

IMPLEMENTASI METODE NAÏVE BAYES CLASSIFIER UNTUK SELEKSI ASISTEN PRAKTIKUM PADA SIMULASI HADOOP MULTINODE CLUSTER

Maryamah¹, Moh. Fadel Asikin², Daisy Kurniawaty³, Selly Kurnia Sari⁴, Imam Cholissodin⁵

^{1,2,3,4,5}Fakultas Ilmu Komputer Universitas Brawijaya

Email: ¹maryamahfaisol02@gmail.com, ²fadelasikin@gmail.com, ³daisyniaa@gmail.com, ⁴sellykurniasr@gmail.com, ⁵imamcs@ub.ac.id

(Naskah masuk: 1 Oktober 2016, diterima untuk diterbitkan: 26 Desember 2016)

Abstrak

Pemilihan asisten pada praktikum dialami oleh berbagai universitas di Indonesia salah satunya di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya (UB). Dalam pemilihan asisten praktikum ada beberapa proses yang harus dilalui. Proses pemilihan yang ada di FILKOM masih dalam bentuk manual. Adapun proses yang dijalani dalam pemilihan asisten praktikum diantaranya adalah tes administrasi, tes *live coding*, dan tes mengajar. Dalam penentuan penerimaan asisten praktikum tersebut berdasarkan hasil tes yang telah dilakukan. Kendala yang dihadapi adalah kemiripan hasil tes pada ketentuan tertentu yang menyebabkan kerancuan proses pemilihan asisten praktikum. Dari permasalahan tersebut penulis melihat suatu peluang untuk membuat sistem yang mampu menyeleksi dengan cara melakukan hasil klasifikasi tertinggi terhadap hasil tes yang dilakukan oleh calon asisten praktikum. Penelitian ini menggunakan *Hadoop* dengan menerapkan metode *Naïve Bayes* yang dianggap mampu menghasilkan klasifikasi yang akurat, sehingga dapat mempermudah seorang dosen dalam memilih asisten praktikum dengan kualitas yang baik.

Kata kunci: *big data, klasifikasi, naïve bayes, hadoop*

Abstract

Selection of the lab assistants experienced by various universities in Indonesia one of them in the Faculty of Computer Science (FILKOM) University of Brawijaya (UB). In the selection of lab assistant, there is some process to be followed. The election process is in FILKOM still in manual form. The process according to which the election is a test lab assistant include administration, test live coding, and test of teaching. In determining the lab assistant acceptance is based on the tests that have been carried out. Obstacles encountered is the similarity test results on the specific provisions that caused confusion electoral process lab assistant. Of these issues, the authors saw an opportunity to create a system that is able to select a way to the highest classification results against the results of tests conducted by the lab assistant candidates. This study uses Hadoop with Naïve Bayes applying methods as may be capable of producing an accurate classification, so as to facilitate a lecturer in choosing a lab assistant with good quality.

Keywords: *big data, classification, naïve bayes, hadoop*

1. PENDAHULUAN

Perkembangan teknologi saat ini membawa kita pada perubahan hampir dari semua aspek di kehidupan. Dalam menentukan sesuatu, manusia pasti dihadapkan banyak pilihan. Dengan adanya pilihan yang benar, akan sangat mempengaruhi kehidupan manusia itu sendiri. Seiring berkembangnya teknologi, perlu dikembangkan dan juga ditingkatkan suatu sumber daya manusia agar semakin berkualitas. Permasalahan seleksi menentukan asisten praktikum banyak dialami di berbagai universitas di Indonesia, khususnya di Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya.

Proses seleksi pada asisten praktikum, di Filkom masih menggunakan cara yang manual. Proses pemilihan asisten praktikum dilakukan

dengan menjalani beberapa tes yang sudah ditentukan, yaitu melalui tiga tahap tes diantaranya tes administrasi, tes *live coding*, dan tes mengajar. Untuk penentuannya dilihat dari hasil tes yang sudah dilakukan sebelumnya. Kemiripan hasil nilai tes pada ketentuan tertentu dapat mengakibatkan kerancuan pada proses pemilihan asisten praktikum, serta dapat berpengaruh pada kualitasnya. Dan sangat diharapkan agar pemilihan asisten praktikum terpilih secara tepat. Untuk menghindari kerancuan hasil, sangat dibutuhkannya suatu sistem yang mampu membantu dalam seleksi pemilihan asisten praktikum baru dengan hasil seleksi yang tepat.

Pada penelitian oleh (Kalamasyah, 2014) yaitu tentang aplikasi untuk pemilihan asisten praktikum dan juga lab yang berbasis web menggunakan metode pengerjaan SDLC *waterfall* pada tahap implementasi. Orang yang terlibat dalam aplikasi tersebut adalah seorang laboran sebagai admin yang

mempunyai akses penuh dalam mengelola informasi seleksi, melakukan *input* nilai, dan melihat hasil perbandingan. Untuk perhitungan seleksi menggunakan metode *Analytic Hierarchy Process*. Hasil dari penelitian tersebut, laboran dapat mengetahui nilai paling tinggi sampai nilai paling rendah dari semua hasil seleksi. Hasil seleksi di pilih dengan nilai tes asisten baru dengan nilai tertinggi.

Selanjutnya penelitian yang dilakukan oleh (Sholihah, 2013), membahas mengenai penentuan jumlah UKT di Universitas Trunojoyo Madura. Penentuan UKT tersebut berdasarkan kriteria yang sesuai. Dalam pembuatan sistem, peneliti menggunakan metode *Naïve Bayes Classifier* (NBC). Hasil yang didapatkan pada skenario satu yaitu 90%, dan skenario hasil dua data kategori 78%.

Kemudian penelitian *Naive Bayes Classifier* oleh (Kusumadewi, 2009) adalah penggunaan alat ukur antropometri sebagai variabel masukan dalam pengklasifikasian status gizi manusia. Untuk mendapatkan hasil yang optimal terdapat beberapa pedoman antropometri yaitu usia dan berat badan, tinggi dan panjang badan, serta lingkaran lengan atas. Dalam penelitian ini didapatkan 5 golongan untuk status gizi, diantaranya berat kurang, berat normal, obesitas kurang, obesitas sedang, dan obesitas berat. Untuk klasifikasi status gizi metode yang digunakan adalah metode *Naive Bayes Classifier* dengan hasil akurasi cukup tinggi yaitu 93,2%.

Berdasarkan permasalahan di atas, maka dibuat suatu sistem yang mampu menyeleksi dengan cara melihat hasil klasifikasi tertinggi terhadap hasil tes yang dilakukan oleh calon asisten praktikum. Penelitian ini menggunakan konsep *Big Data* yang seiring berjalannya waktu data yang digunakan akan semakin besar sebagai data histori yang dikumpulkan. Pada konsep tersebut menggunakan *Hadoop* yang dapat memproses data dalam ukuran besar. Metode yang digunakan adalah *Naïve Bayes* karena dianggap mampu menghasilkan klasifikasi yang akurat, sehingga dapat mempermudah seorang dosen dalam memilih asisten praktikum dengan kualitas yang baik.

2. DASAR TEORI

2.1 Big Data

Konsep *Big Data* pada dasarnya adalah sebuah “lautan data”, dengan banyaknya informasi dan sarana untuk menganalisisnya (Trivu dan Ivan, 2014). *Big Data* merupakan sebuah kombinasi teknologi yang dapat mengelola data yang beragam dalam jumlah besar, dengan kecepatan dan waktu yang tepat dalam keperluan analisis dan reaksi. *Big Data* memiliki tiga karakteristik, yaitu *volume*, *velocity*, dan *variety*.

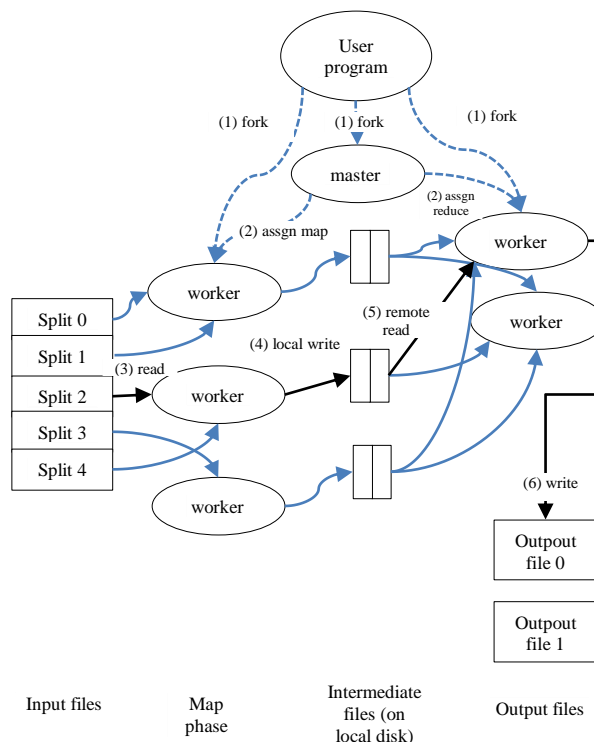
Big Data dideskripsikan sebagai suatu dataset dan teknik analisis di dalam suatu aplikasi yang

memiliki jumlah data sangat besar dan juga sangat kompleks. *Big Data* membutuhkan suatu penyimpanan, manajemen, analisis, dan juga teknologi visualisasi (Chen, Chiang, & Storey, 2012, p 1161 dalam (Irem, 2016). Menurut McAfee & Brynjolfsson, 2012 dalam (Irem, 2016), menerangkan bahwa untuk memanggil data yang begitu besar, dibutuhkan beberapa kategori, terdapat tiga kategori yang memenuhi, diantaranya *volume*, *velocity*, dan *variety*. Dengan masing-masing penjelasan *volume* merupakan jumlah pada data, *velocity* menerangkan tentang kecepatan pencarian informasi data, kemudian *variety* adalah macam-macam jenis data misalnya dalam bentuk txt, doc, dan lain sebagainya.

2.1.1 Mapreduce

Mapreduce merupakan suatu *framework* yang digunakan pada aplikasi dan program yang dikenalkan oleh google untuk menjalankan pekerjaan komputasi yang terdistribusi dan dijalankan pada *cluster*. Konsep yang digunakan dalam Mapreduce adalah fungsi Map dan Reduce untuk *functional programming* (Industri, 2013).

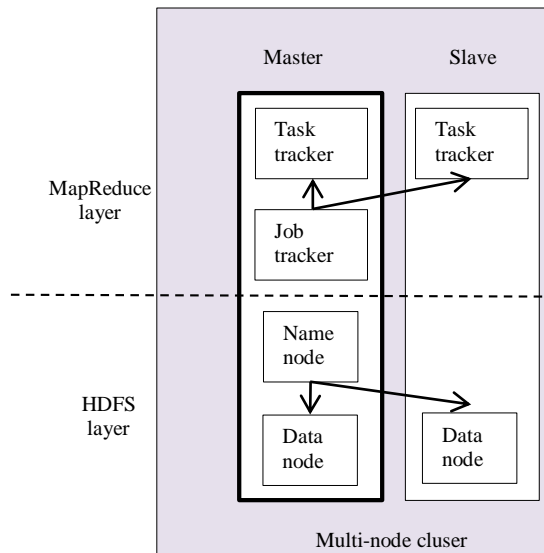
Hadoop Mapreduce termasuk bagian dari susunan kerja yang dibuat agar lebih mudah dalam melakukan suatu perhitungan. Hadoop Mapreduce ini membuat user untuk bertugas lebih ringan dan berjalan secara sejajar dalam node berbeda, menghemat waktu dalam mengeksekusi keseluruhan.



Gambar 1. Mapreduce

2.1.2 Multi Node

Dalam hadoop multi node ini dibutuhkan dua buah komputer, komputer pertama digunakan pada cluster sedangkan komputer kedua digunakan pada slave. Dua komputer ini dikonfigurasi yang merupakan dari mesin dua *single node*.



Gambar 2. Hadoop Multi Node

2.2 Klasifikasi

Klasifikasi merupakan penentuan objek ke dalam suatu kategori atau kelas. Penentuan objek menggunakan beberapa model (Han, 2006). Dalam memulai suatu klasifikasi data dengan membangun sebuah *rule* klasifikasi dengan algoritma tertentu yang digunakan pada data *training* dan data *testing*.

Pada penelitian ini untuk klasifikasi dalam perhitungannya menggunakan metode *Naïve Bayes Classifier*.

2.3 Naive Bayes Classifier

Naïve Bayes Classifier merupakan penyederhanaan dari teorema Bayes, penemu metode ini adalah seorang ilmuwan Inggris yang bernama Thomas Bayes. Algoritma dalam metode Naïve Bayes didasarkan dengan teknik klasifikasi (Kusumadewi, 2009) dapat dibuktikan bahwa saat kecepatan sangat tinggi dan bersamaan diaplikasikan dalam suatu database dengan jumlah data yang besar, naive bayes mempunyai akurasi dan juga kecepatan yang tinggi (Nugroho, 2009).

Metode *Naive Bayes* dengan prinsip teorema *Bayes* mempunyai atribut yang saling berhubungan satu sama lain. Pendekatan yang digunakan teorema bayes yaitu menghitung probabilitas sebuah kejadian pada kondisi tertentu (Lukito & Chrismanto, 2015). Dasar dari teorema *Bayes* dinyatakan dalam persamaan (Bustami, 2013).

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (1)$$

Keterangan:

X : Data kelas yang belum diketahui.

H : Hipotesis dari data X yaitu suatu kelas

Spesifik.

$P(H|X)$: Probabilitas Hipotesis H berdasarkan kondisi X .

$P(H)$: Probabilitas Hipotesis H

$P(X|H)$: Probabilitas X berdasarkan kondisi H

$P(X)$: Probabilitas X

Pada rumus di atas dapat dijelaskan bahwa teorema naive bayes dibutuhkan sebuah petunjuk sebagai proses penentu kelas yang sesuai dengan sampel. Sehingga dibutuhkan kesesuaian terhadap teorema bayes sebagai berikut:

$$P(C|F1 \dots Fn) = \frac{P(C)P(F1 \dots Fn|C)}{P(F1 \dots Fn)} \quad (2)$$

Keterangan:

C : Sebagai kelas

$F1 \dots Fn$: Petunjuk atau syarat kondisi

3. IMPLEMENTASI

Dalam penyelesaian masalah seleksi pemilihan sistem praktikum di Filkom Universitas Brawijaya, data pada penelitian ini berupa atribut hasil *live* koding, mata kuliah, hasil mengajar, dan kelas. Kemudian hasil klasifikasi yang didapatkan adalah sangat disarankan, disarankan, dan yang terakhir tidak disarankan.

Implementasi berisi kode program dan tahapan-tahapan yang dilakukan pada *hadoop*. Implementasi pada permasalahan seleksi pemilihan asisten praktikum menggunakan bahasa pemrograman java. Dalam pengimplementasian menggunakan tiga class, satu class sebagai map, satu class lain sebagai reduce, dan class lainnya sebagai class main.

3.1 Kode Program

Berikut kode program dari kedua class, yaitu class Map.java dan class Reduce.java.

1. Map.java

```
public void map(LongWritable key, Text value,
Context context) throws IOException,
InterruptedException {
    if(test_input==null)
        test_input=context.getConfiguration().get(
            "test_input").split("\\,");
    String[] input=value.toString().split("\\,");
    for(int j=0;j<input.length;j++){
        if(j==input.length-1){
```

```

        if(outcome_count.containsKey(input[j]))
            outcome_count.put(input[j],
            outcome_count.get(input[j])+1);
        else
            outcome_count.put(input[j],
            (double) 1);
    }
    else{
        if(input[j].contentEquals(test_input[j])){

            if(!inputs.containsKey(j+" "+input
            t[j]))

                inputs.put(j+" "+input[j], 0);

            if(features_count.containsKey(j+
            " "+input[j]+" "+input[input.length-1]))

                features_count.put(j+" "+input[j
            ]+" "+input[input.length-1],
            features_count.get(j+" "+input[j]+" "+inp
            ut[input.length-1])+1);
            else
                features_count.put(j+" "+input[j
            ]+" "+input[input.length-1], (double) 1);
        }
    }
    ++count;
}

```

2. Reduce.java

```

public class NBCReduce extends
Reducer<IntWritable, Text, IntWritable, Text>{
    public void reduce(IntWritable key,
    Iterable<Text> values, Context context) throws
    IOException, InterruptedException{
        Double
        out_value=Double.NEGATIVE_INFINITY;
        String out_key=null;
        HashMap<String,Integer>
        final_output=new
        HashMap<String,Integer>();
        for(Text value:values){
            if(final_output.containsKey(v
            alue.toString()))
                final_output.put(value.toStri

```

```

        ng(),
        final_output.get(value.toString()+1);
        else
            final_output.put(value.toStri
            ng(), 1);
    }

    for(Entry<String,Integer>
    output:final_output.entrySet()){
        if(output.getValue()>out_value){

            out_value=(double)
            output.getValue();
            out_key=output.getKey();
        }
    }

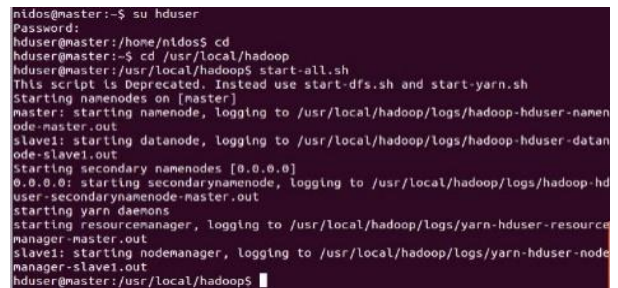
    context.write(null, new
    Text(out_key));
}
}

```

3.2 Langkah-langkah pada Hadoop

Langkah-langkah dimulai dari start sampai mengeluarkan output dari metode *naïve bayes*:

1. Ketik start-all.sh



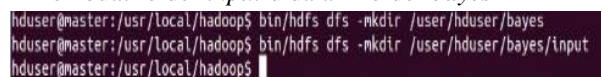
```

nidos@master:~$ su hduser
Password:
hduser@master:~/home/nidos$ cd
hduser@master:~$ cd /usr/local/hadoop
hduser@master:~/usr/local/hadoop$ start-all.sh
This script is deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-namen
ode-master.out
slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-datan
ode-slave1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-master.out
Starting yarn daemons
Starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourc
e-manager-master.out
slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-nod
e-manager-slave1.out
hduser@master:~/usr/local/hadoop$

```

Gambar 3. Start-all.ssh

2. Membuat folder *bayes* di *hduser*, kemudian membuat folder *input* didalam folder *bayes*



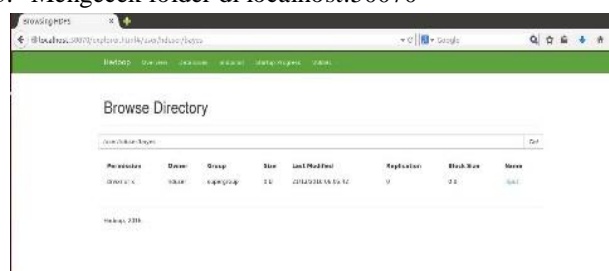
```

hduser@master:~/usr/local/hadoop$ bin/hdfs dfs -mkdir /user/hduser/bayes
hduser@master:~/usr/local/hadoop$ bin/hdfs dfs -mkdir /user/hduser/bayes/input
hduser@master:~/usr/local/hadoop$

```

Gambar 4. Membuat Folder

3. Mengecek folder di localhost:50070



Gambar 5. Cek Folder

4. *Compile* semua file java

```
hduser@master:/usr/local/hadoop$ bin/hdfs com.sun.tools.javac.Main NBC*.java
Note: NBCDriver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
hduser@master:/usr/local/hadoop$
```

Gambar 6. *Compile* File Java

5. Mengecek pada direktori apakah file sudah menghasilkan *.class

Gambar 7. *Class* Pada Direktori

6. Membuat file *.jar dari hasil *.class

```
hduser@master:/usr/local/hadoop$ jar cf bayes.jar NBC*.class
hduser@master:/usr/local/hadoop$
```

Gambar 8. File *.jar

7. Mengecek pada direktori



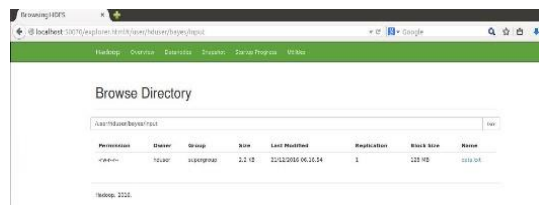
Gambar 9. Direktori

8. Memasukkan file data ke folder *input* localhost

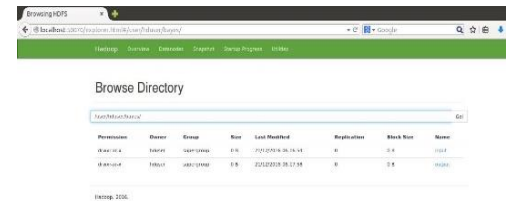
```
hduser@master:/usr/local/hadoop$ bin/hdfs dfs -copyFromLocal /home/nidos/downloads/data.txt /user/hduser/bayes/input
hduser@master:/usr/local/hadoop$
```

Gambar 10. File Pada Folder *Input*

9. Mengecek apakah file data sudah di localhost:50070

Gambar 11. File Pada *Localhost*10. Membuat folder *output* di localhost:50070

```
hduser@master:/usr/local/hadoop$ bin/hdfs dfs -mkdir /user/hduser/bayes/output
hduser@master:/usr/local/hadoop$
```

Gambar 12. Folder *Output*11. Mengecek folder *output* apakah sudah ada di localhost:50070Gambar 13. Folder *Output*

12. Mengecek secara manual folder apa saja yang ada di localhost:50070

```
hduser@master:/usr/local/hadoop$ bin/hdfs dfs -ls /user/hduser/bayes
Found 2 items
drwxr-xr-x - hduser supergroup 0 2016-12-21 06:16 /user/hduser/bayes/input
drwxr-xr-x - hduser supergroup 0 2016-12-21 06:17 /user/hduser/bayes/output
hduser@master:/usr/local/hadoop$
```

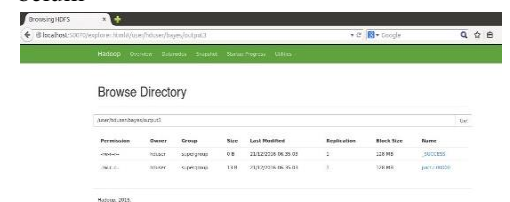
Gambar 14. Cek Folder

13. Proses menjalankan program

```
hduser@master:/usr/local/hadoop$ bin/hadoop jar bayes.jar NBCDriver lulus,rpl,bak /user/hduser/bayes/input/data.txt /user/hduser/bayes/output3
16/12/21 06:34:54 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id
16/12/21 06:34:54 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=
16/12/21 06:34:55 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/12/21 06:34:55 INFO input.FileInputFormat: Total input paths to process : 1
16/12/21 06:34:55 INFO mapreduce.JobSubmitter: number of splits:1
16/12/21 06:34:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local10059613_0001
16/12/21 06:34:57 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
16/12/21 06:34:57 INFO mapreduce.Job: Running job: job_local10059613_0001
16/12/21 06:34:57 INFO mapreduce.LocalJobRunner: OutputCommitter set in config null
16/12/21 06:34:57 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
16/12/21 06:34:57 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
16/12/21 06:34:58 INFO mapred.LocalJobRunner: Waiting for map tasks
16/12/21 06:34:58 INFO mapred.LocalJobRunner: Starting task: attempt_local10059613_0001_m_0000000_0
16/12/21 06:34:58 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
16/12/21 06:34:58 INFO mapred.Task: Using ResourceCalculatorProcessTree: [ ]
16/12/21 06:34:59 INFO mapred.MapTask: Processing split: hdfs://master:9000/user/hduser/bayes/input/data.txt:0:2292
16/12/21 06:34:58 INFO mapreduce.Job: Job job_local10059613_0001 running in uber mode : false
16/12/21 06:34:58 INFO mapreduce.Job: map 0% reduce 0%
16/12/21 06:34:59 INFO mapred.MapTask: Processing split: hdfs://master:9000/user/hduser/bayes/output3/data.txt:0:2292
16/12/21 06:34:59 INFO mapred.MapTask: mapreduce.task.sort.mb: 100
16/12/21 06:34:59 INFO mapred.MapTask: soft limit at 83886080
```

Gambar 15. Menjalankan Program

14. Melihat apakah hasil sudah keluar atau belum

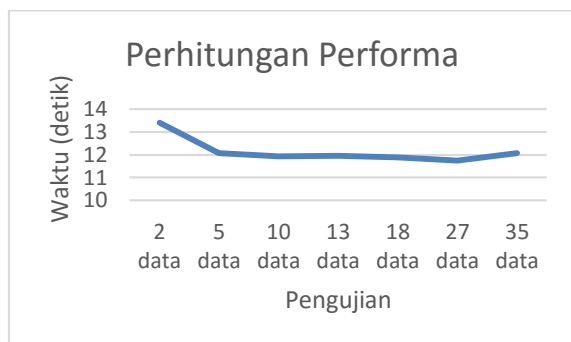


Gambar 16. Cek Hasil

4. PENGUJIAN DAN ANALISIS

Pengujian yang pertama adalah performa, hal ini dilakukan dengan menghitung waktu komputasi dari data. Dalam perhitungannya yaitu dihitung masing-masing data berapa lama waktu saat komputasi dijalankan. Pengujian ini dilakukan sebanyak 7 kali, kemudian dari 35 data tersebut dilakukan perhitungan rata-ratanya. Data yang digunakan diasumsikan akan terus disimpan setiap

kali ada perekrutan baru pada asisten praktikum sebagai data histori sekaligus data latih, sehingga seiring berjalannya waktu, maka data tersebut akan semakin besar atau masuk pada konteks *Big Data*. Pada Gambar 17 ditunjukkan bahwa waktu komputasi yang dibutuhkan sangat fluktuatif, karena memang datanya memang masih tergolong masih sedikit dan hanya sebagai simulasi dalam lingkungan hadoop multinode cluster. Selain itu, lama waktu juga tergantung dari aktifitas komputer.



Gambar 17. Grafik Pengujian Performa

Pengujian yang kedua adalah akurasi, hal ini dilakukan untuk melihat kualitas hasil data yang diuji terhadap ukuran seberapa besar kesesuaian dengan data aktual. Akurasi yang dihasilkan yaitu jumlah seluruh nilai yang benar dibagi dengan jumlah data. Hasil yang didapatkan 80 %.

5. KESIMPULAN DAN SARAN

Kesimpulan yang didapat adalah bahwa dengan menggunakan hadoop multinode dan metode *Naïve Bayes* yang digunakan untuk klasifikasi pemilihan asisten praktikum dapat diperoleh hasil yang cukup baik yaitu sebesar 80%. Saran yang dapat dilakukan dalam pengembangan selanjutnya yaitu, jumlah data yang digunakan dalam pengujian lebih banyak lagi dan adanya pengujian jumlah node pada PC slaves. Kemudian menerapkan metode lainnya yang ada pada *Big Data*. Dapat juga dilakukan *hybrid* atau penggabungan dengan 2 metode atau lebih agar hasil yang didapatkan lebih optimal.

6. DAFTAR PUSTAKA

- (2013, Oktober). Diambil kembali dari Apache TM Hadoop @ homepage: <http://hadoop.apache.org/>.
- BUSTAMI. 2013. Penerapan Algoritma Naive Bayes untuk Mengklasifikasi Data Nasabah Asuransi.
- HAN, J. K. 2006. *Concept and Techniques*. Morgan Kaufmann.
- INDUSTRI, M. 2013. *Definisi cloud Computing*. Diambil kembali dari Cloud Computing: Meruvian.org

IREM, D. 2016. Classifying Multi-Destination Trips in Austria with Big Data. *Elsevier*, 2211-9736.

KALAMSYAH, S. A. 2014. Aplikasi Pendukung Keputusan Seleksi Asisten Praktikum dan Lab Menggunakan Metode Analytical Hierarchi Process (Studi Kasus: Lab Informatika Universitas Telkom).

KUSUMADEWI, S. 2009. *Klasifikasi Status Gizi Menggunakan Naive Bayesian Classification*.

LUKITO, Y., & CHRISMANTO, A. R. 2015. Perbandingan Metode-Metode Klasifikasi Untuk Indoor Positioning System. *Jurnal Teknik Informatika dan Sistem Informasi*, 2.

SHOLIHAH, A. 2013. *Sistem Penentuan Uang Kuliah Tunggal (UKT) Menggunakan Metode Naive Bayes Classifier*.

SOEMARSO. 2003. *Akuntansi Suatu Pengantar*. Jakarta: Salemba Empat.