OPEN ACCESS
23
https://humicpress.org/journals/index.php/ijohce
Intl. Journal of Human Centric Engineering
Vol. 1, No. 1, Mar 2024. pp.23-41

Ballot Counting System Prototype with Hadoop Platform using Map and Reduce

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

[1,2,3]*Faculty of Informatics,*[1,2,3]*Telkom University*
Bandung

[1]anandamuhammad@students.telkomuniversity.ac.id, [2]hilalnuha@gmail.com,
[3]sidikprabowo@telkomuniversity.ac.id

Abstract

Indonesia is among the nations practicing a democratic system, where the head of state is elected through general elections. During these elections, every citizen possessing a National Identity Card (KTP) holds equal voting rights. Counting votes accurately amidst the vast number of ballots poses a challenge. The ultimate objective is to develop a vote counting implementation using Hadoop, utilizing a form C1 image as input. The numerical data from the C1 form is processed into INT data types employing artificial neural networks (ANN) generated through the Restricted Boltzmann Machine (RBM), trained with MNIST data. These ANN models are uploaded onto a website for image storage in the database. The website's data is then crawled using Apache Nutch, and the obtained results are processed through Hadoop's MapReduce algorithm. Testing reveals efficient performance, with the recapitulation of results using Apache Nutch and Hadoop employing multimode and fair scheduler scheduling algorithms taking one minute and twenty-five seconds. Conversely, Hadoop configured with a single-node setup and FIFO scheduling algorithm achieves a quicker time of 67 seconds.

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

Abstrak

Indonesia merupakan salah satu negara yang menganut sistem demokrasi, di mana kepala negara dipilih melalui pemilihan umum. Dalam pemilihan umum ini, setiap warga negara yang memiliki Kartu Tanda Penduduk (KTP) memiliki hak suara yang sama. Menghitung suara secara akurat di tengah-tengah jumlah surat suara yang sangat banyak menjadi sebuah tantangan. Tujuan utamanya adalah untuk mengembangkan implementasi penghitungan suara menggunakan Hadoop, dengan memanfaatkan gambar formulir C1 sebagai input. Data numerik dari formulir C1 d Indonesia merupakan salah satu negara yang menganut sistem demokrasi, di mana kepala negara dipilih melalui pemilihan umum. Dalam pemilihan umum ini, setiap warga negara yang memiliki Kartu Tanda Penduduk (KTP) memiliki hak suara yang sama. Menghitung suara secara akurat di tengah-iproses menjadi tipe data INT dengan menggunakan jaringan syaraf tiruan (JST) yang dihasilkan melalui Restricted Boltzmann Machine (RBM), yang dilatih dengan data MNIST. Model ANN ini diunggah ke situs web untuk penyimpanan gambar dalam database. Data situs web kemudian dirayapi menggunakan Apache Nutch, dan hasil yang diperoleh diproses melalui algoritme MapReduce Hadoop. Pengujian menunjukkan kinerja yang efisien, dengan rekapitulasi hasil menggunakan Apache Nutch dan Hadoop yang menggunakan algoritme penjadwalan multimode dan penjadwalan yang adil membutuhkan waktu satu menit dua puluh lima detik. Sebaliknya, Hadoop yang dikonfigurasi dengan pengaturan node tunggal dan algoritma penjadwalan FIFO mencapai waktu yang lebih cepat yaitu 67 detik.

Kata Kunci: Hadoop, mapreduce, apache nutch, Restricted Boltzmann Machine , formulir C1, job.

I. INTRODUCTION

In Indonesia, a democratic system is upheld, where the head of state is elected by the populace through general elections. During elections, all citizens possessing a Resident Identity Card (KTP) are granted equal voting privileges. The Indonesian population required to have a KTP in 2017 was 189 million people [1]. This amount makes voter data in elections, especially elections to elect the president and vice president, big data because it contains a very large amount of data. Based on the Election Law (UU) and General Election Commission (KPU) regulations, the final official and legal results are manual calculations determined by the KPU. The KPU must count and recapitulate all votes manually starting from the level of the Polling Place (TPS), Voting Committee (PPS), Subdistrict Election Committee (PPK), Regency/City Regional General Election Commission (KPUD), Provincial KPUD, to the Central KPU . The final results can only be announced to the public a few days later after voting day. Because of this, this final project will create a quick count system with samples in the form of images from the C1 form to count and recapitulate votes. The system created inputs the C1 form using a website then the data from the website is crawled using Apache Nutch and then the results of the crawl are input and calculated using Hadoop using the fair scheduling algorithm.

Hadoop is a distributed file system that is intended to carry out jobs for data that is in the large category (Big Data) [2]. Hadoop itself has two basic architectures, namely mapreduce and Hadoop Distributed File System (HDFS), which makes Hadoop very optimal in handling big data [3].

This final project implements changing the C1 form image into an INT which is stored in the website database. This data was crawled using Apache Nutch, and used as a job in Hadoop and tested Conpletion Time, CPU Time, Memory Usage and turnaround time as a time performance analysis of the system being created.

A. *Formulation of the problem*

Based on the background described above, there are problems that arise in this final assignment, namely as follows:

1. How to implement the calculation and recapitulation of election votes using Hadoop.
2. How to convert C1 image data into an integer that can be read by a computer.
3. What is the performance of voice calculation time obtained using Hadoop.

B. *Scope of problem*

The limitations of the problems in this final assignment are:

1. The data used is C1 which is used for counting votes at the TPS
2. The C1 was photographed using a Smartphone camera with a resolution of 12 Megapixels which produces images of around 3000 x 4000 pixels
3. The image format used is .jpg
4. The position of C1 in the image must be the same.
5. The crawled text entered in Hadoop is text that contains a table of the number of votes
6. The scheduler used in Hadoop is a fair scheduler

C. *Objective*

The aim of this final assignment is:

1. Create a website that can be accessed by smartphones and computers to input data in the form of C1 form.
2. Make the C1 form image that is input can be processed into integer data that can be stored in the database.
3. Make voice recapitulation calculations using Hadoop, and analyze its performance.

II. THEORETICAL BASIS

A. *Restricted Boltzmann Machine*

Restricted Boltzmann Machine (RBM) is an Artificial Neural Network (ANN) method and is also a Deep Learning method that uses more than one layer [4]. RBM is a stochastic neural network, which means that every connected neuron that is activated has an element of probability [5] [6]. RBM usually has two layers, namely the visible layer which is the state to be observed and the hidden layer which is the feature detectors and bias unit or denoted by b [5]. Next, each visible unit is connected to all hidden units which are represented by weights or denoted by W [5]. The visible layer only has connections to the hidden layer, there are no connections between neurons in the visible and hidden layers. This relationship allows for two-way information transfer, which is why ANN is called Restricted [7] [8].

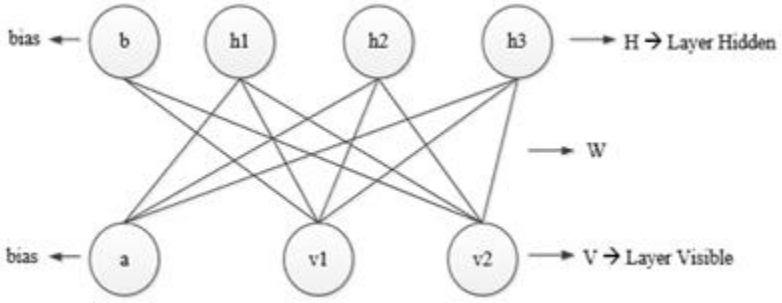

Fig.1. layers in RBM [9]

The architecture in Figure 1 shows the RBM used to train ANN using a dataset via visible and hidden layers.

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

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

Equation 1 is the sigmoid function used for visible unit activation and equation 2 is the softmax function used for hidden unit activation which will produce probability [5].

B. MNIST Dataset

MNIST The data used is a dataset for recognizing human handwriting. In the Mnist dataset there are 600 samples from 60,000 handwriting data for learning [5] [6]. The MNIST dataset is used as a training set to

train the RBM used. The MNIST dataset is used as a training set to train the RBM which is used below. Figure 2 is an example of an MNIST test image and Figure 3 is an example of MNIST training data.

Fig. 2. MNIST Test Data

Fig. 3. MNIST Training Data

C. Apache Nutch

Apache nutch is an open source web crawler that is commonly used for crawling websites [10]. Apache Nutch uses a distributed system like Hadoop. Apache nutch has parsing and indexing capabilities, making it possible to use it as a search engine as needed. In this final project, Apache Nutch crawls HTML data on websites which are created in the form of binary files which are converted into text.

D. MapReduce Application

MapReduce is a program model based on distributed computing [11]. MapReduce programming has two important functions, namely the Map function and the Reduce function. The map function functions to take input data as a set of key and value pairs of those keys and produce output as a set of key and value pairs [11]. The reduce function functions to receive the key and value pair produced by the map function and then set the value of the key [11]. The following figure 4 is the data flow of the map and reduce algorithms.

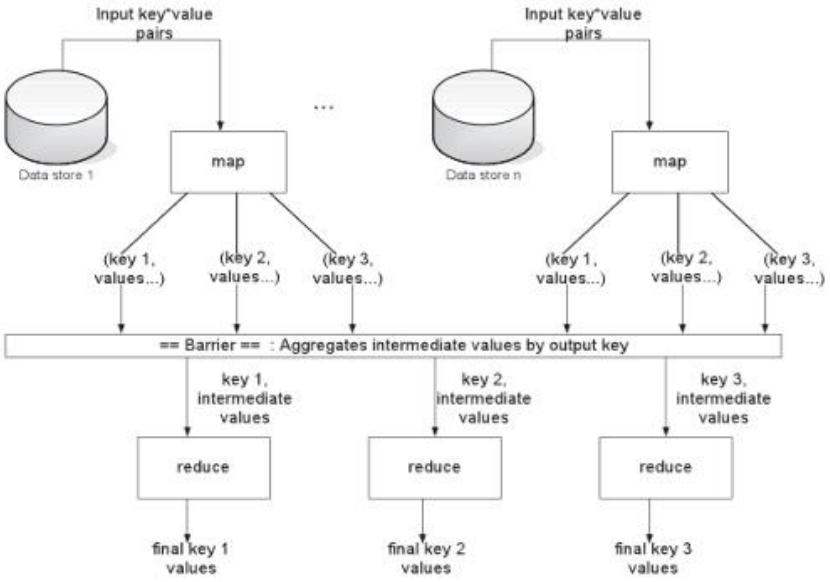

Fig. 4. Data flow in the mapreduce algorithm [12]

E. Hadoop

Hadoop is an open source software framework that is used to process large data [11]. Hadoop has four modules, namely:

1. Hadoop Distributed File System (HDFS)

HDFS is a distributed storage system that can store files distributed on HDFS nodes. Figure 5 shows the HDFS architecture

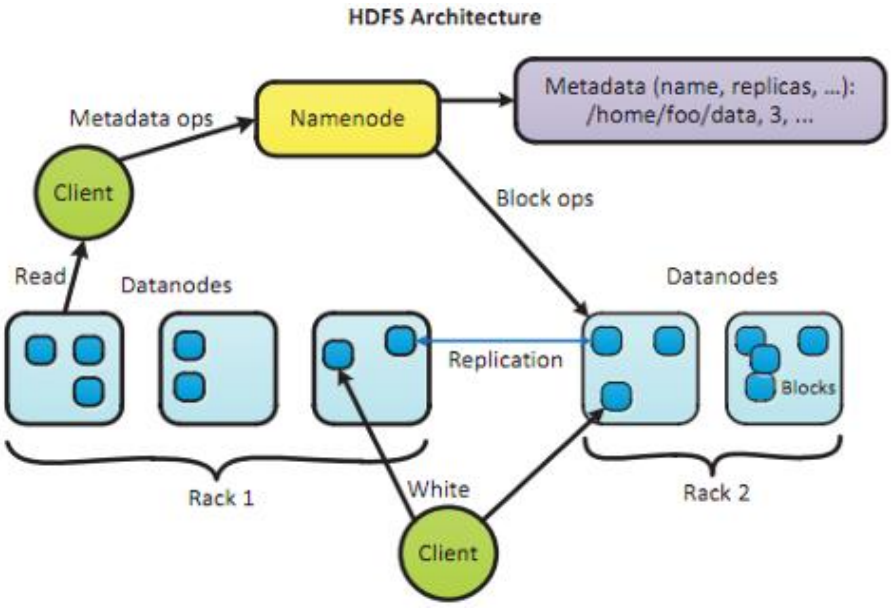

Fig. 5. HDFS Architecture [3]

2. Hadoop Common

   Hadoop Common are libraries and tools required by other Hadoop modules.

3. Folder Reduce

   Map reduce is a program model for processing techniques based on distributed computing.

4. Hadoop Yarn

   Hadoop yarn is used to manage the resources that will be used.

F. Jobs and Job Scheduling

A job is a job assigned by a user to be done by the Hadoop system. The jobs themselves can be given by users connected to the server or given by the server directly. Jobs in Hadoop are divided into two, namely map and reduce [11]. Figure 6 shows the Hadoop layer with multi-nodes.

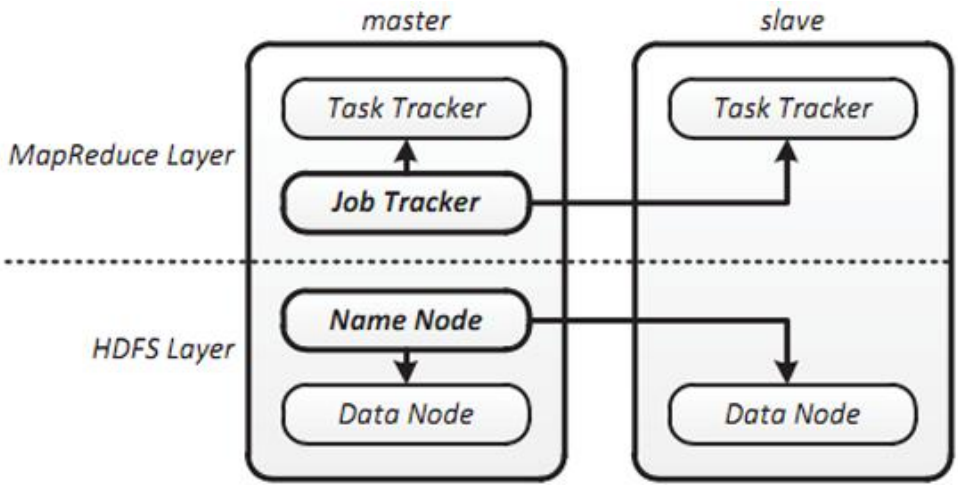

Fig. 6. Layers in Hadoop [3]

A map is the initial part of a job in Hadoop which aims to build the information contained in the data into data that is ready to be processed according to the job that will be processed by the server [4]. Meanwhile, reduce is a process carried out by the server [12] to carry out processes according to the work given by the user or server on data that has gone through the map process [12].

Job scheduling is a way in Hadoop to organize data processing so that the work done is as expected. In default conditions, Hadoop uses the First In First Out (FIFO) algorithm in job scheduling [12].

G. Fair Scheduling

Fair scheduling is a job scheduling algorithm in Hadoop that allocates the same resources to each given job [13]. In the fair scheduler, if there is a job being done, this algorithm will provide an empty slot to be filled by the new job. In this way, each job will get the same resources and small jobs will not have to wait long to be executed. Figure 7 shows an illustration of the data pool used by fair scheduling where m is the number of resources shared.

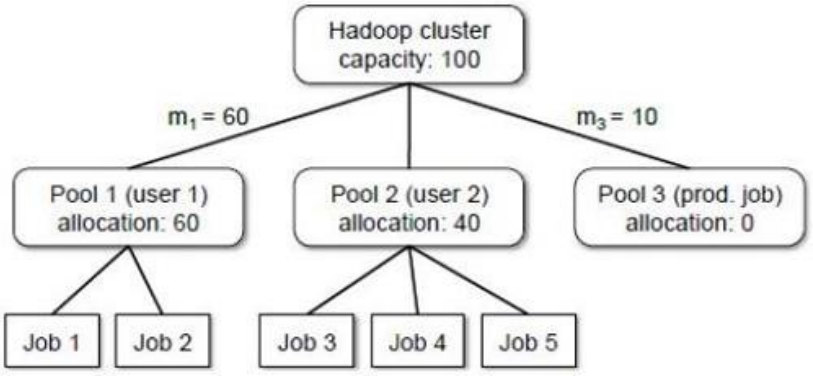

Fig. 7. Pooldata used by fair scheduler [13]

H. Test Parameters

The parameters used for testing in this final project are accuracy tests for MNIST data testing on PHP, average completion time tests, average CPU time, average memory usage, and turnaround time on Hadoop.

1. Test the accuracy of MNIST testing data

   This parameter measures the accuracy of each number from one to nine against the testing data held by MNIST. There are 350 testing data used. The unit used is the percentage (%) of success in predicting image accuracy.

2. Accuracy test on sample form C1

   This parameter is measured by the level of accuracy of six C1 form samples that have been photographed.

3. Average Completion Time [14]

   This parameter measures the average time to complete one job from all the jobs entered. Because only one job is used, the time to complete the given job is measured.

4. Average Memory Usage [14]

   This parameter is measured from the average memory usage in completing the jobs that have been input.

5. Average CPU Time [11]

   This parameter is measured by the amount of CPU usage time in completing the job, where this parameter is based on CPU Time [15].

6. Turnaround Time

   This parameter measures the total time required to complete one scenario. This value will be obtained from the total amount of time needed to complete a scenario [15]. Because there is only one job used in this final project, the scenarios used are the time to complete the job and the time used to complete the crawl results. The units used are minutes or seconds.

III. BUILT SYSTEM

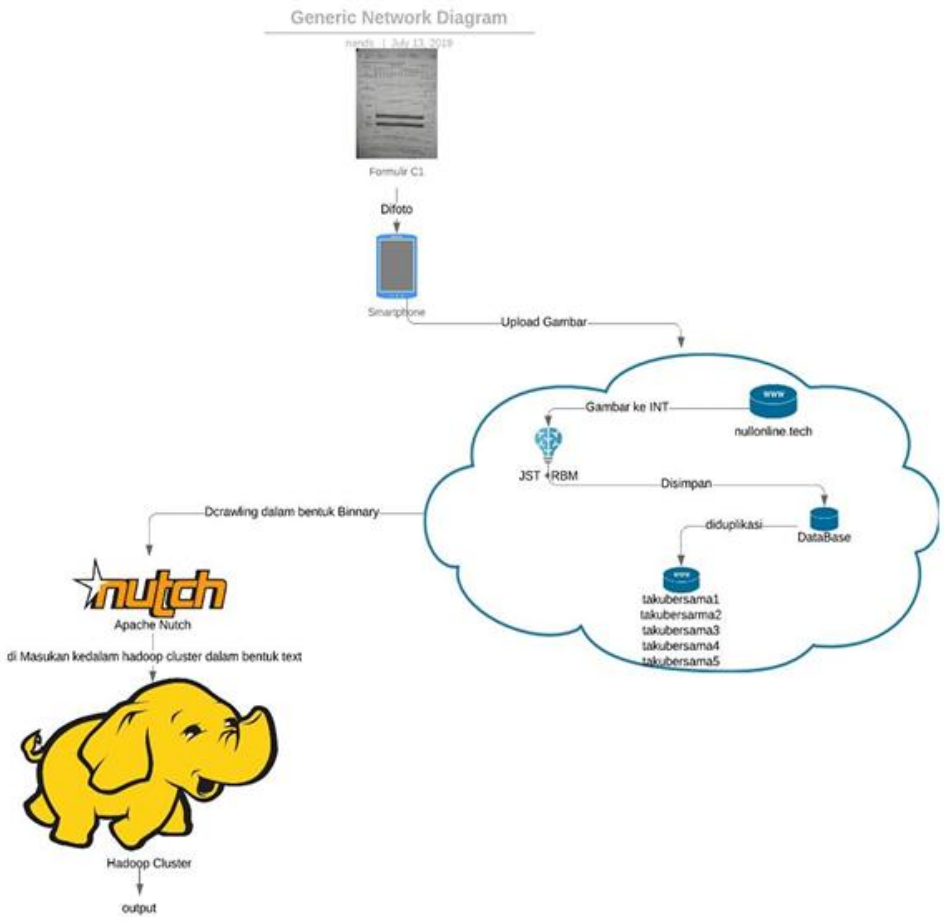

Fig. 8. Whole System Design

Figure 8 is the overall system design where the user takes a photo of the C1 form using a smartphone camera and then the photo is uploaded to the website www.nullonline.tech where on the website there is an ANN from RBM which is used to process the C1 form into INT data. Next, the integer data is stored in a database and file on the website www.nullonline.tech. After becoming saved text, the sound from C1 is crawled by Apache Nutch in text form so that it can become a job that will be processed by the mapreduce algorithm in Hadoop. This final project also prepares five other websites which are duplicates of database tables from the website www.nullonline.tech to store the number of other votes to test the crawling performance of Apache Nutch and Hadoop.

A. System planning

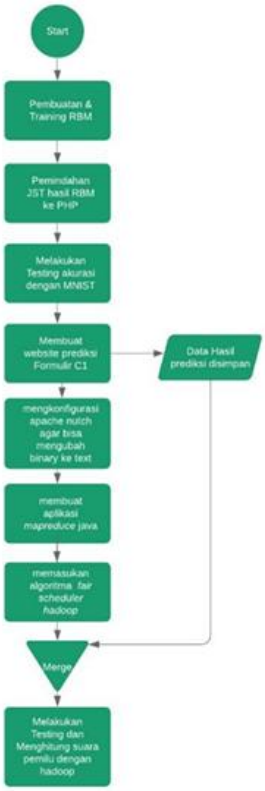

Fig. 9. System Design Flowchart

Figure 9 is a flowchart of the system design that will be created in this final project and will be explained as follows:

1. Create an RBM algorithm using the Matlab programming language with 78,400 visible layers, 1000 hidden layers, and 100 epochs.
2. Train the RBM that has been created with the number training dataset from MNIST.
3. Create a website using the PHP programming language with an image upload feature in .jpg format.
4. Moving ANN training results from RBM from Matlab to PHP.
5. Calculating the accuracy of digits 0 – 9 with the MNIST dataset.
6. Creating a website with JST as a result of RBM training can read the number of votes on the C1 form used in elections.
7. Save the results of reading the number of votes on form C1 in the database and files on the website that has been created
8. Crawling the website that has been created to make data on the number of votes into text form (.txt) so that it can be used as a job by Hadoop.
9. Create a mapreduce application to be installed on Hadoop.
10. Incorporating fair scheduler into YARN on Hadoop.
11. Perform vote calculations with text files that have been crawled with Hadoop

B. Hadoop and Nutch Architecture

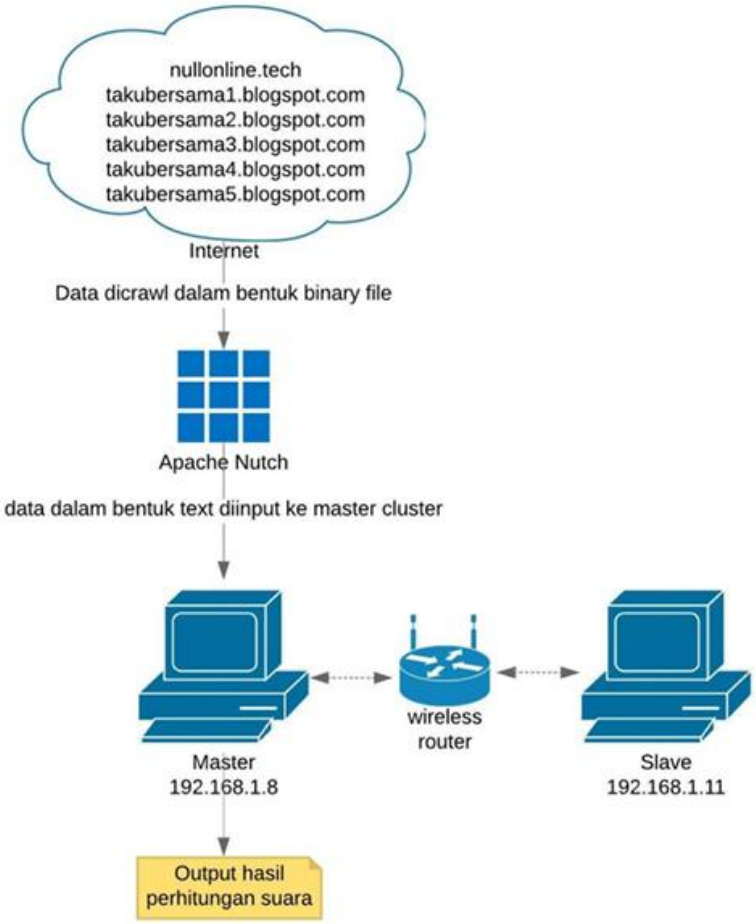

Fig. 10. Hadoop and Nutch Architecture

Figure 10 above is the architecture of Hadoop and Apache Nutch that will be created. Using two clusters. One cluster acts as the master cluster and one cluster acts as the slave cluster. The scheduling algorithm used is a fair scheduler with a total of six websites that will be crawled to count the number of C1 forms on the website that will be crawled.

Hardware Components

The hardware components used in this final project are as follows:

1. One server computer that acts as a master server with an Intel Core I5 – 5200U 2.20 GHz processor, 6 GB RAM and 500 GB hard disk.
2. One server computer that acts as a slave server with an Intel Core I5 – 5200U 2.20 GHz processor, 4 GB RAM and 1 TB hard disk.
3. Two smartphones for taking photos form the C1 which has a camera with a resolution of 12 megapixels.
4. One wireless router to create a network between master and slave in a Hadoop cluster.

Software Components

The software components used in this final assignment are as follows:

1. Operating system
   The operating system used on the Hadoop and Apache Nutch cluster computers is Linux Ubuntu 18.04 LTS and to create RBM and websites is Windows 10 Education.
2. Hadoop Server
   The Hadoop server used is Hadoop 3.2.0.
3. Matlab
   To create and train layers in ANN with RBM.
4. Java
   Java is used to support the use of the Hadoop distributed file system, and create executors for given jobs. A Hadoop cluster will require a Java Run Time Environment (JRE) and a Java Development Kit (JDK) with the java version.
5. Open SSH
   Open SSH because the Hadoop server runs on top of the SSH server.
6. PHP
   PHP is used to create a website that will be used to identify the number of votes on form c1.
7. Mozilla Firefox
   Mozilla Firefox will be used as a browser for the portal to monitor Hadoop in real time and access the websites that have been created.

C. Map and Reduce Application

This final assignment discusses the map and reduce applications which have quite an important role in vote calculations. The class map functions to search for keywords, namely paslon01, paslon02, and invalid votes along with the values of these keywords from the HTML table that has been crawled previously. The reduce class functions to add up the values for each keyword that has been given by the class map. The map and reduce applications in this final project are written in the Java programming language. Figure 11 is the pseudo code of the class map and figure 12 is the pseudo code of the reduce class in this final project.

```
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
```

Fig. 11. Pseudo 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)
```

Fig. 12. Pseudo Code Reduce

IV. EVALUATION

A. *Test result*

With the implementation of the system as explained in the previous chapter, there are test results, namely

a) *Accuracy Test in MNIST Data Testing*

TABLE I
MNIST DATA ACCURACY TEST

Number	Number of Digits	Correctness of Predictions	Accuracy(%)
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

Table 1 shows the level of accuracy for each number is different. And this system produces a total accuracy of MNIST testing images on average of 78%. From these results the trained RBM layer is used to identify the numbers in the KPU C1 form.

b) Accuracy Test on Form C1

TABLE II
FORM C1 ACCURACY TEST

Test No	Data In C1			Prediction			Correctness of Predictions	Amount
	0	5	4	5	5	4	7	18
1	1	6	5	5	5	5	Total(%)	38.88889
	0	0	3	5	1	7		
	x	5	1	5	7	5		
2	1	5	2	5	7	5		
	x	x	4	5	7	5		
	x	5	7	1	7	5		
3	1	6	4	7	5	5		
	x	x	3	7	5			
	0	5	4	5	5	5		
4	1	6	5	5	7	5		
	0	0	3	2	2	1		
	x	5	1	5	2	7		
5	1	5	2	5	5	5		
	x	x	4	1	6	5		
	x	5	1	7	3	5		
6	1	5	2	5	5	7		
	x	x	4	7	5	5		

From testing 6 C1 photo samples, it was found that the total accuracy in predicting the C1 form was 38.9%. The following images 13 and 14 are images of the sample testing carried out

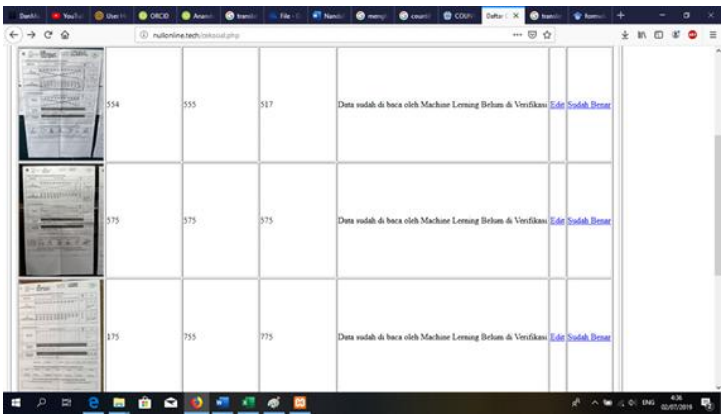

Fig. 13. Sample Identification Figure C1

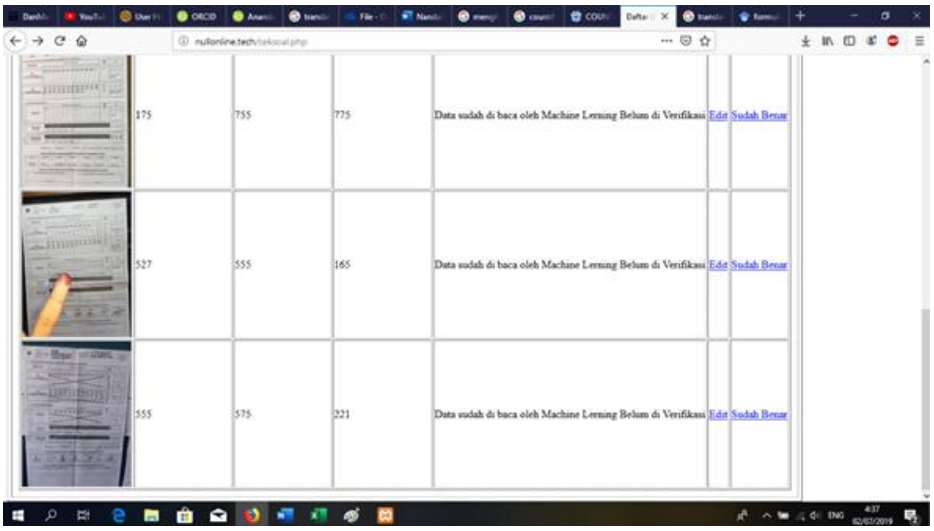

Fig. 14. Sample Identification Figure C1

c) *Test Completion Time, CPU Time, and Memory Usage*

```
File System Counters
        FILE: Number of bytes read=17094
        FILE: Number of bytes written=478705
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=3708
        HDFS: Number of bytes written=51
        HDFS: Number of read operations=8
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Data-local map tasks=1
        Total time spent by all maps in occupied slots (ms)=2456
        Total time spent by all reduces in occupied slots (ms)=2471
        Total time spent by all map tasks (ms)=2456
        Total time spent by all reduce tasks (ms)=2471
        Total vcore-milliseconds taken by all map tasks=2456
        Total vcore-milliseconds taken by all reduce tasks=2471
        Total megabyte-milliseconds taken by all map tasks=2514944
        Total megabyte-milliseconds taken by all reduce tasks=2530304
```

Fig. 15. Job Counter Single Node FIFO

Figure 15 shows the completion time of Hadoop with the FIFO algorithm and single node configuration. Completion time is the total time required for the map algorithm plus the total time used for the reduce algorithm which is required to carry out the input job, which is 4927 milliseconds or four seconds.

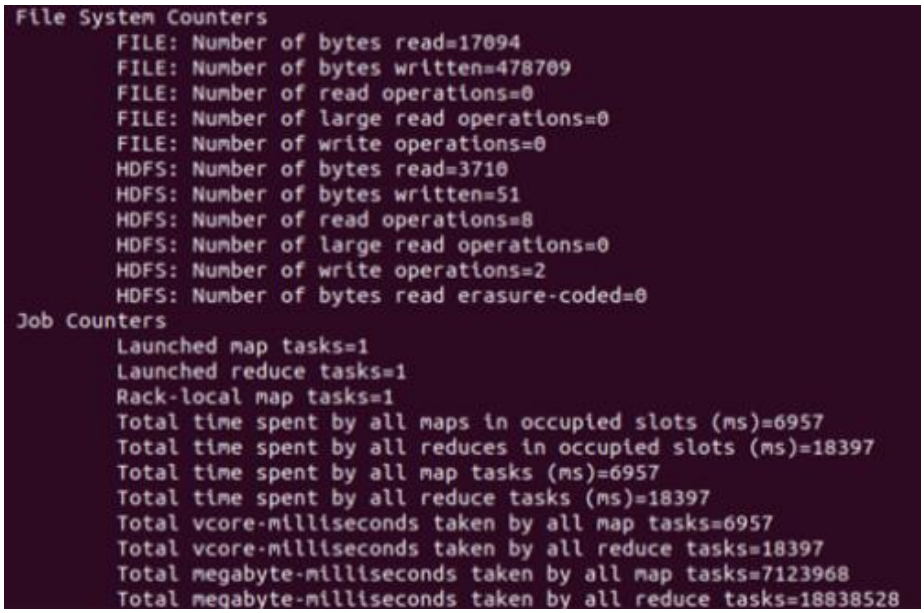

Fig. 16. Job Counter Multi Node fair scheduling

Figure 16 shows the completion time of Hadoop with the fair scheduling algorithm and multi-node configuration. Completion time is the total time required for the map algorithm plus the total time used for the reduce algorithm required to carry out the input job, which is 25354 milliseconds or 25 seconds.

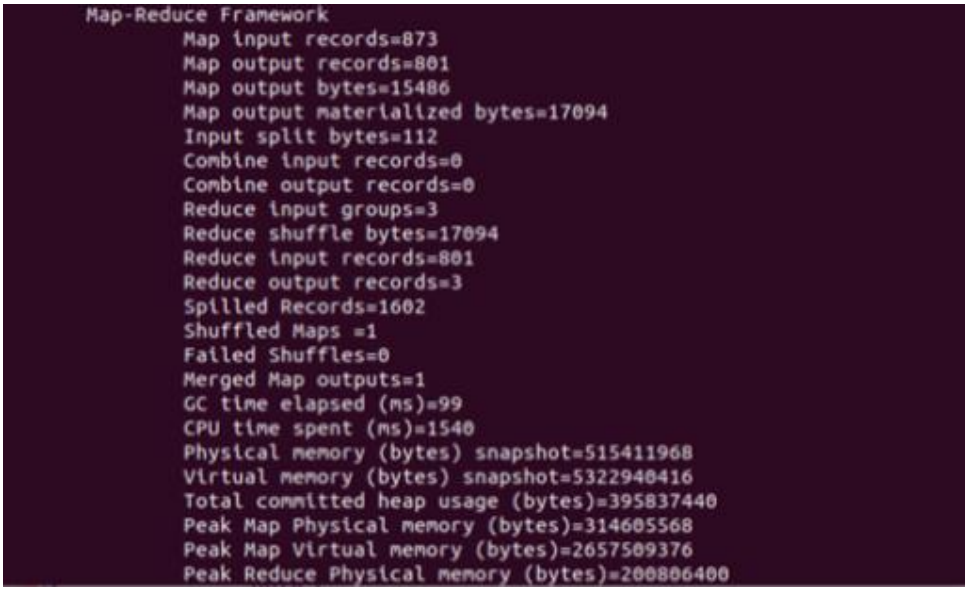

Fig. 17. Map-Reduce Framework Single node FIFO

Figure 17 is a picture that shows the CPU time and memory usage used by a single node cluster with the FIFO algorithm in executing a given job, the input job has a CPU time of 1540 milliseconds or 1.5 seconds, physical memory usage of 515411968 bytes and Virtual Memory usage is 5322940416 bytes.

```
Map-Reduce Framework
        Map input records=873
        Map output records=801
        Map output bytes=15486
        Map output materialized bytes=17094
        Input split bytes=114
        Combine input records=0
        Combine output records=0
        Reduce input groups=3
        Reduce shuffle bytes=17094
        Reduce input records=801
        Reduce output records=3
        Spilled Records=1602
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=146
        CPU time spent (ms)=1700
        Physical memory (bytes) snapshot=419643392
        Virtual memory (bytes) snapshot=5321777152
        Total committed heap usage (bytes)=332398592
        Peak Map Physical memory (bytes)=266727424
        Peak Map Virtual memory (bytes)=2655784960
        Peak Reduce Physical memory (bytes)=152915968
        Peak Reduce Virtual memory (bytes)=2665992192
```

Fig. 18. Map-Reduce Framework Multi node Fair scheduler

Figure 18 is a picture that shows the CPU time and memory usage used by a multi node cluster with the fair scheduler algorithm in executing a given job, the input job has a CPU time of 1700 milliseconds or 1.7 seconds, physical memory usage of 419643392 bytes and Virtual Memory usage is 5321777152 bytes.

d) *Turnaround Time*

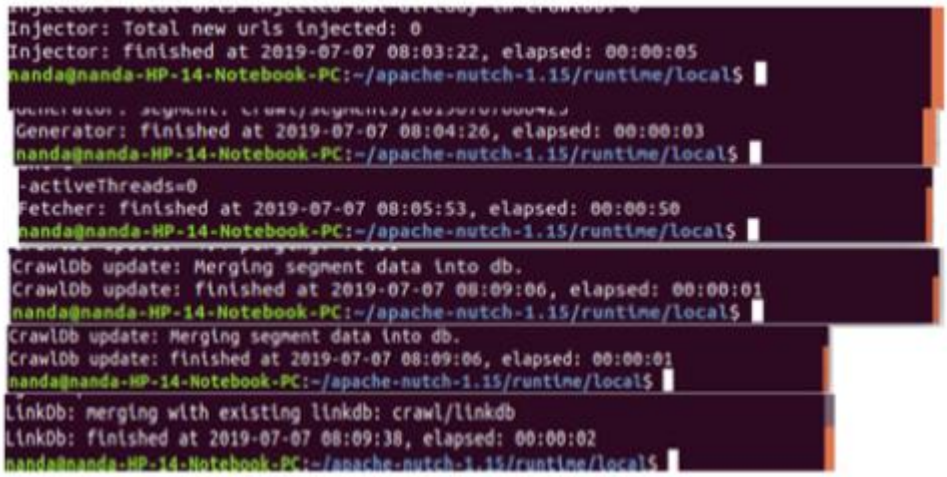

Fig. 19. Website Crawling Time Results

Figure 19 above is a test of the time used when crawling five websites consisting of injection 5 seconds, generating 3 seconds, fetching 50 seconds, updating 1 second and indexing using linkDB 2 seconds so the total time needed for crawling data is 1 minute 2 seconds if plus the completion time on a single node cluster with the FIFO algorithm of 4.9 seconds, the total turnaround time required is 67 seconds. Meanwhile, in a

multi-node cluster with a fair scheduler algorithm with a completion time of 25 seconds, a turnaround time of 87 seconds will be produced.

B. *Analysis of Test Results*

From the test results above, it was found that the accuracy obtained when testing the accuracy of MNIST data testing in PHP was not too high, this was due to the different pixel readings in the Matlab and PHP programming languages. The accuracy test on the C1 form produces a fairly low figure, namely 38.9%, this is due to several factors, firstly, namely, the ANN resulting from RBM training from the MNIST dataset image does not really match the C1 form image because the C1 form has an RGB image format while the MNIST dataset image has a black and white. The second is that at some polling stations there are those who write the number "0" with the letter "X", making it impossible for JST to predict the number written down. The third is that the position of the photos is not the same, making the auto crop installed on the website incorrect in cropping the numbers in the C1 form image. This can be seen from the same C1 but photographed by different people producing different predictions.

In testing completion time, CPU time, turnaround time, it shows that Hadoop with a multi node configuration with a fair scheduling algorithm produces worse results on these parameters compared to Hadoop with a single node configuration with a FIFO scheduling algorithm. This is caused by using jobs that are too small resulting in worse performance in distributed systems such as Hadoop with multi nodes. Meanwhile, in testing Hadoop memory usage with a multi node configuration with a fair scheduling algorithm, it produces better results than Hadoop with a single node configuration with a FIFO scheduling algorithm. This is because memory usage in a distributed system is better than in a non-distributed system.

V. CONCLUSION

A. *Conclusion*

Based on the analysis carried out on testing the system built in this final project, it can be concluded that:

1. There are differences in pixel reading in PHP and Matlab which affect image recognition
2. There is a similarity between the numbers nine and four, eight and three, five and six, and seven and one so that the accuracy obtained with these numbers is the smallest.
3. On the C1 form, the location of the images in each image is not the same, making auto crop to predict the image unable to show the correct image containing the numbers.
4. The difference between the images in the MNIST training and testing images which are in black and white format and the C1 form images which are in red green blue (RGB) format makes the scan results not as accurate as the testing scans in MNIST
5. On some c1 forms that write the number 0 with the letter "X" it is recognized as the number 5 because it has the fullest pixel value.
6. The jobs given to Hadoop have a small size so that the multi node Hadoop configuration with the fair scheduling algorithm produces worse completion time and CPU time than the Hadoop configuration with a single node and FIFO algorithm.
7. Due to the influence of a distributed system, the use of virtual memory and physical memory in Hadoop with a multi node configuration with a fair scheduling algorithm is better than Hadoop

with a single node configuration with a FIFO algorithm. Although it is not very significant because the use of jobs is small.

B. *Suggestion*

The development of this final project can use a dataset that has RGB colors so that it can better recognize photos from smartphone cameras.

REFERENCES

[1] "Kominfo," Ministry of Communication and Information, 11 09 2017. [Online]. Available: https://kominfo.go.id/content/detail/10577/besar-175-juta-wni-telah-rekam-data-kependungan/0/berita. [Accessed 17 17 2019]

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

[3] S. Prabowo and M. Abdurohman, "Implementation of Scheduling Algorithms for processing Big Data with Hadoop," Indonesian Journal of Computing, 2017.

[4] A. Ahmad, "Getting to Know Artificial Intelligence, MachineLearning, Neural Networks, and Deep Learning," no. Islamic Light Foundation, Indonesian Technology Journal, 2017.

[5] S. "Restricted Boltzmann Machines (RBM) Algorithm for Handwritten Number Recognition," National Seminar on Information Technology, "The Future of Computer Vision, no. National Seminar on Information Technology, 2017

[6] S. and M., "Analysis of the Effect of Activation Function, Learning Rate and Momentum in Determining Mean Square Error (MSE) in Restricted Boltzmann Machines (RBM) Neural Networks," JITE (Journal of Informatics and Telecommunication Engineering), 2019

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

[8] R. Hrasko, AG Pacheco and RA 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 tunneling," 2001.

[10] YM Wijaya, Advanced System Bigdata Technology behind 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 MultiUser MapReduce Clusters," Electrical Engineering and Computer Sciences, 2009.

[14] M. Afif, B. Erfianto and S. Prabowo, "Comparative Analysis of Fair Scheduler and Capacity Scheduler Performance on the Hadoop Job Scheduling System," 2018.

[15] JV Gautam, HB Prajapati, VK Dabhi and S. Chaudhary, "Empirical Study of Job Scheduling Algorithms in Hadoop," CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume 17, No 1, 2017.