalah :

1. Satu buah komputer sebagai *master* dan *slave* dengan menggunakan sistem operasi Linux Ubuntu versi 14.04.1 LTS 64 bit.
2. Lingkungan pengujian ini menggunakan versi Apache Hadoop-2.7.3.
3. Menggunakan *cluster* Hadoop *multinode.*
4. *Slave* yang mengakses *job* ke *master* berjumlah 5 *node*.
5. Menggunakan algoritme penjadwalan *Fair Share Scheduling* dan *Capacity Scheduling.*
6. Parameter yang diuji adalah nilai *Job Fail Rate, Latency,* dan *Throughput.*
7. Jumlah *job* yang digunakan untuk pengujian yaitu, 5 *jobs*, 10 *jobs*, dan 15 *jobs.*

8. Pengujian *job scheduling* ini dengan melakukan pengiriman *job* pada Hadoop yaitu, *Job wordcount dan job wordmean.*

1.6 Sistematika Penulisan

Sistematika penulisan ini ditujukkan untuk memahami mengenai gambaran umum tentang penulisan pembuatan laporan skripsi secara garis besar yang meliputi beberapa bab sebagai berikut:

BAB I Pendahuluan

Pada bab ini akan dibahas mengenai hal-hal yang menjadi latar belakang dilakukan penelitian sebagai alasan pemilihan judul, identifikasi dan perumusan masalah, penentuan tujuan, penentuan batasan masalah yang digunakan selama melakukan penelitian, manfaat penelitian ini dari segi pengguna dan dari segi penulis, dan sistematika penulisan.

BAB II Landasan Kepustakaan

Pada bab ini akan menjelaskan tentang pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah yang berkaitan dengan analisis perbandingan terhadap algoritme *Fair Share Scheduling* dengan *Capacity Scheduling* untuk menunjukkan hasil kinerja terbaik pada proses pengiriman *job* di Hadoop *cluster*.

BAB III Metode Penilitian

Pada bab ini akan menjelaskan tentang metode dan langkah kerja yang akan digunakan pada penelitian yang terdiri dari studi literature, perancangan perangkat lunak, implementasi perangkat lunak, pengujuian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV Implementasi dan Pengujian

Pada bab ini akan membahas tentang proses instalasi dan konfigurasi sistem Hadoop untuk melakukan analisis algoritme *Fair Share Scheduling* dengan *Capacity Scheduling* terhadap karakteristik *job* .

BAB V Hasil dan Analisis

Pada bab ini akan memuat tentang hasil pengujian dan analisis terhadap sistem yang telah direalisasikan dan diimplementasikan.

BAB VI Penutup

Pada bab ini akan dijelaskan mengenai kesimpulan yang dapat diambil dari hasil penelitian yang telah dilakukan dan saran-saran yang diperoleh dari pembuatan serta pengujian sistem untuk melakukan perkembangan sistem lebih lanjut.