Prototipe Sistem Perhitungan Surat Suara Dengan Platform Hadoop Menggunakan Map dan Reduce

Ananda Muhammad Nuriawan[1], Hilal H. Nuha[2], Sidik Prabowo[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[4]Divisi Digital Service PT Telekomunikasi Indonesia
[1]anandamuhammad@students.telkomuniversity.ac.id, [2] hilalnuha@gmail.com,
[3]sidikprabowo@telkomuniversity.ac.id

Abstrak

Negara Indonesia merupakan salah satu negara yang menganut sistem demokrasi dimana kepala negara dipilih oleh rakyat melalui pemilihan umum atau di singkat pemilu. Di dalam pemilu, setiap warga negara yang memiliki Kartu Tanda Penduduk (KTP) memiliki hak suara yang sama. Dengan jumlah tersebut menghitung hasil suara pada pemilu bukanlah hal mudah. Tugas akhir ini membuat sebuah implementasi perhitungan suara denganm menggunakan Hadoop dengan gambar formulir C1 sebagai inputnya. Angka pada gambar C1 tersebut diolah menjadi tipe data INT dengan menggunakan jaringan saraf tiruan (JST) yang dibuat dengan menggunakan *Restricted Boltzmann Machine* (RBM) dengan data latih dari MNIST. JST yang telah dibuat kemudian disimpan ke dalam *website* agar bisa menyimpan gambar ke dalam *database* yang ada di dalam *website*. Data yang disimpan di *website* tersebut di-*crawling* menggunakan Apache Nutch yang kemudian hasil dari *crawling* tersebut menjadi *job* yang diolah menggunakan algoritme *mapreduce* pada Hadoop. Pengujian sistem yang telah dibuat ini menunjukkan hasil yang cukup cepat pada rekapitulasi suara yang dilakukan dengan Apache Nutch dan Hadoop dengan *multimode* dan algoritme scheduling *fair scheduler* yaitu 85 detik. Sedangkan, Hadoop dengan konfigurasi *single node* dan algoritme *scheduling* FIFO menghasilkan waktu yang lebih baik yaitu 67 detik.

Kata kunci : Hadoop, *mapreduce*, apache nutch, Restricted Boltzmann Machine , formulir C1, job

Abstract

The Indonesian country is one of the countries that adheres to a democratic system in which the head of state is elected by the people through general elections or in short elections. In elections, every citizen who has a National Identity Card (KTP) has the same voting rights. With this number counting the vote results in elections is not easy. The final task is to make an implementation of vote counting by using Hadoop with a picture of form C1 as its input. The numbers in figure C1 are processed into INT data types using artificial neural networks (ANN) made using the Restricted Boltzmann Machine (RBM) with training data from MNIST. ANN that has been created is saved to the website so that it can save images into the database on the website. The data stored on the website is crawled using Apache Nutch, then the results of crawling are jobs that are processed using the Hadoop mapreduce algorithm. In testing the system that has been made this results in quite fast results on the sound recapitulation conducted with Apache Nutch and Hadoop with multimode and fair scheduler scheduling algorithms which are one minute twenty five seconds. Whereas Hadoop with a single node configuration and FIFO scheduling algorithm produces a better time of 67 seconds.

Keywords: Hadoop, *mapreduce*, apache nutch, Restricted Boltzmann Machine , C1 form, job

1. Pendahuluan dan Latar Belakang

Negara Indonesia merupakan salah satu negara yang menganut sistem demokrasi dimana kepala negara dipilih oleh rakyat melalui pemilihan umum atau di singkat pemilu. Di dalam pemilu, setiap warga negara yang memiliki Kartu Tanda Penduduk (KTP) memiliki hak suara yang sama. Penduduk Indonesia wajib KTP pada tahun 2017 adalah sebanyak 189 juta jiwa [1]. Jumlah sebanyak itu menjadikan data pemilih dalam pemilu, khususnya Pemilu untuk memilih presiden dan wakil presiden, sebagai *big data* karena memiliki jumlah data yang sangat banyak. Berdasarkan undang-undang (UU) Pemilu dan peraturan Komisi Pemilihan Umum (KPU), hasil akhir yang resmi dan sah adalah perhitungan manual yang ditetapkan oleh KPU. KPU harus menghitung dan merekapitulasi seluruh suara secara manual mulai dari tingkat Tempat Pemungutan Suara(TPS), Panitia Pemungutan Suara(PPS), Panitia Pemilihan Kecamatan(PPK), Komisi Pemilihan Umum Daerah(KPUD) Kabupaten/ Kota, KPUD Provinsi, hingga KPU Pusat. Hasil akhir baru bisa diumumkan ke publik beberapa hari kemudian setelah hari pemungutan suara . Karena hal tersebut, tugas akhir ini akan membuat sebuah sistem *quick count* dengan sample berupa gambar dari formulir C1 untuk menghitung dan merekapitulasi suara. Sistem yang dibuat menginputkan formulir C1 dengan menggunakan website lalu data dari website tersebut di *crawling* dengan menggunakan Apache Nutch dan kemudian hasil dari *crawling* tersebut diinput dan dihitung dengan menggunakan Hadoop dengan menggunakan algoritme scheduling *fair scheduling*.

Hadoop merupakan salah satu sistem file terdistribusi yang diperuntukan untuk mengerjakan *job* untuk data yang masuk kategori besar (*Big Data*) [2]. Hadoop sendiri memiliki dua arsitektur dasar yaitu *mapreduce* dan *Hadoop Distributed File System* (HDFS) yang membuat Hadoop sangat optimal dalam menangani *big data* [3].

Tugas akhir ini mengimplementasikan pengubahan gambar formulir C1 menjadi sebuah INT yang disimpan di dalam database website. Data tersebut di*crawling* menggunakan Apache Nutch, dan digunakan sebagai *job* pada Hadoop dan mengujikan *Conpletetion Time*, *CPU Time*, *Memory Usage* dan *turnaround time* sebagai analisis performansi waktu pada sistem yang dibuat.

1.1. Perumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, terdapat permasalahan yang muncul dalam tugas akhir ini, yaitu sebagai berikut :

1. Bagaimana implementasi perhitungan dan rekapitulasi suara pemilu dengan menggunkana Hadoop.
2. Bagaimana cara mengubah data gambar C1 menjadi integer yang dapat di baca komputer.
3. Bagaimana performansi waktu perhitungan suara yang didapat dengan menggunakan Hadoop.

1.2. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

1. Data yang digunakan adalah C1 yang digunakan untuk perhitungan suara di TPS
2. C1 di foto menggunakan kamera *SmartPhone* dengan resolusi 12 *Megapixel* yang menghasilkan gambar sekitar 3000 x 4000 *pixel*
3. Format gambar yang digunakan adalah .jpg
4. Posisi C1 pada gambar harus sama
5. Teks hasil *crawling* yang dimasukan di Hadoop adalah teks yang berisi tabel jumlah suara
6. Scheduler yang digunakan pada Hadoop adalah *fair scheduler*

Tujuan

Tujuan dari tugas akhir ini adalah :

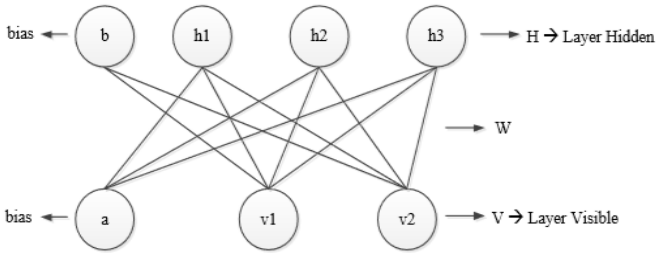
1. Membuat Website yang dapat diakses oleh *smartphone* dan komputer untuk menginputkan data berupa formulir C1.
2. Membuat gambar formulir C1 yang diinputkan dapat diolah menjadi data *integer* yang dapat disimpan dalam database.
3. Membuat perhitungan rekapitulasi suara menggunakan Hadoop, dan menganalisis performansinya .

2. Landasan Teori

2.1. *Restricted Boltzmann Machine*

Restricted Boltzmann Machine (RBM) adalah salah satu metode Jaringan Syaraf Tiruan (JST) dan juga merupakan salah satu *Deep Learning* yang menggunakan *layer* lebih dari satu [4]. RBM merupakan jaringan saraf yang bersifat stokastik yang berarti setiap neuron saling terhubung yang diaktivasi memiliki unsur probabilitas [5] [6]. RBM biasanya memiliki dua *layer* yaitu *visible layer* merupakan *state* yang akan diobservasi dan *hidden layer* merupakan *feature detectors* serta unit bias atau dilambangkan dengan b [5]. Selanjutnya masing-

masing *visible unit* terhubung ke semua *hidden unit* yang diwakili oleh bobot atau dilambangkan dengan W [5]. *Visible layer* hanya memiliki koneksi ke *hidden layer*, tidak ada koneksi antar neuron di visible dan hidden layer. Hubungan ini memungkinkan untuk transfer informasi dua arah karena itu JST ini di sebut *Restricted* [7] [8].



Gambar 1 *layer* pada RBM **[9]**

Arsitektur pada gambar 1 menunjukan RBM yang digunakan untuk melatih JST menggunakan dataset melalui layer *visible* dan *hidden*.

$$F(X_i) = \frac{1}{1+Exp(-X_i)} \qquad i = 1,2,3,\ldots.k \qquad (1)$$

$$F(X_i) = \frac{Exp(X_i)}{\sum_{j=0}^{k} Exp(X_j)} \qquad i = 1,2,3\ldots.k \qquad (2)$$

Persamaan 1 adalah fungsi sigmoid yang digunakan untuk aktivasi *visible* unit dan persamaan 2 adalah fungsi *softmax* yang digunakan untuk aktivasi *hidden* unit yang akan menghasilkan probabilitas [5].

2.2. MNIST Data set

MNIST Data yang digunakan merupakan dataset untuk mengenali tulisan tangan manusia. Di dalam dataset Mnist terdapat 600 sample dari 60000 data tulisan tangan untuk pembelajaran [5] [6]. Dataset MNIST ini dipakai untuk menjadi *training set* untuk melatih *RBM* yang digunakan. Dataset MNIST ini dipakai untuk menjadi *training set* untuk melatih *RBM* yang digunakan berikut gambar 2 Adalah contoh dari gambar test MNIST dan gambar 3 adalah contoh dari *data training* MNIST

Gambar 2 Data Test MNIST

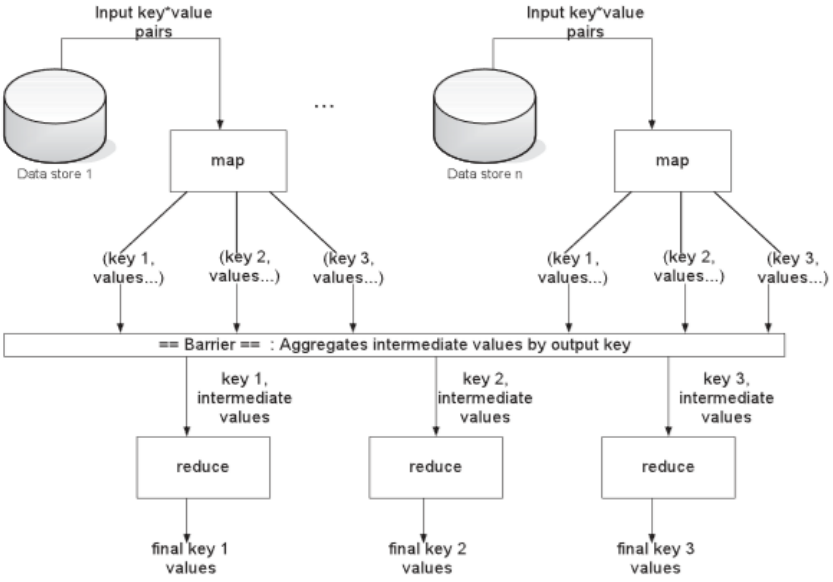
Gambar 3 Data Training MNIST

2.3. Apache Nutch

Apache nutch adalah *web crawler open source* yang biasa digunakan untuk crawling *websites* [10]. Apache nutch menggunakan sistem yang terdistribusi seperti pada hadoop**.** Apache nutch memiliki kemampuan *parsing*, dan juga *indexing* sehingga memungkinkan untuk dipakai sebagai mesin pencari sesuai kebutuhan. Apache Nutch di dalam tugas akhir ini meng-*crawling* data HTML di *website* yang dibuat dalam bentuk *binary* file yang di*convert* ke dalam bentuk teks.

2.4. *MapReduce Application*

MapReduce adalah model program yang dibuat berdasarkan komputasi terdistribusi [11]. Pemrograman MapReduce memiliki dua fungsi penting yaitu fungsi *Map* dan fungsi *Reduce*. Fungsi *map* berfungsi untuk mengambil input data sebagai satu set pasangan kunci dan nilai dari kunci tersebut dan menghasilkan output sebagai satu set pasangan kunci dan nilainya [11]. Fungsi *reduce* berfungsi untuk menerima pasangan kunci dan nilainya yang dihasilkan oleh fungsi *map* lalu mengatur nilai dari kunci tersebut [11]. Berikut gambar 4 adalah alur data dari algoritme *map* dan *reduce.*

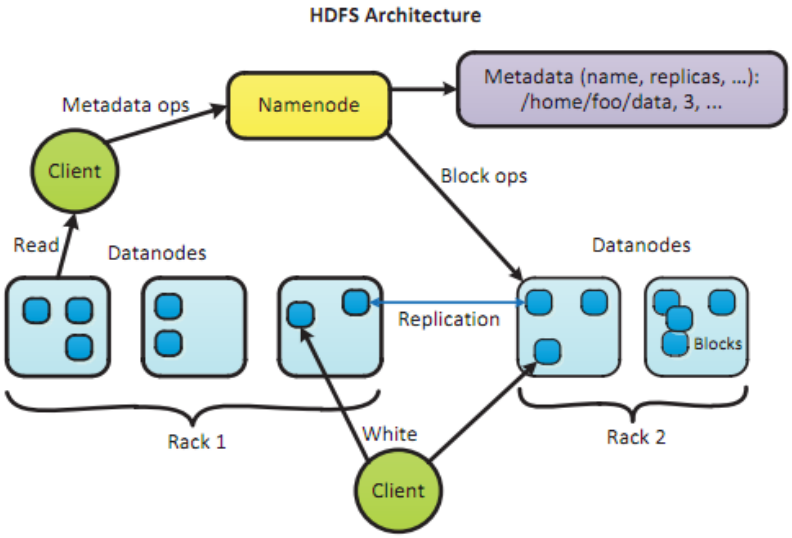


Gambar 4 Alur data pada algoritme *mapreduce* [12]

2.5. Hadoop

Hadoop adalah *open source software framework* yang digunakan untuk memproses data yang besar atau *big data* [11]. Hadoop mempunyai empat modul yaitu :

1. Hadoop *distributed file system* (HDFS)
   HDFS adalah sistem penyimpanan terdistribusi yang dapat menyimpan file secara terdistribusi pada HDFS node. Gambar 5 menunjukkan tentang arsitektur HDFS

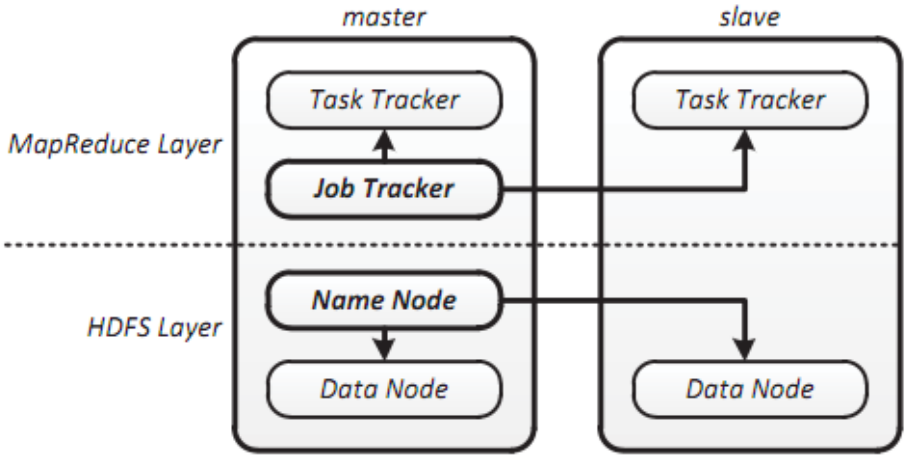


Gambar 5 Arsitektur HDFS **[3]**

2. Hadoop *common*
   Hadoop common adalah *libraries* dan *tools* yang dibutuhkan oleh modul Hadoop lainnya.
3. *Map reduce*
   Map reduce adalah sebuah model program untuk teknik pengolahan berdasarkan komputasi terdistribusi.
4. Hadoop Yarn
   Hadoop yarn digunakan untuk mengatur *resource* yang akan digunakan.

2.6. Job dan *Job Scheduling*

Job adalah pekerjaan yang diberikan oleh *user* untuk dikerjakan oleh sistem Hadoop. *Job* sendiri dapat diberikan oleh *user* yang terhubung ke *server* atau diberikan oleh *server* secara langsung. *Job* di dalam Hadoop dibagi menjadi dua yaitu *map* dan *reduce* [11]. Gambar 6 menunjukan lapisan Hadoop dengan *multi – node*.



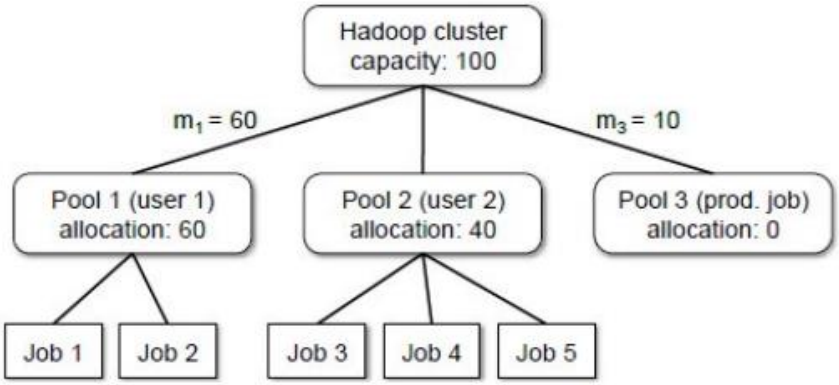
Gambar 6 Lapasin pada Hadoop **[3]**

Map adalah bagian awal dari *job* pada Hadoop yang bertujuan untuk membangun informasi yang terdapat pada data, menjadi data yang siap untuk diproses sesuai dari *job* yang akan diproses oleh server [4]. Sedangkan *reduce* adalah suatu proses yang dikerjakan oleh *server* [12] untuk melakukan proses sesuai dengan pekerjaan yang diberikan oleh *user* atau *server* pada data yang telah melewati proses *map* [12].

Job scheduling adalah cara di dalam Hadoop untuk melakukan pengolaan pada pemrosesan data agar pekerjaan yang dikerjakan sesuai dengan yang diharapkan. Dalam kondisi *default*, Hadoop menggunakan algoritme *First In First Out* (FIFO) dalam *job scheduling* ini [12].

2.7. Fair Scheduling

Fair scheduling adalah salah satu algoritme *job scheduling* pada Hadoop yang mengalokasikan *resource* yang sama pada setiap *job* yang diberikan [13]. Pada *fair scheduler*, jika ada *job* yang dikerjakan, maka algoritme ini akan memberikan slot kosong untuk diisi *job* baru tersebut. Dengan cara tersebut, setiap *job* akan mendapatkan resource yang sama dan dan *job* yang kecil tidak perlu menunggu lama untuk dieksekusi. Gambar 7 menunjukkan ilustrasi *pool* data yang dipakai oleh *fair scheduling* dimana *m* adalah jumlah dari *resource* yang dibagikan.



Gambar 7 Pooldata yang dipakai oleh *fair scheduler* [13**]**

2.8. Parameter Pengujian

Parameter yang digunakan untuk pengujian pada tugas akhir ini adalah uji akurasi terhadap *testing* data MNIST pada PHP, uji *average completion time*, *average cpu time*, *average memory usage*, dan *turnaround time* pada Hadoop.

1. Uji akurasi data *testing* MNIST
   Parameter ini diukur akurasi dari setiap angka dari angka satu sampai sembilan terhadap data *testing* yang dimiliki oleh MNIST. Terdapat 350 data *testing testing* yang di digunakan. Satuan yang digunakan adalah persentase (%) keberhasilan prediksi ketepatan gambar.

2. Uji akurasi pada sample formulir C1
   parameter ini diukur tingkat akurasi dari enam sample formulir C1 yang telah difoto.

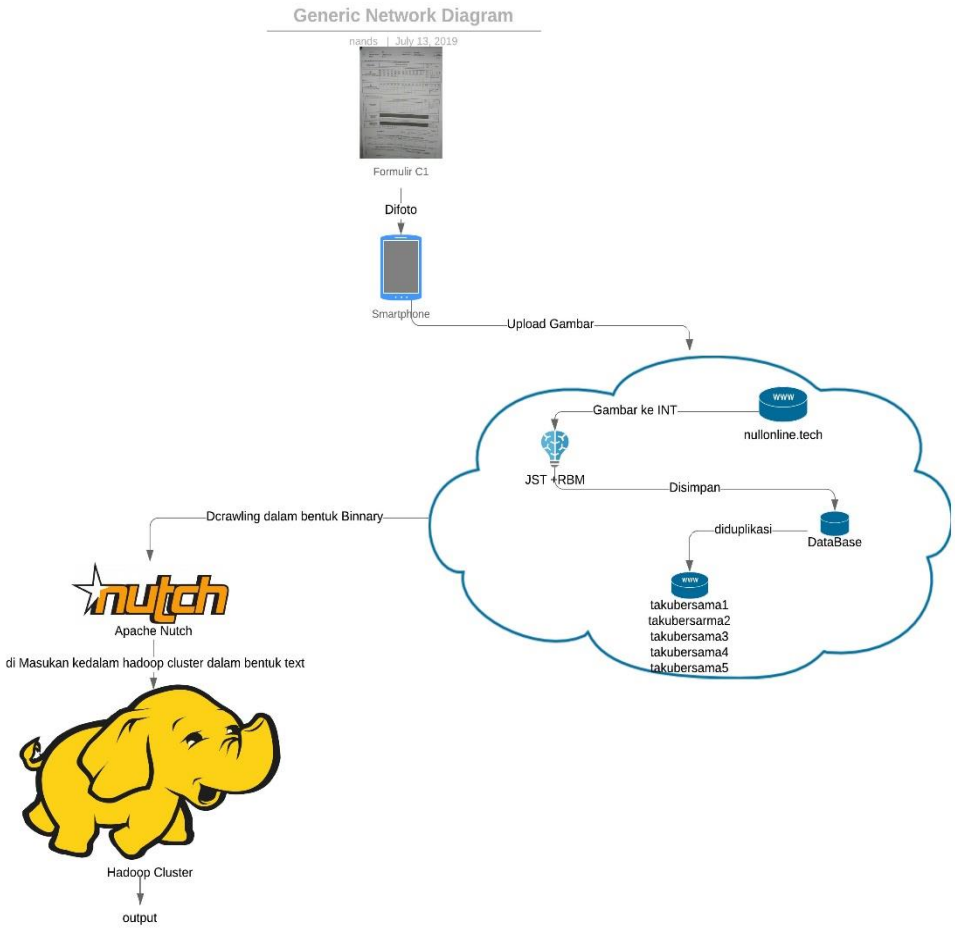
3. *Average Completion Time* [14]
   Parameter ini diukur waktu rata-rata penyelesaian satu *job* dari semua *job* yang diinputkan. Karena job yang dipakai hanya satu maka diukur waktu dalam menyelesaikan job yang diberikan.

4. *Average Memory Usage* [14]
   Parameter ini diukur dari rata-rata penggunaan memory dalam menyelesaikan job yang telah diinputkan.

5. *Average CPU Time* [11]
   Parameter ini diukur dari jumlah waktu penggunaan CPU dalam menyelesaikan job yang mana parameter ini dilandaskan pada CPU Time [15].

6. *Turnaround Time*
   Parameter ini diukur total waktu yang dibutuhkan untuk menyelesaikan satu skenario. Nilai ini akan diperoleh dari jumlah keseluruhan waktu yang dibutuhkan untuk menyelesaikan suatu skenario [15]. Karena job yang di pakai dalam tugas akhir ini hanya satu job maka scenario yang dipakai adalah waktu dalam menyelesaikan *job* dan waktu yang dipakai dalam menyelesaikan hasil *crawling*. Satuan yang dipakai adalah menit atau detik.

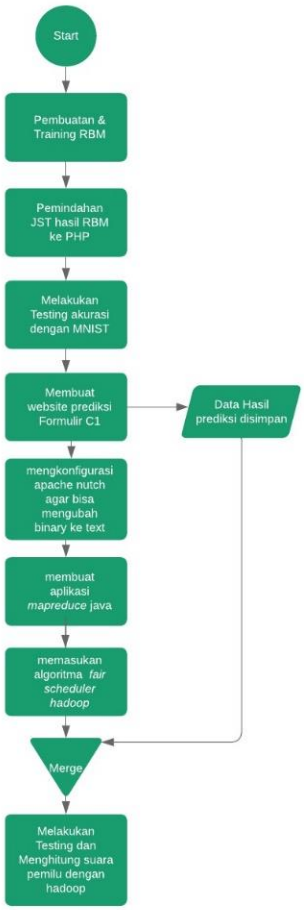
3. Sistem yang Dibangun



Gambar 8 Perancangan Seluruh Sistem

Gambar 8 adalah perancangan sistem secara keseluruhan dimana user memfoto formulir C1 menggunakan kamera *smartphone* dan kemudian foto tersebut di upload ke website www.nullonline.tech dimana dalam website tersebut terdapat *JST* dari *RBM* yang digunakan untuk mengolah formulir c1 menjadi data INT. Selanjutnya data integer tersebut disimpan didalam database dan file di dalam website www.nullonline.tech. Setelah menjadi text yang disimpan, suara dari C1 tersebut di crawling oleh apache nutch dalam bentuk text agar bisa menjadi *job* yang akan diolah oleh algoritme *mapreduce* pada Hadoop. Tugas akhir ini juga menyapkan lima website lain yang merupakan duplikasi dari tabel database dari website www.nullonline.tech untuk menyimpan jumlah suara lainnya untuk menguji performansi crawling dari apache nutch dan Hadoop.

3.1. Perancangan Sistem



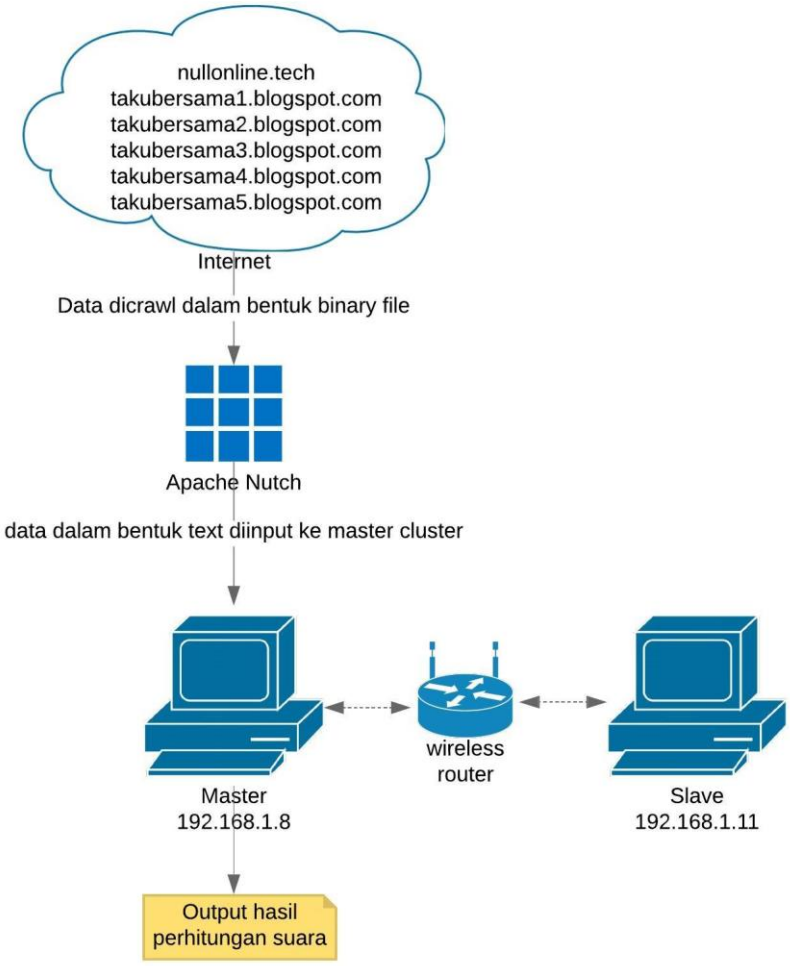
Gambar 9 Flow chart perancangan sistem

Gambar 9 adalah *flowchart* dari perancangan sistem yang akan dibuat dalam tugas akhir ini dan akan dijelaskan sebagai berikut:

1. Membuat algoritme RBM menggunakan Bahasa pemogramam Matlab dengan *visible layer* sebanyak 78.400, *hidden layer* sebanyak 1000, dan *epoch* sebanyak 100.
2. Melatih RBM yang telah dibuat dengan dataset training angka dari MNIST.
3. Membuat website dengan Bahasa pemograman PHP dengan fitur upload gambar dengan format .jpg.
4. Memindahkan JST hasil latih RBM dari Matlab ke PHP.
5. menghitung akurasi digit 0 – 9 dengan dataset MNIST.
6. Membuat website dengan JST hasil latih RBM tersebut dapat membaca jumlah suara pada formulir c1 yang digunakan pada pemilu.
7. Menyimpan hasil dari pembacaan jumlah suara pada formulir c1 di database dan file di dalam website yang telah dibuat

8. Melakukan *crawling* pada website yang telah dibuat untuk membuat data jumlah suara mejadi bentuk teks (.txt) agar bisa di jadikan job oleh Hadoop.
9. Membuat aplikasi *mapreduce* untuk dipasangkan pada Hadoop.
10. Memasukan *fair scheduler* ke dalam YARN pada Hadoop.
11. Melakukan perhitungan suara dengan file text yang telah di crawling dengan hadoop

3.2. Arsitektur Hadoop dan Nutch



Gambar 10 Arsitektur Hadoop dan Nutch

Gambar 10 di atas adalah arsitektur dari Hadoop dan apache nutch yang akan di buat. Menggunakan dua cluster. Satu cluster berperan menjadi *master cluster* dan satu *cluster* berperan sebagai *slave cluster*. Algoritme *scheduling* yang digunakan adalah *fair scheduler* dengan total enam website yang akan di crawling untuk menhitung jumlah formulir C1 yang ada di dalam wesite yang akan di*crawling* tersebut.

Komponen Perangkat Keras

Komponen perangkat keras yang digunakan dalam tugas akhir ini adalah sebagai berikut :

1. Satu komputer *server* yang berperan sebagai *master server* dengan prosesor *Intel Core I5 – 5200U* 2.20 GHz, RAM 6 GB, dan Harddisk 500 GB.
2. Satu komputer *server* yang berperan sebagai *slave server* dengan prosesor *Intel Core I5 – 5200U* 2.20 GHz, RAM 4 GB, dan Harddisk 1 TB.
3. Dua *smartphone* untuk memfoto formulir C1 yang mempunyai kamera dengan resolusi 12 *megapixel*.

4. Satu *wireless* router untuk membuat jaringan antara master dan slave dalam cluster Hadoop.

Komponen Perangkat Lunak

Komponen perangkat lunak yang digunakan dalam tugas akhir ini adalah sebagai berikut :

1. Sistem Operasi
   Sistem operasi yang digunakan pada komputer *cluster* Hadoop dan Apache Nutch adalah Linux Ubuntu 18.04 LTS dan untuk membuat RBM dan website adalah Windows 10 Education.

2. *Server* Hadoop
   Server Hadoop yang digunakan adalah Hadoop 3.2.0.
3. Matlab
   Untuk membuat dan melatih *layer* pada JST dengan RBM.
4. Java
   Java digunakan untuk menunjang penggunaan Hadoop *distributed file system,* dan membuat eksekutor untuk *job* yang diberikan. Pada Hadoop *cluster* akan diperlukan Java *Run Time Environment* (JRE) dan *Java Development Kit* (JDK) dengan versi java .
5. Open SSH
   Open SSH karena server Hadoop berjalan di atas server SSH.
6. PHP
   PHP digunakan untuk membuat website yang akan digunakan untuk mengidentifikasi jumlah suara pada form c1.
7. Mozilla Firefox
   Mozilla Firefox akan digunakan sebagai *browser* untuk portal untuk *memonitoring* Hadoop secara *real – time* dan mengakses website yang telah dibuat.

3.3. *Map* dan *Reduce* Application

Tugas akhir ini membahas aplikasi *map* dan *reduce* memiliki peran yang cukup penting dalam perhitungan suara. *Class map* berfungsi untuk mencari kata kunci yaitu paslon01, paslon02, dan suara tidak sah beserta nilai dari kata kunci tersebut dari tabel HTML yang telah di*crawler* sebelumnya. *Class reduce* berfungsi untuk menjumlahkan nilai pada masing -masing kata kunci yang telah diberikan oleh *class map*. Aplikasi *map* dan *reduce* dalam tugas akhir ini ditulis dalam Bahasa pemograman java. Gambar 11 adalah pseudo code dari *class map* dan gambar 12 adalah pseudo code dari *class reduce* dalam tugas akhir ini.

```
map(key : String, value : String)
  key <- document name
  value <- document contents
  i: integer
  word : String
  i <- 0
  Procedure Map (key, value)
        for each word x in value:
                if x is empty
                        continue
                else if x is number
                        if i equal to 1
                                i <- i+1
                                word <- "Paslon01"
                                output(word,x)
                        else if i equal to 2
                                i <- i+1
                                word <- "Paslon02"
                                output(word,x)
                        else if i equal to 3
                                i <- 1
                                word <- "Suara Tidak Sah"
                                output(word,x)
                else
                        continue
```

Gambar 11Pseudo code Map

```
reduce(key: Srting, value : integer)
  values <- a list of value
  sum : integer
  sum <- 0
  procedure Reduce (key, values)
        for each value in values
                sum <- value+sum
        output(key, sum)
```

Gambar 12 Pseudo code Reduce

4. Evaluasi

4.1 Hasil Pengujian

Dengan implementasi system seperti yang dijelaskan pada bab sebelumnya terdapat hasil pengujian yaitu :

1. Uji akurasi pada Testing data MNIST

Tabel 1 Uji akurasi data MNIST

Angka	Jumlah Digit	Kebenaran Prediksi	Akurasi(%)
0	44	41	93,18182
1	34	30	88,23529
2	28	21	75
3	46	35	76,08696
4	33	21	63,63636
5	31	19	61,29032
6	36	32	88,88889
7	36	24	66,66667
8	32	28	87,5
9	30	22	73,33333
TOTAL	350	273	78

Tabel 1 menunjukan tingkat akurasi setiap angka berbeda -beda. Dan system ini menghasilkan akurasi total dari gambar *testing* MNIST rata - rata sebesar 78%. Dari hasil tersebut *layer* RBM yang telah dilatih digunakan untuk mengidentifikasi angka yang ada didalam formulir c1 KPU.

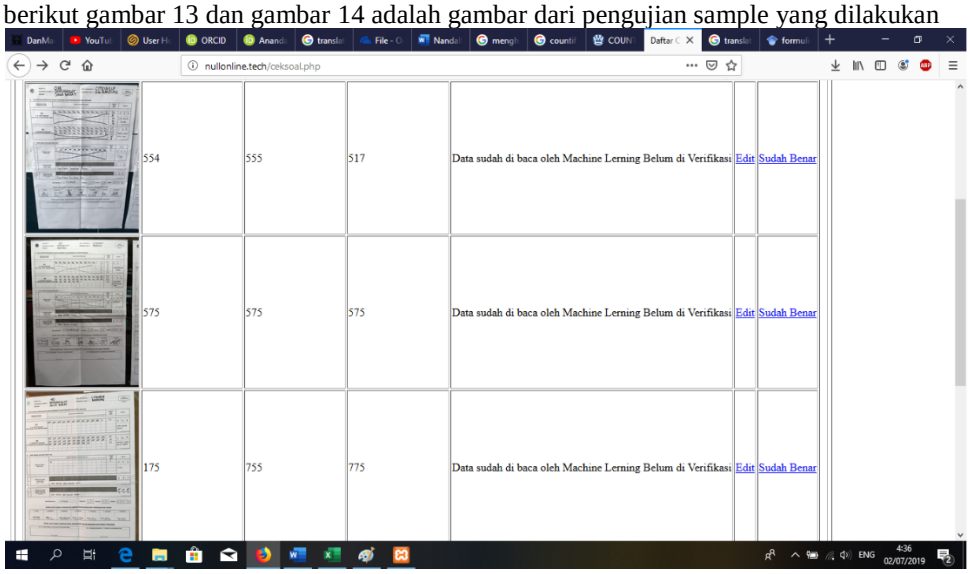
2. Uji akurasi pada Formulir C1

Tabel 2 uji akurasi formulir C1

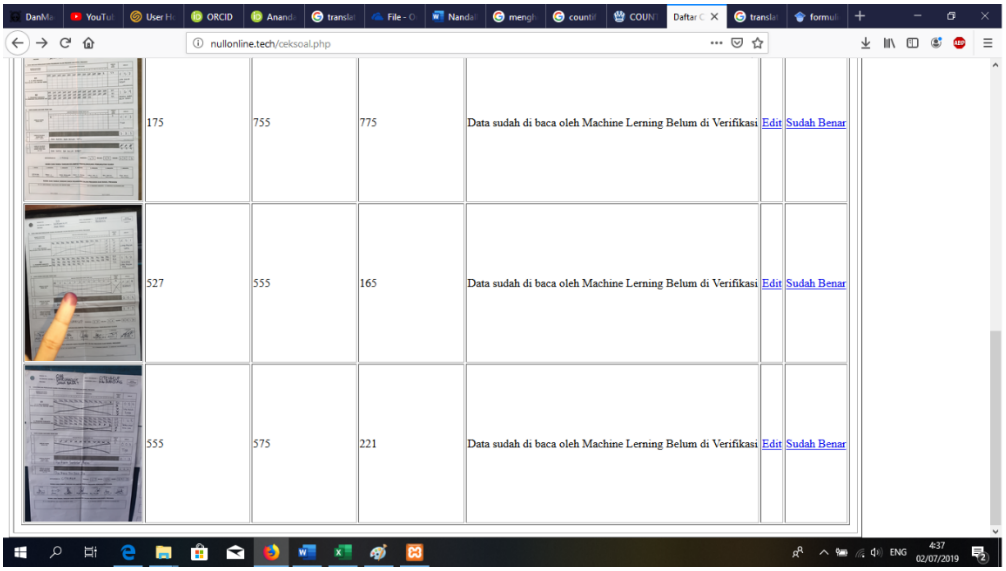
No Pengujian	Data Dalam C1			Prediksi			kebenaran prediksi	Jumlah
1	0	5	4	5	5	4	7	18
	1	6	5	5	5	5	Total(%)	38,88889
	0	0	3	5	1	7		
2	x	5	1	5	7	5		
	1	5	2	5	7	5		
	x	x	4	5	7	5		
3	x	5	7	1	7	5		
	1	6	4	7	5	5		
	x	x	3	7	5	5		
4	0	5	4	5	5	5		
	1	6	5	5	7	5		

	0	0	3	2	2	1		
5	x	5	1	5	2	7		
	1	5	2	5	5	5		
	x	x	4	1	6	5		
6	x	5	1	7	3	5		
	1	5	2	5	5	7		
	x	x	4	7	5	5		

dari pengujian 6 sample foto C1 tersebut didapatkan akurasi total pada prediksi formulir C1 adalah 38.9%. berikut gambar 13 dan gambar 14 adalah gambar dari pengujian sample yang dilakukan

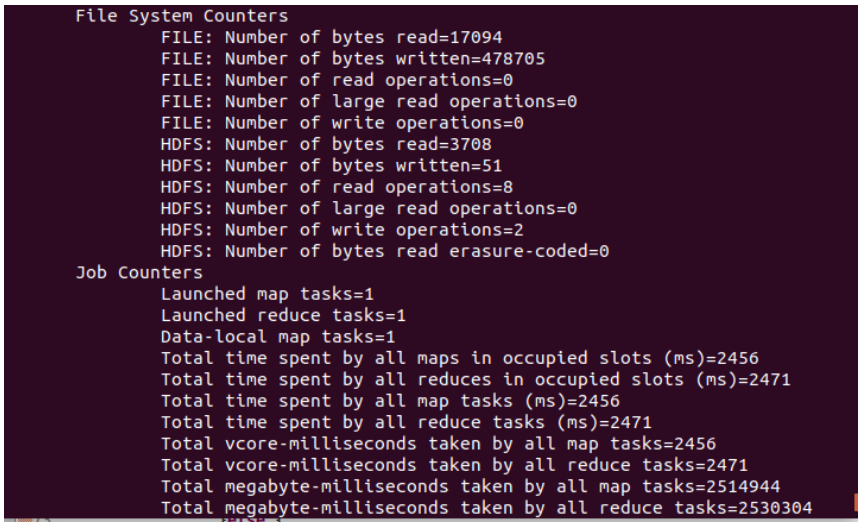


Gambar 13 sample pengidetifikasian gambar C1



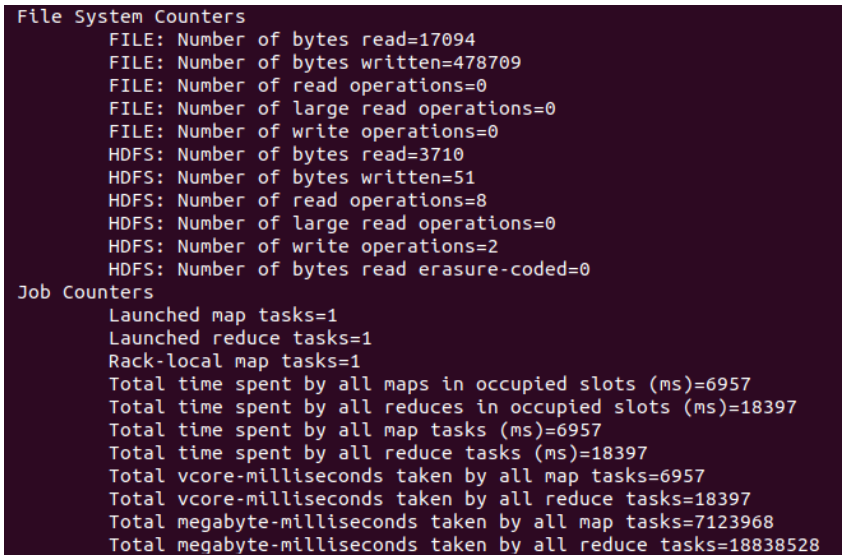
Gambar 14 sample pengidetifikasian gambar C1

2. Uji *Completion Time, CPU Time, dan Memory Usage*



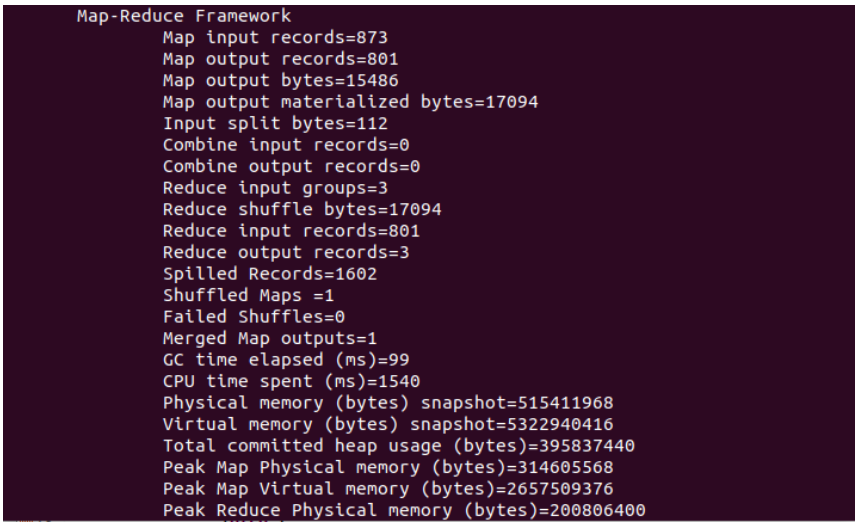
Gambar 15 Job Counter Single Node FIFO

Gambar 15 menunjukan *completion time* dari Hadoop dengan algoritme FIFO dan konfigurasi *single node.Completion time* adalah total waktu yang dibutuhkan untuk algoritme *map* ditambah dengan total waktu yang digunakan untuk algoritme *reduce* yang dibutuhkan dalam mengerjakan job yang di inputkan adalah 4927 milidetik atau empat detik.



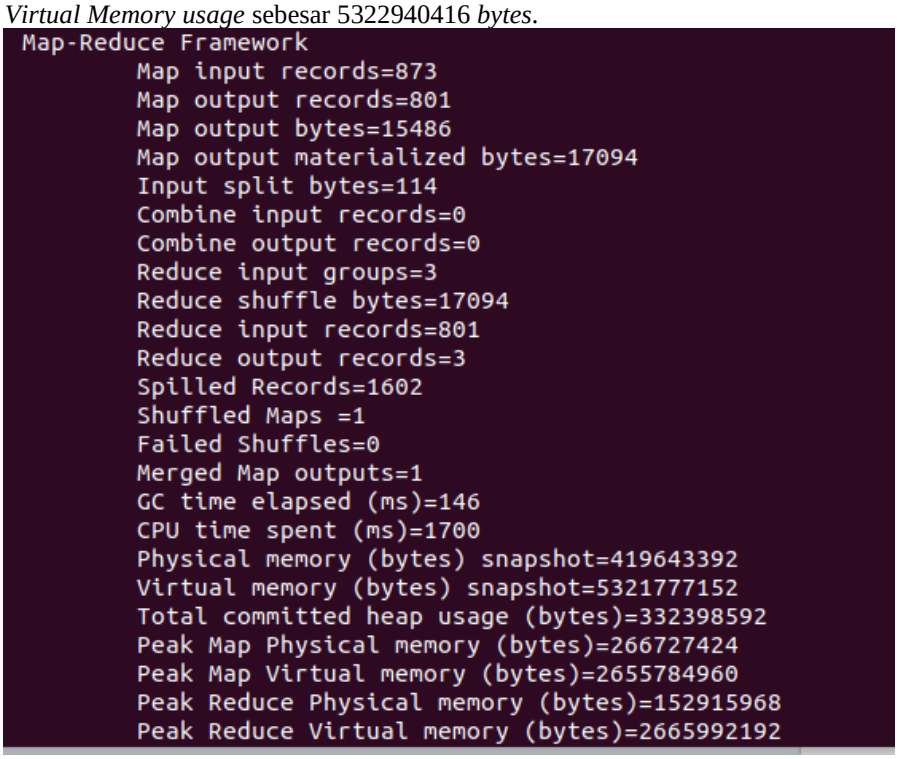
Gambar 16 Job Counter Multi Node fair scheduling

Gambar 16 menunjukan *completion time* dari Hadoop dengan algoritme *fair scheduling* dan konfigurasi *multi node.Completion time* adalah total waktu yang dibutuhkan untuk algoritme *map* ditambah dengan total waktu yang digunakan untuk algoritme *reduce* yang dibutuhkan dalam mengerjakan job yang di inputkan adalah 25354 milidetik atau 25 detik.



Gambar 17 Map-Reduce Framework Single node FIFO

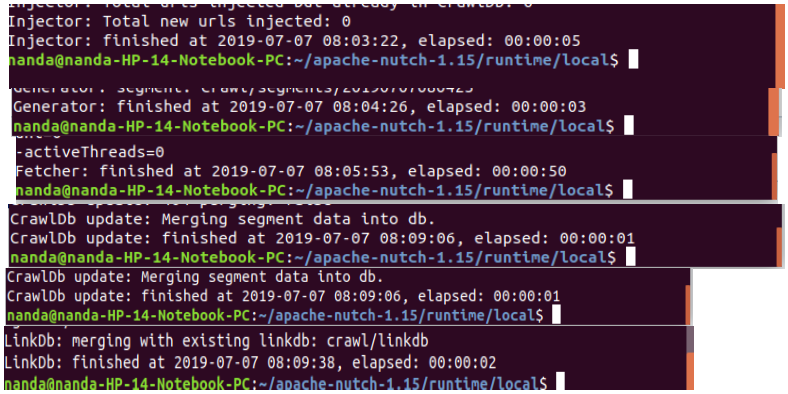
Gambar 17 adalah gambar yang menunjukan *CPU time* dan *memory usage* yang digunakan oleh *cluster single node* dengan algoritme FIFO dalam mengeksikusi *job* yang diberikan, *job* yang diinputkan memiliki *CPU time* sebesar 1540 milidetik atau 1,5 detik, *physical memory usage* sebesar 515411968 *bytes* dan *Virtual Memory usage* sebesar 5322940416 *bytes*.



Gambar 18 Map-Reduce Framework Multi node Fair scheduler

Gambar 18 adalah gambar yang menunjukan *CPU time* dan *memory usage* yang digunakan oleh *cluster multi node* dengan algoritme *fair scheduler* dalam mengeksikusi *job* yang diberikan, *job* yang diinputkan memiliki *CPU time* sebesar 1700 milidetik atau 1,7 detik, *physical memory usage* sebesar 419643392 *bytes* dan *Virtual Memory usage* sebesar 5321777152 *bytes*.

3. *Turnaraund Time*



Gambar 19 Hasil Waktu Crawling Website

Gambar 19 diatas adalah pengujian waktu yang digunakan saat crawling lima website yang terdiri dari *injection* 5 detik, *generating* 3 detik, *fetching* 50 detik, *updated* 1 detik dan *indexing* menggunakan linkDB 2 detik jadi total waktu yang dibutuhkan untuk crawling data adalah 1 menit 2 detik jika ditambah dengan *completion time* pada *cluster single node* dengan algoritme FIFO sebesar 4.9 detik maka total waktu *turnaround time* yang dibutuhkan adalah 67 detik. Sedangkan pada *cluster multi node* dengan algoritme *fair scheduler* dengan *completion time* sebesar 25 detik maka akan dihasilkan *turnaround time* sebesar 87 detik.

4.2 Analisis Hasil Pengujian

Dari hasil uji di atas didapatkan bahwa akurasi yang didapatkan saat uji akurasi pada *testing data* MNISTdi PHP tidak terlalu tinggi hal tersebut dikarenakan pembacaan *pixel* yang berbeda pada Bahasa pemograman Matlab dan PHP. Dari uji akurasi pada formulir C1 menghasilkan angka yang cukup rendah yaitu 38.9 % hal ini dikarenakan beberapa faktor yang pertama yaitu JST hasil latih RBM dari gambar dataset MNIST tidak terlalu cocok dengan gambar formulir C1 karena formulir C1 memiliki format gambar RGB sedangkan gambar dataset MNIST memiliki format black and white. Yang kedua adalah di beberapa TPS ada yang menuliskan angka "0" dengan huruf "X" menjadikan JST tidak dapat memprediksi angka yang dituliskan. Yang ketiga adalah posisi foto yang tidak sama membuat auto crop yang dipasang pada website tidak tepat dalm *cropping* angka dalam gambar formulir C1, hal ini bisa dilihat dari C1 yang sama tetapi difoto oleh orang yang berbeda menghasilkan prediksi yang berbeda.

Pada pengujian *completion time*, *CPU time*, *turnaround time* menunjukan bahwa Hadoop dengan konfigurasi *multi node* dengan algoritme penjadwalan *fair scheduling* menghasilkan hasil yang lebih buruk pada parameter tersebut dibandingkan dengan Hadoop dengan konfigurasi *single node* dengan algoritme penjadwalan FIFO. Hal ini di sebabkan karena penggunaan job yang terlalu kecil mengakibatkan performa yang lebih buruk pada sitem yang terdistribusi seperti Hadoop dengan *multi node.* Sedangkan pada pengujian *memory usage* Hadoop dengan konfigurasi *multi node* dengan algoritme penjadwalan *fair scheduling* menghasilkan hasil yang lebih baik daripada Hadoop dengan konfigurasi *single node* dengan algoritme penjadwalan FIFO. Hal tersebut karena penggunaan memory pada sistem yang terdistribusi lebih baik dibandingkan dengan sistem yang tidak terdistribusi.

5. Kesimpulan

Berdasarkan analisis yang dilakukan pada pengujian pada system yang dibangun pada tugas akhir ini dapat disimpulkan bahwa :

1. Terdapat perbedaan pembacaan pixel pada PHP dan Matlab yang mempengaruhi pengenalan pada gambar
2. Terdapat kemiripan antara angka Sembilan dengan empat, delapan dengan tiga, lima dengan enam, dan tujun dengan satu sehingga akurasi yang di dapat pada angka tersebut paling kecil.
3. Pada formulir c1 letak gambar yang tidak sama pada setiap gambar membuat auto crop untuk memprediksi gambar tidak dapat menjukan gambar yang tepat berisi angka.
4. Perbedaan gambar pada gambar latih dan testing MNIST yang berformat black and white dan gambar formulir c1 yang berformat red green blue(RGB) membuat hasil pemindaian menjadi tidak seakurat pemindaian testing pada MNIST
5. Pada beberapa formulir c1 yang menulis angka 0 dengan huruf "X" dikenali sebagai angka 5 karena bernilai pixel paling penuh.

6. Job yang diberikan kepada Hadoop mempunyai ukuran yang kecil sehingga pada konfigurasi Hadoop *multi node* dengan algoritme *fair scheduling* menghasilkan *completion time* dan *CPU time* yang lebih buruk daripada konfigurasi Hadoop dengan *single node* dan algoritme FIFO.
7. Karena pengaruh sistem yang terdistribusi penggunaan memori virtual dan memori fisik pada Hadoop dengan konfigurasi *multi node* dengan algoritme *fair scheduling* lebih baik daripada Hadoop dengan konfigurasi *single node* dengan algoritme FIFO. Walaupun tidak terlalu signifikan karena penggunaan *job* yang kecil.

5.1. Saran

Pengembangan tugas akhir ini dapat menggunakan *dataset* yang memiliki warna RGB sehingga bisa lebih mengenali foto dari kamera *smartphone* dengan lebih baik.

Daftar Pustaka

[1] "Kominfo," Kementrian Komunikasi dan Informatika, 11 09 2017. [Online]. Available: https://kominfo.go.id/content/detail/10577/lebih-175-juta-wni-telah-rekam-data-kependudukan/0/berita. [Accessed 17 17 2019].

[2] G. Jagdev, B. Singh and M. Mann, "Big Data Proposes an Innovative Concept for Contesting Elections in Indian," *International Journal of Scientific and Technical Advancements* , 2015.

[3] S. Prabowo and M. Abdurohman, "Implementasi Algoritme Penjadwalan untuk pengolaan Big Data dengan Hadoop," *Indonesian Jurnal of Computing,* 2017.

[4] A. Ahmad, "Mengenal Artificial Intelligence, MachineLearning, Neural Network, dan Deep Learning," no. Yayasan Cahaya Islam, Jurnal Teknologi Indonesia, 2017.

[5] S. "Algoritme Restricted Boltzmann Machines (RBM) untuk Pengenalan Tulisan Tangan Angka," *Seminar Nasional Teknologi Informatika, "The Future of Computer Vision,* no. Seminar Nasional Teknologi Informatika, 2017.

[6] S. and M. , "Analisis Pengaruh Fungsi Aktivasi, Learning Rate Dan Momentum Dalam Menentukan Mean Square Error (MSE) Pada Jaringan Saraf Restricted Boltzmann Machines (RBM)," *JITE (Journal of Informatics and Telecommunication Engineering),* 2019.

[7] M. Lankvist, L. Karlsson and A. Loutfi, "A review of unsupervised feature learning and deep learningfor time-series modeling," 2014.

[8] R. Hrasko, A. G. Pacheco and R. A. Krohling, "Time Series Prediction using Restricted Boltzmann Machines and Backpropagation," *Information Technology and Quantitative Management (ITQM 2015),* 2015.

[9] T. Yamada, M. Akazawa, T. Asai and Y. Amemia, "Boltzmann machine neural network devices using single-electron tunnelling," 2001.

[10] Y. M. Wijaya, Teknologi Bigdata Sistem Canggih dibalik Facebokk, Yahoo, Google, IBM, Badung: Nilacakra, 2019.

[11] M. Usama, M. Liu and M. Chen, "Job schedulers for Big data processing in Hadoop environment: testing," *Digital Communications and Network,* 2018.

[12] T. White, Hadoop: The Definitive Guide, Third Edition, Cambridge: O'Reilly, 2012.

[13] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker and I. Stoica, "Job Scheduling for Multi-User MapReduce Clusters," *Electrical Engineering and Computer Sciences,* 2009.

[14] M. Afif, B. Erfianto and S. Prabowo, "Analisis Perbandingan Performa Fair Scheduler dan Capacity Scheduler pada Sistem Job Scheduling Hadoop," 2018.

[15] J. V. Gautam, H. B. Prajapati, V. K. Dabhi and S. Chaudhary, "Empirical Study of Job Scheduling Algorithms in Hadoop," *CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume 17, No 1,* 2017.