

IMPLEMENTATION OF NAIVE BAYES CLASSIFIER METHOD AND ADABOOST ALGORITHM FOR PREDICTION OF CHRONIC KIDNEY DISEASE

Adhi Indra Irawan
Informatika, Fakultas MIPA,
Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
adhiindra@student.uns.ac.id

Ristu Saptono
Informatika, Fakultas MIPA,
Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
ristu.saptono@staff.uns.ac.id

Afrizal Doewes
Informatika, Fakultas MIPA,
Universitas Sebelas Maret Surakarta
Jl. Ir. Sutami No 36 A Surakarta
afrizal.doewes@staff.uns.ac.id

ABSTRACT

Problems that often occur in the medical dataset are many attributes that have missing values. Naïve Bayes method is known to provide good accuracy compared to other methods in dealing with missing values. However, when the results obtained are still not satisfactory then boosting with AdaBoost is used to improve its performance. This study discusses the application of Naïve Bayes method and AdaBoost algorithm to classify chronic kidney disease (CKD). From the result obtained by calculating the confusion matrix, the Naïve Bayes method achieved the accuracy of 0.95 and F1-score of 0.958. While the combination of AdaBoost and Naïve Bayes managed to improve the accuracy of 0.98 and F1-score of 0.984. When the missing values are replaced, the accuracy of Naïve Bayes decreased to 0.945 and F1-score of 0.954, while AdaBoost successfully increased the accuracy to 0.9825 and F1-score of 0.986. This shows that the Naïve Bayes method has good ability in dealing with missing values and AdaBoost algorithm were managed to improve the Naïve Bayes performance by increasing the accuracy..

Keywords: AdaBoost, CKD, Missing Value, Naïve Bayes

1. PENDAHULUAN

Ginjal dirancang untuk menjalankan sejumlah fungsi yang penting, yaitu mempertahankan lingkungan ekstraselular melalui ekskresi produk limbah dan elektrolit yang dan memproduksi hormon untuk mengontrol hemodinamik ginjal, merangsang produksi sel darah merah, dan mempertahankan homeostasis tulang normal [1].

Penyakit ginjal kronik (*chronic kidney disease*, CKD) merupakan penyakit yang menyerang ginjal dan menjadi masalah kesehatan di masyarakat yang terus meningkat. Menurut studi dari *Global Burden of Disease* tahun 2010, CKD berada pada peringkat 27 dalam daftar jumlah total kematian di seluruh dunia pada tahun 1990, kemudian naik menjadi peringkat 18 pada tahun 2010 [2].

Teknik data mining dengan klasifikasi merupakan salah satu metode yang paling sering digunakan dalam memecahkan masalah prediksi secara umum. *Naive bayes classifier* merupakan salah satu dari sepuluh algoritma klasifikasi yang populer yang memiliki keunggulan antara lain mudah untuk dibangun dan tidak membutuhkan skema parameter yang rumit sehingga dapat diaplikasikan dalam dataset yang besar [3]. Keunggulan lainnya yaitu ketika menghadapi atribut yang banyak memiliki *missing value*, *naive bayes* masih bisa menghasilkan akurasi yang baik

dibandingkan *classifier* lain seperti KNN dan *desicion tree* C4.5 [4].

Saputra [5] melakukan komparasi algoritma data mining C4.5, *naive bayes*, *neural network*, dan *logistic regression* untuk memprediksi penyakit *tuberculosis*. Dari hasil evaluasi dan validasi pada penelitian ini, algoritma *naive bayes* memiliki nilai akurasi dan AUC paling tinggi dibandingkan algoritma lain yang dikomparasikan.

Algoritma *boosting* merupakan algoritma iteratif yang memberikan bobot yang berbeda pada distribusi training data pada setiap iterasi. Bobot akan ditambahkan pada setiap contoh-contoh kesalahan klasifikasi dan akan diturunkan pada setiap klasifikasi yang benar. Penggunaan *adaboost* dalam meningkatkan akurasi dimanfaatkan oleh Korada, et al [6]. Pengujian dilakukan terhadap *Maize Expert System* yang merupakan sistem pakar untuk mendeteksi penyakit pada tanaman jagung. Dengan menggunakan *Naive bayes* sebagai *base learner* dan *Adaboost* untuk metode *boosting*, hasil akurasi yang diperoleh meningkat sebanyak 33% sehingga kesalahan klasifikasi dapat dikurangi.

Penggunaan *naive bayes* dan *Adaboost* untuk klasifikasi teks berupa sentimen *review* restoran yang dilakukan oleh Utami dan Satria Wahono [7]. Hasil dari penelitian ini yaitu algoritma *Adaboost* mampu meningkatkan akurasi metode *naive bayes* dan *information gain* yang semula hanya 70% menjadi 99,5% atau naik sekitar 29,5%.

Dalam penelitian ini akan dilakukan pengujian dan pengukuran kinerja metode *Naïve Bayes Classifier* dan algoritma *AdaBoost* dalam memprediksi penyakit ginjal kronik berdasarkan dataset yang masih memiliki *missing value* dan dataset yang sudah dilakukan pengisian *missing value*.

2. DATA MINING

Data mining merupakan suatu proses penemuan pola dan pengetahuan atau informasi yang menarik dari data dengan jumlah yang besar. *Data mining* memiliki beberapa nama alternatif seperti *knowledge mining from data*, *knowledge extraction*, *data/pattern analysis*, *data archaeology*, *data dredging*, dan salah satu terminologi yang populer disebut dengan *knowledge discovery from data* (KDD). *Data mining* sebagai rangkaian suatu proses dapat dibagi menjadi beberapa tahap [8]:

1. *Data cleaning* (pembersihan data). Bertujuan untuk membuang data yang tidak konsisten dan *noise*. Termasuk didalamnya penanganan terhadap *missing*

value yang terdapat dalam data (bisa diisi dengan nilai yang paling sesuai atau diabaikan begitu saja) [9].

2. *Data integration* (integrasi data). Proses dimana terjadi penggabungan data dari berbagai macam sumber.
3. *Data selection* (pemilihan data). Proses dimana hanya data yang relevan untuk dianalisis yang diambil dari database.
4. *Data transformation* (transformasi data). Data ditransformasi dan dikonsolidasikan menjadi suatu format yang sesuai untuk digali dengan menjalankan operasi penjumlahan atau agregasi.
5. *Data mining* (penggalian data). Suatu proses dengan menerapkan metode cerdas untuk ekstraksi pola data.
6. *Pattern evaluation* (evaluasi pola). Bertujuan untuk mengidentifikasi mana yang menjadi pola yang benar-benar menarik berdasarkan “penilaian ketertarikan”.
7. *Knowledge presentation* (presentasi pengetahuan). Dimana proses visualisasi dan teknik representasi pengetahuan digunakan untuk menyajikan pengetahuan atau informasi yang telah digali kepada pengguna.

Beberapa metode *data mining* yang banyak digunakan antara lain: klasifikasi dimana dipelajari suatu pola-pola dari data yang telah lalu (kumpulan informasi, variabel, fitur) dengan tujuan untuk menempatkan *instance* baru (dengan label yang belum diketahui) ke grup atau kelasnya yang sesuai. Selanjutnya *cluster analysis* digunakan untuk mengklasifikasi suatu barang, kejadian, atau konsep kedalam kelompok yang sama (*cluster*) atau memiliki karakteristik yang mirip. Berbeda dengan klasifikasi, *clustering* ini memiliki label kelas yang belum diketahui. Kemudian *association rule* yang mining bertujuan untuk menemukan hubungan yang menarik (afinitas) antara variabel (item) dalam *database* besar. Dikarenakan kesuksesan penggunaannya dalam menangani masalah bisnis, sehingga biasa disebut dengan *market-basket analysis* [9].

3. KLASIFIKASI

Klasifikasi adalah suatu bentuk analisis data yang mengekstrak model yang menggambarkan kelas-kelas data. Suatu *classifier*, atau model klasifikasi (*classification model*), memprediksi label berkategori (*classes*) [8]. Tahapan dalam klasifikasi dibagi menjadi berikut:

1. Tahap pembelajaran. Disini tiap *record* data dari *training set* yang nilai atributnya saling berhubungan dianalisis dengan menggunakan suatu algoritma klasifikasi sehingga dapat menghasilkan suatu model pembelajaran atau *classifier* yang tepat.
2. Tahap klasifikasi. Pada tahap ini data tes digunakan untuk mengetahui ketepatan atau akurasi dari aturan-aturan klasifikasi yang berlaku pada model yang dihasilkan. Apabila tingkat akurasi yang diperoleh sesuai dengan nilai yang ditentukan, maka model

tersebut dapat digunakan untuk mengklasifikasikan data *record* lain yang data kelasnya belum diketahui atau diujikan (dalam pembelajaran mesin, data tersebut juga dikenal sebagai data *unknown* atau *previously unseen data*).

4. NAIVE BAYES CLASSIFIER

Naive Bayes Classifier merupakan salah satu teknik klasifikasi yang menggunakan metode probabilitas sederhana berdasarkan teorema bayes dengan asumsi ketidaktergantungan (*independent*) yang tinggi. Teorema bayes yang digunakan sebagai dasar algoritma ini merupakan suatu teori yang dikemukakan oleh ilmuwan Inggris Thomas Bayes yang memprediksi probabilitas dimasa depan berdasarkan pengalaman dimasa sebelumnya. Adapun rumus dari *naive bayes* sebagai berikut:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \dots\dots\dots(4.1)$$

Dimana X merepresentasikan vektor masukan yang berisikan fitur. Sedangkan C_i merepresentasikan label kelas. Dengan asumsi bahwa nilai variabel dalam tiap kelas saling independen yang kuat (*naive*) satu dengan yang lainnya dan nilai $P(X)$ dalam setiap kelas bernilai konstan, maka model persamaan *Naive Bayes* untuk klasifikasi dapat disederhanakan menjadi:

$$P(C_i|X) = \prod_{k=1}^n P(x_k|C_i)P(C_i) \dots\dots\dots(4.2)$$

Jika dalam suatu atribut ternyata memiliki nilai kontinyu, maka perhitungan probabilitasnya memakai distribusi *Gauss* dengan mean μ dan standar deviasi σ , yang dirumuskan sebagai berikut:

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \dots\dots\dots(4.3)$$

Sehingga nilai $P(x_k|C_i)$ berubah menjadi:

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \dots\dots\dots(4.4)$$

Beberapa studi mengenai algoritma klasifikasi menunjukkan bahwa *Naive Bayes Classifier* memiliki performa yang sebanding dengan *decision tree* dan *neural network classifiers* tertentu. Selain itu, metode ini juga menunjukkan akurasi dan kecepatan yang tinggi ketika digunakan dalam basis data yang berukuran besar [8].

5. ADABOOST

Algoritma *AdaBoost* pertama kali diperkenalkan pada tahun 1995 oleh Freund dan Schapire, telah banyak memecahkan berbagai masalah praktis dari algoritma *boosting* sebelumnya [10].

Pada dasarnya metode *boosting* (*AdaBoost*) dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi dari suatu model, tetapi hasil klasifikasi atau prediksi yang dipilih adalah model yang memiliki nilai bobot paling besar. Dengan demikian, setiap model yang dibangun tersebut memiliki atribut berupa nilai bobot.

Teknik pembobotan dalam algoritma *AdaBoost* adalah sebagai berikut [3]:

Input:
 Dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$;
 Algoritma pembelajaran dasar \mathcal{L} ;
 Jumlah iterasi T

Proses:
 Inisialisasi nilai bobot
 $D_1(i) = \frac{1}{m}$ untuk $i = 1, \dots, m$
 for $t = 1, \dots, T$:
 Latih weak learn h_t dari \mathcal{D} dengan menggunakan distribusi D_t
 $h_t = \mathcal{L}(\mathcal{D}, D_t)$;
 Hitung error dari h_t
 $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$;
 Menentukan bobot dari h_t
 $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$;
 $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(x_i) \neq y_i \end{cases}$
 Update distribusi, dimana Z_t sebuah faktor normalisasi yang mengaktifkan D_{t+1} menjadi distribusi
 $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
 end for.

Output:
 $H_{(x)} = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$

Setelah hipotesis lemah h_t diperoleh, *AdaBoost* memilih parameter α_t seperti pada persamaan diatas. Secara intuitif, α_t mengukur bobot yang akan diberikan pada h_t . Perlu diketahui bahwa $\alpha_t \geq 0$ jika $\epsilon_t \leq 0,5$ (dengan asumsi tidak ada kehilangan dalam keumuman), sehingga nilai α_t menjadi lebih besar dan nilai ϵ_t menjadi semakin kecil. Distribusi D_t kemudian diperbarui nilainya (*update*) dengan menggunakan aturan seperti pada persamaan diatas. Pengaruh dari aturan ini yaitu untuk meningkatkan bobot dari contoh yang diklasifikasikan salah (*misclassified*) oleh h_t dan untuk mengurangi bobot pada contoh yang sudah diklasifikasi benar. Hipotesis akhir H adalah voting terbanyak terbobot dari hipotesis lemah T dimana α_t merupakan bobot yang diberikan kepada h_t [10].

6. METODOLOGI

6.1 Pengumpulan Data

Penelitian ini dimulai dengan pengumpulan data berupa dataset yang diperoleh dari UCI Machine Learning Repository [11] berupa data stadium awal penyakit ginjal kronik terhadap orang India.

6.2 Analisis dan Perancangan

6.2.1 Proses Data Cleaning

Proses *data cleaning* dilakukan sebelum data dibagi menjadi *data training* dan *data testing*, yaitu dengan *replace* data yang hilang atau kosong (*missing value*) dengan mean untuk atribut numerik dan modus untuk atribut nominal. Mean adalah nilai rata-rata dari suatu informasi, sedangkan modus dalam suatu data adalah nilai yang paling sering muncul.

6.2.2 Pelatihan Naive Bayes dan AdaBoost

Proses pelatihan dengan menghitung probabilitas masing-masing atribut yang menyertai berdasarkan kelasnya. Untuk menghindari terjadinya probabilitas yang bernilai 0, maka dilakukan proses *smoothing* dengan menggunakan rumus *laplacian correction* seperti dibawah.

$$P(X = x_k | C_i) = \frac{N_{ik} + p}{N_i + p \cdot N_k} \dots \dots \dots (6.1)$$

Dimana N_{ik} merupakan jumlah kejadian yang muncul di kolom k dari baris i pada *data training*, N_i adalah jumlah kemunculan kejadian pada *data training* dari kelas C_i , sedangkan N_k adalah jumlah kejadian yang muncul pada kolom k yang terdapat dalam *data training*, dan p merupakan *arbitrary probability*, disini nilai $p = 1$.

Kemudian setelah dilakukan pelatihan dengan menggunakan metode *naive bayes*, pelatihan selanjutnya dengan menggunakan metode *boosting AdaBoost*. Pada dasarnya, metode *boosting* ini bertujuan untuk meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi dari suatu model, tetapi hasil klasifikasi atau prediksi yang dipilih adalah model yang memiliki atribut nilai bobot terbesar. Jumlah perulangan yang dilakukan dalam percobaan menggunakan metode *AdaBoost* dibatasi sebanyak 10 kali. Pembagian *data training* dan *testing* untuk pelatihan menggunakan validasi *10-fold cross validation* yang kemudian hasil pengukuran kinerja tersebut saling dibandingkan antara metode *Naive Bayes* dan metode *Naive Bayes* yang telah dilakukan *boosting* menggunakan *AdaBoost*.

6.2.3 Proses Cross Validation

Dalam *k-fold cross validation*, data awal dibagi secara acak menjadi k subset saling eksklusif (berdiri sendiri) atau disebut dengan *fold*, dari ukuran yang kira-kira sama.

Proses *training* dan *testing* dilakukan sebanyak k kali eksperimen. Pada setiap iterasi, dimana satu partisi digunakan sebagai *data testing* dan memanfaatkan sisa partisi lainnya sebagai *data training*. Sebagai contoh apabila terdapat subset D_1, D_2, \dots, D_k , maka untuk iterasi pertama, subset D_1 digunakan sebagai *data testing* sedangkan sisanya D_2, \dots, D_k digunakan sebagai *data training* untuk memperoleh model pertama, begitu juga untuk iterasi kedua, maka D_2 digunakan untuk *data testing* dan sisanya D_1, D_3, \dots, D_k digunakan untuk *data training*, begitu seterusnya sampai subset terakhir D_k .

Hasil perkiraan akurasi *cross validation* diperoleh dari jumlah keseluruhan klasifikasi yang benar dari iterasi k , dibagi dengan jumlah total *tuple* dalam data awal. Secara

umum, *stratified 10-fold cross validation* dianjurkan untuk memperkirakan akurasi (meskipun daya komputasi memungkinkan untuk menggunakan *fold* yang lebih banyak) dikarenakan bias dan variansi yang relatif rendah [8].

6.3 Pengembangan Aplikasi

Ruang lingkup perangkat yang digunakan dalam mengimplementasikan sistem adalah sebuah platform komputer berbasis Intel® Core™ i5-2410M CPU @ 2.3 GHz, RAM 2GB, dan sistem operasi Linux Mint 17.3 Cinnamon 64-bit. Sedangkan lingkungan pengembangan aplikasi dengan bahasa pemrograman Python 2.7.6. dan *library* WEKA 3.9.0.

6.4 Pengujian dan Analisis Hasil

Pengujian dilakukan untuk mengukur keakuratan hasil dari tiap model yang diusulkan. Pengukuran kinerja ini menggunakan perhitungan nilai *accuracy*, *precision*, *recall*, dan *F₁-score* (disebut juga sebagai *F-score* atau *F-measure*) untuk mengetahui seberapa besar perbedaan antara metode *naive bayes* dengan metode *naive bayes* dan *AdaBoost* dalam menangani kasus prediksi penyakit ginjal kronik. *Accuracy* didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Definisi dari *precision* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data sebenarnya, sedangkan *recall* adalah proporsi dari kasus positif kejadian sebenarnya yang diprediksi positif benar [12]. *F₁-score* atau *F-measure* merupakan rata-rata harmonik dari nilai *precision* dan *recall* [8], dimana *F₁-score* ini mencapai nilai terbaik pada 1 dan terburuk pada 0 (*range* penilaian dari 1-0). Tabel 1 menunjukkan model *confusion matrix* untuk dua kelas.

Tabel 1. Confusion matrix untuk 2 kelas

		Kelas Prediksi	
		yes	no
Kelas Sebenarnya	yes	True Positive (TP)	False Negative (FN)
	no	False Positive (FP)	True Negative (TN)

Berdasarkan tabel 1 maka rumus untuk masing-masing nilai adalah:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(6.2)$$

$$precision = \frac{TP}{TP+FP} \dots\dots\dots(6.3)$$

$$recall = \frac{TP}{TP+FN} \dots\dots\dots(6.4)$$

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \dots\dots\dots(6.5)$$

True positive adalah jumlah *record* positif yang diklasifikasikan sebagai positif oleh *classifier*. *True negative* adalah jumlah *record* negatif yang diklasifikasikan

sebagai negatif oleh *classifier*. *False positive* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, sedangkan *false negative* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif oleh *classifier*.

Pengujian dilakukan melalui dua skenario, yaitu:

1. Pengujian terhadap dataset tanpa melalui proses penggantian *missing value*
2. Pengujian terhadap dataset melalui proses pengisian *missing value* (mean dan modus)

7. HASIL DAN PEMBAHASAN

7.1 Gambaran Umum Dataset CKD

Dataset yang digunakan berjumlah 400 *record* data berisikan 25 atribut dengan pembagian sebanyak 250 *record* (62,5%) sebagai kelas CKD dan 150 *record* (37,5%) sebagai kelas NotCKD. Total sebanyak 1012 *missing value* terdapat pada data yang diteliti. Parameter dataset yang diteliti dapat dilihat pada tabel 2.

Tabel 2. Parameter dataset

Atribut	Tipe	Keterangan
Age	Numerik	Umur pasien
Blood Pressure	Numerik	Tekanan darah
Specific Gravity	Nominal	Kepekatan urine
Albumin	Nominal	Kadar albumin urine
Sugar	Nominal	Kadar gula urine
Red Blood Cells	Nominal	Jumlah sel darah merah di urine
Pus Cell	Nominal	Nanah/pus di urine
Pus Cell Clumps	Nominal	Gumpalan pus di urine
Bacteria	Nominal	Bakteri di urine
Blood Glucose Random	Numerik	Jumlah gula darah yang diambil acak
Blood Urea	Numerik	Kadar urea di darah
Serum Creatinine	Numerik	Pengukuran kreatinin dalam darah
Sodium	Numerik	Pengukuran sodium di darah
Potassium	Numerik	Pengukuran potasium di darah
Hemoglobin	Numerik	Pengukuran hemoglobin
Packed Cell Volume	Numerik	Pengukuran hematocrit
White Blood Cell Count	Numerik	Jumlah sel darah putih
Red Blood Cell Count	Numerik	Jumlah sel darah merah
Hypertension	Nominal	Riwayat hipertensi

Diabetes Mellitus	Nominal	Riwayat diabetes mellitus
Coronary Artery Disease	Nominal	Riwayat penyakit arteri koroner
Appetite	Nominal	Nafsu makan
Pedal Edema	Nominal	Pembengkakan pada kaki akibat cairan yang berlebih pada jaringan tubuh
Anemia	Nominal	Riwayat anemia
Class	Nominal	Output kelas yang dicari; CKD (hasil positif) atau NotCKD (hasil negatif)

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$F_1 = \frac{2 \times 230}{2 \times 230 + 0 + 20}$$

$$F_1 = \frac{460}{480} = 0,958$$

Percobaan selanjutnya yaitu mengkombinasikan metode *naive bayes* dengan algoritma *AdaBoost*. Pengujian dilakukan dengan melakukan optimalisasi perulangan (*iteration*) sebanyak 10 kali. Tabel 4 menunjukkan hasil *confusion matrix* dari percobaan dengan *AdaBoost* yang dilakukan.

Tabel 4. Confusion matrix AdaBoost dengan data ber-missing value

	Prediksi CKD	Prediksi NotCKD	Class Recall
Aktual CKD	242	8	0,968
Aktual NotCKD	0	150	1
Class Precision	1	0,949	

7.2 Hasil Pengujian

7.2.1 Pengujian Naive Bayes dan AdaBoost dengan Dataset Ber-missing Value

Data yang digunakan dalam penelitian ini merupakan data asli tanpa ada proses *data cleaning* berupa penggantian *missing value*. Dari jumlah data sebanyak 400 data klasifikasi dengan kelas CKD sebanyak 250 *record* dan NotCKD sebanyak 150 *record*, data yang diprediksi sesuai sebagai CKD sebanyak 230 *record*, sedangkan data yang diprediksi CKD tetapi kenyataannya NotCKD sebanyak 20 *record*. Kemudian untuk data yang diprediksi NotCKD dan semuanya sesuai berjumlah 150 *record*, data yang diprediksi NotCKD tetapi kenyataannya CKD berjumlah 0 *record*. Hasil *confusion matrix* dari percobaan yang dilakukan dapat dilihat pada tabel 3.

Tabel 3. Confusion matrix model naive bayes dengan data ber-missing value

	Prediksi CKD	Prediksi NotCKD	Class Recall
Aktual CKD	230	20	0,92
Aktual NotCKD	0	150	1
Class Precision	1	0,882	

Nilai akurasi dan F_1 -score dari *confusion matrix* diatas menjadi sebagai berikut:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{230 + 150}{230 + 150 + 0 + 20}$$

$$accuracy = \frac{380}{400} = 0,95$$

Dari jumlah total 400 data dengan klasifikasi kelas CKD 250 *record* dan kelas NotCKD 150, pada iterasi ke-5 dari algoritma *AdaBoost* menghasilkan data yang diprediksi tepat sebagai CKD sebanyak 242, sedangkan data yang diprediksi CKD, tetapi pada kenyataannya merupakan NotCKD sebanyak 8. Kemudian data yang diprediksi tepat sebagai NotCKD sebanyak 150 dan tidak ada yang diprediksi salah untuk kelas NotCKD. Sehingga akurasi dan F_1 -score dari model *confusion matrix* diatas adalah sebagai berikut:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{242 + 150}{242 + 150 + 0 + 8}$$

$$accuracy = \frac{392}{400} = 0,98$$

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$F_1 = \frac{2 \times 242}{2 \times 242 + 0 + 8}$$

$$F_1 = \frac{484}{492} = 0,984$$

7.2.2 Pengujian Naive Bayes dan AdaBoost dengan Pengisian Missing Value

Percobaan berikutnya yaitu dengan melakukan *data cleaning* berupa pengisian atribut yang memiliki *missing*

value dengan mean untuk data numerik dan modus untuk data nominal.

Pengukuran terhadap 400 data klasifikasi, metode naive bayes menghasilkan data yang diprediksi tepat sebagai CKD sejumlah 229, dan data yang diprediksi sebagai CKD akan tetapi ternyata termasuk kedalam kelas NotCKD sebanyak 21. Kemudian data yang diprediksi tepat sebagai kelas NotCKD sebanyak 149, dan data yang diprediksi sebagai NotCKD tetapi masuk kedalam kelas CKD sebanyak 1. Hasil *confusion matrix* pengukuran ini dapat diamati pada tabel 5.

Tabel 5. Confusion matrix naive bayes data dengan pengisian missing value

	Prediksi CKD	Prediksi NotCKD	Class Recall
Aktual CKD	229	21	0,916
Aktual NotCKD	1	149	0,993
Precision Class	0,996	0,876	

Berdasarkan *confusion matrix* diatas, maka nilai akurasi dan F_1 -score metode ini yaitu:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{229 + 149}{229 + 149 + 1 + 21}$$

$$accuracy = \frac{378}{400} = 0,945 = 94,5\%$$

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$F_1 = \frac{2 \times 229}{2 \times 229 + 1 + 21}$$

$$F_1 = \frac{458}{480} = 0,954$$

Kemudian percobaan berikutnya yaitu melakukan uji coba terhadap dataset dengan menerapkan algoritma AdaBoost dengan optimalisasi perulangan sebanyak 10 kali. Hasil *confusion matrix* dari pengujian yang dilakukan dapat diamati pada tabel 6.

Tabel 6. Confusion matrix adaboost iterasi ke-4

	Prediksi CKD	Prediksi NotCKD	Class Recall
Aktual CKD	244	6	0,976
Aktual NotCKD	1	149	0,993
Class Precision	0,996	0,961	

Penjabaran mengenai akurasi dan F_1 -score berdasarkan *confusion matrix* diatas adalah sebagai berikut:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{244 + 149}{244 + 149 + 1 + 6}$$

$$accuracy = \frac{393}{400} = 0,9825$$

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$F_1 = \frac{2 \times 244}{2 \times 244 + 1 + 6}$$

$$F_1 = \frac{488}{495} = 0,986$$

7.3 Analisis Hasil Pengujian

Analisis pengujian dilakukan dengan membandingkan hasil yang diperoleh dari tiap metode yang diusulkan. Hasil uji dataset asli tanpa mengalami penggantian *missing value* ketika diuji dengan menggunakan *Naïve Bayes* (NB) menghasilkan akurasi sebesar 0,95, nilai *precision* sebesar 1, *recall* sebesar 0,958, dan F_1 -score sebesar 0,958. Sedangkan ketika dataset diubah melalui proses *data cleaning* dengan penggantian *missing value* berupa mean untuk data numerik dan modus untuk data nominal, maka hasil pengujian dengan metode *Naïve Bayes* (NBmv) menunjukkan akurasi sebesar 0,945, nilai *precision* sebesar 0,996, *recall* sebesar 0,916, dan F_1 -score sebesar 0,954. Dari hasil tersebut dapat diamati ketika dataset dilakukan proses *data cleaning*, nilai akurasi, *precision*, *recall*, dan F_1 -score justru mengalami penurunan, meskipun nilainya tidak begitu besar.

Ketika dilakukan pengisian *missing value* terjadi kesalahan prediksi untuk kelas yang seharusnya termasuk ke dalam CKD, diprediksi menjadi NotCKD sebanyak 21 data. Kesalahan prediksi juga terjadi untuk data yang seharusnya masuk kedalam kelas NotCKD, tetapi diprediksi menjadi CKD sebanyak 1 data. Berbeda hasilnya

ketika dilakukan pengujian menggunakan dataset asli tanpa ada pengisian *missing value*, kesalahan prediksi untuk kelas yang seharusnya masuk kedalam CKD, tetapi diprediksi sebagai NotCKD sebanyak 20 data. Sedangkan untuk kelas NotCKD tidak terjadi kesalahan prediksi. Hal ini menunjukkan bahwa data yang semula memiliki *missing value* pada atributnya setelah dilakukan pengisian terhadapnya, membuat *classifier* memprediksi hasil yang berbeda. Sehingga dengan adanya pengisian *missing value* tersebut berpengaruh terhadap hasil prediksi dari *classifier*. Penurunan yang terjadi dikarenakan penggunaan metode pengisian *missing value* dengan nilai konstan yaitu mean dan modus. Pengisian menggunakan nilai konstan memiliki keunggulan dalam penentuan nilai harapan yang secara relatif memiliki kestabilan yang tinggi. Sedangkan kelemahannya yaitu dalam ragam yang diperoleh dengan metode ini tidak sesuai dengan data yang sebenarnya dan korelasi antar peubah dapat memberikan informasi yang menyesatkan. Adapun metode pengisian *missing value* lain yang dapat digunakan sebagai alternatif yang lebih baik yaitu metode *hot deck* seperti penelitian yang dilakukan oleh Hendrawati [13].

Kemudian analisis selanjutnya membandingkan metode *Naïve Bayes* yang di-boosting dengan algoritma *AdaBoost*. Dari pengujian dengan optimalisasi perulangan sebanyak 10 kali, didapatkan hasil untuk algoritma *AdaBoost* menggunakan dataset asli tanpa proses penggantian *missing value* (NBAd) dengan nilai akurasi sebesar 0,98, *precision* sebesar 1, nilai *recall* sebesar 0,968, dan *F₁-score* sebesar 0,984. Sedangkan hasil uji coba algoritma *AdaBoost* dataset yang telah mengalami penggantian *missing value* (NBAdmv), nilai akurasi yang diperoleh sebesar 0,9825, *presicion* sebesar 0,996, *recall* sebesar 0,976, dan *F₁-score* sebesar 0,986.

Untuk hasil pengujian dengan menggunakan dataset asli, kesalahan prediksi untuk data yang seharusnya masuk ke dalam kelas CKD, tetapi diprediksi sebagai NotCKD sebanyak 8 data, dan tidak ada yang diprediksi salah untuk kelas NotCKD. Namun, ketika dilakukan pengisian *missing value* data yang seharusnya diprediksi sebagai kelas CKD, tetapi masuk kedalam kelas NotCKD sebanyak 6 dan untuk data yang seharusnya masuk kedalam kelas NotCKD tetapi diprediksi sebagai CKD sebanyak 1 data. Hal ini juga menunjukkan bahwa metode pengisian *missing value* yang digunakan sangat berpengaruh pada hasil prediksi dari suatu *classifier*. Rekapitulasi hasil pengujian dapat dilihat pada tabel 7.

Tabel 7. Rekapitulasi hasil pengujian

	Accuracy	Precision	Recall	F ₁
NB	0,95	1	0,92	0,958
NBmv	0,945	0,996	0,916	0,954
NBAd	0,98	1	0,968	0,984
NBAdmv	0,9825	0,996	0,976	0,986

Dari tabel 7 diatas dapat disimpulkan bahwa penambahan algoritma *AdaBoost* secara umum mampu memberikan hasil yang baik dengan meningkatkan nilai akurasi, *recall*, dan *F₁-score*. Nilai *precision* tidak mengalami perubahan untuk uji coba berdasarkan dataset asli tanpa proses data cleaning (NB dan NBAd) maupun untuk dataset yang melalui proses *data cleaning* (NBmv dan NBAdmv). Peningkatan nilai akurasi terbesar terjadi ketika dilakukan pengujian terhadap dataset yang melalui proses *data cleaning*, dengan kenaikan sebesar 0,0375 yaitu yang semula (NBmv) 0,945 menjadi 0,9825 (NBAdmv). Sedangkan untuk pengujian menggunakan dataset asli (NB dan NBAd) mengalami kenaikan sebesar 0,03 yaitu dari 0,95 meningkat menjadi 0,98.

8. KESIMPULAN DAN SARAN

Dari hasil penelitian dapat disimpulkan bahwa metode *Naïve Bayes Classifier* cukup sukses dalam mengklasifikasikan data CKD. Meskipun dataset yang diteliti memiliki banyak *missing value*, *Naïve Bayes* dapat memperoleh nilai akurasi sebesar 0,95 dan *F₁-score* sebesar 0,958. Sedangkan apabila dataset tersebut digantikan atau diisi *missing value*-nya dengan mean dan modus, hasil akurasi metode *Naïve Bayes* ini justru mengalami penurunan sebesar 0,005 yaitu menjadi 0,945. Untuk nilai *F₁-score* hanya mengalami penurunan yang sangat rendah menjadi 0,954. Hal ini juga membuktikan bahwa metode *Naïve Bayes* cukup handal menghadapi data set yang memiliki banyak *missing value* sesuai dengan penelitian yang dilakukan oleh Liu, Lei, dan Wu [4].

Penggunaan metode *boosting (AdaBoost)* juga terbukti mampu meningkatkan performa dari metode *Naïve Bayes*. Dari 10 kali perulangan yang diujicobakan, nilai akurasi untuk algoritma *AdaBoost* pada dataset asli sebesar 0,98 dan *F₁-score* sebesar 0,984. Sedangkan untuk pengujian dengan dataset yang sudah melalui proses penggantian *missing value* nilai akurasi yang diperoleh meningkat menjadi 0,9825 dan *F₁-score* sebesar 0,986.

Adapun saran untuk penelitian selanjutnya adalah:

1. Melakukan klasifikasi dengan menggunakan metode yang lain seperti *desicion tree*, *neural network*, atau *logistic regression*.
2. Melakukan penelitian dengan memanfaatkan metode *ensemble* yang lain seperti *bagging* maupun *random forest*.
3. Melakukan pengujian terhadap keterkaitan antara variabel yang digunakan.
4. Melakukan penelitian dengan menggunakan metode pengisian *missing value* yang lain seperti *hot deck*, *regression*, *expectation maximisation*, dan *multiple imputation*.

9. DAFTAR PUSTAKA

- [1] Reilly Jr., R.F., Perazella, M.A., (2005). *Nephrology In 30 Days*. Singapore: McGraw-Hill Education (Asia),.
- [2] Jha, V., et al. (2013). Chronic kidney disease: Global dimension and perspectives. *The Lancet* 382. 260-272.
- [3] Wu, X., et al. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*.
- [4] Liu, Peng., Lei, Lei., Wu, Naijun. (2005). A Quantitative Study of the Effect of Missing Data in Classifier. *Computer and Information Technology CIT 2005*. 28-33.
- [5] Saputra, R.A. (2014). Komparasi Algoritma Klasifikasi Data Mining Untuk Memprediksi Penyakit Tuberculosis (TB): Studi Kasus Puskesmas Karawang. *Proceedings Seminar Nasional Inovasi dan Tren SNIT*. 1-8.
- [6] Korada, N.K., Kumar, N.S.P., Deekshitulu, Y.V.N.H. (2012). Implementation of Naive Bayesian Classifier and Ada-Boost Algorithm Using Maize Expert System. *International Journal of Information Sciences and Techniques (IJIST)* 2. 63-75.
- [7] Utami, L.D., Satria Wahono, R. (2015). Integrasi Metode Information Gain Untuk Seleksi Fitur Dan Adaboost Untuk Mengurangi Bias Pada Analisis Sentimen Review Restoran Menggunakan Algoritma Naive Bayes. *Journal of Intelligent Systems* 1. 120-126.
- [8] Han, J., Kamber, M., Pei, J. (2012). *Data Mining Concepts and Techniques, 3rd ed*. Watham: Morgan Kauffman.
- [9] Turban, E., Sharda, R., Delen, D. (2011). *Data Mining Methods, in: Decision Support and Business Intelligence Systems*. New Jersey: Pearson Education. 216-228.
- [10] Freund, Y., Schapire, R.E. (1999). A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence* 14. 771-780.
- [11] Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
- [12] Powers, D.M.W. (2011). Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies* 2. 37-63.
- [13] Hendrawati, T., 2015. Kajian Metode Imputasi Dalam Menangani Missing Data. *Prosiding Seminar Nasional Matematika dan Pendidikan Matematika*. 637-642.