Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) p-ISSN: 2355-7699
Vol. 4, No. 2, Juni 2017, hlm. 81-86 e-ISSN: 2528-6579

KLASIFIKASI SEPEDA MOTOR BERDASARKAN KARAKTERISTIK KONSUMEN DENGAN METODE K-NEAREST NEIGHBOUR PADA BIG DATA MENGGUNAKAN HADOOP SINGLE NODE CLUSTER

Nanda Agung Putra[1], Ardisa Tamara Putri[2], Dhimas Anjar Prabowo[3], Listiya Surtiningsih[4], Raissa Arniantya[4], Imam Cholissodin[6]

[1,2,3,4,5,6]Fakultas Ilmu Komputer Universitas Brawijaya
Email: [1]nandoooo1412@gmail.com, [2]ardisatamar@gmail.com, [3]dhimasanjar@gmail.com, [4]listiyasurtiningsih@gmail.com, [5]rarniantya@gmail.com, [6]imamcs@ub.ac.id

(Naskah masuk: 6 Januari 2017, diterima untuk diterbitkan: 7 Mei 2017)

Abstrak

Penelitian ini mengusulkan sebuah klasifikasi terhadap sepeda motor berdasarkan karakteristik konsumen. Sepeda motor memiliki beberapa jenis dan merk yang berbeda sehingga menyebabkan banyaknya pilihan yang dimiliki konsumen. Konsumen akan memilih sepeda motor yang diinginkannya berdasarkan latar belakang yang berbeda. Pada penelitian ini, Konsumen akan dikelompokkan berdasarkan sepeda motor yang dibeli sehingga penjual dapat mengetahui karakteristik konsumen yang membeli suatu jenis atau merk tertentu. Karakterisitik konsumen dapat ditentukan dengan usia, jenis kelamin, pendapatan, status pernikahan dan jumlah anak. Berdasarkan karakteristik tersebut perlu dilakukan pengelompokan untuk menentukan merk sepeda motor. Dalam penelitian ini metode yang digunakan yakni K-Nearest Neighbour (K-NN). K-NN merupakan algoritma yang umum digunakan untuk klasifikasi dan mencari kelas dari data uji dengan mayoritas kelompok yang memiliki jarak terdekat. Dataset yang digunakan dalam penelitian ini yaitu karakteristik konsumen. Uji coba dengan dataset tersebut menghasilkan merk sepeda motor dari data uji yang sudah ditentukan.

Kata kunci: *k-nearest neighbor, klasifikasi, k-nearest neighbor classification, sepeda motor.*

Abstract

This research proposed a classification of motorcycle based on customer's characteristics. Motorcycles have different type and brand so that customers have many choices. Customer will choose motorcycle which they want to be based on different background. In this study, the customer will be grouped by motorcycle were purchased so that the seller can know characteristics of customers who buy certain type or brand. Characteristics of customers can be determined by age, gender, income, status and number of children. Based on these characteristic, we have to group for specifying motorcycle's type. In this research, the method used K-Nearest Neighbor (K-NN). K-NN algorithm is commonly used for classifying and searching for a group of test data with the majority of the group that has the shortest distance. The dataset used in this project is the final consumer characteristics. Trials with the dataset produce motorcycle brand of test data that has been determined.

Keywords: *k-nearest neighbor, classification, k-nearest neighbor classification, motorcycle.*

1. PENDAHULUAN

Sepeda motor adalah kendaraan yang memiliki dua roda dimana mesin bekerja sebagai penggeraknya (Cossalter, 2006). Sepeda motor memiliki 2 roda yang sebaris sehingga sepeda motor dapat tetap stabil pada kecepatan tinggi yang disebabkan oleh gaya giroskopik. Sedangkan kestabilan pada kecepatan rendah bergantung pada pengaturan setang oleh pengendara. Sepeda motor sangat populer di Indonesia karena memiliki harga yang murah dan terjangkau untuk kalangan masyarakat dan biaya operasional yang cukup rendah.

Sepeda motor memiliki beberapa jenis dan merk yang berbeda. Hal ini menyebabkan banyaknya pilihan yang dimiliki konsumen. Konsumen akan memilih sepeda motor yang diinginkannya berdasarkan latar belakang yang berbeda. Maka dari itu mengetahui preferensi konsumen akan sangat menguntungkan penjual. Konsumen akan dikelompokkan berdasarkan sepeda motor yang dibeli sehingga penjual dapat mengetahui karakteristik konsumen yang membeli suatu jenis atau Merk tertentu.

Pengelompokkan dilakukan dengan metode KNN. Algoritma ini bekerja dengan cara mencari kelompok data uji berdasarkan k data tetangga terdekatnya. Algoritma ini menerima masukan berupa parameter dan nilai k.

Pada penelitian yang dilakukan oleh Nouvel (2015), klasifikasi kendaraan roda empat menggunakan metode K-Nearest Neighbour (K-NN) dengan jumlah data sebanyak 14 memiliki tingkat akurasi sebesar 78,57% dan RMSE dari 0,23. Jika

jumlah data sebanyak 1728 data, tingkat akurasi sebesar 95,78%, RMSE 0,19 dan ROC daerah 0.99. Pengujian itu dilakukan untuk membuktikan bahwa tingkat akurasi yang dihasilkan dipengaruhi oleh jumlah data latih yang digunakan. Jika semakin banyak data yang dilatih maka semakin tinggi juga tingkat akurasinya.

Permasalahan tersebut melatarbelakangi peneliti untuk melakukan penelitian mengenai pengelompokan sepeda motor berdasarkan karakteristik konsumen dengan metode K-NN. Penelitian ini akan menghasilkan sistem pengelompokkan sepeda motor berdasarkan karakteristik dari konsumen.

2. DASAR TEORI

2.1 Karakteristik Pembeli

Data yang digunakan adalah data karakteristik pembeli yang diadopsi dari GitHub dengan jumlah data sebesar 2500 data. Data terdiri dari 5 parameter yaitu

1. Usia
   Parameter yang menyimpan usia dari pembeli.
2. Pendapatan
   Parameter yang menyimpan pendapatan per bulan dari pembeli.
3. Status Pernikahan
   Parameter yang menyimpan status pernikahan pembeli. Status pernikahan dibagi menjadi 4 yaitu *Divorce*, *Single*, *Married* dan *Widowed*.
4. Jenis Kelamin
   Parameter yang menyimpan jenis kelamin pembeli. Jenis kelamin dibagi menjadi 2 yaitu *Male* dan *Female*.
5. Jumlah Anak
   Parameter yang menyimpan jumlah anak yang dimiliki oleh pembeli.

Data dari GitHub tersebut diadopsi dengan mengganti kelas data mobil, menjadi kelas data sepeda motor sebagai data simulasi, dengan mempertimbangkan kesetaraan harga mobil dan motor dari yang paling mahal sampai yang murah. Kelas data motor yang digunakan dibagi menjadi 4 kelas antara lain Vario, Mio, Next dan Beat.

2.2 Konsep Big Data

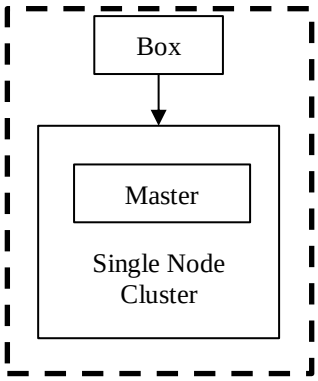
Big data merupakan istilah umum untuk sekumpulan data yang jumlahnya sangat besar dan kompleks sehingga tidak mudah untuk ditangani atau proses hanya dengan metode pemrosesan data biasa. Terdapat tiga masalah utama yang diselesaikan oleh *big data*, antara lain (Pawitra, 2016):

1. *Volume*
   Ukuran data yang disimpan atau diproses.
2. *Velocity*
   Kecepatan membuat data. Kecepatan data dibuat umumnya berbanding lurus dengan volume data.
3. *Variety*
   Keberagaman data yang diolah. Dari segi format maupun struktur data.

Teknologi yang berkaitan dengan big data akan memudahkan proses pengumpulan data-data yang sebelumnya tidak bisa atau sulit untuk dikumpulkan.

2.2.1 Single Node

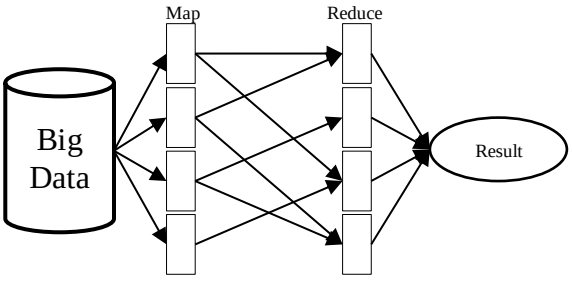
Hadoop *single node* menggunakan 1 mesin / computer saja dalam melakukan prosesnya. Secara default Hadoop dikonfigurasikan untuk berjalan pada mode *non-distributed* (berdiri sendiri). Komputer didesain sebagai *master* bukan *slave* sehingga semua proses dilakukan dalam satu mesin seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur Single Cluster

2.2.2 Mapreduce

Pada Gambar 2, Mapreduce bertujuan untuk memproses data yang memiliki ukuran yang besar secara terdistribusi dan parallel dalam kluster yang terdiri atas ribuan computer. Dalam prosesnya, mapreduce dibagi menjadi 2 proses yaitu map dan reduce. Map berfungsi dalam pengumpulan informasi dari data-data yang terdistribusi dalam tiap komputer. Keluaran dari proses map akan digunakan dalam proses reduce. Proses reduce berfungsi dalam penggabungan atau pengelompokan berdasarkan kata kunci (Dean & Ghemawat, 2004).



Gambar 2. Cara Kerja MapReduce

2.3 KNN

K-Nearest Neighbor (K-NN) merupakan algoritma untuk menentukan kelas objek data uji berdasarkan K objek pada data latih yang terdekat (mirip). Algoritma ini termasuk instance-based learning dan merupakan salah satu teknik lazy learning. Dasar Algoritma K-Nearest Neighbour (Brammer, 2007):

a. Tentukan data latih yang paling dekat dengan data uji.

b. Kelas yang paling sering muncul dari *k* data latih yang terdekat akan dipilih.

2.3.1 Inisialisasi

Menentukan parameter *K*, dimana *K* merupakan jumlah dari tetangga terdekat, nilai *K* untuk menguji data uji ditentukan berdasarkan nilai *K* optimum pada saat *training*.

2.3.2 Alokasi Data

Alokasikan data uji dengan cara menghitung jarak setiap atribut data uji terhadap data latih dengan persamaan (1).

$$d(x_1, x_2) = (x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 \quad (1)$$

dimana x_{ij} menyatakan koordinat titik x_i pada dimensi ke-*j* dan *d* merupakan jarak. Setelah menghitung jarak data uji terhadap data latih, urutkan jarak setiap atribut data uji, kelompokkan menurut jarak yang terdekat.

2.3.3 Klasifikasi

Klasifikasi kelompok data uji dilakukan dengan cara mengumpulkan kategori *Y* (klasifikasi nearest neighbour). Lalu, memilih kategori mayoritas dari *K* data yang ditentukan.

3. IMPLEMENTASI

3.1 Kode Program

Proses ini merupakan proses map dalam *MapReduce* kNN. Proses map sendiri berfungsi dalam pengumpulan informasi data-data yang menjadi input dalam proses kNN. Keseluruhan proses map dapat dilihat dalam Kode Program 1.

```
1  @Override
2  public void map(Object kunci, Text
3  hasil, Context konteks) throws
4  IOException, InterruptedException
5  {
6          String baris = hasil.toString();
7          StringTokenizer token = new
8  StringTokenizer(baris, ",");
9
10         double umur =
11 normalisasi(token.nextToken(), min_umur,
12 max_umur);
13
14         double penghasilan =
15 normalisasi(token.nextToken(),
16 min_penghasilan, max_penghasilan);
17 String status_pernikahan =
18 token.nextToken();
19 String kelamin = token.nextToken();
20         double anak =
21 normalisasi(token.nextToken(), minanak,
22 maxanak);
23         String motor =
24 token.nextToken();
25         double jarak = totaljarak(umur,
26 penghasilan, status_pernikahan,
27 kelamin,anak, umur2, penghasilan2,
28 status_pernikahan2, kelamin2, anak2);
29
30         KnnMap.put(jarak, motor);
31         if (KnnMap.size() > K)
32         {
33
34         KnnMap.remove(KnnMap.lastKey());
35         }
36 }
```

Penjelasan dari Kode Program 1:

1. Baris 1-8 merupakan proses tokenisasi yaitu
2. proses memecah baris menjadi beberapa kata.
3. Baris 8-21 merupakan proses inisialisasi parameter. Terdapat beberapa parameter yang perlu dinormalisasi terlebih dahulu yaitu *age*, *income* dan *children*.
4. Baris 23-28 merupakan proses perhitungan jarak data uji dan data latih.
5. Baris 31 merupakan proses pembuatan *TreeMap* dengan jarak sebagai sebuah *key* dan model sepeda motor sebagai *value*.
6. Baris 32-37 memproses *TreeMap* agar hanya memuat K data. Apabila TreeMap memiliki data lebih dari K data maka akan menghapus data yang tidak diperlukan.

Proses ini merupakan proses reduce dalam *MapReduce* kNN. Proses reduce sendiri berfungsi dalam penggabungan dan pengelompokan berdasarkan *key* atau kata kunci. Keseluruhan proses reduce dapat dilihat pada Kode Program 2.

```
1  public void reduce(NullWritable kunci,
2  Iterable<DoubleString> hasil, Context
3  konteks) throws IOException,
4  InterruptedException
5  {
6          for (DoubleString hsl : hasil)
7          {
8                  String motor =
9  hsl.getmotor();
10                 double jarak =
11 hsl.gettotaljarak();
12
13                 KnnMap.put(jarak,
14 motor);
15                 if (KnnMap.size() > K)
16                 {
17
18         KnnMap.remove(KnnMap.lastKey());
19                 }
20         }
21                 List<String> baris = new
22 ArrayList<String>(KnnMap.values());
```

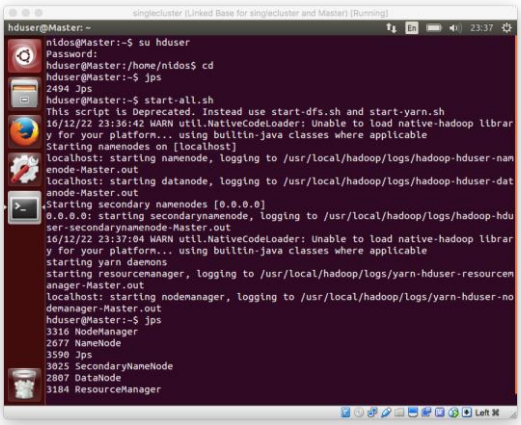
```
23
24                  Map<String, Integer>
25  frekuensi = new HashMap<String,
26  Integer>();
27
28              for(int i=0; i< baris.size();
29  i++)
30              {
31                  Integer frek =
32  frekuensi.get(baris.get(i));
33                  if(frek == null)
34                  {
35
36  frekuensi.put(baris.get(i), 1);
37                  } else
38                  {
39
40  frekuensi.put(baris.get(i), frek+1);
41                  }
42              }
43
44              String MotorMuncul = null;
45              int frekuensi_maks = -1;
46              for(Map.Entry<String,
47  Integer> masukan: frekuensi.entrySet())
48              {
49                  if(masukan.getValue() >
50  frekuensi_maks)
51                  {
52                      MotorMuncul =
53  masukan.getKey();
54                      Frekuensi_maks =
55  masukan.getValue();
56                  }
57              }
58
59          konteks.write(NullWritable.get(),
60  new Text(MotorMuncul));
61          }
62  }
```

Penjelasan dari Kode Program 2 :

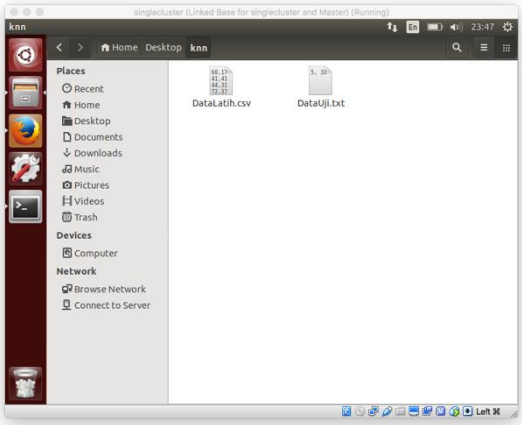
1. Baris 1-20 merupakan proses pembuatan *TreeMap* berdasarkan data dari objek *DoubleString* dengan jarak sebagai *key* dan model sepeda motor sebagai *value* dan mengatur agar *TreeMap* tidak melebihi K data.
2. Baris 21-22 merupakan proses menyimpan nilai dari *TreeMap* dalam sebuah *ArrayList*.
3. Baris 24-25 merupakan proses inisialisasi sebuah *HashMap*.
4. Baris 27-41 merupakan proses menghitung banyaknya frekuensi dati tiam model sepeda motor. *HashMap* digunakan untuk menyimpan nilai frekuensi tersebut sebagai *value* dan model sepeda motor sebagai *key*.
5. Baris 43-57 merupakan proses memeriksa *HashMap* untuk memperoleh model sepeda motor yang memiliki frekuensi paling tinggi.
6. Baris 59-62 merupakan proses penyimpanan hasil perhitungan frekuensi tertinggi ke dalam *context*.

3.2 Hadoop Single Node

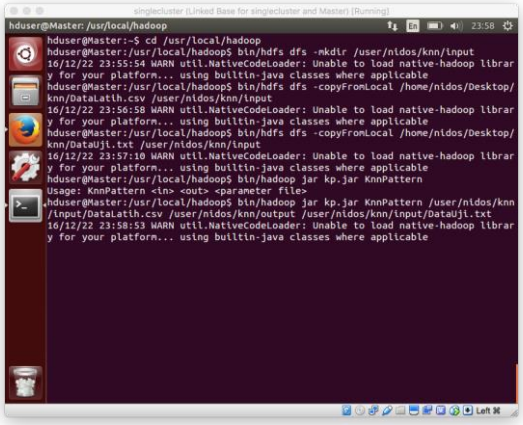
Langkah utama Hadoop Single Node yang perlu dilakukan dapat dilihat pada Gambar 3 hingga 6.



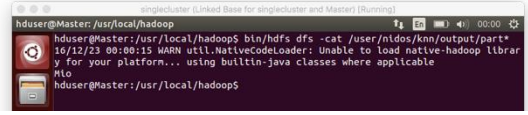
Gambar 3. Jalankan hadoop (start-all.sh)



Gambar 4. Folder Data Latih dan Data Uji



Gambar 5. Proses Klasifikasi



Gambar 6. Contoh Hasil Klasifikasi

Dengan penjelasan sebagai berikut:

- Langkah 1: Pada Gambar 3, masuk sebagai hduser, cek kondisi apakah hadoop telah berjalan atau belum dengan jps, jika belum maka jalankan hadoop dengan start-all.sh, lalu cek kebali kondisi hadoop dengan jps
- Langkah 2: Pada Gambar 4, menyiapkan file data latih dan data uji, dengan data latih berupa file .csv dan data uji berupa file .txt
- Langkah 3: Pada Gambar 5, masuk ke direktori hadoop dengan path /user/local/hadoop, lalu membuat direktori input pada hdfs, dan memasukkan file data latih serta data uji kedalam direktori input yang telah dibuat, selanjutnya cek bentuk format penjalanan programnya dengan bin/hadoop jar kp.jar KnnPattern, kemudian jalankan program dengan masukkan path file data latih, path direktori output, dan path file data uji dengan urutan <data latih> <output> <data uji>, lalu tekan enter dan tunggu program berjalan, dan lihat apakah program berjalan dengan benar tanpa error
- Langkah 4: Pada Gambar 6, menampilkan output hasil program dengan memasukkan path direktori output yang ditentukan pada saat menjalankan program pada langkah ke-3

Berdasarkan Gambar 3 hingga 6, Hadoop Single node dapat berjalan dengan baik dalam melakukan komputasi klasifikasi sepeda motor digunakan kode program KnnPattern.java dan data latih yang didapatkan dari sumber ini (https://github.com/matt-hicks/MapReduce-KNN) dengan membuat analogi kesetaraan tingkat harga mobil, yang disesuaikan dengan urutan harga motor dalam penelitian ini sebagai kelas, baik untuk data latih maupun data uji.

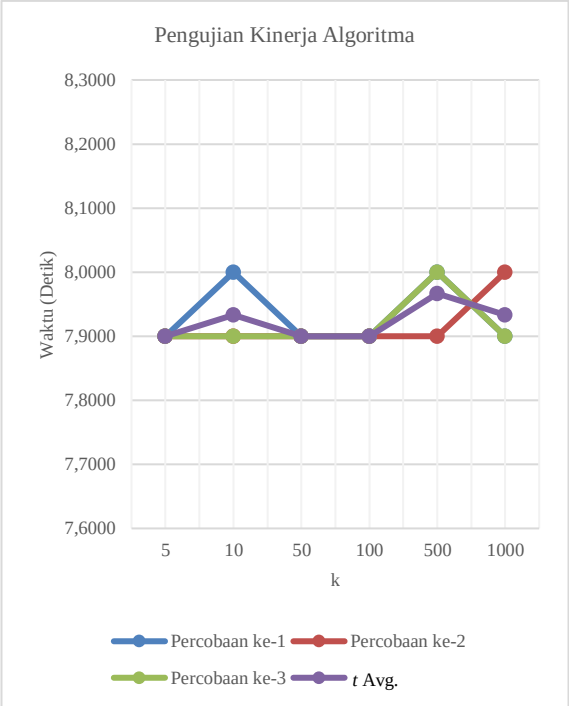
4. PENGUJIAN DAN ANALISIS

4.1 Pengujian Kinerja Algoritma

Pengujian kinerja algoritma dilakukan dengan cara menghitung seberapa cepat waktu yang dibutuhkan untuk mengolah data dengan nilai *k* yang bervariasi. Hasil dari pengujian dapat dilihat pada Tabel 1 dan Gambar 7.

Tabel 3.1 Hasil Pengujian Kinerja Algoritma

Nilai *k*	Lama Waktu Komputasi (*t*) Percobaan ke-*i*			*t* Avg.
	1	2	3	
5	7.9	7.9	7.9	7.9000
10	8.0	7.9	7.9	7.9333
50	7.9	7.9	7.9	7.9000
100	7.9	7.9	7.9	7.9000
500	8.0	7.9	8.0	7.9667
1000	7.9	8.0	7.9	7.9333



Gambar 7. Grafik Hasil Pengujian Kinerja Algoritma

Pengujian ini dilakukan dengan beberapa nilai k seperti yang dicantumkan pada Tabel 1, dan untuk setiap nilai *k*-nya akan dijalankan sebanyak 3 kali percobaan. Dimana hasil akan dilihat adalah lama waktu komputasi dalam satuan detik. Dari Tabel 1 dan Gambar 7, dapat dilihat bahwa lama waktu komputasi dalam ketiga percobaan tidak terlalu berbeda dengan perbedaan waktu hanya 0.1 detik pada beberapa nilai *k*, atau dapat dikatakan bahwa program dengan nilai *k* sebanyak 5 hingga 1000 dapat berjalan dengan lama waktu yang hampir sama. Namun, walaupun program dapat berjalan dengan waktu yang perbedaannya tidak begitu signifikan, pada klasifikasi atau keluaran tipe kelas untuk beberapa nilai *k* memiliki hasil yang berbeda-beda tergantung kelas apa yang menjadi mayoritas dalam nilai *k* tersebut. Hal ini menunjukkan juga, bahwa hadoop single node cluster memiliki kemampuan pengolahan data yang cepat dan handal.

5. KESIMPULAN DAN SARAN

Pada implementasi hadoop single node menggunakan metode K-Nearest Neighbour (KNN) pada klasifikasi sepeda motor berdasarkan karakteristik konsumen. Klasifikasi dilakukan dengan menentukan jumlah tetangga terdekat dengan cara menghitung jarak antara data uji dan data latih. Hasil klasifikasi diperoleh dari mayoritas kelas/kategori dalam jumlah tetangga terdekat. Dalam implementasi ini, dilakukan pengujian lama waktu komputasi berdasarkan jumlah tetangga terdekat (*k*) yakni sebanyak 5 hingga 1000 dengan percobaan sebanyak 3 kali dan waktu komputasi

pada tiap percobaan tidak terlalu signifikan dengan perbedaan waktu 0.1 detik. Hasil waktu rata-rata terbaik yaitu 7.9 detik pada nilai *k*, masing-masing 5, 50, dan 100.

Adapun saran untuk penelitian selanjutnya, yaitu metode yang digunakan dapat dilanjutkan dengan teknik lain untuk mengetahui perbandingan waktu komputasi terbaik antara hadoop single node cluster dan multi mode cluster, dan juga menambahkan variasi pengujian yang lainnya, misal menggunakan *k*-fold cross-validation.

6. DAFTAR PUSTAKA

BRAMMER., 2007. *Principles of Data Mining.* UK: University of Portsmouth.

COSSALTER, V., 2016. Motorcycle Dynamics.

DEAN, J., & GHEMAWAT, S., 2004. *MapReduce: Simplified Data Processing on Large Clusters.* Google Corp.

NOUVEL, AHMAD, 2015. Klasifikasi Kendaraan Roda Empat Berbasis KNN. Jurnal Bianglala Informatika Vol 3 No 2.

PAWITRA, P. M., 2016. *Paper Basis Data.* Surakarta.

S., RUSSEL, & P, NORVIG., 2010. *Artificial Intelligence A Modern Approach.* New Jersey: Pearson Education, Inc.

X., WU, & V., KUMAR., 2009. *The Top Ten Algorithm in Data Mining.* Chapman and Hall.