Analisis Komparatif Algoritme *Machine Learning* pada Klasifikasi Kualitas Air Layak Minum

G L Pritalia [*1]

[1]Program Studi Sistem Informasi, Departemen Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta

E-mail: generosa.pritalia@uajy.ac.id[1]

Abstrak. Air berperan penting untuk kelangsungan hidup makhluk hidup. Kebutuhan memantau, menilai dan mengklasifikasi kualitas air menjadi penting untuk memahami dampak pembangunan dan industrialisasi. Proses klasifikasi kualitas air telah dilakukan menggunakan metode tradisional seperti WQI dan *Storet* serta metode modern menggunakan *machine learning.* Dalam proses klasifikasi kualitas air menggunakan *machine learning*, terdapat kondisi data yang tidak seimbang (*imbalanced data*) yang dapat menyebabkan metode *machine learning* cenderung memprediksikan kelas mayoritas dan menjadi bias. Selain itu, penggunaan seluruh fitur dalam proses klasifikasi dapat menurunkan performa klasifikasi dan menyebabkan waktu komputasi yang tinggi. Untuk mengatasi permasalahan tersebut, Penelitian ini akan melakukan beberapa pendekatan diantaranya melakukan *resampling* data untuk menyeimbangkan kelas data. Kemudian akan mencari fitur yang paling sesuai dan paling berkontribusi, serta menganalisis perbandingan kinerja algoritme *machine learning* dalam mengklasifikasi air layak minum. Hasil dari penanganan data yang tidak seimbang dan implementasi *feature selection* mampu memberikan peningkatan kerja pada algoritme khususnya metrik akurasi mencapai 24,8% dari penelitian sebelumnya. Kinerja algoritme paling optimal diperoleh dari *Random Forest* mencapai 87%, *Recall* 84%, *Miss rate* 16%, *F-Measure* 85%, dan akurasi *test* 85% dengan menggunakan 7 fitur terbaik. Namun hal lain yang dipertimbangkan adalah tingkat *Miss rate* paling kecil, yaitu 15% diperoleh dari algoritme *Decision Tree.*

Kata kunci: *Machine learning*, algoritme *imbalanced data*, klasifikasi, kualitas air

Abstract. *Water is essential for survival. Currently, there are requirements to monitor, assess, and classify water quality to understand the impact of industrialization. The water quality classification process has been carried out using traditional methods such as WQI and Storet, and machine learning methods. Imbalanced data in machine learning method can make this method have a tendency to predict the majority class and become biased. In addition, using all features in the classification process can degrade classification performance and lead to high computation time. To overcome the above-mentioned problems, this study proposes several approaches, included resampling the data to be balanced, determined the most suitable and contributing features, and compared the performance of machine learning algorithms in classifying potable water. The results of handling unbalanced data and implementing feature selection were able to provide increased work on the algorithm, especially the accuracy metric reached 24.8% from*

previous study. The most optimal algorithm performance was obtained from Random Forest with 87% of precision, 84% of recall, 16% of Miss rate, 85% of F-measure, and 85% of test accuracy, while used seven best features. However, another important aspect is the smallest Miss rate, which was 15%, obtained from Decision Tree algorithm.

Keywords: *Machine learning, algorithm, imbalanced data, classification, water quality*

1. Pendahuluan

Air bersih layak minum merupakan salah satu sumber daya yang sangat penting bagi kehidupan dan pembangunan. Standar kelayakan air bersih bagi masyarakat internasional yang digunakan untuk keperluan rumah tangga adalah sekitar 20 liter per orang per hari [1]. Hak asasi manusia atas air dan sanitasi merupakan hal mendasar bagi terwujudnya kehidupan yang layak dan bermartabat [2], [3]. Demikian pula dalam target *Sustainable Development Goals* (*SDGs*), pemenuhan hak atas air dan sanitasi diatur dalam tujuan keenam, yaitu menjamin ketersediaan dan pengelolaan air bersih dan sanitasi yang berkelanjutan untuk semua [4]. Target tersebut diharapkan dapat tercapai pada tahun 2030 secara universal terhadap pemenuhan hak atas air bersih dan sanitasi yang layak.

Penurunan kualitas air merupakan tantangan yang signifikan dalam konteks pembangunan perkotaan dan pertumbuhan penduduk, terutama dalam pengaturan pengelolaan air limbah yang memadai [5]. Kustanto berpendapat bahwa semakin tinggi aktivitas manusia dan industri yang kompleks, maka semakin tinggi tingkat pencemaran yang mempengaruhi ekosistem perairan dan sanitasi yang layak serta akses ke air minum yang aman untuk konsumsi manusia [6]. Permasalahan yang terjadi dalam kurun waktu 1 dekade ini adalah 800 juta orang tidak memiliki akses yang memadai ke air minum yang aman [7], dan hampir 2 juta bayi, meninggal setiap tahun karena kurangnya akses terhadap air minum yang aman [8].

Saat ini, ada urgensi kebutuhan untuk memantau, menilai dan mengklasifikasi kualitas air untuk memahami dampak pembangunan dan industrialisasi [9]. Untuk mengetahui bahwa air memiliki kualitas yang sesuai standar kesehatan dapat diketahui dari zat-zat atau mineral yang terkandung didalamnya. Proses klasifikasi kualitas air umumnya dilakukan menggunakan perhitungan rumus secara manual seperti menghitung *Water Quality Index* (WQI) dan *STORET* [10]. Metode tradisional didasarkan pada pengetahuan tentang parameter yang telah diterapkan untuk menilai dan mengklasifikasikan kualitas air layak minum atau tidak. Proses tersebut memakan waktu yang cukup lama dalam perhitungan. Oleh karena itu, diperlukan sistem otomatis yang dapat mempermudah proses klasifikasi kualitas air.

Model *Artificial Intelligent (AI)* salah satunya, yaitu metode *machine learning* merupakan sarana yang cocok untuk pemantauan, pengelolaan, berkelanjutan dan pembuatan kebijakan [11]. Penelitian terkait klasifikasi kualitas air pernah dilakukan oleh Riyantoko menggunakan metode *semi-supervised machine learning* serta menggunakan *dataset water potability* dari *Kaggle dataset* [12]. Hasil penelitian menyatakan bahwa model terbaik yaitu, *Random Forest Classifier* dengan tingkat akurasi tertinggi sebesar 72,81%. Lebih lanjut, penelitian yang dilakukan oleh [10] yang bertujuan untuk mengklasifikasi kualitas air bersih di PDAM Tirta Kencana Kabupaten Jombang. Penelitian tersebut melakukan komparasi algoritme *machine learning KNN* dan *Naïve Bayes*. Berdasarkan hasil studi literaturnya, Rahman dan kawan-kawan menyimpulkan bahwa metode *K-Nearest Neighbor* dan *Naïve Bayes* merupakan metode yang memiliki akurasi cukup tinggi. Dari hasil pengujian diperoleh rata- rata nilai akurasi metode *KNN* sebesar 82.42% dan rata-rata nilai akurasi metode *Naïve Bayes* sebesar 70.32%.

Kehandalan dan kinerja model *machine learning* sangat dipengaruhi oleh ketersediaan sampel baik sampel data air yang layak minum dan tidak layak minum dalam jumlah yang besar dan seimbang. Dalam kasus klasifikasi kualitas air memiliki kondisi data yang tidak seimbang dimana kelas air yang tidak layak minum mendominasi *dataset*. Metode *machine learning* akan cenderung memprediksi kelas mayoritas

dan menjadi sarat akan bias [13]. Untuk mengatasi ketidakseimbangan data (*imbalanced dataset*), ada alternatif upaya yang dapat dilakukan dalam bentuk *resampling*[14].

Selain faktor keseimbangan *dataset,* fitur atau atribut dapat mempengaruhi kinerja model. Penggunaan seluruh fitur dalam proses klasifikasi dapat menurunkan performa klasifikasi dan menyebabkan data berdimensi tinggi dan memiliki waktu komputasi yang tinggi. Pendekatan populer untuk masalah kumpulan data berdimensi tinggi ini adalah dengan mencari proyeksi data ke sejumlah kecil variabel atau fitur yang menyimpan informasi sebanyak mungkin [15]. Istilah tersebut sering dikenal dengan seleksi fitur (*feature selection*).

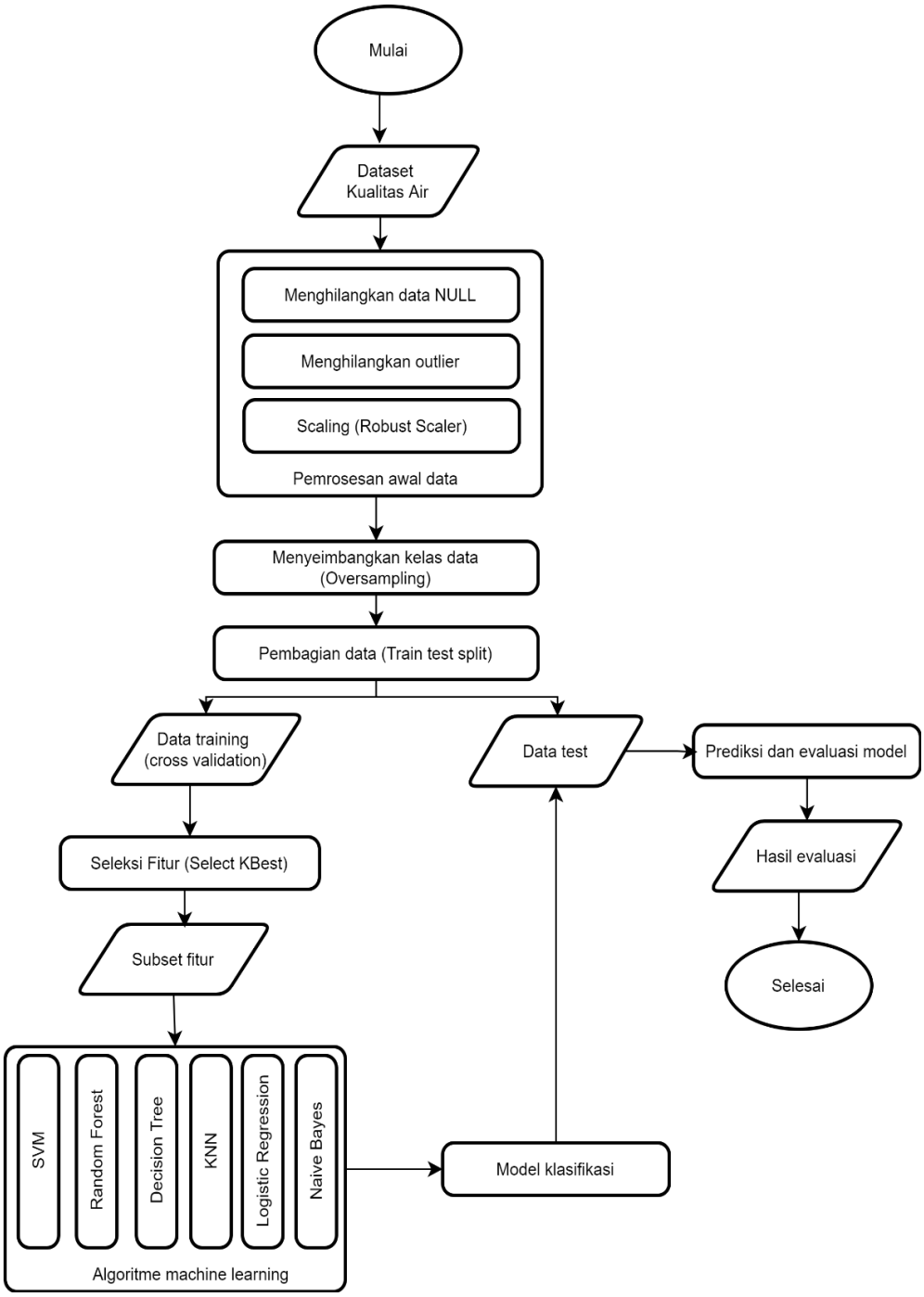
Dalam upaya mengoptimalkan kinerja prediksi dan mengurangi bias, penelitian ini akan melakukan beberapa pendekatan, pertama penelitian ini melakukan proses *resampling* data untuk menyeimbangkan kelas data. Kemudian yang kedua, penelitian ini akan mencari fitur yang paling sesuai dan paling berkontribusi dalam *dataset* kualitas air. Selain itu, tujuan lain dari penelitian ini adalah untuk menganalisis perbandingan kinerja algoritme *machine learning* untuk mengklasifikasi air layak minum dan menentukan parameter yang paling berkontribusi didalamnya.

2. Metode

Penelitian ini menggunakan *dataset* dari repositori Kaggle yang tersedia untuk umum. Terdapat 3276 baris data dan 9 fitur atau parameter dan 1 atribut labelling bernama *Potability* yang tertampil pada Tabel 1 [16].

Tabel 1. Parameter *Dataset* Kualitas Air

Parameter	Deskripsi
PH	PH merupakan indikator kondisi asam atau basa status air. PH optimum yang dibutuhkan berkisar 6,5-8
Hardness	*Hardness* atau kesadahan air merupakan tingkat kemampuan air untuk mengendapkan sabun disebabkan karena adanya ion-ion logam bervalensi seperti kalsium dan magnesium
Solids	*Solid* menggambarkan garam anorganik dan jumlah bahan organik yang berada di larutan dalam air.
Chloramines	Klorin dan kloramin adalah disinfektan utama yang digunakan dalam sistem air umum. Kloramin terbentuk ketika amonia ditambahkan ke klorin untuk mengolah air minum
Sulfate	*Sulfate* ditemukan di mineral, tanah, batuan. *Sulfate* dapat menyebabkan rasa yang pekat dengan kadar yang tinggi serta dapat menyebabkan efek pencahar pada konsumen yang tidak terbiasa.
Conductivity	Konsentrasi jumlah kation atau ion bermuatan positif dan anion atau ion bermuatan negatif di dalam air. Peningkatan konsentrasi ion meningkatkan konduktivitas listrik di air.
Organic Carbon	*Organic carbon* berasal dari bahan atau senyawa organik yang terurai di air
Trihalomethanes	*Trihalomethanes* (THMs) terbentuk dalam air minum terutama sebagai akibat dari klorinasi bahan organik
Turbidity	Tingkat kekeruhan yang ada di air
Potability	Indikator air layak diminum dan tidak layak diminum



Gambar 1. Alur kerja penelitian

Penelitian menggunakan *Jupyter Notebook* IDE (*Python versi 3.7*). *Jupyter Notebook* merupakan aplikasi *open-source* berbasis *website* yang dibangun menggunakan bahasa pemrograman Python [17] serta Library Scikit-learn versi 0.21.3 digunakan untuk mendukung pemrograman. Diagram alur kerja penelitian ini ditunjukkan pada Gambar 1. Langkah Pertama ada beberapa tahapan untuk pemrosesan data

awal. Algoritme *machine learning* cenderung dipengaruhi oleh data yang mengandung *noise*. *Noise* harus dikurangi sebanyak mungkin untuk menghindari kompleksitas yang tidak perlu dalam model yang disimpulkan dan meningkatkan efisiensi algoritme [18].

Tahapan proses data awal meliputi: menghilangkan data yang NULL atau kosong pada 1434 baris data, selanjutnya menghilangkan *outlier* atau pencilan. *Outlier* (pencilan) dapat didefinisikan sebagai titik ekstrim suatu data yang letaknya jauh dari pusat data dan menyimpang dari data yang lain serta terjadi kemungkinan besar dapat berpengaruh terhadap analisis regresi [19]. Salah satu cara mengetahui data *outlier,* yaitu dengan menggunakan Metode *Boxplot*. Konsep metode ini adalah menggunakan nilai dari jangkauan interkuartil atau *Interquartile Range* (IQR) yang merupakan selisih antara kuartil 1 terhadap kuartil 3. Tahapan pemrosesan data yang terakhir, yaitu *scaling. Scaling* adalah suatu cara untuk mengubah nilai data ke kisaran nilai data baru dengan menggunakan pendekatan tertentu yang bertujuan untuk menghindari fitur yang memiliki nilai lebih besar mendominasi fitur yang memiliki nilai lebih kecil. *Robust scaler* memiliki kinerja yang baik dalam menjaga penyebaran variabel setelah proses scaling yang memiliki tingkat kecondongan (*skewness*) yang tinggi dan *robust scaler* dapat melakukan penskalaan pada data yang mengandung outlier, sehingga dalam penelitian ini menggunakan *Robust scaler* [20].

Penelitian ini juga mengusulkan adanya *resampling* untuk menyeimbangkan kelas data. Terdapat dua alternatif cara untuk menyeimbangkan kelas data. Cara pertama, yaitu *oversampling* atau menambah jumlah data dengan mensintesis data, sedangkan cara kedua, yaitu *undersampling* atau mengurangi jumlah data pada kelas mayor. Namun, melakukan *undersampling* dapat terjadi kemungkinan hilangnya informasi yang dibutuhkan saat proses klasifikasi data sehingga penelitian ini memilih melakukan *oversampling* dengan menggunakan teknik sampling *Random Over Sampling* (ROS) [21]. Metode ini terdiri dari peningkatan ukuran kelompok pengamatan dari kelas minoritas dengan memilih secara acak data yang akan direplikasi.

Setelah kedua kelas data memiliki jumlah data yang seimbang, *dataset* dibagi menjadi data untuk *training* (data pelatihan) dan data untuk *testing* (data pengujian) dengan skema pengujian 80%: 20 dan 75%: 25%. Penelitian ini juga menggunakan metode validasi *K-fold cross validation* pada set pelatihan. Metode *K-fold Cross Validation* merupakan metode pembagian data ke dalam K bagian secara acak. K atau lipatan, dalam penelitian ini berjumlah 5. Data akan diproporsikan sebagai *training set* maupun *tes*t *set* pada masing-masing lipatan (*fold*). Hal tersebut untuk memvalidasi reliabilitas hasil.

2.1. Feature Selection

Feature Selection pada dasarnya adalah proses memilih beberapa fitur yang informatif dan relevan dari kumpulan fitur yang lebih besar yang menghasilkan karakterisasi pola beberapa kelas yang lebih baik [15]. Ada sejumlah teknik pemilihan fitur termasuk metode filter, *wrapper*, dan *embedded*. Metode Filter bekerja tanpa membawa pengklasifikasi. Hal ini membuat metode filter sangat efisien secara komputasi [22]. *Univariate Feature Selection (SelectKBest)* masuk dalam kategori metode filter dalam proses *feature selection*. *SelectKBest* merupakan algoritme pemilihan fitur yang digunakan untuk meningkatkan akurasi prediksi atau untuk meningkatkan kinerja pada *dataset* dimensi tinggi. *SelectKBest* termasuk dalam bagian *Univariate Feature selection* yang memilih fitur terbaik berdasarkan uji statistik *univariat* atau uji ANOVA. Uji statistik dapat digunakan untuk memilih fitur-fitur yang memiliki hubungan paling kuat dengan variabel *output*. SelectKBest menghapus semua kecuali fitur yang memiliki skor tertinggi [23]. SelectKBest memilih fitur top K yang memiliki relevansi maksimum dengan variabel target [24].

2.2. Algoritme Machine Learning

Algoritme yang digunakan dalam penelitian ini adalah *Support Vector Machine, Logistic Regression, Naïve Bayes, KNN*, dan *Random Forest*. *Logistic Regression, Random Forest,* dan *Naïve Bayes* baik

untuk kumpulan data besar dan Data berdimensi tinggi [24]. *Support Vector Machine* kuat terhadap *overfitting* dan berkinerja baik pada masalah dimensi data yang sangat tinggi [25]. *Overfitting* terjadi ketika model tampil sangat baik pada set pelatihan *(training set)* tetapi tidak dapat digeneralisasi ke data baru (test set) [26]. Sementara itu, *Decision Tree* dan *KNN* sangat mudah dipahami [27], dan seringkali memberikan kinerja yang baik tanpa banyak penyesuaian parameter, dan menghasilkan performa akurasi yang tinggi [28]. Masing-masing algoritme di *setting* parameter yang sesuai untuk mendapatkan performa yang maksimal dari setiap algoritme. Pencarian parameter terbaik dari setiap algoritme dilakukan menggunakan metode *grid search.*

2.2.1. Support Vector Machine (SVM)

SVM dapat menyelesaikan masalah terkait klasifikasi dan regresi. SVM bekerja dengan mengelompokkan data pelatihan dengan menggunakan *hyperplane* paling optimal. *Hyperplane* adalah bidang pemisah antara dua kelas dengan jarak maksimal. Bagian dari data pelatihan yang terletak pada *input space* disebut sebagai *support vector*. Algoritme SVM menggunakan kombinasi C, Gamma, dan kernel sebagai parameter. Nilai C dan Gamma yang digunakan dalam penelitian ini berkisar antara 0,001 hingga 1000. Kernel yang diterapkan untuk SVM adalah Linear, Poly, dan Fungsi Basis Radial.

2.2.2. Logistic Regression

Logistic Regression adalah model linier untuk klasifikasi. Dalam model ini, probabilitas menggambarkan bahwa kemungkinan hasil dari percobaan tunggal adalah model yang menggunakan fungsi logistik. Model *Logistic Regression* menggunakan *solver—Newton-cg, lbfgs,* dan *liblinear* sebagai parameter.

2.2.3. Naïve Bayes

Naïve Bayes merupakan teknik klasifikasi dengan prinsip probabilitas sederhana dalam pengkombinasian pengetahuan sebelumnya dengan pengetahuan baru. *Naïve Bayes* sangat efisien adalah karena mereka mempelajari parameter dengan melihat setiap fitur secara independen dan mengumpulkan statistik setiap kelas secara sederhana dari setiap fitur. Pada penelitian ini menetapkan parameter *smoothing* variabel mulai dari 0,000000001 hingga 100. *Smoothing* adalah metode untuk menghindari nilai nol dalam model probabilitas.

2.2.4. K-Nearest Neighbors (KNN)

Metode klasifikasi algoritme *KNN* adalah salah satu metode klasifikasi yang memiliki konsistensi yang kuat [10]. Algoritme *KNN* melakukan klasifikasi terhadap objek berdasarkan pada data pembelajaran yang jaraknya paling dekat dengan mencari kelompok objek tersebut. Cara kerja algoritme *KNN*, yaitu sampel diklasifikasikan berdasarkan suara terbanyak (majority voting) dari tetangga terdekat yang diukur dengan fungsi jarak. *Euclidean Distance* merupakan rumus jarak yang paling sering digunakan dan merupakan *default setting* untuk parameter metrik yang ada pada *library scikit-learn* untuk algoritme *KNN*. Penelitian ini menggunakan parameter *n_neighbors* atau jumlah tetangga terdekat mulai dari 1-50.

2.2.5. Random Forest

Random Forest pada dasarnya adalah sekumpulan *decision tree* atau pohon keputusan, dimana setiap pohon sedikit berbeda dari yang lainnya. Gagasan utama tentang *Random Forest* adalah bahwa setiap pohon mungkin melakukan pekerjaan prediksi yang relatif baik, tetapi kemungkinan besar akan terlalu cocok pada sebagian data. Jika kita membangun banyak pohon, yang semuanya bekerja dengan baik dan *overfitting* dengan cara yang berbeda, kita dapat mengurangi jumlah overfitting dengan merata-ratakan hasilnya. Pendekatan *Random Forest* dilakukan melalui penggabungan *Decision Tree.* Hasil

(*output*) akhir diperoleh dengan konsep *averaging* (mengambil nilai rata-rata) untuk meningkatkan akurasi dan mengontrol overfitting. Pengaturan parameter pada *Random Forest* menggunakan 'n_estimators' dari 100 hingga 500. 'n_estimators' merupakan jumlah pohon yang ada di *Forest*.

2.2.6. Decision Tree

Decision Tree merupakan salah satu algoritme paling populer untuk klasifikasi karena hasilnya yang bisa dipahami dalam bentuk kaidah keputusan. Pada dasarnya, *Decision Tree* mempelajari hierarki pertanyaan "*if-else*", yang mengarah pada keputusan. Proses pada *Decision Tree* adalah mengubah bentuk data (tabel) menjadi bentuk *Tree*, Mengubah *Tree* menjadi *Rule*, dan menyederhanakan *Rule.* Algoritme yang digunakan untuk pembentukan *Decision Tree* diantaranya: ID3,CART,C4.5. Algoritme ini menyederhanakan hubungan kompleks antara input variabel dan variabel target dengan membagi yang asli memasukkan variabel ke dalam sub kelompok yang signifikan. Penelitian ini menetapkan parameter gini dan *entropy*, serta *max_depth* antara 1-50 sebagai kedalaman maksimum *tree*.

2.3. Metriks Performa

Model klasifikasi yang telah dibangun kemudian diujikan pada data test. Dari penguji tersebut akan muncul hasil prediksi. Hasil prediksi klasifikasi kemudian dinilai untuk mengukur kinerja suatu model klasifikasi. Kinerja dari model klasifikasi mendeskripsikan seberapa baik sistem dalam melakukan klasifikasi terhadap data. *Confusion* matriks merupakan suatu metode
yang biasa digunakan untuk mengukur kinerja suatu model klasifikasi. Terdapat empat istilah sebagai representasi hasil proses klasifikasi:

- *True positive* (TP) mengacu pada sampel data kualitas air yang benar diklasifikasikan sebagai air layak minum.
- *True negative* (TN) adalah jumlah sampel yang diprediksi benar sebagai air yang tidak layak diminum.
- *False Negatif* (FN) adalah jumlah sampel data sebagai air tidak layak minum yang salah diklasifikasikan sebagai air yang layak minum.
- *False Positif* (FP) mengacu pada jumlah sampel data kualitas air layak minum yang salah dideteksi menjadi tidak layak minum.

Pengukuran performa pada penelitian ini terdiri dari Akurasi, *Recall, Precision, Miss rate, F-measure* [22].

- Akurasi merupakan persentase kualitas air yang layak minum dan tidak layak minum yang diklasifikasikan dengan benar. Akurasi dinotasikan sebagai berikut:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- *Recall* menunjukkan performa model dalam memprediksi kelas positif atau air layak minum terhadap data yang sebenarnya adalah positif atau air layak minum. *Recall* dirumuskan sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

- *Precision* juga dikenal sebagai tingkat prediksi positif adalah jumlah sampel data kualitas air layak minum yang diprediksi di antara hasil prediksi positif atau layak minum. Berikut ini merupakan rumus *precision*:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

- *Miss rate* atau *false negative rate* (FNR) merupakan tingkat kesalahan dalam memprediksi sampel yang seharusnya tidak layak minum menjadi sampel air yang layak untuk diminum. Tingkat kesalahan ini sebaiknya ditekan seminimal mungkin karena dapat menimbulkan isu serius di masa mendatang.

$$Miss\ rate = \frac{FN}{TP+FN} \quad (4)$$

- *F-measure* atau F1-*score* merupakan harmonisasi dari *metrik precision* dan *recall*. *F-measure* dinotasikan sebagai berikut:

$$F\text{-}measure = \frac{2\ x\ precision\ x\ recall}{precision+recall} \quad (5)$$

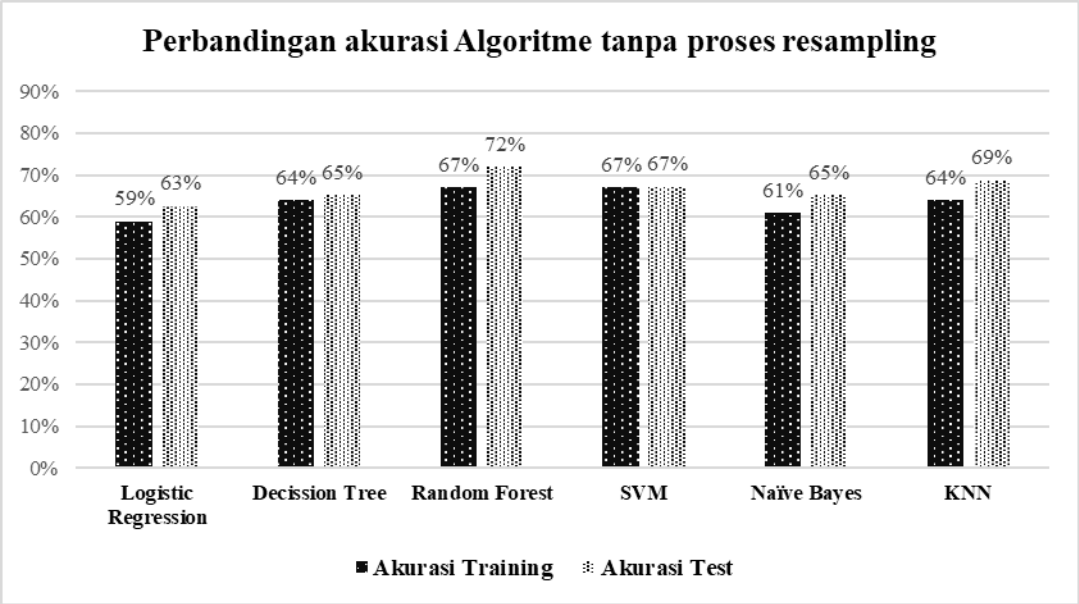
3. Hasil dan Pembahasan

Pada penelitian ini menggunakan skema pengujian dengan proporsi sampel uji untuk data *training* dan data test sebesar 75%:25% dan 80%:20%. Tabel 2 menunjukkan hasil proporsi sampel uji yang berbeda dengan menggunakan algoritme dengan pengaturan *baseline*. Hasilnya menunjukkan sebagian besar algoritme menghasilkan akurasi yang lebih baik ketika dilakukan pembagian data dengan skema 80% untuk data *training* dan 20% untuk data test.

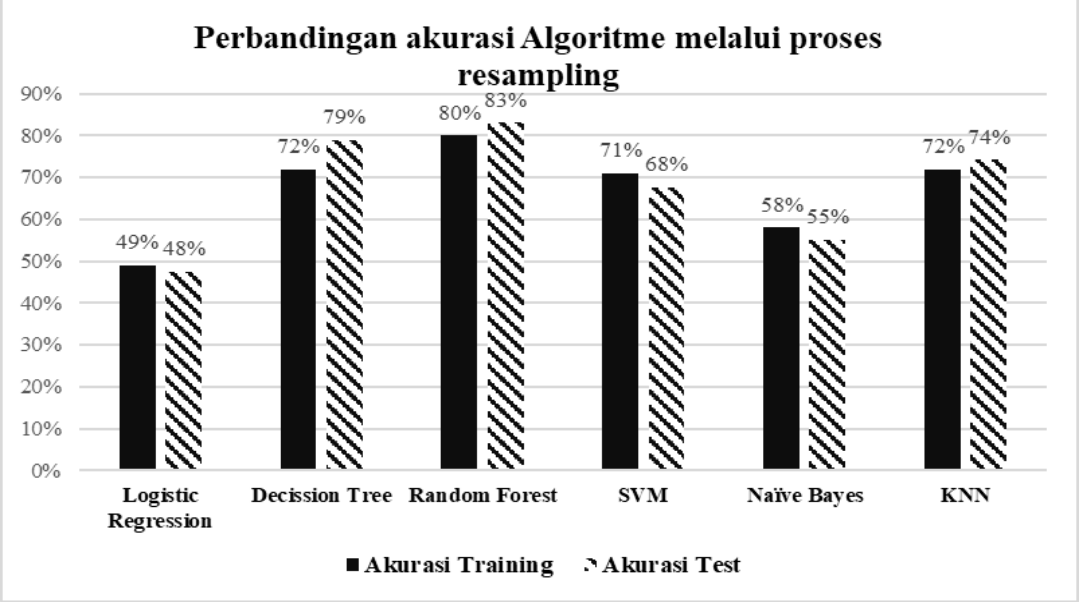
Tabel 2. Hasil untuk proporsi sampel uji yang berbeda

Algoritme *machine learning* (*baseline*)	Akurasi Test (75%:25%)	Akurasi Test (80%:20%)
Decision Tree	67%	65%
Logistic Regression	63%	63%
Random Forest	71%	72%
SVM	68%	67%
Naïve Bayes	63%	65%
KNN	68%	69%

Pengimplementasian proses *resampling* bertujuan untuk mengoptimalkan kinerja prediksi. Gambar 2 menampilkan hasil perbandingan akurasi algoritme tanpa melalui proses *resampling*. Algoritme *Random Forest* dan SVM mencapai akurasi 67% pada data *training*, sementara itu akurasi tertinggi untuk data test diperoleh ketika mengimplementasikan metode *Random Forest* yang mencapai 72%. Dari pengujian tanpa dilakukan proses *resampling* diperoleh hasil bahwa algoritme *random forest* memiliki kinerja paling baik pada akurasi *training* maupun akurasi *test*.



Gambar 2. Perbandingan performa akurasi Algoritme tanpa proses *resampling*



Gambar 3. Perbandingan performa akurasi Algoritme melalui proses resampling

Lebih lanjut, Gambar 3 menampilkan hasil pengujian akurasi Algoritme melalui proses *resampling*. Algoritme *Random Forest* memiliki hasil akurasi tertinggi, yaitu mencapai 80% pada akurasi *training* dan 83% pada akurasi test. Melihat pada Gambar 2 dan Gambar 3, Algoritme *Logistic regression* dan *Naïve Bayes* memiliki hasil kinerja akurasi test yang baik tanpa melalui proses *resampling*, sedangkan keempat Algoritme lainnya cenderung mengalami peningkatan kinerja akurasi ketika dilakukan proses *resampling*,

bahkan kinerja akurasi Algoritme *Decision Tree* meningkat 22%. Berdasarkan hasil pengujian dapat disimpulkan bahwa rata-rata algoritme memperoleh hasil akurasi terbaik ketika diimplementasikan proses *resampl*ing.

Tabel 3 merupakan hasil komparasi keenam algoritme *machine learning* baik menggunakan *feature selection* maupun tidak menggunakan *feature selection*. Algoritme *Decision Tree* memiliki kinerja akurasi test yang lebih tinggi dan *miss rate* yang lebih rendah ketika tidak menggunakan *feature selection*, namun performa akurasi *training* lebih baik ketika menggunakan *feature selection*. Lebih lanjut, dengan menggunakan 8 fitur, *Decision tree* memiliki kinerja yang lebih fit antara akurasi *training* dan akurasi *testing*. Fit mengindikasikan bahwa model tersebut dapat bekerja dan menghasilkan performa yang sepadan dan baik antara data yang di tes maupun data yang di *training* [24]. Sementara itu Algoritme *Logistic Regression* dan *KNN* memperoleh hasil pengujian paling tinggi dengan menggunakan 6 fitur yang terseleksi meliputi *Sulfate, Chloramines, Solids, Conductivity, Trihalomethanes*, dan *Ph.* Pada studi kasus ini, performa dari Algoritme *Logistic Regression* tidak lebih tinggi dari kelima algoritme *machine learning* lainnya yang diujikan. Performa algoritme *Naïve Bayes* mampu menghasilkan kinerja yang optimal dengan menggunakan jumlah fitur paling sedikit, yaitu 5 fitur meliputi *Sulfate, Chloramines*, *Solids, Conductivity, Trihalomethanes*. Hasil pengujian menggunakan 5 fitur tersebut mencapai akurasi test 57%.

Tabel 3. Komparasi performa Algoritme *Machine Learning*

Algoritme	**Feature selection**	**Jumlah fitur**	**Precision (%)**	**Recall (%)**	**Miss Rate (%)**	**F measure (%)**	**Akurasi training (%)**	**Akurasi test (%)**
Decision Tree	Tidak	9	72	85	**15**	78	72	79
	SelectKBest	8	71	81	19	76	76	76
Logistic Regression	Tidak	9	46	49	51	48	48	48
	SelectKBest	6	50	52	48	51	51	50
Random Forest	Tidak	9	85	83	17	84	80	83
	SelectKBest	**7**	**87**	**84**	**16**	**85**	**80**	**85**
SVM	Tidak	9	63	71	29	67	71	68
	SelectKBest	8	69	65	35	67	72	65
Naïve Bayes	Tidak	9	65	56	44	60	58	55
	SelectKBest	5	72	56	44	63	57	57
KNN	Tidak	9	69	78	22	73	72	74
	SelectKBest	6	70	81	19	75	74	77

Pada pengujian algoritme SVM hasil menunjukkan bahwa SVM memiliki kinerja yang baik tanpa melalui proses *feature selection*. Dengan menggunakan keseluruhan fitur, SVM mampu mencapai 68% pada akurasi test. Sementara itu, *Random Forest* menunjukkan performa yang baik mencapai *precision* 87%, *Recall* 84%, *Miss rate* 16%, F *Measure* 85%, *akurasi training* 80%, dan *akurasi test* 85%. Hasil pengujian tersebut diperoleh dengan pengaturan atau *best* parameter *'n_estimators*': 350 dan melalui proses *feature selection* dengan 7 fitur terbaik meliputi: *Sulfate, Chloramines, Solids, Conductivity, Trihalomethanes, Ph,* dan *Organic carbon.* Dari keseluruhan pengujian pada keenam algoritme *machine learning,* Algoritme *Random Forest* memiliki performa yang paling optimal pada metrik performa

precision, Recall, F-Measure, akurasi training, dan akurasi test. Akan tetapi, ada penilaian penting lainnya yang menjadi perhatian pada penelitian ini, yaitu performa dari *miss rate*. *Miss rate* atau *false negative rate (FNR*) merupakan indikasi dari banyaknya prediksi data kualitas air yang seharusnya air tidak layak minum kemudian diprediksi menjadi air yang layak minum dapat menjadi isu serius [22]. Hasil pengujian *miss rate* dengan persentase kesalahan prediksi *FNR* paling kecil diperoleh dari algoritme *Decision Tree* dengan rasio kesalahan 15%.

Hasil dari penanganan data yang tidak seimbang dan implementasi *feature selection* mampu memberikan peningkatan kerja pada algoritme *machine learning*. Tabel 4 menampilkan perbandingan performa akurasi algoritme penelitian terdahulu dengan penelitian saat ini yang menggunakan *dataset* kualitas air yang sama. Secara umum, hampir seluruh algoritme *machine learning* yang diuji pada penelitian ini mengalami peningkatan secara signifikan khususnya pada metrik akurasi, bahkan Algoritme *Decision Tree* mengalami peningkatan akurasi mencapai 24,8%. Model klasifikasi melalui proses penanganan *imbalanced data* dan *feature selection* dapat dijadikan sebagai alternatif cara untuk meningkatkan dan mengoptimalkan kinerja klasifikasi air yang layak minum dan air yang tidak layak minum.

Tabel 4. Perbandingan performa akurasi penelitian sebelumnya dan penelitian saat ini

Algoritme Machine Learning	Akurasi penelitian sebelumnya [12]	Akurasi penelitian saat ini
Decision Tree	63,28%	79%
Logistic Regression	51,86%	50%
Random Forest	72,81%	85%
SVM	54,37%	68%
Naïve Bayes	57%	57%
KNN	72,03%	77%

4. Kesimpulan

Dalam klasifikasi Dalam klasifikasi kualitas air, kondisi *imbalanced data* dapat menyebabkan metode *machine learning* cenderung memprediksi kelas mayoritas dan menjadi bias. Selain itu, penggunaan seluruh fitur dalam proses klasifikasi dapat menurunkan performa klasifikasi dan menyebabkan waktu komputasi yang tinggi. Penelitian ini mengusulkan proses *resampling* data untuk menyeimbangkan kelas data menggunakan metode *Random Oversampling*. Penelitian ini juga melakukan pencarian fitur yang paling sesuai dan paling berkontribusi menggunakan *Univariate Feature Selection (SelectKBest)*. Serta, menganalisis perbandingan kinerja pada enam algoritme *machine learning*. Performa setiap model dibandingkan dan dipilih yang paling optimal dan handal dalam mengklasifikasi kualitas air yang layak minum dan tidak layak minum.

Berdasarkan penelitian yang telah dilakukan, ada beberapa kesimpulan sebagai berikut: Implementasi proses resampling menggunakan *Random Oversampling* menjadikan sebagian besar kinerja pada Algoritme meningkat, salah satunya Algoritme *Decision Tree* yang meningkat 22% pada metrik akurasi. Lebih lanjut, pengujian Algoritme dengan proses *feature selection* menghasilkan performa paling optimal dari Algoritme *Random Forest* mencapai *precision* 87%, *Recall* 84%, *Miss rate* 16%, *F Measure* 85%, *akurasi training* 80%, dan *akurasi test* 85%. Hasil pengujian tersebut diperoleh dengan pengaturan atau *best* parameter *'n_estimators*': 350 dan melalui proses *feature selection* dengan 7 fitur terbaik. Untuk *performa miss rate* dengan persentase kesalahan prediksi FNR paling kecil diperoleh dari algoritme *Decision Tree* dengan rasio kesalahan 15%. Hasil dari penanganan data yang tidak seimbang dan implementasi *feature selection* mampu memberikan peningkatan kerja pada algoritme *machine learning* mencapai 24,8%. Waktu komputasi (*computational time*) algoritme tidak dipelajari pada penelitian ini, kedepannya matrik waktu komputasi bisa dijadikan pertimbangan sebagai komponen evaluasi model klasifikasi

Referensi

[1] E. De Buck, V. Borra, E. De Weerdt, A. Vande Veegaete, dan P. Vandekerckhove, "A systematic review of the amount of water per person per day needed to prevent morbidity and mortality in (post-)disaster settings," *PLoS One*, vol. 10, no. 5, 2015, doi: 10.1371/journal.pone.0126395.

[2] WHO, "The Human Right to Water and Sanitation Media brief," *UN-Water Decad. Program. Advocacy Commun. Water Supply Sanit. Collab. Counc.*, no. April 2011, hal. 1–8, 2011, [Daring]. Tersedia pada: http://www.un.org/waterforlifedecade/pdf/human_right_to_water_and_sanitation_media_brief.pdf .

[3] UNU-INWEH, *Water Security & the Global Water Agenda. The UN-Water analytical brief*, vol. 53, no. 9. 2013.

[4] U. Nations, "Sustainable Development Goal (SDG)." https://sdgs.un.org/ (diakses Feb 14, 2022).

[5] P. Luo *et al.*, "Water quality trend assessment in Jakarta: A rapidly growing Asian megacity," *PLoS One*, vol. 14, no. 7, hal. 1–17, 2019, doi: 10.1371/journal.pone.0219009.

[6] A. Kustanto, "Water quality in Indonesia: The role of socioeconomic indicators," *J. Ekon. Pembang.*, vol. 18, no. 1, hal. 47–62, 2020, doi: 10.29259/jep.v18i1.11509.

[7] A. K. Makarigakis dan B. E. Jimenez-Cisneros, "UNESCO's contribution to face global water challenges," *Water (Switzerland)*, vol. 11, no. 2, 2019, doi: 10.3390/w11020388.

[8] A. M. Graboski, J. Martinazzo, S. C. Ballen, J. Steffens, dan C. Steffens, *Nanosensors for water quality control*. Elsevier Inc., 2020.

[9] S. Kar, V. S. Rathore, P. K. Champati ray, R. Sharma, dan S. K. Swain, "Classification of river water pollution using Hyperion data," *J. Hydrol.*, vol. 537, hal. 221–233, 2016, doi: 10.1016/j.jhydrol.2016.03.047.

[10] M. A. Rahman, N. Hidayat, dan A. Afif Supianto, "Komparasi Metode Data Mining K-Nearest Neighbor Dengan Naïve Bayes Untuk Klasifikasi Kualitas Air Bersih (Studi Kasus PDAM Tirta Kencana Kabupaten Jombang)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Vol. 2, No. 12, Desember 2018, hlm. 6346-6353 e-ISSN*, vol. 2, no. 12, hal. 925–928, 2018.

[11] Tiyasha, T. M. Tung, dan Z. M. Yaseen, "A survey on river water quality modelling using artificial intelligence models: 2000–2020," *J. Hydrol.*, vol. 585, no. January, hal. 124670, 2020, doi: 10.1016/j.jhydrol.2020.124670.

[12] P. A. Riyantoko, "Analisis Sederhana Pada Kualitas Air Minum Berdasarkan Akurasi Model Klasifikasi Dengan Menggunakan Lucifer Machine Learning," *Semin. Nas. Sains Data 2021 (SENADA 2021)*, vol. 2021, no. Senada, hal. 12–18, 2021.

[13] V. García, J. S. Sánchez, dan R. A. Mollineda, "On the effectiveness of preprocessing methods when dealing with different levels of class imbalance," *Knowledge-Based Syst.*, vol. 25, no. 1, hal. 13–21, 2012, doi: 10.1016/j.knosys.2011.06.013.

[14] F. Charte, A. J. Rivera, M. J. del Jesus, dan F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," *Neurocomputing*, vol. 163, hal. 3–16, 2015, doi: 10.1016/j.neucom.2014.08.091.

[15] Z. M. Hira dan D. F. Gillies, "BioMed Research International (J BIOMED BIOTECHNOL)," *Comput. Math. Methods Med.*, vol. 2015, no. 1, hal. 2–4, 2015, [Daring]. Tersedia pada: http://dx.doi.org/10.1155/2015/.

[16] WHO, "Guidelines for Drinking-water Quality," vol. 1, no. 3rd, 2006.

[17] D. Setiabudidaya, "Jupyter notebook app: Alternatif teknologi pembelajaran fisika berbasis web browser," in *Annual Research Seminar (ARS)*, 2015, vol. 1, no. 1, hal. Annual Research Seminar (ARS), [Daring]. Tersedia pada: https://www.scoutsecuador.org/site/sites/default/files/%5Bbiblioteca%5D/5.1 Conservacion de alimentos y Recetas

sencillas.pdf%0Ahttp://publications.lib.chalmers.se/records/fulltext/245180/245180.pdf%0Ahttps://hdl.handle.net/20.500.12380/245180%0Ahttp://dx.d.

[18] J. Han, M. Kamber, dan J. Pei, *Data Mining Concepts and Techniques Third*. 2012.

[19] I. Pardoe, *Applied Regression Modeling: A Business Approach*. 2012.

[20] D. Sarkar, R. Bali, dan T. Sharma, *Practical Machine Learning with Python*. Berkely: Apress, 2018.

[21] H. Li, J. Li, P. C. Chang, dan J. Sun, "Parametric prediction on default risk of Chinese listed tourism companies by using random oversampling, isomap, and locally linear embeddings on imbalanced samples," *Int. J. Hosp. Manag.*, vol. 35, hal. 141–151, 2013, doi: 10.1016/j.ijhm.2013.06.006.

[22] M. Raihan-Al-Masud dan M. Rubaiyat Hossain Mondal, "Data-driven diagnosis of spinal abnormalities using feature selection and machine learning algorithms," *PLoS One*, vol. 15, no. 2, hal. 1–21, 2020, doi: 10.1371/journal.pone.0228422.

[23] B. George, "A study of the effect of random projection and other dimensionality reduction techniques on different classification methods.," *A Biannu. J. Interdiscip. Stud. Res.*, vol. XVIII, no. 01, 2017.

[24] A. C. Mueller dan S. Guido, *Introduction to machine learning with Python*. 2016.

[25] J. D. Kelleher, B. Mac Namee, dan A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics : Algorithms, Worked Examples, and Case Studies*, Second. Cambridge, Massachusetts: The MIT Press, 2020.

[26] A. Zheng dan A. Casari, *Feature Engineering for Machine Learning and Data Analytics*. Sebastopol, CA: O'Reilly Media, Inc, 2018.

[27] Y. Y. Song dan Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Arch. Psychiatry*, vol. 27, no. 2, hal. 130–135, 2015, doi: 10.11919/j.issn.1002-0829.215044.

[28] I. Handayani, "Application of K-Nearest Neighbor Algorithm on Classification of Disk Hernia and Spondylolisthesis in Vertebral Column," *Indones. J. Inf. Syst.*, vol. 2, no. 1, hal. 57, 2019, doi: 10.24002/ijis.v2i1.2352.