Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 1 (2020) 28 - 36 ISSN Media Elektronik: 2580-0760

LL-KNN ACW-NB: Local Learning K-Nearest Neighbor in Absolute Correlation Weighted Naïve Bayes untuk Klasifikasi Data Numerik

Azminuddin I. S. Azis[1], Budy Santoso[2], Serwin[3]
[1,2,3]Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Ichsan Gorontalo
[1]azminuddinazis@unisan.ac.id, [2]budysantoso@unisan.ac.id, [3]serwin_pattinjo@unisan.ac.id

Abstract

Naïve Bayes (NB) algorithm is still in the top ten of the Data Mining algorithms because of it is simplicity, efficiency, and performance. To handle classification on numerical data, the Gaussian distribution and kernel approach can be applied to NB (GNB and KNB). However, in the process of NB classifying, attributes are considered independent, even though the assumption is not always right in many cases. Absolute Correlation Coefficient can determine correlations between attributes and work on numerical attributes, so that it can be applied for attribute weighting to GNB (ACW-NB). Furthermore, because performance of NB does not increase in large datasets, so ACW-NB can be a classifier in the local learning model, where other classification methods, such as K-Nearest Neighbor (K-NN) which are very well known in local learning can be used to obtain sub-dataset in the ACW-NB training. To reduction of noise/bias, then missing value replacement and data normalization can also be applied. This proposed method is termed "LL-KNN ACW-NB (Local Learning K-Nearest Neighbor in Absolute Correlation Weighted Naïve Bayes)," with the objective to improve the performance of NB (GNB and KNB) in handling classification on numerical data. The results of this study indicate that the LL-KNN ACW-NB is able to improve the performance of NB, with an average accuracy of 91,48%, 1,92% better than GNB and 2,86% better than KNB.

Keywords: *naïve bayes, k-nearest neighbor, absolute correlation coefficient, local learning, attribute weighting*

Abstrak

Algoritma *Naïve Bayes* (NB) masih dalam daftar 10 besar algoritma *Data Mining* karena kesederhanaan, efisiensi, dan kinerjanya. Dalam menangani klasifikasi pada data numerik, distribusi *Gaussian* dan pendekatan *kernel* dapat diterapkan pada NB (GNB dan KNB). Namun dalam proses klasifikasi NB, atribut-atribut dianggap independen, sedangkan asumsi tersebut tidak selalu tepat pada banyak kasus. *Absolute Correlation Coefficient* dapat menentukan korelasi antar atribut dan bekerja pada atribut numerik, sehingga dapat diterapkan untuk *attribute weighting* pada GNB (ACW-NB). Selain itu, karena kinerja NB tidak meningkat dalam *dataset* yang besar, maka ACW-NB dapat menjadi pengklasifikasi dalam model *local learning*, yang mana metode klasifikasi lainnya, yaitu *K-Nearest Neighbor* (K-NN) yang memang terkenal sangat baik dalam *local learning* dapat digunakan untuk memperoleh sub-dataset pelatihan ACW-NB. Untuk mereduksi *noise/bias*, maka *missing value replacement* dan *data normalization* diterapkan pula. Metode yang diusulkan ini diistilahkan dengan "LL-KNN ACW-NB (*Local Learning K-Nearest Neighbor in Absolute Correlation Weighted Naïve Bayes*)," dengan tujuan untuk meningkatkan kinerja NB (GNB dan KNB) dalam menangani klasifikasi pada data numerik. Hasil penelitian menunjukkan bahwa LL-KNN ACW-NB mampu meningkatkan kinerja NB, yaitu dengan akurasi rata-rata sebesar 91,48%, lebih baik 1,92% daripada GNB dan lebih baik 2,86% daripada KNB.

Kata kunci: naïve bayes, k-nearest neighbor, absolute correlation coefficient, local learning, attribute weighting

1. Pendahuluan

Alagaritma *Naïve Bayes* (NB) masih terus berada dalam daftar 10 besar algoritma *Data Mining* karena kesederhanaan, efisiensi, dan kinerjanya [1]. Standar NB bekerja pada data nominal, menggunakan pendekatan distribusi probabilitas, dan sangat baik untuk *dataset* yang besar. Agar dapat bekerja pada data numerik, pendekatan distribusi *Gaussian* dapat diterapkan pada NB untuk menghitung probabilitas dari atribut numerik, dinamakan *Gaussian Naive Bayes* (GNB) [2]. Pendekatan *kernel* dapat pula diterapkan pada NB (KNB) untuk menangani klasifikasi pada data numerik. Namun KNB dapat menyebabkan terjadinya

Diterima Redaksi : 24-10-2020 | Selesai Revisi : 08-01-2020 | Diterbitkan Online : 01-02-2020

curse of dimensionality dan kompleksitas komputasinya jauh lebih besar daripada GNB, karena pendekatan *kernel* melakukan transformasi data ke dimensi yang lebih tinggi (*feature space*). Dengan demikian GNB menjadi pilihan yang lebih efisien untuk menangani klasifikasi pada data numerik.

Dalam proses klasifikasi NB, atribut-atribut dianggap tidak saling terkait (independen), setiap atribut dianggap sama pentingnya, padahal asumsi tersebut tidak selalu tepat dalam banyak kasus [2]-[6]. Dalam mengatasi masalah tersebut, pendekatan *structure extension*, *feature selection*, ataupun *attribute weighting* pada NB terbukti mampu meningkatkan kinerja NB. Secara rinci ditunjukkan pada Tabel 1.

Tabel 1. Pengembangan Algoritma Naïve Bayes

Year, 1st Author	Proposed Method	Categories	Ref
1994, Langley	Greedy Bayesian	Feature Selection	[7]
1997, Friedman	TAN	Structure Extension	[8]
1997, Kohavi	Wrappers–NB	Feature Selection	[9]
1999, Nurnberger	ANFIS–NB	Structure Extension	[6]
2001, Zhang	ANB	Structure Extension	[10]
2002, Ratanamahatana	DT–NB	Feature Selection	[11]
2004, Zhang	WNB–G-HC	Attribute Weighting	[12]
2005, Jiang	Evolutional NB	Feature Selection	[13]
2005, Webb	AODE–NB	Structure Extension	[14]
2007, Hall	DT–CFS–WNB	Attribute Weighting	[15]
2007, Deng	RS–WNB	Attribute Weighting	[16]
2008, Zhang	Cloning NB	Data Expansion	[17]
2009, Jiang	Hidden NB	Structure Extension	[18]
2011, Wu	DEA–WNB	Attribute Weighting	[3]
2011, Lin	PSO–WNB	Attribute Weighting	[4]
2014, Taheri	Adaptive WNB	Attribute Weighting	[5]
2015, Wu	AIS–WNB	Attribute Weighting	[19]
2015, Muktamar	CNBC	Attribute Weighting	[20]
2015, Asmono	AC–WNB	Attribute Weighting	[2]
2016, Zhang	GR & DT–WNB	Feature Selection	[21]
2016, Jiang	DF–WNB	Attribute Weighting	[1]
2017, Song	Multinomial NB	Attribute Weighting	[22]
2017, Zhu	NB–DT–J48	Attribute Weighting	[23]
2018, Sun	LPNB	Structure Extension	[24]
2018, Wu	TAN	Structure Extension	[25]
2018, Yu	CAVW	Attribute Weighting	[26]

Akhir-akhir ini, umumnya pendekatan yang diterapkan untuk mengatasi masalah independensi atribut pada NB adalah *attribute weighting* (ditunjukkan pada Tabel 1). Metode *Absolute Correlation Coefficient* (ACC) bekerja pada atribut numerik dan dapat menentukan kekuatan hubungan antar atribut, sehingga dapat digunakan untuk *attribute weighting* pada NB. Pendekatan ini dinamakan *Absolute Correlation – Weighted Naïve Bayes* (ACW-NB) [2].

Furey, *et al*., menggunakan nilai *absolute* dari *coefficient* dalam penelitian yang dilakukan Golub [27] sebagai metode untuk *feature selection* pada *Support Vector Machine* dalam menangani klasifikasi kanker [28]. Sedangkan Zhang, menggunakan *correlation coefficient* untuk meningkatkan kinerja *Weighted Naïve Bayes* [12]. Begitupun Asmono, Wahono dan Syukur, mirip seperti penelitian yang dilakukan oleh Furey, namun nilai *absolute* dari *coefficient* dalam penelitian yang dilakukan Golub digunakan untuk *attribute weighting* pada GNB dalam menangani prediksi cacat *software* [2]. Dengan demikian penerapan ACC untuk *attribute weighting* pada GNB (ACW-NB) telah terbukti mampu meningkatkan kinerja GNB.

Sementara itu, karena dependensi (ketergantungan) atribut dalam *sub-dataset* pelatihan tentu lebih lemah daripada di seluruh *dataset* pelatihan [18] dan karena kinerja NB tidak meningkat dalam *dataset* yang besar [29], maka ACW-NB dapat menjadi pengklasifikasi dalam model *Local Learning* (LL). Dalam LL, metode klasifikasi lainnya dapat digunakan untuk memperoleh *sub-dataset* pelatihan (ditunjukkan pada Tabel 2).

Kohavi mengusulkan metode *NBTree* dengan menggunakan *Decision Tree* (DT) sebagai LL [29]. Sementara Xie *et al*., mengusulkan metode *Selective Neighborhood Based Naive Bayes* (SNNB) dengan menggunakan *K-Nearest Neighbor* (K-NN) sebagai LL [30]. Begitupun Frank, mengusulkan metode *Local Weighted Naive Bayes* (LWNB) dengan menggunakan K-NN sebagai LL [31].

ACW-NB bekerja pada data numerik, begitupun standar K-NN, sedangkan DT membutuhkan diskretisasi. Dengan demikian K-NN lebih tepat digunakan daripada DT untuk LL pada ACW-NB. Selain itu, K-NN merupakan algoritma yang terkenal sangat baik dalam LL [18].

Tabel 2. Pendekatan Local Learning pada Naïve Bayes

Year, 1st Author	Method	Classifiers	LL	Ref
1996, Kohavi	NBTree	Naïve Bayes	Decision Tree	[29]
2000, Zheng	LBR – NB	LBR	Naïve Bayes	[32]
2002, Xie	SNNB	Naïve Bayes	k-NN	[30]
2003, Frank	LWNB	Naïve Bayes	k-NN	[31]
2018, Safri	NB – k-NN	k-NN	Naïve Bayes	[33]

Walaupun NB cukup kuat dalam menangani *missing value* dan *noisy data*, namun tentu saja lebih efisien apabila masalah *missing value* dan *noisy data* dapat ditangani sebelum NB bekerja. Dengan begitu *noise/bias* dapat direduksi, sehingga efisiensi dan kinerja NB dapat meningkat.

Adanya *missing value* dapat menurunkan efisiensi dan akurasi model klasifikasi [34]. Namun membuangnya bisa jadi menghilangkan informasi yang penting, sehingga mengakibatkan bias. Pendekatan imputasi merupakan strategi yang efisien untuk menangani masalah tersebut, umumnya dengan melakukan *Missing Value Replacement* (MVR) menggunakan pendekatan *mean/mode* [34], [35].

Sementara adanya *outlier* dapat menyebabkan *noise* [36], berdampak buruk pula terhadap kinerja suatu

model klasifikasi [37]. *Outlier* sebaiknya dibuang dengan cara mendeteksinya lebih dahulu, misalnya dengan model prediksi. Namun cara lain yang klasik dan umum diterapkan untuk mereduksi *noise* yaitu dengan melakukan *Data Normalization* (DN) menggunakan pendekatan *Min-Max Normalization* [37]. Oleh karena itu, penerapan MVR dan DN dapat diterapkan dalam tahap pra pengolahan data.

Berdasarkan berbagai latar belakang yang telah dikemukakan, penelitian ini bertujuan untuk meningkatkan kinerja NB (GNB dan KNB) dalam menangani klasifikasi pada data numerik melalui penerapan algoritma K-NN untuk LL, algoritma ACC untuk *attribute weighting*, pendekatan *mean/mode* untuk MVR, dan metode *Min-Max Normalization* untuk DN. Metode yang diusulkan ini kami istilahkan "LL-KNN ACW-NB (*Local Learning K-Nearest Neighbor in Absolute Correlation – Weighted Naïve Bayes*)."

Metode yang diusulkan tersebut diaplikasikan pada 11 *dataset* dengan karakter yang berbeda-beda (ditunjukkan pada Tabel 3). Hal ini agar metode yang diusulkan ini dapat teruji dengan baik. Seluruh *dataset* tersebut dikumpulkan dari *UCI Machine Learning Repository*.

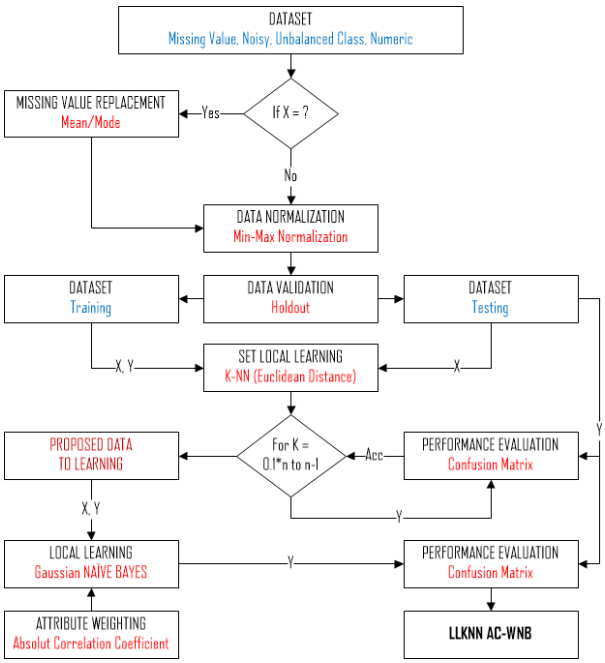
Tabel 3. Karakteristik Dataset

Code	Dataset	Ins.	Att.	MV	Classes		Type
D1	Cleveland	303	14	6	0	54,13%	int, real,
					1	18,15%	bit, ord.
					2	11,88%	
					3	11,55%	
					4	4,29%	
D2	Statlog	270	14	0	1	55,56%	int, real,
					2	44,44%	bit, ord.
D3	Horze Colic	368	28	1927	1	33,70%	int, real,
					2	66,30%	bit, ord.
D4	Hepatitis	155	20	157	1	20,65%	int, real,
					2	79,35%	bit.
D5	Labor	57	17	326	1	64,91%	int, real,
					2	35,09%	bit, ord.
D6	Hypothyroid	3163	26	5329	1	4,77%	int, real,
					2	95,23%	bit.
D7	Newthyroid	215	6	0	1	69,77%	int, real.
					2	16,28%	
					3	13,95%	
D8	BCW Original	699	10	16	1	65,50%	int.
					2	34,50%	
D9	BCW Diagnostic	569	31	0	1	62,74%	real.
					2	37,26%	
D10	BCW Prognostic	198	34	4	1	23,74%	int, real.
					2	76,26%	
D11	Vehicle	846	19	0	1	25,77%	int.
					2	25,06%	
					3	25,65%	
					4	23,52%	

2. Metode Penelitian

Penelitian ini merupakan penelitian eksperimental. *Tools* yang digunakan dalam melakukan eksperimen yaitu Matlab. Dipandang dari jenis informasi yang diolah, penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakukan terhadap data, penelitian ini merupakan penelitian konfirmatori. *Machine Learning* dan klasifikasi dalam *Data Mining* merupakan subjek penelitian ini. Sedangkan objek penelitian ini adalah algoritma NB. Penelitian ini dilaksanakan selama satu tahun lebih, dari Juli 2018 hingga Oktober 2019.

Secara keseluruhan, metode yang diusulkan ditunjukkan pada Gambar 1. Dimulai dari pengumpulan *dataset*, jika terdapat *missing value* pada *dataset*, maka dilakukan MVR. Selanjutnya data dinormalisasikan menggunakan *Min-Max Normalization*. Selanjutnya validasi data menggunakan teknik *Holdout*, dengan komposisi 80% data latih dan 20% data uji. Selanjutnya K-NN bertugas membuat *sub-dataset* LL yang nantinya akan digunakan ACW-NB dalam pelatihannya. K-NN melakukan pelatihan menggunakan data latih dari $K = 0.1*n$ hingga $n-1$, di mana n adalah banyaknya data. Setiap iterasi K, K-NN dievaluasi menggunakan data uji. Data dalam K yang memiliki akurasi terbaik yang merupakan *sub-dataset* LL. Selanjutnya ACC melakukan *attribute weighting* pada GNB dalam pelatihannya menggunakan *sub-dataset* LL. Langkah terakhir adalah mengevaluasi metode yang diusulkan ini menggunakan teknik *Confusion Matrix*.



Gambar 1. Metode yang Diusulkan

Metode yang diusulkan ini (LL-KNN ACW-NB) dikomparasi kinerjanya dengan beberapa metode lainnya yang diuji coba pula dalam penelitian ini (ditunjukkan pada Tabel 4).

2.1 Missing Value Replacement

Pendekatan yang digunakan untuk menangani MVR yaitu *mean/mode*. *Missing value* dari atribut bertipe numerik diganti dengan nilai *mean* (4). Sedangkan

missing value dari atribut bertipe kategorikal (nominal, binominal, ordinal) diganti dengan nilai *mode*.

Tabel 4. Metode yang Diuji Coba

Kode	Metode
M1	GNB (*Gausian Naïve Bayes without MVR & DN*)
M2	KNB (*Kernel Naïve Bayes without MVR & DN*)
M3	GNB (*Gausian Naïve Bayes*)
M4	K-NN (*K-Nearest Neighbor*)
M5	ACW-NB (*Absolute Correlation – Weighted GNB*)
M6	LL-KNN (*Local Learning K-NN*)
M7	LL-KNN-NB (*Local Learning K-NN – GNB*)
M8	LL-KNN ACW-NB (*Proposed Method*)

2.2 Data Normalization

Pendekatan yang digunakan untuk menangani DN yaitu *Min-Max Normalization* (1) yang didefenisikan sebagai berikut:

$$x'_i = \frac{(x_i - x_{min})}{(x_{max} - x_{min})}((n_{max} - n_{min}) + n_{min}) \quad (1)$$

Notasi x_i menyatakan data ke-*i* dari atribut *x*, x_{min} menyatakan nilai minimum dari atribut *x*, x_{max} menyatakan nilai maksimum dari atribut *x*, n_{min} menyatakan jangkauan minimum ke n_{max} menyatakan jangkauan maksimum untuk hasil normalisasi data.

2.3 Data Validation

Teknik yang digunakan untuk validasi data yaitu *Holdout* dengan komposisi 80% data latih dan 20% data uji.

2.4 Gaussian Naïve Bayes

Hasil keputusan klasifikasi metode NB didefinisikan pada Persamaan (2) berikut ini.

$$y' = \underset{y_k}{\operatorname{argmax}}\, P(y_k) \prod_{i=1}^{m} P(x_i|y_k) \quad (2)$$

Notasi *y'* menyatakan label *class* hasil keputusan klasifikasi suatu data uji/prediksi. $P(y_k)$ menyatakan probabilitas label *class* (y_k, *k = 1, 2, …, j)*, yang mana *j* adalah banyaknya label *class*. $P(x_i|y_k)$ menyatakan probabilitas atribut (x_i, *i = 1, 2, …, m)* pada label *class* (y_k), yang mana *m* adalah banyaknya atribut.

Untuk menangani data numerik, maka distribusi *Gaussian* dapat diterapkan, didefenisikan pada Persamaan (3) berikut ini.

$$P(x_i|y_k) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

Notasi μ menyatakan nilai *mean* yang didefinisikan pada Persamaan (4), sedangkan notasi σ adalah nilai *standard deviation* yang didefinisikan pada Persamaan (5). $P(x_i|y_k)$ dari distribusi *Gaussian* kemudian dapat diterapkan pada Persamaan NB (2), yang mana penentuan $P(y_k)$ sama seperti standar NB.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (4)$$

$$\sigma = \left(\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right)^{1/2} \quad (5)$$

2.5 Absolute Correlation Coefficient

Absolute Correlation Coefficient (ACC) (9) dapat menentukan kekuatan antar atribut dan bekerja pada atribut bertipe numerik. Metode ini menggunakan nilai μ (4) dan σ (5). Dasarnya adalah *Correlation Coefficient* (6) yang dapat menentukan kekuatan hubungan antara dua variabel numerik [38].

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2} \quad (6)$$

Notasi $\bar{x}$ menyatakan nilai *mean* (4) dari *x*, sedangkan $\bar{y}$ adalah nilai *mean* (4) dari *y*.

Beberapa penelitian telah menggunakan dan mengembangkan *Correlation Coefficient* tersebut. Guyon *et al*., mengusulkan metode *weighting* [39] dengan menggunakan koefisien (7) dari penelitian yang dilakukan Golub [27], didefinisikan sebagai berikut.

$$w_j = \frac{(\mu_j(+) - \mu_j(-))}{(\sigma_j(+) + \sigma_j(-))} \quad (7)$$

Notasi μ_j menyatakan *mean* (4) dan σ_j menyatakan *standard deviation* (5) dari atribut ke-*j* untuk *class* (+) dan *class* (–) masing-masing. w_j dengan nilai positif yang besar menunjukkan kekuatan hubungan yang kuat dengan *class* (+), sebaliknya w_j dengan nilai negatif yang besar menunjukkan kekuatan hubungan yang kuat dengan *class* (–).

Zhang, meningkatkan kinerja *Weighted Naïve Bayes* menggunakan *Correlation Coefficient* pula [12]. Sementara Pavlidis, *et al*., mengusulkan *associated coefficients* (8) yang didefinisikan sebagai berikut [40].

$$w_i = \frac{(\mu_i(+) - \mu_i(-))^2}{(\sigma_i(+)^2 + \sigma_i(-)^2)} \quad (8)$$

Furey, *et al*., menggunakan nilai *absolute* dari w_i (7) sebagai metode untuk *feature selection* pada *Support Vector Machine* dalam menangani klasifikasi kanker [28]. Begitupun Asmono, Wahono & Syukur, menggunakan nilai *absolute* dari w_i (7) sebagai metode *attribute weighting* pada *Naïve Bayes* dalam menangani prediksi cacat *software* [2]. Metode *weighting* tersebut kemudian dinamakan *Absolute Correlation Coefficient* (9), didefinisikan sebagai berikut.

$$w_j = \left|\frac{(\mu_{jc} - \mu_{j\bar{c}})}{(\sigma_{jc} - \sigma_{j\bar{c}})}\right| \quad (9)$$

Notasi w_j menyatakan *weight* dari atribut ke-*j*, μ_{jc} menyatakan nilai *mean* (4) dari atribut ke-*j* pada *class c*, $\mu_{j\bar{c}}$ menyatakan nilai *mean* (4) dari atribut ke-*j* pada *class* non *c*; σ_{jc} menyatakan nilai *standard deviation* (5) dari atribut ke-*j* pada *class c*; dan $\sigma_{j\bar{c}}$ menyatakan

nilai *standard deviation* (5) dari atribut ke-*j* pada *class* non *c*.

2.6 Absolute Correlation Weighted Naïve Bayes

Attribute weighting untuk NB disebut *Weighted Naive Bayes* (WNB), sehingga Persamaan NB (2) dapat diubah menjadi persamaan (10) berikut ini.

$$y' = \underset{y_k}{\operatorname{argmax}} P(y_k) \prod_{i=1}^{m} P(x_i|y_k)^{w_i} \quad (10)$$

Nilai w_i (*weight*) pada Persamaan (10) di atas dapat mengggunakan pendekatan ACC (9), sehingga menjadi *Absolute Correlation Weighted Naïve Bayes* (ACW-NB) [2]. Metode ACW-NB mampu bekerja pada atribut numerik, dapat menentukan kekuatan hubungan antar atribut, dan terbukti secara signifikan mampu meningkatkan kinerja NB [2]. Sedangkan K-NN terkenal sangat baik dalam LL [18] (LL-KNN) dan terbukti pula mampu meningkatkan kinerja NB [30], [31]. Dengan demikian, kombinasi keduanya akan lebih meningkatkan lagi kinerja NB. Secara rinci, algoritma ACW-NB adalah sebagai berikut:

1. Hitung $P(x_i|y_k)$, yaitu probabilitas setiap atribut (x_i) pada setiap label *class* (y_k) menggunakan distribusi *Gaussian* (3).
2. Hitung L_k, yaitu *weight likelihood* setiap label *class* menggunakan Persamaan (11) berikut ini, yang mana w_i diperoleh menggunakan metode ACC (9).

$$L_k = \prod_{i=1}^{m} P(x_i|y_k)^{w_i} \quad (11)$$

3. Hitung $P(y_k)$, yaitu probabilitas setiap label *class* (y_k) menggunakan Persamaan (12) berikut ini, yang mana L_k menyatakan *weight likelihood* label *class* ke-*k*, sedangkan $L_{\bar{k}}$ menyatakan *weight likelihood* label *class* lainnya. $P(y_k)$ inilah yang mengganti $P(y_k)$ standar NB.

$$P(y_k) = \frac{L_k}{L_k + L_{\bar{k}}} \quad (12)$$

4. Akhirnya $P(x_i|y_k)$ yang diperoleh dari distribusi *Gaussian* (proses/langkah 1) dan $P(y_k)$ yang diperoleh dari proses/langkah 3 dapat diterapkan pada Persamaan NB (2).

2.7 Local Learning K-NN

Ide dasar dari LL adalah membangun model klasifikasi dari *sub-dataset* pelatihan saja (*local learning*) daripada menggunakan seluruh *dataset* pelatihan [18]. DT [29] dan K-NN [30]**,** [31] merupakan algoritma yang dapat digunakan untuk LL. ACW-NB bekerja pada atribut numerik, sehingga K-NN yang lebih tepat untuk LL dari pada DT yang membutuhkan diskretisasi dalam mengolah atribut numerik.

Hasil keputusan klasifikasi metode K-NN didefinisikan pada Persamaan (13) berikut ini.

$$y' = \underset{v}{\operatorname{argmax}} \sum_{D \in D_z}^{k} I(v = y_c) \quad (13)$$

Notasi *y'* menyatakan label *class* hasil keputusan klasifikasi suatu data uji/prediksi. *v* menyatakan jumlah data yang masuk dalam *class* (y_c, *c = 1, 2, …, p*), yang mana *p* menyatakan banyaknya *class*. Sedangkan $d(x', x_j)$ merupakan jarak antara data prediksi/uji (*z*) ke setiap data latih (L_j) yang disimpan dalam *D* dapat dihitung menggunakan salah satu metode pengukuran jarak (*dissimilarity*), umumnya menggunakan *Euclidean* yang didefinisikan pada Persamaan (14) berikut ini.

$$D(x', x) = \|x' - x\|_2 = \sqrt{\sum_{i=1}^{n} (x' - x_i)^2} \quad (14)$$

Prosedur LL yang kami gunakan berdasarkan penelitian yang telah dilakukan oleh Xie *et al*., [30]. K-NN bertugas membuat *sub-dataset* LL yang nantinya akan digunakan ACW-NB dalam pelatihannya. K-NN melakukan pelatihan menggunakan data latih dari *k = 0.1*n* hingga *n-1*, di mana *n* adalah banyaknya data. Setiap iterasi *K*, K-NN dievaluasi menggunakan data uji. Data dalam *K* yang memiliki akurasi terbaik yang merupakan *sub-dataset* LL. Selanjutnya ACW-NB melakukan pelatihan dan klasifikasi menggunakan *sub-dataset* LL tersebut.

2.8 Method Evaluation

Pengukuran kinerja suatu model klasifikasi dapat dilakukan menggunakan pendekatan *Confussion Matrix* untuk memperoleh *accuracy*, *precision*, *recall* (*sensitivity* dan *specificity*), dan *F-Measure* yang ditunjukkan pada Tabel 5 berikut ini.

Tabel 5. Confusion Matrix

	Actual +	Actual -	Precision
Predicted +	TP	FP	TP/(TP+FP) *
Predicted -	FN	TN	TN/(TN+FN)
Recall	TP/(TP+FN)	TN/(TN+FP)	
F-Measure	(2*Precision*Sensitivity)/(Precision+Sensitivity)		
Accuracy	(TP+TN) / (TP+TN+FN+FP)		

Keterangan: True (T); False (F); Positive (P); Negative (N)

3. Hasil dan Pembahasan

Pada tahap MVR, *missing value* diganti nilainya dengan nilai *mode* dari atribut yang bertipe ordinal atau nominal. Sedangkan atribut yang bertipe numerik (*integer* dan *real*), *missing value* diganti dengan nilai *mean* (4) dari atribut tersebut. Pada atribut bertipe *integer*, nilai *mean* yang diperoleh kemudian dibulatkan sebagai hasil normalisasi data. Sedangkan pada atribut bertipe *real*, nilai *mean* yang diperoleh tidak perlu dibulatkan sebagai hasil normalisasi data.

Selanjutnya pada tahap DN menggunakan teknik *Min-Max Normalization* (1) dalam jangkauan [0, 1]. Misalnya data ke-1 dari atribut *x* (x_1) = 7, data

maksimum dari atribut x = 10, dan data minimum dari atribut x = 0, maka hasil normalisasi data ke-1 dari atribut x (x'_1) adalah sebagai berikut.

$$x'_1 = \frac{7-1}{10-0}((1-0)+0) = 0{,}7$$

Setelah MVR dan DN dilakukan, prosedur terakhir dalam pra pengolahan data adalah *data validation* menggunakan teknik *HoldOut* dengan komposisi data latih sebesar 80% dan data uji sebesar 20%.

Setelah pra pengolahan data, langkah selanjutnya adalah melakukan pemodelan melalui pelatihan dan pengujian terhadap metode-metode yang diuji coba (ada 8 metode yang diuji coba yang ditunjukkan pada Tabel 4). Berdasakan hasil beberapa percobaan yang telah dilakukan, GNB yang memang hanya untuk menangani data numerik bahkan kurang baik untuk data ordinal yang telah di transformasi dengan teknik *encoding*. Sementara itu, *unbalanced class* jadi masalah serius yang bahkan bisa menyebabkan *error* jika probabilitas *Gaussian* suatu atribut pada suatu *class* tidak diperoleh. Masalah ini sebenarnya bisa diselesaikan oleh KNB atau bisa pula dengan pendekatan *ensemble* untuk mereduksi *unbalanced class*. Namun penelitian ini hanya fokus pada pengembangan GNB. Selain itu, kompleksitas komputasi KNB yang relatif besar tidak efisien dibandingkan GNB.

Contohnya pada *dataset* yang memiliki *class* 1 yang jauh lebih besar daripada *class* 2 dan lebih parah lagi dengan *missing value* yang begitu banyak. Hal ini menyebabkan data tidak terdistribusi dengan baik. Misalnya saja pada atribut *A class 1* hanya memiliki data = 2, selebihnya adalah *missing value*, padahal atribut *A* merupakan data ordinal dengan nilai 1, 2, atau 3. *Missing value* pada atribut *A* tidak boleh diganti dengan nilai *mean* karena bersifat ordinal. Tapi jika *missing value* diganti dengan nilai *mode* pada atribut *A*, maka *missing value* diganti menjadi nilai 2. Dengan demikian, *class 1* pada atribut *A* hanya memiliki nilai = 2, sehingga GNB tidak dapat menanganinya.

Kasus seperti ini terjadi pada beberapa *dataset*, yaitu D3, D4, D5, D6, dan D11 yang tidak bisa ditangani GNB. Dengan demikian, *dataset* yang digunakan tinggal berjumlah enam, yaitu D1, D2, D7, D8, D9, dan D10. Secara rinci, kinerja *accuracy*, *precision*, *specificity*, *sensitivity*, dan lama proses (detik) tiap-tiap metode yang diuji coba pada setiap *dataset* tersebut ditunjukkan pada Tabel 6.

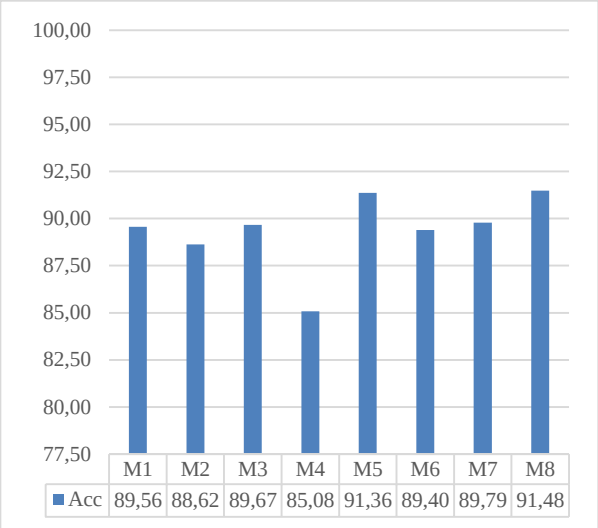
Hasil evaluasi menunjukkan bahwa akurasi rata-rata yang terbaik dari seluruh *dataset* diberikan oleh metode M8 (LL-KNN ACW-NB) atau metode yang diusulkan, yaitu sebesar 91,48%. Pada setiap *dataset*, LL-KNN ACW-NB memberikan akurasi yang paling tinggi, kecuali pada *dataset* D9, yaitu M4 (K-NN) dengan akurasi 96,46%, selisih 1,77% dengan LL-KNN ACW-NB yang sebesar 94,69%. *Dataset* D9 memilki dimensi yang cukup besar (31 atribut dengan 569 *instances*), memilki masalah *unbalanced class*, tidak ada *missing value*, dan semua atribut bertipe numerik.

Tabel 6. Hasil Evaluasi

DS	Method	Accuracy	Precision	Specificity	Sensitivity	Times
D1	M1	85,96 %	93,55 %	82,86 %	90,91 %	0,34
	M2	82,46 %	90,32 %	80,00 %	86,36 %	0,51
	M3	86,44 %	93,75 %	83,33 %	91,30 %	0,06
	M4	83,05 %	93,75 %	78,95 %	90,48 %	0,21
	M5	86,44 %	93,75 %	83,33 %	91,30 %	0,19
	M6	84,75 %	100 %	78,05 %	100 %	6,67
	M7	86,44 %	93,75 %	83,33 %	91,30 %	6,70
	M8	86,44 %	93,75 %	83,33 %	91,30 %	6,79
D2	M1	92,59 %	93,33 %	93,33 %	91,67 %	0,36
	M2	90,74 %	93,33 %	90,32 %	91,30 %	0,61
	M3	92,59 %	93,33 %	93,33 %	91,67 %	0,08
	M4	90,74 %	93,33 %	90,32 %	91,30 %	0,21
	M5	94,44 %	93,33 %	96,55 %	92,00 %	0,21
	M6	92,59 %	96,67 %	90,63 %	95,45 %	5,07
	M7	92,59 %	93,33 %	93,33 %	91,67 %	5,10
	M8	94,44 %	93,33 %	96,55 %	92,00 %	5,18
D7	M1	97,67 %	100 %	96,77 %	100 %	0,33
	M2	100 %	100 %	100 %	100 %	0,47
	M3	97,67 %	100 %	96,77 %	100 %	0,09
	M4	69,77 %	100 %	81,08 %	NaN	0,14
	M5	100 %	100 %	100 %	100 %	0,20
	M6	93,02 %	100 %	90,91 %	100 %	2,37
	M7	97,67 %	100 %	96,77 %	100 %	2,40
	M8	100 %	100 %	100 %	100 %	2,46
D8	M1	95,56 %	95,45 %	97,67 %	91,84 %	0,34
	M2	96,30 %	97,73 %	96,63 %	95,65 %	0,60
	M3	95,68 %	95,60 %	97,75 %	92,00 %	0,06
	M4	93,53 %	97,80 %	92,71 %	95,35 %	0,14
	M5	95,68 %	95,60 %	97,75 %	92,00 %	0,34
	M6	93,53 %	98,90 %	91,84 %	97,56 %	47,26
	M7	96,40 %	97,80 %	96,74 %	95,74 %	47,30
	M8	96,40 %	97,80 %	96,74 %	95,74 %	47,45
D9	M1	93,81 %	92,96 %	97,06 %	88,89 %	0,55
	M2	95,58 %	94,37 %	98,53 %	91,11 %	1,21
	M3	93,81 %	92,96 %	97,06 %	88,89 %	0,09
	M4	96,46 %	98,59 %	95,89 %	97,50 %	0,30
	M5	94,69 %	92,96 %	98,51 %	89,13 %	0,43
	M6	95,58 %	97,18 %	95,83 %	95,12 %	21,39
	M7	93,81 %	92,96 %	97,06 %	88,89 %	21,42
	M8	94,69 %	92,96 %	98,51 %	89,13 %	21,65
D10	M1	71,79 %	33,33 %	37,50 %	80,65 %	0,28
	M2	66,67 %	44,44 %	33,33 %	81,48 %	0,76
	M3	71,79 %	33,33 %	37,50 %	80,65 %	0,06
	M4	76,92 %	0,00 %	NaN	76,92 %	0,16
	M5	76,92 %	66,67 %	50,00 %	88,89 %	0,16
	M6	76,92 %	0,00 %	NaN	76,92 %	7,34
	M7	71,79 %	33,33 %	37,50 %	80,65 %	7,36
	M8	76,92 %	66,67 %	50,00 %	88,89 %	7,43
Avg	M1	89,56 %	84,77 %	84,20 %	90,66 %	0,37
	M2	88,62 %	86,70 %	83,14 %	90,99 %	0,69
	M3	89,67 %	84,83 %	84,29 %	90,75 %	0,07
	M4	85,08 %	80,58 %	87,79 %	90,31 %	0,19
	M5	91,36 %	90,39 %	87,69 %	92,22 %	0,25
	M6	89,40 %	82,13 %	89,45 %	94,18 %	15,02
	M7	89,79 %	85,20 %	84,12 %	91,37 %	15,05
	M8	91,48 %	90,75 %	87,52 %	92,84 %	15,16

Walaupun M4 (K-NN) memberikan akurasi terbaik pada *dataset* D9, namun secara rata-rata dari seluruh *dataset*, M4 (K-NN) memberikan akurasi terburuk, yaitu sebesar 85,08%, disusul metode M2 (KNB) sebesar 88,62%. Sedangkan akurasi 100% diberikan

oleh metode LL-KNN ACW-NB, M5 (ACW-NB), dan M2 (KNB) pada *dataset* D7 yang memilki dimensi kecil (6 atribut dengan 215 *instances*), memilki masalah *unbalanced class*, tidak ada *missing value*, dan semua atribut bertipe numerik. Secara keseluruhan, akurasi rata-rata dari tiap-tiap metode ditunjukkan pada Gambar 2.

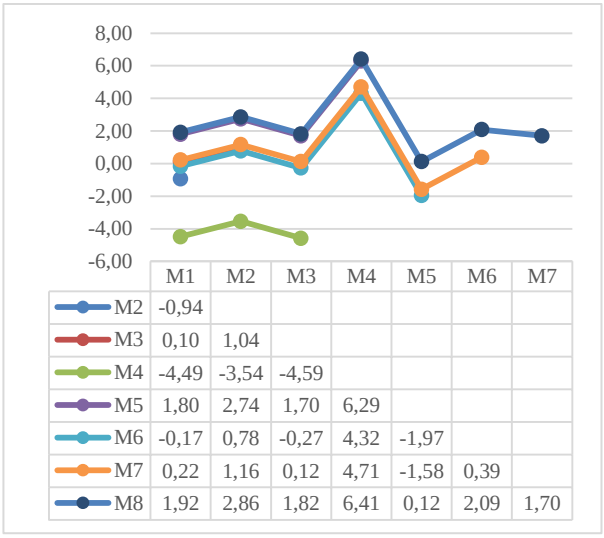


Gambar 2. Akurasi Rata-Rata Setiap Metode

Hasil evaluasi menunjukkan pula bahwa penerapan MVR dan DN memberikan peningkatan kinerja GNB. Hal ini dapat ditunjukkan melalui akurasi rata-rata yang diperoleh metode M3 (GNB) sebesar 89,67%, lebih baik 0,10% daripada M1 (GNB tanpa MVR dan DN). Selisihnya lebih meningkat lagi ketika ACC diterapkan pada GNB (ACW-NB) dan terus meningkat hingga diterapkannya LL pada metode M8 (LL-KNN ACW-NB), hingga selisih 1,92% dengan M1 (GNB tanpa MVR dan DN) dan 1,82% dengan metode M3 (GNB).

Sedangkan penerapan K-NN untuk LL, yaitu pada metode M6 (LL-KNN) menunjukkan pula akurasi yang lebih baik 4,32% daripada metode M4 (K-NN). Selanjutnya lebih baik lagi ketika LL-KNN diterapkan pada GNB, yaitu pada metode M7 (LL-KNN-NB), selisih 4,71% dengan metode M4 (K-NN), 0,12% dengan metode M3 (GNB), dan 0,22% dengan metode M1 (GNB tanpa MVR dan DN). Secara lengkap, komparasi antar metode-metode ditunjukkan pada Gambar 3.

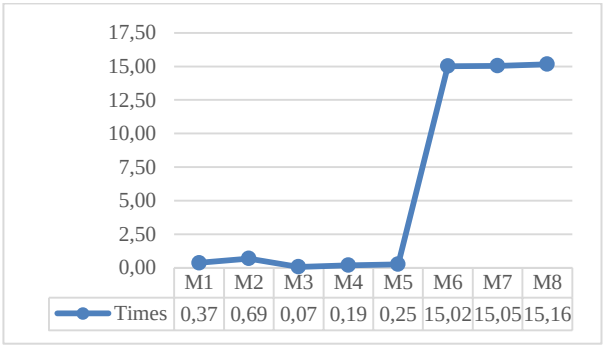
Namun kinerja akurasi biasanya berlawanan dengan kinerja kompleksitas komputasi. Hal ini dapat ditunjukkan melalui waktu (dalam satuan detik) yang dibutuhkan suatu metode dalam melakukan prosesnya. Penerapan MVR dan DN memang justru menurunkan waktu proses, ditunjukkan pada lama proses rata-rata metode M3 (GNB) sebesar 0,07 detik, lebih baik 0,29 detik daripada metode M1 (GNB tanpa MVR dan DN) yang lama proses rata-ratanya sebesar 0,37 detik. Bahkan masih tetap lebih baik ketika ACC diterapkan pada GNB, yaitu pada metode M5 (ACW-NB) dengan lama proses rata-rata sebesar 0,25 detik, masih lebih baik 0,11 detik daripada metode M1 (GNB tanpa MVR dan DN).



	M1	M2	M3	M4	M5	M6	M7
M2	-0,94						
M3	0,10	1,04					
M4	-4,49	-3,54	-4,59				
M5	1,80	2,74	1,70	6,29			
M6	-0,17	0,78	-0,27	4,32	-1,97		
M7	0,22	1,16	0,12	4,71	-1,58	0,39	
M8	1,92	2,86	1,82	6,41	0,12	2,09	1,70

Gambar 3. Komparasi Kinerja Antar Metode (Selisih Akurasi)

Sayangnya ketika pendekatan LL-KNN diterapkan, yaitu pada metode M6, waktu proses rata-rata yang dibutuhkan meningkat secara signifikan, yaitu sebesar 15,02 detik, lebih buruk 14,65 detik daripada metode M1 (GNB tanpa MVR dan DN), 14,94 detik daripada M3 (GNB), dan 14,76 detik daripada M5 (ACW-NB). Dengan begitu waktu proses metode-metode selanjutnya, yaitu metode M7 (LL-KNN-NB) dan metode M8 (LL-KNN ACW-NB) tentu saja akan lebih lama. Secara keseluruhan, waktu proses rata-rata tiap-tiap metode ditunjukkan pada Gambar 4 berikut ini.



	M1	M2	M3	M4	M5	M6	M7	M8
Times	0,37	0,69	0,07	0,19	0,25	15,02	15,05	15,16

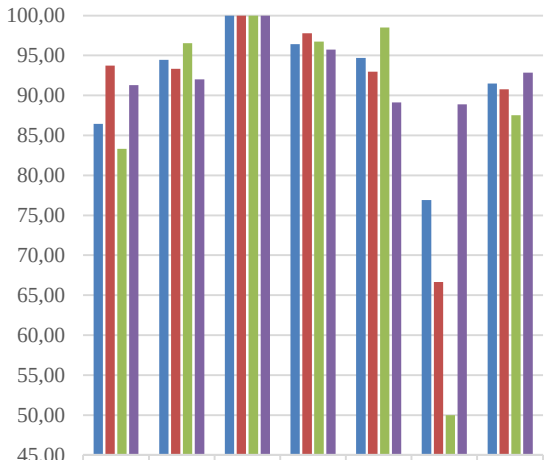
Gambar 4. Waktu Proses Rata-Rata Setiap Metode

Seperti yang telah dijelaskan sebelumnya, metode M8 (LL-KNN ACW-NB) sebagai metode yang diusulkan, menunjukkan kinerja yang lebih baik dari pada metode-metode lainnya. Secara rinci, kinerja LL-KNN ACW-NB ditunjukkan pada Gambar 5.

Hasil penelitian ini mengindikasikan bahwa:

1. Penerapan pendekatan *mean/mode* untuk *Missing Value Replacement* (MVR) dan *Min-Max Normalization* untuk *Data Normalization* (DN) pada *Gaussian Naive Bayes* (GNB + MVR & DN)

menunjukkan akurasi sebesar 89,67%, lebih baik 0,10% dari pada GNB tanpa MVR & DN, dan lebih baik 1,04% daripada *Kernel Naive Bayes* (KNB). Begitupun waktu proses yang dibutuhkan menurun setelah MVR & DN diterapkan, yaitu selisih 0,29 detik. Dengan demikian penerapan MVR & DN mampu meningkatkan kinerja akurasi dan efisiensi NB (GNB dan KNB).



	D1	D2	D7	D8	D9	D10	Avg
Acc	86,44	94,44	100,00	96,40	94,69	76,92	91,48
Pre	93,75	93,33	100,00	97,80	92,96	66,67	90,75
Spe	83,33	96,55	100,00	96,74	98,51	50,00	87,52
Sen	91,30	92,00	100,00	95,74	89,13	88,89	92,84

Gambar 5. Kinerja LL-KNN ACW-NB

2. Akurasi rata-rata *Absolute Correlation Weighted Naive Bayes* (ACW-NB) sebesar 91,36%, lebih baik 1,70% daripada GNB + MVR & DN, dan lebih baik 2,74% daripada KNB. Dengan demikian penerapan *Absolute Correlation Coefficient* (ACC) untuk *attribute weighting* pada GNB + MVR & DN mampu meningkatkan kinerja akurasi NB (GNB dan KNB).
3. Akurasi rata-rata *Local Learning K-Nearest Neighbor* (LL-KNN) sebesar 89,40%, lebih baik 4,32% daripada K-NN. Dengan begitu LL-KNN dapat pula diterapkan pada GNB + MVR & DN. Hasilnya, akurasi rata-rata LL-KNN-NB sebesar 89,79%, lebih baik 0,12% daripada GNB + MVR & DN, dan lebih baik 1,16% daripada KNB. Dengan demikian penerapan K-NN untuk *local learning* pada GNB + MVR & DN mampu meningkatkan kinerja akurasi NB (GNB dan KNB).
4. Penerapan pendekatan *mean/mode* untuk MVR dan *Min-Max Normalization* untuk DN pada GNB (GNB + MVR & DN) menunjukkan akurasi yang lebih baik daripada NB (GNB dan KNB). Selanjutnya penerapan K-NN untuk *local learning* pada GNB + MVR & DN (LL-KNN-NB) dan ACC untuk *attribute weighting* pada GNB + MVR & DN (ACW-NB) menunjukkan akurasi yang lebih baik daripada GNB + MVR & DN. Dengan begitu LL-KNN-NB dan ACW-NB dapat disatukan untuk lebih meningkatkan kinerja akurasi NB (GNB dan KNB). Hasilnya, akurasi rata-rata LL-KNN ACW-NB sebesar 91,48%, lebih baik 1,82% daripada GNB + MVR & DN, lebih baik 1,92% daripada GNB, dan lebih baik 2,86% daripada KNB. Dengan demikian penerapan pendekatan *mean/mode* untuk MVR, *Min-Max Normalization* untuk DN, K-NN untuk *local learning*, dan ACC untuk *attribute weighting* pada GNB mampu meningkatkan kinerja akurasi NB (GNB dan KNB).
5. Kompleksitas komputasi (waktu proses) yang dibutuhkan sangat tinggi ketika pendekatan *local learning* menggunakan K-NN diterapkan, selisih 14,90 detik antara LL-KNN ACW-NB dengan ACW-NB. Hal ini karena LL-KNN ACW-NB memiliki sub proses (iterasi) sebanyak *0,1*n* hingga *n-1* (*n* adalah jumlah *instances*) dalam menentukan data latih untuk ACW-NB berdasarkan akurasi disetiap sub prosesnya.

4. Kesimpulan

Berdasarkan indikasi-indikasi dari hasil penelitian, maka disimpulkan bahwa penerapan pendekatan *mean/mode* untuk *missing value replacement*, *Min-Max Normalization* untuk *data normalization*, *K-Nearest Neighbor* untuk *local learning*, dan *Absolute Correlation Coefficient* untuk *attribute weighting* pada *Gaussian Naïve Bayes* yang dinamakan LL-KNN ACW-NB (*Local Learning K-Nearest Neighbor in Absolute Correlation Weighted Naïve Bayes*) mampu meningkatkan kinerja *Naïve Bayes* (*Gaussian Naïve Bayes* dan *Kernel Naïve Bayes*), yaitu dengan akurasi rata-rata sebesar 91,48%, lebih baik 1,92% daripada *Gaussian Naïve Bayes* dan lebih baik 2,86% daripada *Kernel Naïve Bayes*.

Namun kompleksitas komputasi waktu proses yang dibutuhkan untuk menerapkan *local learning* menggunakan K-NN (LL-KNN) sangat tinggi dibandingkan tidak menerapkannya, selisih 14,90 detik antara LL-KNN ACW-NB dengan ACW-NB. Sementara selisih akurasi rata-rata antara LL-KNN ACW-NB dengan ACW-NB tidak berbeda jauh, hanya selisih 0,12%, paling kecil dibandingkan selisih LL-KNN-ACW-NB dengan metode-metode lainnya. Dengan demikian metode ACW-NB lebih disarankan daripada LL-KNN ACW-NB jika mempertimbangkan sisi efisiensi secara keseluruhan. Kompleksitas komputasi waktu proses yang tinggi ini dapat diperbaiki pada penelitian berikutnya melalui strategi *local learning* yang berbeda.

Ucapan Terima Kasih

Penelitian ini didukung dan didanai oleh: (1) Direktorat Riset dan Pengabdian Masyarakat; (2) Kementerian Riset dan Pendidikan Tinggi Republik Indonesia.

Daftar Rujukan

[1] L. Jiang, C. Li, S. Wang, and L. Zhang, "Deep feature weighting for naive Bayes and its application to text classification," *Eng. Appl. Artif. Intell.*, vol. 52, pp. 26–39, 2016.

[2] R. T. Asmono, R. S. Wahono, and A. Syukur, "Absolute Correlation Weighted Naïve Bayes for Software Defect Prediction," *J. Softw. Eng.*, vol. 1, no. 1, pp. 38–45, 2015.

[3] J. Wu and Z. Cai, "Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB)," *J. Comput. Inf. Syst.*, vol. 5, no. 5, pp. 1672–1679, 2011.

[4] J. Lin and J. Yu, "Weighted Naive Bayes Classification Algorithm Based on Particle Swarm Optimization," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011, pp. 444–447.

[5] S. Taheri, J. Yearwood, M. Mammadov, and S. Seifollahi, "Attribute weighted Naive Bayes classifier using a local optimization," *Neural Comput. Appl.*, vol. 24, no. 5, pp. 995–1002, 2014.

[6] A. Nurnberger, C. Borgelt, and A. Klose, "Naive Bayes Classifiers Using Neuro-Fuzzy Learning '," in *ICONIP'99. ANZIIS'99 & ANNES'99 & ACNN'99. 6th International Conference on Neural Information Processing. Proceedings (Cat. No.99EX378)*, 1999, pp. 154–159.

[7] P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," in *Proceedings 10th Conference Uncertainty in Artificial Intelligence*, 1994, pp. 339–406.

[8] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," *Mach. Learn.*, vol. 29, pp. 131–163, 1997.

[9] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.

[10] H. Zhang and C. X. Ling, "An Improved Learning Algorithm for Augmented Naive Bayes," *Adv. Knowl. Discov. Data Min.*, pp. 581–586, 2001.

[11] C. A. Ratanamahatana and D. Gunopulos, "Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection," in *Proceedings Workshop Data Cleaning and Preprocessing (DCAP '02)*, 2002.

[12] H. Zhang and S. Sheng, "Learning Weighted Naive Bayes with Accurate Ranking," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004, pp. 567–570.

[13] L. Jiang, H. Zhang, Z. Cai, and J. Su, "Evolutional Naive Bayes," in *Proceedings First International Symposium on Intelligent Computation and Its Applications (ISICA '05)*, 2005, pp. 344–350.

[14] G. I. Webb, J. R. Boughton, and Z. Wang, "Not So Naive Bayes: Aggregating One-Dependence Estimators," *Mach. Learn.*, vol. 58, no. 1, pp. 5–24, 2005.

[15] M. Hall, "A Decision Tree-Based Attribute Weighting Fiter for Naive Bayes," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2007, pp. 59–70.

[16] W. Deng, G. Wang, and Y. Wang, "Weighted Naive Bayes Classification Algorithm Based on Rough Set," *Comput. Sci.*, vol. 34, pp. 204–206, 2007.

[17] H. Zhang, "Using Instance Cloning to Improve Naive Bayes for Ranking," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 22, no. 6, pp. 1121–1140, 2008.

[18] L. Jiang, H. Zhang, and Z. Cai, "A Novel Bayes Model: Hidden Naive Bayes," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1361–1371, 2009.

[19] J. Wu, S. Pan, Z. Cai, X. Zhu, P. Zhang, and C. Zhang, "Self-adaptive attribute weighting for Naive Bayes classification," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1487–1502, 2015.

[20] B. A. Muktamar, N. A. Setiawan, and T. B. Adji, "Pembobotan Korelasi Pada Naïve Bayes Classifier," in *Seminar Nasional Teknologi Informasi dan Multimedia 2015*, 2015, no. 2, pp. 43–47.

[21] L. Zhang, L. Jiang, C. Li, and G. Kong, "Two Feature Weighting Approaches for Naive Bayes Text Classifiers," *Knowledge-Based Syst.*, vol. 100, pp. 137–144, 2016.

[22] J. Song, K. T. Kim, B. Lee, S. Kim, and H. Y. Youn, "A novel classification approach based on Naïve Bayes for Twitter sentiment analysis," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 6, pp. 2996–3011, 2017.

[23] J. Zhu, J. Xu, C. Zhang, and Y. Gao, "Marine Fishing Ground Prediction Based on Bayesian Decision Tree Model," in *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, 2017, pp. 316–320.

[24] N. Sun, B. Sun, J. D. Lin, and M. Y. Wu, "Lossless Pruned Naive Bayes for Big Data Classifications," *Big Data Res.*, vol. 14, pp. 27–36, 2018.

[25] J. Wu, "A Generalized Tree Augmented Naive Bayes Link Prediction Model," *J. Comput. Sci.*, vol. 27, pp. 206–217, 2018.

[26] L. Yu, L. Jiang, W. Dianhong, and L. Zhang, "Toward naive Bayes with attribute value weighting," *Neural Comput. Appl.*, vol. 5, pp. 1–15, 2018.

[27] T. R. Golub *et al.*, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science (80-.).*, vol. 286, no. 5439, pp. 531–537, 1999.

[28] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[29] R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid," in *Proceedings Second International Conference Knowledge Discovery and Data Mining (KDD '96)*, 1996, pp. 202–207.

[30] Z. Xie, W. Hsu, Z. Liu, and M. L. Lee, "SNNB: A Selective Neighborhood Based Naive Bayes for Lazy Learning," in *Proceedings Sixth Pacific-Asia Conference Knowledge Discovery and Data Mining (KDD '02)*, 2002, pp. 104–114.

[31] E. Frank, M. Hall, and B. Pfahringer, "Locally Weighted Naive Bayes," in *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 249–256.

[32] Z. Zheng and G. I. Webb, "Lazy Learning of Bayesian Rules," *Mach. Learn.*, vol. 41, no. 1, pp. 53–84, 2000.

[33] Y. F. Safri, R. Arifudin, and M. A. Muslim, "K-Nearest Neighbor and Naive Bayes Classifier Algorithm in Determining The Classification of Healthy Card Indonesia Giving to The Poor," *Sci. J. Informatics*, vol. 5, no. 1, pp. 9–18, 2018.

[34] C. Bielza and P. Larrañaga, "Discrete Bayesian Network Classifiers: A Survey," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 5:1-5:43, 2014.

[35] S. Zhang, Z. Jin, and X. Zhu, "Missing data imputation by utilizing information within incomplete instances," *J. Syst. Softw.*, vol. 84, no. 3, pp. 452–459, 2011.

[36] P. H. Abreu, M. S. Santos, M. H. Abreu, B. Andrade, and D. C. Silva, "Predicting Breast Cancer Recurrence Using Machine Learning Tehniques: A Systematic Review," *ACM Comput. Surv.*, vol. 49, no. 3, pp. 52:1-52:40, 2016.

[37] M. M. Suarez-Alvarez, D.-T. Pham, M. Y. Prostov, and Y. I. Prostov, "Statistical approach to normalization of feature vectors and clustering of mixed datasets," in *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2012, vol. 468, no. 2145, pp. 2630–2651.

[38] R. J. Freund and W. J. Wilson, *Statistical Methods (2nd ed.)*. Academic Press, 2003.

[39] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machine," *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002.

[40] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, "Gene functional classification from heterogeneous data," in *Proceedings of the fifth annual international conference on Computational biology - RECOMB '01*, 2001, no. 212, pp. 1–11.