e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Perbandingan Algoritma *Random Forest*, *Naïve Bayes*, Dan *Decision Tree* Dengan *Oversampling* Untuk Klasifikasi Bakteri *E. Coli*

Alvita I. Kusumarini [1)], Pandu A. Hogantara[2)], Muammar Fadhlurohman[3)], Nurul Chamidah[4)]
Program Studi Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya No.1, Pd. Labu, Kec. Cilandak, Jakarta Selatan 12450
alvitaizana@upnvj.ac.id[1)], pandu@upnvj.ac.id[2)], muammar@upnvj.ac.id[3)]
Co-Author: nurul.chamidah@upnvj.ac.id[4)]

Abstrak. Berbagai jenis bakteri dapat dibedakan berdasarkan klasifikasi bakteri yang juga tergantung varietas dari bakteri tersebut. Suatu hal penting dalam mengidentifikasi bakteri adalah melalui karakteristik yang dapat diamati pada bakteri tersebut yang memanfaatkan ciri bentuk serta melalui perwarnaan sifat dari bakteri itu sendiri. Karakteristik yang dapat diamati dari suatu bakteri dapat diklasifikasi dengan memanfaatkan algoritma-algoritma klasifikasi. Pada penelitian ini dataset yang digunakan yaitu dataset E. coli yang merupakan data sekunder. Kemudian dilakukan praproses data dengan menghilangkan kolom ID yang sifatnya unik di tiap sampel data. Lalu membagi data terlebih dahulu menjadi 70% data latih dan 30% data uji. Data latih akan dibagi lagi menggunakan *k-fold cross validation* dengan nilai $k = 10$ yang dimana akan menghasilkan data latih sebanyak 211 sampel dan data validasi sebanyak 24 sampel. Untuk membuktikan algoritma dengan akurasi terbaik, maka proses klasifikasi dilakukan menggunakan tiga model algoritma sebagai perbandingan yaitu algoritma *random forest, Naïve Bayes* dan *decision tree* yang sebelumnya juga pernah dilakukan dengan penelitian dan dataset yang berbeda. Pada proses pelatihan model juga akan dilakukan proses *hyperparameter tuning*, yaitu proses pencarian parameter terbaik untuk mendapatkan hasil akurasi yang terbaik. Hasil yang didapatkan menjelaskan bahwa algoritma *random forest* memiliki akurasi tertinggi yaitu sebesar 84%.

Kata Kunci: Klasifikasi, *Random Forest, Naïve Bayes, Decision Tree, E. coli, K-Fold Cross Validation*

1 PENDAHULUAN

Seorang Dokter dari Jerman pada tahun 1884 melakukan pengembangan teknik yang bertujuan untuk menyelidiki perbedaan jenis bakteri melalui ketebalan lapisan penyusun dinding sel pada bakteri atau yang bisa disebut dengan peptidoglikan dengan sistem pewarnaan. Pada percobaannya, bakteri diberi zat warna violet dan yodium yang nantinya dilakukan pembilasan dengan alkohol dan diberi zat warna merah. Apabila bakteri mengalami perubahan warna menjadi berwarna ungu maka bakteri tersebut dikelompokkan ke dalam bakteri gram-positif. Adapun bakteri yang pada waktu tertentu dapat mengalami perubahan yang semula bakteri tersebut merupakan bakteri gram-positif hingga nantinya berubah menjadi bakteri gram-negatif, yang dimana bakteri tersebut dikenal sebagai bakteri gram-variabel. Dinding sel bakteri gram-variabel dapat menyerap warna merah dan mempunyai lapisan penyusun dinding sel yang tidak tebal, bakteri ini disebut dengan bakteri gram-negatif. Lapisan penyusun dinding selnya berada pada ruang periplasmik antara membran plasma dengan membran luar. Contoh bakteri gram-negatif adalah *Leguminosarum, Salmonella, Aeruginosa, Rhizobium, Helicobacter Pylori, Azotobacter, Escherichia Coli*, dan *Neisseria Gonorrchoeae* [1].

Pada penelitian sebelumnya [2] telah dilakukan proses klasifikasi bakteri gram-negatif menggunakan algoritma Naïve Bayes pada dataset E. coli [3] yang terdiri dari 7 fitur dan 8 kelas yang dievaluasi dengan menganalisa tingkat akurasi pada dataset. Pada penelitian tersebut dimaksudkan untuk mengetahui jenis bakteri gram-negatif dengan perhitungan Yes atau No. Evaluasi pada model yang berhasil terbentuk diukur menggunakan *confusion matrix*. Hasil akurasi yang didapatkan dari klasifikasi algoritma Naïve Bayes yaitu 80,93%.

Kemudian penelitian terkait klasifikasi data juga telah dilakukan [4] menggunakan algoritma *Decision Tree* pada data peserta didik. Metodologi yang digunakan yaitu CRISP-DM berupa *Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation,* dan *Deployment*. Pada penelitian [4], keseluruhan dataset dibagi-bagi menggunakan teknik *k-fold cross validation* dengan nilai $k = 10$. Hasil akurasi yang didapatkan dari klasifikasi terhadap data peserta didik adalah sebesar 97,63%.

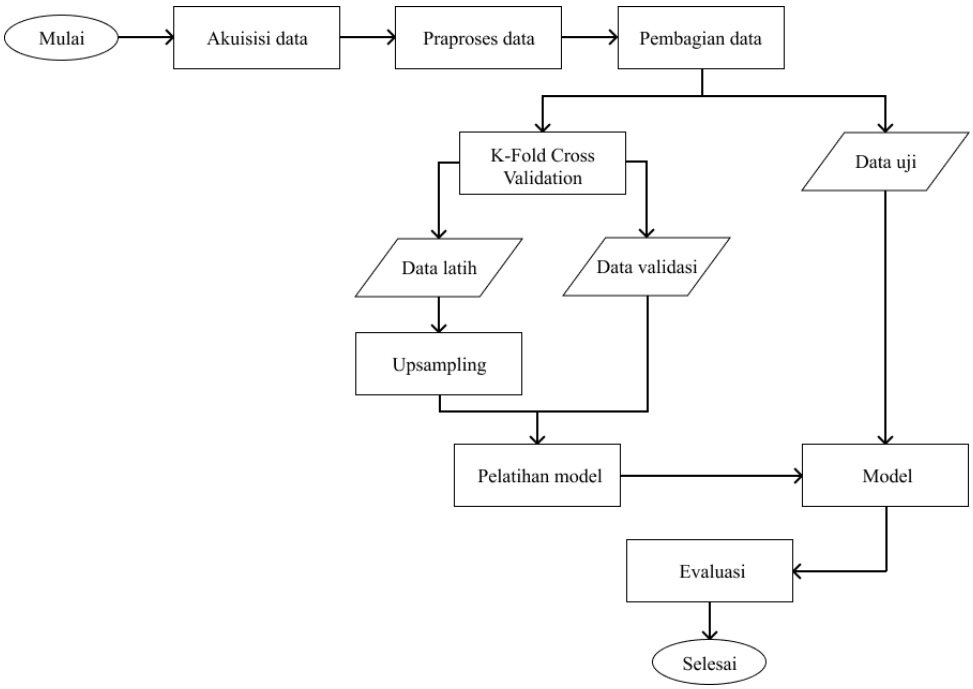
Klasifikasi menggunakan algoritma *random forest* juga sudah pernah dilakukan sebelumnya [5] pada data curah hujan. Percobaan dilakukan terhadap dua skenario. Skenario yang pertama adalah ketika data dibagi menggunakan teknik *k-fold cross validation* dengan nilai $k = 10$, kemudian percobaan kedua dilakukan terhadap keseluruhan dataset. Hasil

akurasi dari algoritma *random forest* dengan pembagian data menggunakan *k-fold cross validation* dengan nilai $k = 10$ mendapatkan akurasi sebesar 71,09%, sedangkan pemodelan menggunakan seluruh data mendapatkan akurasi sebesar 99,45%.

Pada penelitian ini akan dilakukan klasifikasi bakteri E. coli berdasarkan delapan kelas yang dimiliki oleh dataset E. coli dengan penerapan klasifikasi menggunakan algoritma Naïve Bayes, *Decision Tree*, dan *Random Forest* untuk membandingkan performa ketiga algoritma tersebut dan mengetahui manakah algoritma klasifikasi yang dapat menghasilkan akurasi terbaik.

2 TINJAUAN PUSTAKA

Dalam mencapai tujuan penelitian, maka terdapat tahapan-tahapan kerja yang perlu dilakukan yang diilustrasikan pada **Gambar 2.1.**



Gambar 2.1 Metode Penelitian

2.1 Akuisisi Data

Dataset E. coli pada penelitian ini yaitu data sekunder yang diperoleh melalui halaman *website* UCI Machine Learning Repository. Dataset E. coli memiliki jumlah data sebanyak 336 sampel data yang terdiri dari 8 atribut, dengan 7 atribut sebagai fitur data dan 1 atribut sebagai kelas data. Kelas data pada dataset E. coli terdiri dari 8 kelas, yaitu cp, im, imL, imS, imU, om, omL, dan pp. Lebih jelas, sampel data pada dataset ini ditunjukkan pada **Tabel 2.1**.

Tabel 2.1 Sampel Dataset

No	seq_name	mcg	gvh	lip	chg	aac	alm1	alm2	class
1	AAT_ECOLI	0.49	0.29	0.48	0.5	0.56	0.24	0.35	cp
2	EMRA_ECOLI	0.06	0.61	0.48	0.5	0.49	0.92	0.37	im
3	NLPA_ECOLI	0.75	0.55	1.0	1.0	0.40	0.47	0.30	imL
4	ATKC_ECOLI	0.85	0.53	0.48	0.5	0.53	0.52	.035	imS
5	CAIT_ECOLI	0.69	0.43	0.48	0.5	0.59	0.74	0.77	imU
6	FADL_ECOLI	0.78	0.68	0.48	0.5	0.83	0.40	0.29	om
7	MULI_ECOLI	0.77	0.57	1.0	0.5	0.37	0.54	0.01	omL
8	AGP_ECOLI	0.74	0.49	0.48	0.5	0.42	0.54	0.36	pp

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

2.2 Praproses Data

Menurut deskripsi dataset E. coli yang disediakan oleh sumber data, dataset E. coli ini tergolong dataset yang bersih, yaitu pada dataset ini tidak terdapat *missing value*, tidak terdapat duplikasi data, serta data sudah dalam rentang yang seragam dengan nilai di antara 0-1, sehingga tidak perlu dilakukan praproses data yang intensif. Praproses data yang dilakukan hanyalah menghilangkan kolom identitas data (ID) yang bersifat unik untuk tiap sampel datanya dan hanya berfungsi sebagai tanda pengenal untuk tiap sampel data, sehingga kolom tersebut tidak akan memberikan nilai yang berarti bagi model *machine learning*.

2.3 Pembagian Data

Setelah melakukan tahapan akuisisi data dan praproses data, data dibagi menjadi data latih dan data uji dengan rasio 70% untuk data latih dan 30% untuk data uji. Rasio 70%:30% ini dipilih karena keterbatasan jumlah data, sehingga dengan membagi data dengan rasio 70%:30%, maka sampel data akan terkandung seluruhnya ke dalam data latih dan data uji. Selanjutnya, berdasarkan data latih yang sudah dibentuk melalui proses pemisahan awal dengan data uji, maka data latih tersebut akan dibagi lagi menggunakan *k-fold cross validation* dengan jumlah $k = 10$. Melalui proses *k-fold cross validation* ini akan dihasilkan data latih dan data validasi. Selain membagi data, di dalam *k-fold cross validation* juga akan dilakukan *random oversampling* untuk menyeimbangkan jumlah data. Proses *random oversampling* ini dilakukan terhadap data latih saja dan tidak dilakukan terhadap data validasi. Ini adalah pembaruan yang kami lakukan pada penelitian kami, dimana belum ada penelitian yang mencoba melakukan klasifikasi pada dataset E. coli dengan melakukan *oversampling* untuk menyeimbangkan jumlah data.

2.4 Pelatihan Model terhadap Data

Pelatihan model terhadap data akan dilakukan pada data latih dan data validasi yang sudah dibagi menggunakan teknik *k-fold cross validation*. Model yang akan digunakan adalah model klasifikasi Naïve Bayes, *decision tree*, serta *random forest*. Pada proses pelatihan model juga akan dilakukan proses *hyperparameter tuning*, yaitu proses pencarian parameter terbaik sehingga model dapat mengklasifikasi data dengan tingkat akurasi yang baik. Proses *hyperparameter tuning* tersebut akan dilakukan untuk seluruh model yang digunakan.

2.4.1 Naïve Bayes

Algoritma Naïve Bayes dapat dijadikan sebagai pengklasifikasi probabilistik sederhana yang akan memperkirakan himpunan peluang dengan memperhitungkan kemunculan serta kombinasi nilai dalam himpunan data tertentu. Algoritma Naïve Bayes menggunakan teorema Bayes yang menganggap bahwa semua atribut bersifat berdiri sendiri atau tidak berkaitan satu sama lain berdasarkan nilai variabel kelas [6]. Secara matematis, algoritma ini dapat dituliskan berdasarkan **Persamaan 1** [7]**.**

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} \quad 1$$

Dimana, $P(H|E)$ adalah kemungkinan atau peluang hipotesis berdasarkan kondisi (*posterior probability*), $P(E|H)$ adalah peluang parameter E berdasarkan kondisi pada hipotesis H, kemudian $P(H)$ adalah peluang hipotesis H (*prior probability*) dan $P(E)$ adalah peluang parameter E (*prior probability*).

Adapun keunggulan dari algoritma Naïve Bayes adalah sebagai berikut [7]:

1. Tidak memerlukan data latih yang banyak untuk dapat mengestimasi parameter yang dibutuhkan untuk klasifikasi.
2. Efisien dari segi ruang dan waktu.
3. Dapat menangani atribut yang tidak relevan dengan baik.

2.4.2 Decision Tree

Algoritma *Decision tree* merupakan algoritma yang menggunakan pohon keputusan sebagai fungsi prediksi untuk memetakan sebuah data terhadap kelasnya [8]. Struktur pohon keputusan terdiri dari simpul akar (*root node*), simpul internal (*internal node*), dan simpul daun (*leaf node*). Bentuk dari pohon keputusan menyerupai diagram alir (*flow chart*), dimana setiap simpul internal menunjukkan kondisi pengujian, dan setiap simpul daun menunjukkan kelas dari suatu data. Pohon keputusan disusun menggunakan pendekatan *divide-and-conquer*. Setiap jalur di dalam pohon keputusan menunjukkan sebuah aturan yang dapat mengklasifikasi data ke dalam suatu kelas [8]. Ada banyak algoritma *decision tree*, di antaranya adalah ID3, C4.5, C5.0, CART, dan lainnya. Pada penelitian ini, jenis algoritma *decision tree* yang digunakan adalah algoritma CART.

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

CART adalah kependekkan dari *Clasification and Regression Tree*. Algoritma ini diciptakan oleh Breiman, dkk. pada tahun 1984. *Tree* yang akan tercipta dari algoritma CART memiliki karakteristik sebagai *binary tree*, yaitu setiap simpul internalnya hanya memiliki tepat dua anak simpul saja [9]. Adapun keunggulan dari algoritma CART sebagai berikut [9]:

1. Algoritma CART dapat menangani data yang bersifat numerikal maupun kategorikal.
2. Algoritma CART dengan sendirinya akan mengidentifikasi parameter yang signifikan dan akan mengeliminasi parameter yang tidak signifikan.
3. Algoritma CART dapat menangani *outlier*.

2.4.3 Random Forest

Random forest adalah pengklasifikasi yang bersifat *ensemble*, yaitu *random forest* akan menciptakan sebuah hutan (*forest*) menggunakan sejumlah pohon keputusan (*decision tree*). Jumlah suara terbanyak (*voting*) dari seluruh pohon keputusan akan digunakan untuk menentukan kelas dari sebuah input data [10]. Hal ini secara langsung dapat mengatasi masalah ketika melakukan klasifikasi hanya menggunakan satu pohon keputusan saja sering kali tidak optimal, tetapi dengan memasukkan banyak pohon keputusan, maka akan diperoleh nilai akurasi yang optimal secara global [10].

2.5 Evaluasi Model

Hasil klasifikasi oleh sebuah model *machine learning* dapat direpresentasikan ke dalam sebuah matriks atau disebut sebagai *confusion matrix*. Sejumlah k kelas pada *confusion matrix* adalah matriks dengan ukuran $k \times k$ yang setiap *cell* $[i, j](i = 1, \dots, k, j = 1, \dots, k)$ merepresentasikan jumlah kemunculan kelas nyata C_i dan kelas terprediksi C_j [11]. Dalam kasus kelas biner, nilai yang terkandung dalam *confusion matrix* dapat digolongkan ke dalam empat nilai, yaitu TP (*True Positive*) yang merupakan total keseluruhan data pada kelas positif yang benar diklasifikasi sebagai data pada kelas positif, TN (*True Negative*) merupakan total data yang berada pada kelas negatif yang benar diklasifikasi sebagai data pada kelas negatif, FP (*False Positive*) yang merupakan jumlah data pada kelas negatif yang salah diklasifikasi sebagai data kelas positif, serta FN (*False Negative*) yang merupakan jumlah data pada kelas positif yang salah diklasifikasi sebagai data pada kelas negatif [11]. Menggunakan nilai-nilai yang terkandung di dalam *confusion matrix* maka dapat dihitung metrik akurasi keseluruhan dengan rumus berikut [12]:

$$akurasi = \frac{1}{N_T} \sum_{i=1}^{N_C} M_{ii} \quad (2)$$

Dimana, N_T adalah jumlah data uji dan M_{ii} adalah nilai TP untuk setiap kelas.

3 Hasil dan Pembahasan

Dataset E. coli berhasil diakuisisi melalui *website* UCI Machine Learning Repository. Setelah melakukan tahapan akuisisi data, dataset dipraproses dengan menghilangkan kolom seq_name yang merupakan kolom sebagai tanda pengenal untuk tiap sampel data. Data yang sudah dipraproses ditunjukkan pada **Tabel 3.1**.

Tabel 3.1 Dataset Setelah Praproses

No	mcg	gvh	lip	chg	aac	alm1	alm2	class
1	0.49	0.29	0.48	0.5	0.56	0.24	0.35	cp
2	0.06	0.61	0.48	0.5	0.49	0.92	0.37	im
3	0.75	0.55	1.0	1.0	0.40	0.47	0.30	imL
4	0.85	0.53	0.48	0.5	0.53	0.52	.035	imS
5	0.69	0.43	0.48	0.5	0.59	0.74	0.77	imU
6	0.78	0.68	0.48	0.5	0.83	0.40	0.29	om
7	0.77	0.57	1.0	0.5	0.37	0.54	0.01	omL
8	0.74	0.49	0.48	0.5	0.42	0.54	0.36	pp

Setelah dipraproses, data dibagi menjadi data uji dan data latih dengan rasio 70% untuk data latih dan 30% untuk data uji. Hasil dari proses pemecahan ini adalah tercipta data latih sebesar 235 sampel data dan data uji sebesar 101 sampel data. Selanjutnya, data latih akan dibagi lagi menggunakan *k-fold cross validation* dengan jumlah $k = 10$. Melalui

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

proses *k-fold cross validation* ini akan dihasilkan data latih dan data validasi dengan jumlah 211 sampel data untuk data latih dan 24 sampel data untuk data validasi. Selain membagi data, di dalam *k-fold cross validation* juga akan dilakukan *random oversampling* untuk menyeimbangkan data. Proses *random oversampling* ini dilakukan terhadap data latih saja dan tidak dilakukan terhadap data validasi. Pada *fold* 1 hingga *fold* 3 serta *fold* 6 hingga *fold* 10 berisi 720 sampel data setelah dilakukan *upsampling* dan pada *fold* 4 dan *fold* 5 berisi 630 sampel data setelah dilakukan *upsampling*.

Setelah didapatkan data latih dan data validasi, maka dilanjutkan dengan proses *hyperparameter tuning* untuk menemukan parameter terbaik untuk masing-masing model. Model Naïve Bayes tidak memiliki parameter yang perlu di*tuning*, sehingga langsung didapatkan akurasi pelatihan sebesar 75,32%. Selanjutnya, untuk model *decision tree* terdapat beberapa parameter yang dapat di*tuning* yang ditunjukkan pada **Tabel 3.2**. Akurasi yang ditunjukkan merupakan hasil rata-rata akurasi validasi dari seluruh data *fold*.

Tabel 3.2 Parameter Tuning Model *Decision Tree*

Parameter	Nilai Parameter	Akurasi
criterion	gini	79,54%
	entropy	81,74%
max_features	3	75,70%
	4	78,80%
	5	81,74%
max_depth	5	79,60%
	6	81,74%
	7	80,88%
	8	81,32%
	9	80,90%
min_sample_leaf	1	78,31%
	2	78,33%
	3	81,74%

Setelah dilakukan pengujian parameter, hasil terbaik didapatkan rata-rata akurasi pelatihan model sebesar 81,74% dengan menggunakan parameter {*criterion*: entropy, *max_depth*: 6, *max_features*: 5, dan *min_samples_leaf*: 3}.

Sama seperti model *decision tree*, model *random forest* pun memiliki beberapa parameter yang dapat di*tuning* yang ditunjukkan pada **Tabel 3.3**.

Tabel 3.3 Parameter Tuning Model *Random Forest*

Parameter	Nilai Parameter	Akurasi
criterion	gini	86,34%
	entropy	84,27%
n_estimators	100	85,48%
	150	85,48%
	200	85, 92%
	250	86,34%
max_depth	5	84,25%
	6	86,34%
	7	86,24%
	8	85,45%
	9	85,03%

Hasil terbaik model *random forest* mendapatkan rata-rata akurasi pelatihan model sebesar 86,34% setelah dilakukan pengujian parameter di dalam *cross validation* dengan menggunakan parameter {*criterion*: gini, *n_estimators*: 250, *max_depth*: 6, *random_state*: 42}.

e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Tahapan selanjutnya setelah didapatkan parameter terbaik untuk setiap model adalah melakukan evaluasi model terhadap data uji. Metrik yang digunakan untuk menguji model adalah *confusion matrix*. Model pertama yang akan dievaluasi adalah model Naïve Bayes. *Confusion matrix* untuk model Naïve Bayes ditunjukkan pada **Tabel 3.4**.

Tabel 3.4 Confusion Matrix Model Naïve Bayes

		Prediksi							
		cp	**im**	**imS**	**imL**	**imU**	**om**	**omL**	**pp**
Aktual	**cp**	42	0	0	0	0	0	0	1
	im	4	11	0	0	7	0	0	1
	imS	0	0	0	0	0	0	1	0
	imL	0	0	0	0	1	0	0	0
	imU	0	0	0	0	9	0	0	1
	om	0	0	0	0	0	0	0	6
	omL	0	0	0	0	0	0	1	0
	pp	0	0	0	0	0	0	0	16

Berdasarkan *confusion matrix* pada **Tabel 3.4** maka dapat dihitung nilai akurasi terhadap data uji untuk model Naïve Bayes sebagai berikut:

$$akurasi = \frac{42 + 11 + 0 + 0 + 9 + 0 + 1 + 16}{101} = 0.7821 = 78{,}21\%$$

Hasil klasifikasi data oleh model *decision tree* ditunjukkan pada *confusion matrix* pada **Tabel 3.5**.

Tabel 3.5 Confusion Matrix Model Decision Tree

		Prediksi							
		cp	**im**	**imS**	**imL**	**imU**	**om**	**omL**	**pp**
Aktual	**cp**	41	0	0	0	0	2	0	1
	im	1	15	0	0	7	0	0	
	imS	0	0	0	0	0	0	1	0
	imL	0	1	0	0	0	0	0	0
	imU	1	4	0	0	5	0	0	0
	om	0	0	0	0	0	5	0	1
	omL	0	0	0	0	0	0	1	0
	pp	3	2	0	0	1	0	0	10

Berdasarkan *confusion matrix* pada **Tabel 3.5** maka dapat dihitung nilai akurasi terhadap data uji untuk model *decision tree* sebagai berikut:

$$akurasi = \frac{41 + 15 + 0 + 0 + 5 + 5 + 1 + 10}{101} = 0.76223 = 76{,}23\%$$

Terakhir, hasil klasifikasi data oleh model *random forest* ditunjukkan pada *confusion matrix* pada **Tabel 3.6**.

Tabel 3.6 Confusion Matrix Random Forest

		Prediksi							
		cp	**im**	**imS**	**imL**	**imU**	**om**	**omL**	**pp**
Aktual	**cp**	43	0	0	0	0	0	0	

	im	1	17	0	0	5	0	0	1
	imS	0	0	0	0	0	0	1	0
	imL	0	1	0	0	0	0	0	0
	imU	0	3	0	0	6	0	0	1
	om	0	0	0	0	0	6	0	0
	omL	0	0	0	0	0	0	1	0
	pp	1	2	0	0	1	0	0	12

Berdasarkan *confusion matrix* **Tabel 3.6** maka dapat dihitung nilai akurasi terhadap data uji untuk model *random forest* sebagai berikut:

$$akurasi = \frac{43 + 17 + 0 + 0 + 6 + 6 + 1 + 12}{101} = 0.8415 = 84{,}15\%$$

Berdasarkan hasil nilai akurasi yang diperoleh dari ketiga algoritma di atas, Algoritma *random forest* terlihat dapat memprediksi lebih akurat pada data E. coli dibandingkan dengan kedua algoritma lainnya yaitu *decision tree* dan Naïve Bayes. Rata-rata hasil akurasi yang diperoleh algoritma *random forest* sebesar **0.8415** atau **84.15%** yang menunjukkan bahwa *random forest* memiliki hasil akurasi yang lebih besar dibandingkan kedua algoritma lainnya.

4 PENUTUP

4.1 KESIMPULAN

Melalui penelitian yang telah dilakukan terhadap dataset E. coli, dapat disimpulkan bahwa evaluasi dengan membandingkan algoritma yang dipilih sudah terlihat pada bagian akurasi masing-masing dari hasil pemodelan. Algoritma *random forest* mampu memberikan hasil klasifikasi terbaik dibandingkan dengan algoritma klasifikasi Naïve Bayes dan *decision tree*. Pengujian yang dilakukan menggunakan algoritma Naïve Bayes memperoleh hasil akurasi yaitu sebesar 78%, *decision tree* dengan menggunakan parameter {*criterion*: entropy, *max_depth*: 6, *max_features*: 5, *min_samples_leaf*: 3} mendapatkan nilai akurasi sebesar 76%, serta *random forest* dengan menggunakan parameter {*criterion*: gini, *n_estimators*: 250, *max_depth*: 6, *random_state*: 42} mendapatkan nilai akurasi tertinggi, yaitu mendapatkan nilai 84%. Berdasarkan hasil penelitian yang telah diperoleh dapat disimpulkan bahwa *random forest* adalah algoritma yang cocok dan tepat untuk klasifikasi pada dataset E. coli.

4.2 SARAN

Penulis sadar bahwa penelitian yang penulis lakukan ini belum sempurna, sehingga adapun saran penulis untuk penelitian selanjutnya adalah sebagai berikut:

1. Selanjutnya, penelitian lebih lanjut perlu dilakukan dengan menambahkan fitur yang lebih banyak lagi untuk melihat perbedaan hasil klasifikasi sebelum dilakukan penambahan fitur.
2. Dataset E. coli ini adalah dataset yang sifatnya sangat tidak seimbang, yaitu jumlah sampel data tiap kelasnya memiliki jumlah yang tidak sama dalam rentang yang jauh. Sehingga perlu adanya penambahan jumlah data.
3. Penelitian yang serupa dapat dilakukan, namun metode klasifikasi yang digunakan dapat diganti dengan metode klasifikasi lain seperti *K-Nearest Neighbors* yang hasilnya nanti bisa dibandingkan dengan hasil klasifikasi pada penelitian sebelumnya untuk menentukan apakah klasifikasi yang dilakukan lebih baik atau tidak.

References

[1] S. N. Chatterjee and K. Chaudhuri, "Outer Membrane Vesicles: Interaction with Prokaryotes and Eukaryotes," *SpringerBriefs in Microbiology*, pp. 71-79, 2012.

[2] E. Priyanti, "Penerapan Algoritma Naïve Bayes Untuk Klasifikasi Bakteri Gram-Negatif," *Jurnal Teknik Informatika*, vol. 3, no. 2, pp. 68-76, 2017.

[3] K. Nakai, "UCI Machine Learning Repository: Ecoli Data Set," [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Ecoli/. [Accessed 22 Maret 2021].

[4] I. Sutoyo, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI DATA PESERTA DIDIK," *Jurnal Pilar Nusa Mandiri,* vol. 14, no. 2, pp. 217-224, 2018.

[5] A. Primajaya and B. N. Sari, "Random Forest Algorithm for Prediction of Precipitation," *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM),* vol. 1, no. 1, pp. 27-31, 2018.

[6] T. R. Patil and S. S. Sherekar, "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," *International Journal Of Computer Science And Applications,* vol. 6, no. 2, pp. 256-261, 2013.

[7] I. B. A. Peling, I. N. Arnawan, I. P. A. Arthawan and I. G. N. Janardana, "Implementation of Data Mining To Predict Period of Students Study Using Naive Bayes Algorithm," *International Journal of Engineering and Emerging Technology,* vol. 2, no. 1, pp. 53-57, 2017.

[8] S. D. Jadhav and H. P. Channe, "Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques," *International Journal of Science and Research (IJSR),* vol. 5, no. 1, pp. 1842-1845, 2016.

[9] S. Singh and P. Gupta, "COMPARATIVE STUDY ID3, CART AND C4.5 DECISION TREE ALGORITHM: A SURVEY," *International Journal of Advanced Information Science and Technology (IJAIST)* , vol. 27, no. 27, pp. 97-103, 2014.

[10] A. E. Maxwell, T. A. Warner and F. Fang, "Implementation of machine-learning classification in remote sensing: an applied review," *International Journal of Remote Sensing,* vol. 39, no. 9, pp. 2784-2817, 2018.

[11] R. Salla, H. Wilhelmiina, K. Sari, M. Mikaela, M. Pekka and M. Jaakko, "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle," *Behavioural Processes,* vol. 148, pp. 56-62, 2018.

[12] T. Nguyen, I. Hettiarachchi, A. Khatami, L. Gordon-Brown, C. P. Lim and S. Nahavandi, "Classification of Multi-class BCI Data by Common Spatial Pattern and Fuzzy System," *IEEE Access,* vol. 6, pp. 27873-27884, 2018.