

# ANALISIS PERFORMA ALGORITME WEIGHTED NAIVE BAYES CLASSIFIER

Burhan Alfironi Muktamar

Program Studi Teknik Informatika

STMIK Jenderal Achmad Yani

[burhanalfironimuktamar@gmail.com](mailto:burhanalfironimuktamar@gmail.com)

## Abstrak

*Naïve Bayes Classifier merupakan salah satu algoritme klasifikasi dalam data mining. Naïve Bayes Classifier mengadopsi teorema Bayesian untuk memetakan suatu data terhadap kelas dengan memperhitungkan probability dari atribut data tersebut. Dalam beberapa tahun terakhir, banyak penelitian dilakukan untuk mengatasi permasalahan Naïve Bayes Classifier yang hanya bergantung pada distribusi probabilitas. Beberapa algoritme yang telah diusulkan para peneliti terdahulu adalah : Naïve Bayes Classifier with Hybrid-weight (NBCH), Weighted Naïve Bayes on Correlation Coefficient (WNB-CC) dan Correlated Naïve Bayes Classifier (CNBC). Hasil dari penelitian ini adalah informasi perbandingan performa algoritme-algoritme yang termasuk dalam Weighted Naïve Bayes Classifier. Performa masing-masing algoritme dinilai dari tingkat akurasi dan kompleksitas proses. Setelah dilakukan pengujian terhadap 30 data set, dapat diketahui bahwa algoritme CNBC menunjukkan performa yang lebih baik yaitu dengan tingkat akurasi 66,36% dan kompleksitas proses  $O(n^2)$  dibandingkan dengan algoritme NBCH yang memiliki tingkat akurasi 58,9% dan kompleksitas proses  $O(n^3)$  serta dibandingkan dengan algoritme WNB-CC yang memiliki tingkat akurasi 32,92% dan kompleksitas proses  $O(n^2)$ .*

**Kata Kunci:** *Data Mining, Klasifikasi, Naïve Bayes Classifier, Weighted Naive Bayes Classifier.*

## 1. Pendahuluan

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Berhubungan dengan pemanfaatan teknologi *data mining* dalam berbagai bidang, klasifikasi banyak digunakan dalam pengembangan sistem pendukung pengambilan keputusan, sistem prediksi, sistem diagnosis dan lain-lain. Sehubungan dengan besarnya kebutuhan terhadap sistem klasifikasi yang efektif dan efisien, maka klasifikasi yang baik dituntut untuk dapat memberikan hasil klasifikasi yang akurat, pemrosesan yang cepat dan mudah untuk diimplementasikan atau memiliki kompleksitas yang rendah. Dalam pengimplementasian klasifikasi terdapat beberapa algoritme yang dapat digunakan, seperti : *Decision Tree, Naïve Bayes Classifier, Neural Network, Genetic Algorithms, Support Vector Machines* dan lain sebagainya.

Salah satu algoritme yang dapat digunakan dalam proses klasifikasi adalah *Naïve Bayes Classifier* (NBC). *Naïve Bayes Classifier* merupakan

algoritme klasifikasi yang berbasis pada teori keputusan *Bayes* yang menggunakan probabilitas untuk penentuan keputusan. *Naïve Bayes Classifier* merupakan algoritme klasifikasi yang sederhana, efektif, cepat, memiliki tingkat akurasi yang tinggi dan algoritme yang paling banyak digunakan dalam proses klasifikasi dibanding algoritme klasifikasi lainnya. Meskipun *Naïve Bayes Classifier* merupakan algoritme yang paling banyak digunakan, *Naïve Bayes Classifier* masih memiliki kelemahan yang harus diatasi guna meningkatkan performa dari algoritme tersebut.

Kelemahan *Naïve Bayes Classifier* dijelaskan pada penelitian yang dilakukan oleh Nurnberger, dkk (1999) yang menyebutkan bahwa dalam proses klasifikasi, algoritme *Naïve Bayes Classifier* hanya berdasar pada distribusi probabilitas (*prior probability* dan *conditional probability*). Jika hanya menggunakan distribusi probabilitas maka yang menjadi faktor penentu pemetaan suatu kelas hanya berdasar kepada tingkat kemunculan data pada *data set* dengan asumsi bahwa bobot atau prioritas masing-masing atribut adalah sama. Pada saat dilakukan perhitungan *probability* masing-masing atribut terhadap kelas dengan metode *Naive Bayes Classifier*, idealnya juga diperhitungkan seberapa kuat hubungan atribut tersebut dengan kelas. Perhitungan nilai hubungan atribut dengan kelas menyebabkan yang menjadi parameter untuk pemetaan data yang belum diketahui kelasnya terhadap kelas yang ditentukan bukan hanya dari *probability* atribut data tersebut, tetapi juga seberapa kuat hubungan atribut dari data tersebut dengan kelas yang ditentukan. Dalam beberapa tahun terakhir, banyak penelitian dilakukan untuk mengatasi permasalahan *Naïve Bayes Classifier* yang hanya bergantung pada distribusi probabilitas. Pengembangan yang dilakukan untuk mengatasi masalah tersebut adalah dengan melakukan pembobotan atribut pada *Naïve Bayes Classifier*. Beberapa algoritme yang telah diusulkan para peneliti terdahulu adalah : *Naïve Bayes Classifier with Hybrid-weight* (NBCH), *Weighted Naïve Bayes on Correlation Coefficient* (WNB-CC) dan *Correlated Naïve Bayes Classifier* (CNBC).

Algoritme *Naïve Bayes Classifier with Hybrid-weight* (NBCH) diusulkan oleh Geng, dkk (2011) yang menerapkan metode penggabungan *gain ratio* dan koefisien korelasi. *Gain ratio* digunakan untuk mengukur keefektifan kondisi atribut dalam klasifikasi dan koefisien korelasi digunakan untuk menggambarkan hubungan ketergantungan antara atribut dengan kelas. Algoritme ini memberikan

tingkat akurasi yang cukup tinggi, akan tetapi kelemahan dari algoritme ini adalah memiliki kompleksitas yang tinggi dan hampir mendekati kompleksitas algoritme *Decision Tree*. Dengan melakukan pembobotan berdasarkan *information gain*, *entropy* dan koefisien korelasi, proses komputasi dalam klasifikasi menjadi sangat tinggi. Hal ini sangat berpengaruh terhadap kecepatan proses klasifikasi.

Algoritme yang cukup efektif dan efisien serta memiliki akurasi cukup tinggi diusulkan oleh Yao dan Li (2012) yaitu *Weighted Naïve Bayes on Correlation Coefficient* (WNB-CC). Konsep dari pembobotan atribut dengan koefisien korelasi ini adalah dengan memberikan bobot atribut berdasarkan pada nilai koefisien korelasi antara atribut tersebut dengan atribut kelas. Pembobotan atribut dibobotkan pada *conditional probability*, hal ini membuat algoritme ini memiliki kelemahan karena nilai koefisien korelasi berkisar antara -1 sampai dengan 1 maka memungkinkan terjadi *probability* negatif. Kelemahan lain dari algoritme ini adalah pembobotan dengan metode ini tidak selalu akurat untuk semua kondisi misalnya jika frekuensi kemunculan data suatu atribut berbanding terbalik dengan koefisien korelasinya serta jika terjadi *probability* 0 atau *zero probability* pada *conditional probability*, maka pembobotan atribut menjadi tidak efektif lagi.

Algoritme pengembangan lain yang diusulkan adalah *Correlated Naïve Bayes Classifier* (CNBC). Algoritme ini diusulkan oleh Mukhtar, dkk (2015) yang menggunakan teknik pembobotan atribut dengan menggunakan *R Square*. Dalam metode ini menggunakan teknik *joint probability* dalam perhitungan *conditional probability* serta menerapkan metode laplacian untuk mencegah terjadinya *zero probability*.

Dengan adanya beberapa algoritme pembobotan yang merupakan pengembangan dari algoritme *Naïve Bayes Classifier*, ketepatan dalam pemilihan algoritme yang akan digunakan dalam suatu proses klasifikasi menjadi sangat penting. Berhubungan dengan hal tersebut dibutuhkan informasi performa klasifikasi dari masing-masing algoritme *Weighted Naïve Bayes Classifier* untuk berbagai macam tipe data. Hal utama yang perlu diukur untuk menunjukkan performa algoritme tersebut adalah kompleksitas proses dan tingkat akurasi. Untuk menjamin hasil yang diperoleh, maka proses pengukuran performa dilakukan terhadap beberapa data set yang berbeda.

## 2. Metode Penelitian

### 2.1 Jalannya Penelitian

Pada penelitian ini terdapat tiga tahapan, yaitu : pengumpulan data dan *preprocessing*, implementasi dan pengujian. Tahapan pertama adalah pengumpulan *data set* dan *preprocessing* yang akan digunakan sebagai bahan penelitian untuk membandingkan performa algoritme *Weighted Naïve Bayes Classifier*. *Data set* diambil dari *Uci Learning Repository* sebanyak 30 *data set* dengan detail data tercantum pada Tabel 1.

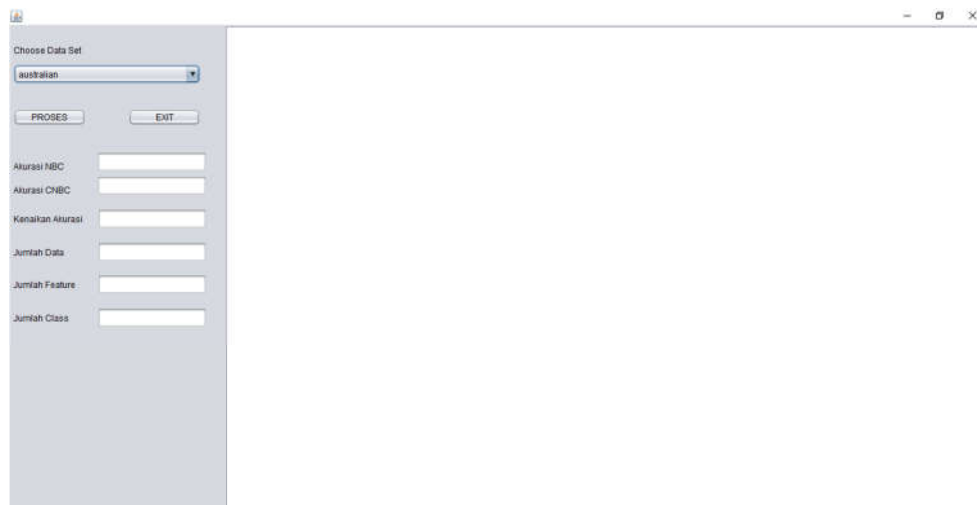
**Tabel 1.** Detail data set

No	Data Set	Type Data	Type Atribut	Data	Atribut
1	Australian	Multivariate	Categorical, Integer	690	14
2	Autompg	Multivariate	Categorical, Real	398	8
3	Bridges	Multivariate	Categorical, Integer	108	13
4	Coil	Multivariate	Categorical, Real	340	17
5	Diabetes	Multivariate	Integer, Real	768	8
6	Ecoli	Multivariate	Real	336	8
7	Fertility	Multivariate	Real	100	10
8	Flag	Multivariate	Categorical, Integer	194	30
9	Haberman	Multivariate	Integer	306	3
10	Heart-c	Multivariate	Categorical, Integer, Real	303	75
11	Heart-h	Multivariate	Categorical, Integer, Real	303	75
12	Hydrodynamics	Multivariate	Real	308	7
13	Iris	Multivariate	Real	150	4
14	Liver	Multivariate	Integer, Real	583	10
15	Lung Cancer	Multivariate	Integer	32	56
16	User Knowledge Modeling	Multivariate	Integer	403	5
17	Monk	Multivariate	Categorical	432	7
18	Monk-1	Multivariate	Categorical	432	7
19	Monk-2	Multivariate	Categorical	432	7
20	Monk-3	Multivariate	Categorical	432	7

Tabel 2. Detail *data set* (lanjutan)

No	Data Set	Tipe Data	Tipe Atribut	Data	Atribut
21	Pop Failures	Multivariate	Integer, Real	540	19
22	Postoperative	Multivariate	Integer, Real	90	8
23	Servo	Multivariate	Categorical, Integer	167	4
24	Spectrometer	Multivariate	Integer, Real	531	102
25	Thoracic	Multivariate	Real	132	5
26	Transfusion	Multivariate	Real	748	5
27	Vehicle-a	Multivariate	Integer	946	18
28	Vehicle-b	Multivariate	Integer	946	18
29	Vehicle-c	Multivariate	Integer	946	18
30	Voting	Multivariate	Categorical	435	16

Tahap kedua adalah tahap metode implementasi yang digunakan untuk menguji performa algoritme-algoritme yang akan dibandingkan. Pada tahapan ini dirancang dan dibangun sebuah aplikasi berbasis *Java Desktop*. Aplikasi ini digunakan untuk membantu dalam pengujian algoritme-algoritme yang akan dibandingkan. Aplikasi ini menggunakan *data set* yang telah dikumpulkan kemudian menggunakan rumus perhitungan masing-masing algoritme untuk memberikan informasi berupa *output* tingkat akurasi masing-masing algoritme.



Gambar 1. Antarmuka aplikasi pengujian

Tahap ketiga adalah tahap pengujian yang digunakan untuk mengetahui hasil tingkat akurasi dari masing-masing algoritme yang dibandingkan. Pengujian

akan difokuskan terhadap dua hal yaitu tingkat akurasi dan kompleksitas proses. Pengujian untuk mendapatkan tingkat akurasi dilakukan dengan metode *10-fold cross validation* menggunakan sistem yang telah dikembangkan terhadap 30 *data set* pada Tabel 1 dan Tabel 2. Pengujian untuk mendapatkan tingkat kompleksitas proses dilakukan dengan analisis *running time* algoritme yang akan dibandingkan untuk mendapatkan waktu proses *Big Oh (O)* berdasarkan pada kompleksitas setiap langkah  $T(n)$ .

## 2.2 Cara Analisis

Analisis dalam penelitian ini digunakan untuk melihat perbandingan tingkat performa klasifikasi antara algoritme WNB-CC, NBCH dan CNBC. Tahap analisis dalam penelitian ini meliputi dua langkah yaitu analisis uji normalitas dan analisis uji hipotesis.

### 2.2.1 One-Sample Kolmogorov-Smirnov Test

Uji *One Sample kolmogorov-Smirnov* digunakan untuk mengetahui apakah distribusi nilai-nilai sampel yang teramati terdistribusi normal. Uji *Kolmogorov-Smirnov* beranggapan bahwa distribusi variabel yang sedang diuji bersifat *continue* dan pengambilan sampel dilakukan secara acak sederhana. Perumusan hipotesis uji *One Sample Kolmogorov-Smirnov* sebagai berikut :

$$H_0 : F(x) = G(x)$$

$$H_1 : F(x) \neq G(x)$$

Prinsip kerja dari uji *Kolmogorov-Smirnov* adalah menghitung selisih absolut antara distribusi frekuensi kumulatif sampel  $F(x)$  dan distribusi frekuensi kumulatif teoritis  $G(x)$  pada masing-masing interval kelas. Uji *One Sample Kolmogorov-Smirnov* merupakan selisih absolut terbesar antara  $F(x)$  dan  $G(x)$  yang disebut dengan deviasi maksimum. Persamaan uji *One Sample Kolmogorov-Smirnov* adalah sebagai berikut:

$$T_{KS} = \max_i \{|F(x_i) - G(x_i)|\}$$

Dengan :

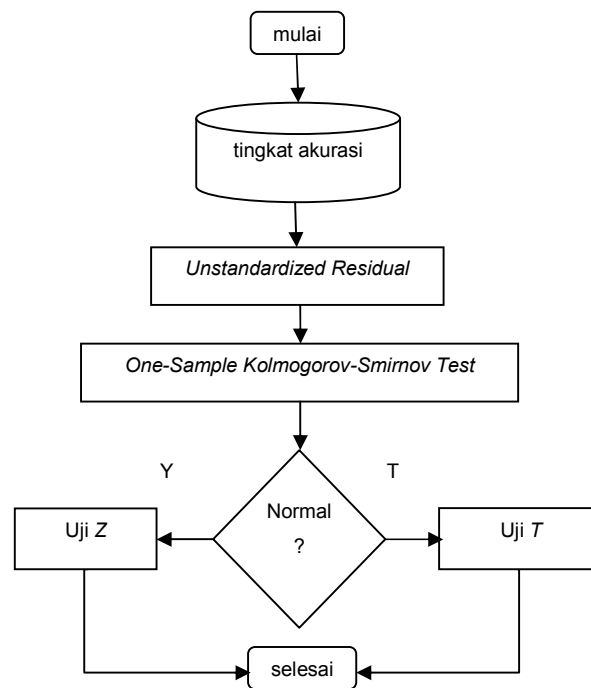
$T_{KS}$  = deviasi maksimum *Kolmogorov-Smirnov*

$F(x_i)$  = fungsi distribusi frekuensi kumulatif sampel data  $x_i$

$G(x_i)$  = fungsi distribusi frekuensi kumulatif teoritis data  $x_i$

$T_{KS}$  kemudian dibandingkan dengan nilai kritis untuk uji *One Sample Kolmogorov-Smirnov* yang diperoleh dari tabel *One Sample Kolmogorov-Smirnov*  $D_\alpha$ . Jika nilai  $T_{KS} \geq D_\alpha$ , maka  $H_0$  ditolak dan berarti data tidak terdistribusi normal.

Jika nilai  $T_{KS} < D_{\alpha}$ , maka  $H_0$  diterima dan berarti data terdistribusi normal (Marsalek & Povalac, 2012).



Gambar 2. Alur analisis

### 2.2.2 Uji Hipotesis

Uji hipotesis merupakan suatu metode pengambilan keputusan yang berdasar pada hasil analisis data. Dalam statistik sebuah hasil bisa dikatakan signifikan secara statistik jika kejadian tersebut hampir tidak mungkin disebabkan oleh faktor yang kebetulan, sesuai dengan batas probabilitas yang sudah ditentukan sebelumnya. Analisis uji hipotesis merupakan pengujian yang dilakukan untuk menentukan significant atau tidaknya suatu hasil perbandingan. Uji hipotesis dilakukan setelah uji normalitas untuk menentukan jenis pengujian yang tepat. Jika hasil uji normalitas menunjukkan bahwa data terdistribusi normal maka uji hipotesis dilakukan dengan menggunakan uji z. Jika hasil uji normalitas menunjukkan bahwa data tidak terdistribusi normal maka uji hipotesis dilakukan dengan menggunakan uji  $T$ . Dalam sebuah pengujian hipotesis perlu dicegah terjadinya dua jenis kesalahan, yaitu (Harinaldi, 2005):

1. Kesalahan tipe I (*type I error*) atau biasa disebut *alpha risk* adalah kesalahan yang terjadi apabila menolak suatu hipotesis yang seharusnya diterima.
2. Kesalahan tipe II (*type II error*) atau biasa disebut *beta risk* adalah

kesalahan yang terjadi apabila menerima suatu hipotesis yang seharusnya ditolak.

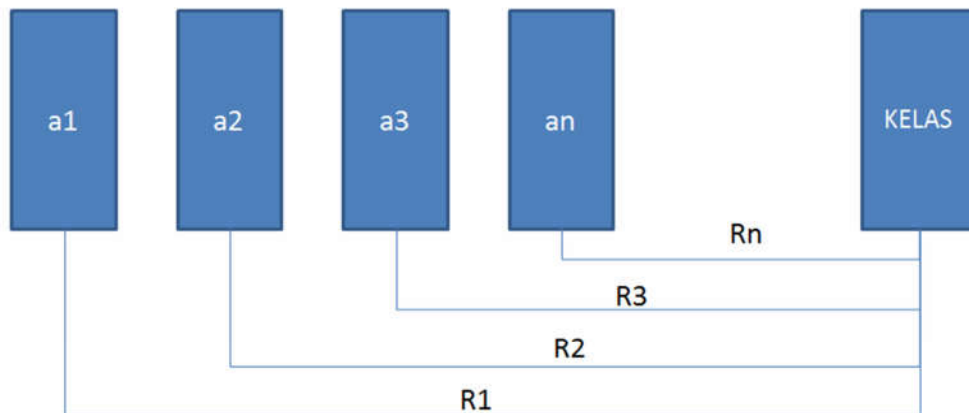
Untuk mencegah terjadinya kesalahan dalam pengujian hipotesis diperlukan prosedur umum sebagai berikut:

1. Penentuan hipotesis nol ( $H_0$ ) dan hipotesis alternatif ( $H_1$ )
2. Pemilihan tingkat kepentingan atau *level of significant* ( $\alpha$ )
3. Penentuan distribusi pengujian yang digunakan
4. Pendefinisian daerah-daerah penolakan (kritis)
5. Pernyataan aturan keputusan (*decision rule*)
6. Perhitungan pada data sampel dan perhitungan rasio uji
7. Pengambilan keputusan secara statistik

### 3. Hasil dan Pembahasan

#### 3.1 Correlated Naïve Bayes Classifier (CNBC)

Algoritme *Correlated Naïve Bayes Classifier* menggunakan metode pembobotan atribut dengan menggunakan *R Square*. Nilai *R Square* merupakan sebuah bobot nilai yang menggambarkan tingkat pengaruh suatu atribut terhadap kelas. Jika suatu *data set* memiliki atribut  $a_1, a_2, a_3, \dots, a_n$  dan masing-masing atribut memiliki bobot  $R_1, R_2, R_3, \dots, R_n$ , maka pembobotan *R Square* pada algoritme *Correlated Naïve Bayes Classifier* ditunjukkan pada Gambar 3.



Gambar 3. Pembobotan *R Square* CNBC

Teknik perhitungan *conditional probability* pada algoritme *Naïve Bayes Classifier* menggunakan *joint probability* untuk menghindari kesalahan perhitungan karena sifat komutatif perkalian. Pada CNBC metode *Laplacian* diperlukan untuk menjamin bahwa setiap bobot akan diperhitungkan dengan



*probability* masing-masing atribut. Persamaan algoritme *Correlated Naïve Bayes Classifier* dapat dijelaskan sebagai berikut :

$$P_{CNBC}(Y|X) = \max \left( \frac{P(Y) \sum_{i=1}^q (P(X_i|Y)^\ell R(i|Y))}{P(X)} \right)$$

Dengan:

$P_{CNBC}(Y|X)$  = *posterior probability*

$P(Y)$  = *prior probability*

$P(X_i|Y)^\ell$  = *probability* masing-masing atribut dengan *Laplacian*

$R(i|Y)$  = bobot atribut dari nilai R Square

$P(X)$  = *probability vector* X tanpa bukti apapun

### 3.2 **Weighted Naïve Bayes on Correlation Coefficient (WNB-CC)**

*Weighted Naïve Bayes on Correlation Coefficient (WNB-CC)* merupakan algoritme pengembangan dari *Naïve Bayes Classifier* yang menggunakan pembobotan atribut menggunakan korelasi. Korelasi merupakan konsep statistik yang menggambarkan hubungan linear antara atribut dengan kelas. Dalam algoritma ini diasumsikan bahwa nilai korelasi selalu dinilaikan positif karena bobot berkisar antara 0 – 1. Persamaan algoritme WNB-CC adalah sebagai berikut :

$$P_{WNB-CC}(Y|X) = \max \left( \frac{P(Y) \sum_{i=1}^q (P(X_i|Y)^{W_i})}{P(X)} \right)$$

Dengan :

$P_{WNB-CC}(Y|X)$  = *posterior probability*

$P(Y)$  = *prior probability*

$P(X_i|Y)^{W_i}$  = *probability* masing-masing atribut dengan *bobot*

$W_i$  = bobot atribut dari koefisien korelasi

$P(X)$  = *probability vector* X tanpa bukti apapun

### 3.3 **Naïve Bayes Classifier with Hybrid-weight (NBCH)**

#### 3.3.1 **Gain Ratio Weight (WGR)**

Pembobotan atribut dengan gain ratio dapat ditunjukkan pada persamaan berikut :

$$W_{GR}(i) = \frac{GR(E, A_i) \times d}{\sum_{i=1}^D GR(E, A_i)}, (i = 1, 2, \dots, d)$$

Dimana  $d$  adalah nilai conditional atribut data set  $E$ .  $W_{GR}$  merupakan *gain ratio* bobot *conditional* atribut  $A_i$  dari data set  $E$ .  $GR(E, A_i)$  adalah *information gain ratio* dari *conditional* atribut  $A_i$  relatif terhadap data set  $E$ . Berdasarkan hal tersebut dapat disimpulkan sebuah metode untuk menghitung  $GR(E, A_i)$  seperti yang ditunjukkan pada persamaan berikut:

$$GR(E, A_i) = \frac{IG(E, A_i)}{SI(E, A_i)}$$

$$IG(E, A_i) = Entropy(E) - \sum_{k=1}^{M_i} \frac{|E_k|}{|E|} Entropy(E_k)$$

$$SI(E, A_i) = - \sum_{k=1}^{M_i} \frac{|E_k|}{|E|} \log_2 \left( \frac{|E_k|}{|E|} \right)$$

Dimana  $IG(E, A_i)$  adalah *information gain* dan  $SI(E, A_i)$  adalah *split information*.  $M_i$  adalah nilai dari *conditional* atribut  $A_i$ .  $E_k$  ( $k = 1, 2, \dots, M$ ) adalah nilai atribut ke- $k$  dari atribut  $A_i$  dan  $|S|$  merupakan ukuran dari data set  $S$ .

### 3.3.2 Correlation Coefficient Weight ( $W_{CC}$ )

Pembobotan atribut dengan koefisien korelasi dapat dihitung dengan persamaan berikut:

$$W_{CC}(i) = \frac{Cov(A_i, C)}{\sqrt{D(A_i) \times D(C)}}, (i = 1, 2, \dots, d)$$

Dimana  $Cov(A_i, C) = E\{[A_i - E(A_i)][C - E(C)]\}$  adalah *covariance* antara  $A_i$  dan  $C$ ,  $D(A_i) = E\{[A_i - E(A_i)]^2\}$  adalah *variance* dari  $A_i$  dan  $D(C) = E\{[C - E(C)]^2\}$  adalah *variance* dari  $C$ .

### 3.3.3 Hybrid Weight (NBC-NBCH)

*Hybrid Weight* merupakan gabungan antara *gain ratio* dan koefisien korelasi. Berdasarkan pembobotan *gain ratio* dan pembobotan koefisien korelasi yang telah dijelaskan, maka didapatkan persamaan untuk *Hybrid Weight* seperti yang ditunjukkan pada persamaan berikut:

$$W_H(i) = -[W_{GR}(i) \times \log_2 W_{GR}(i) + W_{CC}(i) \times \log_2 W_{CC}(i)]$$

## 3.4 Hasil

Setelah dilakukan pengujian perbandingan performa klasifikasi antara algoritme *Correlated-Naïve Bayes Classifier* (CNBC), *Weighted Naïve Bayes on Correlation Coefficient* (WNB-CC), dan *Naïve Bayes Classifier with Hybrid-weight*

(NBCH) yang dinilai dari tingkat akurasi dan waktu proses didapatkan hasil perbandingan seperti yang tercantum pada Tabel 3.1.

Tabel 3. Hasil perbandingan CNBC, WNB-CC dan NBCH

Algoritme	Akurasi (%)	T(n)	Big Oh
CNBC	66,36	$10n^2 + 46n + 2$	$O(n^2)$
WNB-CC	32,92	$10n^2 + 45n + 2$	$O(n^2)$
NBCH	58,9	$17n^3 + 52n^2 + 30n + 2$	$O(n^3)$

Dengan melihat hasil perbandingan pada Tabel 3.1, maka dari tingkat akurasi dan waktu proses algoritme *Correlated Naïve Bayes Classifier* (CNBC) merupakan algoritme yang paling efektif dan efisien. Akurasi dari pengujian terhadap 30 *data set* menunjukkan nilai sebesar 66,36% dengan kompleksitas waktu proses  $O(n^2)$ .

Hasil dari uji normalitas yang dilakukan ditemukan bahwa nilai deviasi maksimum *Kolmogorov-Smirnov* ( $T_{ks}$ ) adalah 0,154. Berdasarkan table *Kolmogorov-Smirnov* untuk  $\alpha=0,01$  dan  $n = 30$  didapatkan nilai  $k$  sebesar 0,29. Karena  $T_{ks} < k$ , maka dapat disimpulkan bahwa data persebaran tingkat akurasi terdistribusi normal sehingga uji hipotesis yang digunakan adalah uji  $z$ . Dari hasil uji  $z$  yang dilakukan didapatkan nilai  $RUz = 2,695$ . Karena  $2,33 < RUz$  maka  $H_0 : \mu_{cnbc} = \mu_{wnb}$  ditolak dan  $H_1 : \mu_{cnbc} > \mu_{wnb}$  diterima sehingga dapat disimpulkan bahwa algoritme *Correlated-Naïve Bayes Classifier* (CNBC) memiliki tingkat akurasi yang lebih tinggi secara signifikan.

#### 4. Penutup

Terdapat beberapa pengembangan algoritme *Naïve Bayes Classifier* dengan metode pembobotan atribut untuk meningkatkan hasil klasifikasi yang lebih baik. Untuk menilai performa dari suatu algoritme dapat dilihat dari tingkat akurasi dan kompleksitas prosesnya. Pada penelitian ini telah dilakukan perbandingan antara algoritme-algoritme yang termasuk dalam *Weighted Naïve Bayes Classifier* dengan hasil yang menunjukkan bahwa algoritme *Correlated Naïve Bayes Classifier* memberikan tingkat akurasi yang lebih tinggi dan kompleksitas proses yang lebih rendah dibandingkan algoritme pembobotan atribut lainnya secara signifikan. Penelitian selanjutnya diharapkan dapat membandingkan algoritme-algoritme yang menggunakan teknik pembobotan atribut dengan algoritme-algoritme pengembangan NBC dengan teknik yang lain.

## Daftar Pustaka

- Geng, C., Guan, H.-Y., & Liu, H.-T. (2011). Arranging a hybrid-weight for attribute in weighted naive Bayesian classifier. *Machine Learning and Cybernetics (ICMLC), 2011 International Conference on*.  
<https://doi.org/10.1109/ICMLC.2011.6016776>
- Harinaldi. (2005). *Prinsip Statistik untuk Teknik dan Sain* (1st ed.). Yogyakarta: Penerbit Erlangga.
- Marsalek, R., & Povalac, K. (2012). Kolmogorov - Smirnov test for spectrum sensing: From the statistical test to energy detection. *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, (1), 97–102.  
<https://doi.org/10.1109/SiPS.2012.58>
- Muktamar, B. A., Setiawan, N. A., & Adji, T. B. (2015). Pembobotan Korelasi Pada Naïve Bayes Classifier. *Seminar Nasional Teknologi Informasi Dan Multimedia 2015*, (2), 43–47.
- Nurnberger, A., Borgelt, C., & Klose, A. (1999). Naive Bayes Classifiers Using Neuro-Fuzzy Learning', 154–159.
- Yao, S., & Li, L. (2012). Weighted Naive Bayesian Classification Algorithm Based on Correlation Coefficients. *International Journal of Advancements in Computing Technology*, 4(20), 29–35.  
<https://doi.org/10.4156/ijact.vol4.issue20.4>