

**PERBANDINGAN REGRESI LOGISTIK, KLASIFIKASI
NAIVE BAYES, DAN RANDOM FOREST**
**(COMPARISON THE LOGISTIC REGRESSION, NAIVE
BAYES CLASSIFICATION, AND RANDOM FOREST)**

SRI WAHYUNI KALUMBANG*, SUBANAR

Abstract. Classification analysis is a method that aims to group a number of observations or observations into certain classes based on features or independent variables from these observations. In Hasti et al. [7] it is stated that statistical researchers in 2003 competed various classification methods and various metrics to measure their goodness in line with the scientific activities carried out by the statisticians above. Therefore, the researcher is interested in comparing 2 statistic methods and the now popular method, namely the random forest, using data sets taken from kaggle.com and using FPR and accuration to see the goodness of the model. The results of this study indicate that the random forest method has better predictive results than the other two methods seen from its accuration.

Keywords: machine learning, classification, classical method, random forest, FPR.

Abstrak. Analisis klasifikasi adalah suatu metode yang bertujuan untuk mengelompokkan sejumlah observasi atau pengamatan ke dalam kelas-kelas tertentu berdasarkan *feature* atau variabel independen dari observasi tersebut. Dalam Hasti dkk [7] disebutkan bahwa para peneliti statistika pada tahun 2003 melakukan kompetisi berbagai metode klasifikasi dan berbagai metrik untuk ukuran kebaikannya sejalan dengan kegiatan ilmiah yang dilakukan para statistikawan di atas. Oleh karena itu, Peneliti mempunyai ketertarikan untuk membandingkan 2 metode statistik dan metode yang sekarang populer yaitu random forest dengan menggunakan data set yang diambil dari *kaggle.com* dan menggunakan FPR dan akurasi untuk melihat kebaikan model. Hasil penelitian ini menunjukkan bahwa metode random forest memiliki hasil prediksi yang lebih baik dibandingkan dua metode lainnya dilihat dari akurasinya.

Kata-kata kunci: machine learning, klasifikasi, metode klasik, random forest, FPR.

1. PENDAHULUAN

Analisis klasifikasi adalah suatu metode yang bertujuan untuk mengelompokkan sejumlah observasi atau pengamatan ke dalam kelas-kelas tertentu berdasarkan *feature* atau variabel independen dari observasi tersebut. Kelas-kelas yang dimaksud merupakan bagian dari suatu variabel yang disebut variabel dependen. Umumnya klasifikasi dapat dilakukan dengan pendekatan statistik maupun pendekatan non statistik.

Salah satu bagian yang paling sulit adalah memilih klasifier yang cocok dan tepat untuk melakukan pengambilan keputusan. Meskipun ada banyak teknik dalam proses klasifikasi data, memilih classifier terbaik menjadi pekerjaan yang menantang. Ini membuat masalah bagi para peneliti yang sering menggunakan lebih banyak data untuk pekerjaan penelitian mereka. Pada penelitian ini, Metode Klasifikasi Bayesian dan Regresi Logistik akan dibandingkan untuk mengetahui klasifier yang sesuai untuk data saat melakukan pengklasifikasian.

Dalam Hasti et all [7] disebutkan bahwa para peneliti statistika pada tahun 2003 melakukan kompetisi berbagai metode klasifikasi dan berbagai metrik untuk ukuran kebaikannya sejalan dengan kegiatan ilmiah yang dilakukan para statistikawan di atas. Peneliti mempunyai ketertarikan untuk membandingkan 2 metode statistik dalam klasifikasi dan metode *machine learning*.

Mengapa *machine learning*? Saat ini *machine learning* sebagai analisis data generasi ke-3 perkembangannya sangat pesat, termasuk terapannya dalam masalah klasifikasi. Namun demikian analisis ini masih relatif sedikit dikaji oleh para peneliti, salah satunya metode random forest. Selain itu, baik metode regresi logistik maupun **Naive Bayes** mempunyai kelemahan dalam hal bias maupun variansi. Menurut Hastie et all [7] metode *bootstrap agregation* atau *bagging* bisa mengatasi masalah-masalah yang ditimbulkan dari penggunaan regresi logistik atau **Naive Bayes**. Menurut Breiman [3], random forest adalah modifikasi metode *bagging* yang lebih sederhana; sebagai akibatnya random forest sangat populer dan bisa diterapkan dibanyak persoalan. Oleh karena itu kajian mengenai *machine learning* melalui random forest untuk data *outcome* biner sangat menarik, dan bahkan perlu diketahui tingkat keakuratan, dan kelebihan-kelebihan metode random forest dibandingkan dengan metode regresi logistik dan klasifikasi **Naive Bayes**, akan tetapi karena tidak dapat dibandingkan secara teoritis maka akan dibandingkan secara empiris.

Data yang digunakan dalam penelitian ini adalah data yang diambil dari *kaggle.com* yaitu *income evaluation*. Data ini dipilih untuk melihat kebaikan dari ketiga metode yang akan dibandingkan.

2. RUMUSAN MASALAH

Berdasarkan uraian latar belakang masalah tersebut, maka permasalahan yang dapat dirumuskan dari penelitian ini adalah sebagai berikut:

- (1) Bagaimana metode regresi logistik, naive bayes, dan random forest bekerja?
- (2) Bagaimana hasil ketepatan dari metode regresi logistik, naive bayes, dan random forest?

3. LANDASAN TEORI

3.1. Analisis Klasifikasi. Klasifikasi adalah suatu metode analisis data yang membentuk model (selanjutnya disebut classifier) yang dapat digunakan untuk mendeskripsikan dan memprediksi kelas dari suatu data, atau dengan kata lain memberi label pada data berdasarkan suatu variabel dependen atau kelas tertentu. Secara umum klasifikasi data terdiri dari dua tahap, yaitu learning step di mana algoritma klasifikasi akan membentuk suatu classifier dengan menganalisa dataset yang ada, atau sering disebut dengan training set, yang mana training set tersebut telah mengandung label dari masing-masing data. Tahap kedua *testing step* yaitu classifier yang telah terbentuk pada learning step tadi digunakan untuk melakukan klasifikasi atau prediksi terhadap dataset baru yang tidak diketahui nilai dari variabel dependen atau labelnya.

3.2. Supervised dan Unsupervised Learning. *Supervised learning* adalah pembelajaran terarah/terawasi. Pada *supervised learning* ada guru yang membuat kunci jawaban, sedangkan pada *unsupervised learning* tidak ada guru yang mengajar.

3.3. Decision Tree. Decision Tree adalah suatu model klasifikasi dengan menggunakan suatu struktur berbentuk pohon untuk mengambil keputusan. Dalam pembentukan pohon keputusan, diperlukan suatu ukuran yang disebut ukuran pemilihan atribut (*attribute selection measure*). Ukuran yang digunakan untuk pemilihan atribut tersebut yaitu information gain.

4. METODE

4.1. Analisis Regresi Logistik. Analisis regresi logistik adalah analisis regresi di mana variabel respon hanya memiliki dua kemungkinan nilai (*dikotomous*), misalkan penelitian tentang peluang seorang ibu melahirkan bayi perempuan ada dua kemungkinan "Ya" jika ibu melahirkan bayi perempuan dan "Tidak" jika ibu melahirkan bayi laki-laki. Model regresi logistik adalah sebagai berikut:

$$y_i = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + \epsilon \quad (4.1)$$

$$y = \begin{cases} 1, & \text{untuk menyatakan sukses} \\ 0, & \text{untuk menyatakan gagal} \end{cases}$$

4.2. Metode Klasifikasi Bayes. Metode klasifikasi Bayes memprediksi probabilitas keanggotaan kelas seperti probabilitas yang dimiliki tuple tertentu ke kelas tertentu.

4.2.1. Algoritma Klasifikasi Bayes.

- (1) Misalkan D menjadi set pelatihan tuple dan label kelas yang terkait. Masing-masing tuple diwakili oleh vektor atribut n-dimensi, $X = (x_1, x_2, \dots, x_n)$, menggambarkan n pengukuran dilakukan pada tupel dari n atribut, masing-masing A_1, A_2, \dots, A_n .
- (2) Misalkan ada m kelas, C_1, C_2, \dots, C_m . Jika diberikan tuple X , maka metode klasifikasi akan memprediksi bahwa X termasuk dalam kelas yang memiliki probabilitas posterior tertinggi dengan bersyarat X . Artinya, Naive Bayesian

classifier memprediksi bahwa tuple X termasuk kelas C_i jika dan hanya jika

$$P(C_i|X) > P(c_j|X)$$

untuk $1 \leq j \leq m, j = i$

maka, dapat dimaksimalkan $P(C_i|X)$, Kelas C_i dimana $P(C_i|X)$ dimaksimalkan disebut *maximum posteriori hypothesis*. Berdasarkan teorema Bayes diperoleh:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (4.2)$$

- (3) Karena $P(X)$ bernilai konstan untuk semua kelas, maka hanya $P(X|C_i)P(C_i)$ yang perlu dimaksimalkan. Jika probabilitas prior setiap tidak diketahui, umumnya akan diasumsikan bahwa kelas-kelas tersebut memiliki peluang yang sama, atau $P(C_1) = P(C_2) = \dots = P(C_m)$, maka akan dimaksimalkan $P(X|C_i)$. Jika tidak, kami memaksimalkan $P(X|C_i)$. Namun jika $P(C_i)$ diketahui, maka akan dimaksimalkan $P(X|C_i)$. Probabilitas prior setiap kelas juga dapat diestimasi dengan persamaan berikut

$$P(C_i) = |C_i.D|/|D|,$$

dimana $|C_i.D|$ adalah jumlah *tupel training* yang termasuk kelas C_i dalam D.

- (4) Diasumsikan bahwa nilai-nilai dari setiap atribut saling independen satu dengan lainnya atau disebut dengan asumsi *class-conditional independence*. Dengan begitu

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i) \end{aligned} \quad (4.3)$$

Nilai-nilai $P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i)$ dapat diestimasi dari *tupel training*. Perhatikan masing-masing nilai tersebut berbeda tergantung pada tipe atribut masing-masing:

- (a) Jika A_k berskala kategorik, maka $P(x_k|C_i)$ adalah jumlah tupel kelas C_i di D yang memiliki nilai x_k untuk A_k , dibagi dengan $|C_i.D|$, jumlah tupel kelas C_i di D.
 - (b) Jika A_k berskala kontinu, dengan atribut bernilai kontinu biasanya diasumsikan memiliki distribusi Gaussian dengan mean μ dan standar deviasi σ .
- (5) Untuk memprediksi label kelas X, perlu dihitung nilai $P(X|C_i)P(C_i)$ untuk setiap kelas C_i . Metode klasifikasi ini akan memprediksi label kelas dari X adalah C_i jika dan hanya jika

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j)$$

untuk $1 \leq j \leq m, j = i$

Dengan kata lain, label kelas yang diprediksi adalah kelas C_i yang memiliki nilai $P(X|C_i)P(C_i)$ maksimum.

4.3. Random Forest. Random forest adalah suatu metode klasifikasi yang dihasilkan dengan cara menggabungkan sejumlah pohon keputusan. Dalam random forest, dibentuk pohon keputusan secara rekursif dengan cara memilih sebanyak m variabel secara acak dari data, kemudian dipilih variabel atau split-point terbaik, kemudian dilakukan split berdasarkan variabel tersebut.

4.3.1. Algoritma Random Forest. Langkah-langkah dalam klasifikasi dengan metode random forest adalah sebagai berikut.

- (1) Menentukan parameter yang digunakan. Pada random forest, terdapat dua parameter utama yaitu n yang menunjukkan banyaknya pohon keputusan yang digunakan, serta m yang menunjukkan banyaknya *feature* yang dipertimbangkan dalam setiap split. Nilai n *default* yang digunakan adalah 100, sedangkan nilai m *default* yang digunakan adalah akar kuadrat dari banyaknya *feature* yang ada.
- (2) Membentuk pohon keputusan (*decision tree*). Setiap pohon keputusan dibentuk dengan menggunakan sebanyak m variabel yang dipilih secara acak. Dari m variabel yang dipilih acak tersebut akan dilakukan proses pemilihan *split-point* terbaik berdasarkan nilai *Gain*. Nilai *Gain* tersebut dapat dihitung dengan persamaan berikut

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i)$$

Dengan perhitungan nilai Entropy yang dapat dilihat pada rumus sebagai berikut :

$$Entropy(S) = \sum_{i=1}^n -p_i \times \log_2 p_i$$

dengan S adalah himpunan kasus, A adalah atribut, n adalah jumlah partisi atribut A , $|S_i|$ adalah jumlah kasus pada partisi ke- i , $|S|$ adalah jumlah kasus dalam $|S|$, dan p_i adalah proporsi dari $|S_i|$ terhadap $|S|$.

- (3) Melakukan perulangan terhadap langkah 2 sebanyak n kali, sehingga terbentuk sebanyak n pohon keputusan.
- (4) Melakukan klasifikasi terhadap data testing menggunakan setiap pohon keputusan yang terbentuk. Hasil klasifikasi adalah kelas yang paling banyak dihasilkan oleh pohon-pohon keputusan tersebut.

5. Syntax Program Python

```
1 # -*- coding: utf-8 -*-
2 """income.ipynb
3
4 Automatically generated by Colaboratory.
5
6 Original file is located at
7     https://colab.research.google.com/drive/1
8     VA6y5dKiNiXLi8EBDQ-MF6eOV4ZiPfnXZ
9
10 import numpy as np
11 import pandas as pd
12 import os
13 from scipy import stats
14 from sklearn.preprocessing import MinMaxScaler
15 from sklearn.preprocessing import LabelEncoder
16
17 from google.colab import files
18
19 uploaded = files.upload()
20
21 heart = pd.read_csv("/content/income_evaluation.csv",
22                     delimiter=";")
23 heart.head(10)
```

```
23 heart
24
25 heart.describe()
26
27 heart.columns = ['age', 'workclass', 'fnlwgt', 'education',
28     'education-num',
29     'marital-status', 'occupation', 'relationship',
30     'race', 'sex',
31     'capital-gain', 'capital-loss', 'hours-per-week',
32     'native-country',
33     'income']
34 X_data = heart[['age","workclass","fnlwgt","education",
35     "education-num","marital-status","occupation",
36     "relationship","race","sex","capital-gain","capital-
37     loss","hours-per-week","native-country"]]
38 X_data.head()
39
40 sc= MinMaxScaler()
41 datasetNum = heart.iloc[:,[0,2,4,10,11,12]]
42 datasetNum = sc.fit_transform(datasetNum)
43 datasetNum = pd.DataFrame(datasetNum, columns = ["A1","A2",
44     "A3","A4","A5","A6"])
45
46 datasetCat = heart.iloc[:,[1,3,5,6,7,8,9,13]]
47 datasetCat = np.array(datasetCat)
48 label_encoder = LabelEncoder()
49 datasetCat[:, 0] = label_encoder.fit_transform(datasetCat
50     [:, 0])
51 datasetCat[:, 1] = label_encoder.fit_transform(datasetCat
52     [:, 1])
53 datasetCat[:, 2] = label_encoder.fit_transform(datasetCat
54     [:, 2])
55 datasetCat[:, 3] = label_encoder.fit_transform(datasetCat
56     [:, 3])
57 datasetCat[:, 4] = label_encoder.fit_transform(datasetCat
58     [:, 4])
```

```
 4 datasetCat[:, 5] = label_encoder.fit_transform(datasetCat
  5     [:, 5])
 6 datasetCat[:, 6] = label_encoder.fit_transform(datasetCat
  7     [:, 6])
 8 datasetCat[:, 7] = label_encoder.fit_transform(datasetCat
  9     [:, 7])
10 datasetCat = pd.DataFrame(datasetCat, columns = ["A7","A8",
11     "A9","A10","A11","A12","A13","A14"])
12
13 Y = heart[["income"]]
14 Y.head(10)
15
16 np.min(Y)
17
18 X = pd.concat([datasetNum, datasetCat], axis = 1)
19
20 y = label_encoder.fit_transform(heart.iloc[:, 14])
21
22 """Split Dataset"""
23
24 from sklearn.model_selection import train_test_split
25 X_train, X_test, y_train, y_test = train_test_split(X,Y,
26     test_size=0.20, random_state=42)
27 y_test
28
29 """Logistic Regression"""
30
31 from sklearn.linear_model import LogisticRegression
32
33 lr = LogisticRegression()
34 lr.fit(X_train,y_train)
35
36 y_predict_lr = lr.predict(X_test)
37
38 from sklearn.metrics import confusion_matrix
39 cm = confusion_matrix(y_test, y_predict_lr)
```

```
78 cm
79
80 from sklearn.metrics import classification_report
81
82 print(classification_report(y_test, y_predict_lr))
83
84 """Random Forest Classifier"""
85
86 from sklearn.ensemble import RandomForestClassifier
87
88 rfc = RandomForestClassifier(random_state=0)
89 # fit the model
90 rfc.fit(X_train, y_train)
91
92 y_predict_rfc = rfc.predict(X_test)
93 y_predict_rfc
94
95 from sklearn.metrics import confusion_matrix
96 cm = confusion_matrix(y_test, y_predict_rfc)
97 cm
98
99 from sklearn.metrics import classification_report
100
101 print(classification_report(y_test, y_predict_rfc))
102
103 """Naive Bayes"""
104
105 from sklearn.naive_bayes import GaussianNB
106 modelnb = GaussianNB()
107 nbtrain = modelnb.fit(X_train, y_train)
108
109 y_pred = nbtrain.predict(X_test)
110 y_pred
111
112 from sklearn.metrics import confusion_matrix
113 confusion_matrix(y_test, y_pred)
```

```

114
115 from sklearn.metrics import classification_report
116 print(classification_report(y_test, y_pred))

```

6. Studi Kasus

Dataset yang digunakan pada studi kasus ini berjudul *income evaluation* yang diperoleh dari *Website kaggle*. Data yang digunakan merupakan data sekunder. *Income evaluation dataset* berukuran 32561 dengan 15 variabel, yang terdiri dari *income* sebagai variabel respon dan 14 variabel lainnya sebagai variabel prediktor . Langkah pertama yang dilakukan adalah mengimport package data yang akan digunakan(*numpy*, *pandas*, *os*, *scipy*, *sklearn.preprocessing*, *sklearn.tree*, *subprocess*, dan *Ipython.display*). Setelah mengimport packagge yang akan digunakan, selanjutnya mengimport file excel dalam format *csv*.

Karena rentang antar variabel data besar maka perlu dilakukan normalisasi menggunakan in-Max normalization. Setelah dilakukan normalisasi, dataset kemudian dibagi menjadi 2 yaitu data training dan data testing. Proporsi yang digunakan yaitu 80% data training dan 20% data testing.

6.1. Klasifikasi dengan Metode Regresi Logistik. Penerapan metode regresi logistik ini menggunakan modul *LogisticRegression*. Data train yang digunakan memiliki nilai akurasi sebesar 0.88. Setelah menerapkan *package* tersebut pada setiap baris data testing diperoleh *confusion matrix* yang menggambarkan ketepatan klasifikasi yaitu:

	Disease	No Disease
Disease	4583	359
No Disease	893	678

TABEL 1. *confusion matrix*

$$FPR = 0.07$$

$$Akurasi = 0.80$$

(6.1)

6.2. Klasifikasi dengan Metode Naive Bayes. Penerapan metode **Naive Bayes** ini menggunakan modul *GaussianNB*. Data train yang digunakan memiliki nilai akurasi sebesar 0.80. Setelah menerapkan *package* tersebut pada setiap baris data testing diperoleh *confusion matrix* yang menggambarkan ketepatan klasifikasi yaitu:

	Disease	No Disease
Disease	4708	234
No Disease	1008	563

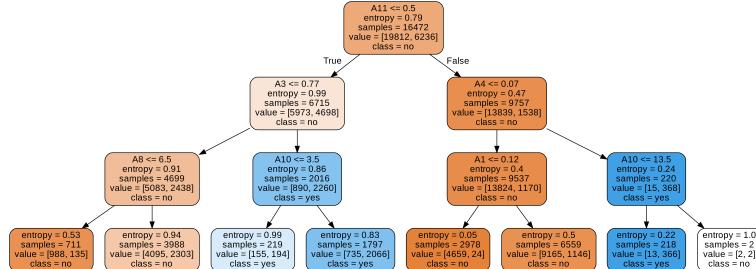
TABEL 2. *confusion matrix*

$$FPR = 0.04$$

$$Akurasi = 0.81$$

(6.2)

6.3. Klasifikasi dengan Metode Random Forest. Penerapan metode **Random Forest** ini menggunakan modul *RandomForestClassifier*. Metode Random Forest menggunakan gabungan dari sejumlah Decision Tree untuk menghasilkan suatu prediksi. Dalam implementasi Random Forest pada kasus ini digunakan modul *RandomForestClassifier* pada *package sklearn*. Ditentukan parameter banyaknya *Decision Tree* yang digunakan (*n*) adalah 100, serta banyaknya *feature* yang dipertimbangkan untuk setiap split (*m*) adalah $\sqrt{14} = 3.74 \approx 4$. Sebagai ilustrasi, ditampilkan salah satu Decision Tree yang dihasilkan, sebagai berikut. Untuk kemudahan ilustrasi, digunakan parameter *max-depth* = 3.

FIGURE 1. Ilustrasi *Decision Tree* pada *Income evaluation dataset*Tabel *Confusion Matrix* untuk *income evaluation dataset*

	Disease	No Disease
Disease	4610	332
No Disease	553	1018

TABEL 3. *confusion matrix*

$$FPR = 0.06$$

$$Akurasi = 0.86$$

(6.3)

Data ini juga diujicobakan pada n estimator yang berbeda yaitu 10, 20 dan 50. Berikut grafik n estimator dan nilai akurasinya:

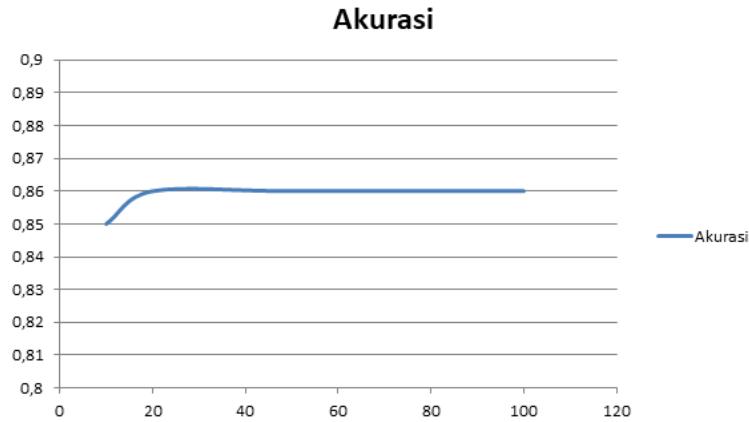


FIGURE 2. grafik untuk *income evaluation dataset*

Dari grafik diatas dapat dilihat bahwa pada saat $n = 10$ nilai akurasinya 0.85, $n = 20$ nilai akurasinya 0.86, dan pada saat $n = 50$ nilai akurasinya 0.86.

7. PENUTUP

Berdasarkan studi kasus yanng dilakukan diperoleh:

	Income Evaluation	
	Akurasi	FPR
Regggresi Logistik	0.80	0.07
Naïve Bayes	0.81	0.04
Random Forest	0.86	0.06

TABEL 4. tabel perbandingan tiga metode

Dari tabel di atas dapat disimpulkan metode random forest lebih baik dibandingkan dua metode lainnya dilihat dari nilai akurasinya.

Referensi

- [1] Aisah, Lely Nur.(2019). *Implementasi Naive Bayes dan Random Forest untuk Analisis Sentimen terhadap Data Imbalanced Review Produk Kosmetik pada Platform Online Sociolla*, Skripsi, Program Studi S11 Statistika FMIPA UGM, Yogyakarta.
- [2] Berry, Michael W. and Murray Browne. (2006). *Lecture Notes in DATA MINING*. World Scientific, USA.
- [3] Breiman, Leo. (2001). *Machine Learning*. Berkeley : University of California.
- [4] Dangeli, Pratap.(2017). *Statistics for Machine Learning*. Packt Publishing Ltd, UK.
- [5] Gladence, L. Mary, Karthi, M., Anu, V. Maria. (2015). *A statistical comparison of logistic regression and different bayes classification methods for machine learning*.ARPN Journal of Engineering and Applied Sciences: Vol. 10, No. 14,(AUGUST 2015).
- [6] Han, Jiawei and Kamber, Micheline. (2012). *Data Mining concepts and Techniques, Third Edition*. Elsevier Inc. All rights reserved, USA.
- [7] Hasti, Trevor, Tibshirani, Robert, Friedman, Jerome. (2009). *The Elements of Statistical Learning, Second Edition*. Springer, California.
- [8] Hosmer, David W.(1989). *Applied Logistic Regression*. John Wiley & Sons, Inc., Canada.
- [9] Hosmer, David W.(2013). *Applied Logistic Regression, Third Edition*. John Wiley & Sons, Inc., Canada.
- [10] James, Gareth, et al. (2017). *An Introduction to Statistical Learning, Eighth Edition*. Springer, New York Heidelberg Dordrecht London.
- [11] Setyawan, M.Y.H., Awangga, R.M., Efendi, S.R. (2018). *Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot*, Proceedings of the 2018 International Conference on Applied Engineering, ICAE 2018, art. no.8579372, . Cited 4 times.
- [12] Telnoni, P.A., Budiantan, R., Qana'a, M.(2019). *Comparison of Machine Learning Classification Method on Text-based Case in Twitter*, Proceeding - 2019 International Conference on ICT for Smart Society: Innovation and Transformation Toward Smart Region, ICISS 2019, art. no. 8969850,.
- [13] Tsangaratos, P., Ilia, I. (2016). *Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size*, Catena, 145, pp. 164-179. Cited 125 times.

SRI WAHYUNI KALUMBANG* (Penulis Korespondensi)

Departemen Matematika, Fakultas MIPA, Universitas Gadjah Mada, Indonesia
srikalumbang31@gmail.com

SUBANAR

Departemen Matematika, Fakultas MIPA, Universitas Gadjah Mada, Indonesia
subanar@ugm.ac.id