

# Klasifikasi Kualitas Air Sungai Citarum Menggunakan Metode K-Nearest Neighbors, Support Vector Machine, dan Random Forest

1<sup>st</sup> Delatifa Putri Sugandi  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

delatifaps@student.telkomuniversity.ac.id

2<sup>nd</sup> Meta Kallista  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

3<sup>rd</sup> Ig. Prasetya Dwi Wibawa  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

prasdwiwawa@telkomuniversity.ac.id

**Abstrak** — Sungai merupakan sumber kehidupan di lingkungan sekitar baik untuk keperluan rumah tangga, pertanian maupun industri. Karena sungai memiliki peranan yang begitu penting, maka perlu dilakukan pemeliharaan kualitas air agar tidak berdampak buruk bagi lingkungan sekitar. Padahal, kualitas air Sungai Citarum pernah menjadi sungai paling tercemar dan kotor di dunia. Kualitas air sungai dapat ditentukan dengan menghitung nilai indeks pencemaran secara manual. Namun, metode perhitungan manual dianggap membutuhkan waktu yang lama. Oleh karena itu, diperlukan suatu teknik yang efektif untuk mengukur kualitas air sungai. *Machine learning* dipilih sebagai salah satu teknik yang diusulkan untuk mengklasifikasikan kualitas air sungai. Tiga metode klasifikasi diterapkan dalam penelitian ini. *K-Nearest Neighbors*, *Support Vector Machine*, dan *Random forest*. Setelah dilakukan pengujian, didapatkan kesimpulan bahwa menggunakan *random forest* memberikan hasil yang lebih baik antara metode *K-Nearest Neighbors* dan metode *Support Vector Machine* dengan akurasi sebesar 99,24% dan jika model *random forest* tersebut ditambahkan dengan metode *AdaBoost* maka akurasi meningkat menjadi 99,34%.

**Kata kunci**— Klasifikasi, Kualitas Air Sungai, KNN, SVM, *Random forest*.

## I. PENDAHULUAN

Air merupakan sumber kehidupan bagi semua makhluk hidup yang ada di bumi, khususnya manusia[1]. Hampir 71% permukaan bumi ditutupi oleh air. Air adalah hal yang paling penting bagi semua makhluk hidup seperti tanaman, hewan dan manusia. Air sungai juga memiliki peran yang penting dalam kehidupan. Sungai melakukan banyak hal, seperti memberikan air untuk pertanian, industri, tumbuhan, dan hewan serta untuk minum. Selain itu, banyak industri dan masyarakat yang membuang limbah industri atau dapur ke sungai, menyebabkan pencemaran. Karena peran penting air dalam kehidupan, dikatakan bahwa kualitas air dapat menunjukkan tingkat kesehatan seseorang.

Keputusan Menteri Lingkungan Hidup No. 115 Tahun 2003 tentang Pedoman Penetapan Kualitas Air Saat Ini menjelaskan metode indeks pencemaran air (IP) untuk

menempatkan kualitas air sungai dalam empat kategori: baik, cemar ringan, cemar sedang, dan cemar berat[1].

Pada umumnya, kualitas air sungai dapat diketahui dengan melakukan perhitungan manual menggunakan metode perhitungan indek pencemaran (IP) dan terdapat beberapa metode lainnya. Namun, perhitungan manual dirasa kurang efektif karena membutuhkan waktu yang cukup lama dan ketelitian yang cukup dalam proses perhitungannya. Oleh karena itu, dibutuhkan suatu sistem yang dapat mengklasifikasikan kualitas air. Teknologi bernama pembelajaran mesin saat ini sudah banyak diimplementasikan untuk membantu kehidupan manusia. Yang salah satunya dapat dijadikan sebagai solusi ini adalah sebuah metode klasifikasi menggunakan *machine learning*.

Pada penelitian ini, dilakukan percobaan untuk melakukan klasifikasi kualitas air sungai dengan menggunakan *machine learning*. Algoritma yang akan dilakukan adalah algoritma *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), dan *Random forest*. Percobaan ini dilakukan untuk mengetahui algoritma terbaik pada dataset kualitas air sungai Citarum.

## II. KAJIAN PUSTAKA

### A. Klasifikasi

Klasifikasi adalah salah satu metode *supervised learning*, yaitu pembelajaran data yang melibatkan penggunaan kumpulan data yang telah diberi label[2]. *Supervised Learning* adalah salah satu metode pembelajaran mesin yang menggunakan kumpulan data (*training data*) yang telah diberi label data untuk melakukan proses pembelajaran mesin. Ini memungkinkan mesin pembelajaran untuk mengidentifikasi dan mengklasifikasikan data yang dimasukkan untuk memprediksi dan mengklasifikasikan [3].

### B. *Synthetic Minority Oversampling Technique* (SMOTE)

Ketidakeimbangan data terjadi ketika satu kelas memiliki jumlah yang jauh lebih tinggi dari yang lain, yang menyebabkan penurunan kinerja klasifikasi pada kelas minoritas [4]. Performa algoritma pembelajaran mesin sering

dievaluasi berdasarkan keakuratan hasil prediksi, namun hal ini tidak sesuai jika terjadi ketidakseimbangan kelas [4]. SMOTE adalah metode *oversampling* di mana data kelas minoritas direplikasi menggunakan data agregat dari replikasi data kelas minoritas. Sampling SMOTE mengambil instance kelas minoritas, kemudian menemukan k tetangga terdekat dari setiap *instance*, dan kemudian membuat *instance* komposit alih-alih menduplikasi instance kelas minoritas [5].

### C. K-Nearest Neighbors (KNN)

Metode *K-Nearest Neighbors* (KNN) merupakan metode klasifikasi klasik yang paling sederhana. Metode KNN sering juga disebut dengan *Instance Based Learning*, KNN melakukan klasifikasi terhadap objek berdasarkan jarak antara objek tersebut dengan objek lain. Metode KNN menggunakan prinsip ketetangaan untuk memprediksi kelas yang baru[4].

Metode pengajaran mesin yang diawasi yang paling sederhana adalah KNN[6]. Metode KNN biasanya menemukan data yang paling mirip dengan menggunakan rumus jarak geometri [7]. Nilai K ideal untuk algoritma ini bergantung pada data; nilai K yang lebih tinggi biasanya dapat mengurangi efek noise pada klasifikasi, tetapi juga dapat membuat batas klasifikasi lebih berbahaya. Beberapa metode, seperti *Euclidean Distance* dan *Manhattan Distance* [1], dapat digunakan untuk mengukur kedekatan antara data baru dan data pelatihan.

Berikut adalah formula untuk jarak Manhattan:

$$d(x, X) = \sum_{i=1}^n |x_i - X_i| \quad (1)$$

### D. Support Vector Machine (SVM)

Algoritma SVM mengubah data pelatihan awal ke dimensi yang lebih tinggi dengan menggunakan pemetaan nonlinier. Dalam hal ini, akan mencari *hyperplane* untuk memisahkan secara linier untuk dimensi baru, dan dengan pemetaan nonlinier yang tepat ke dimensi yang lebih tinggi, data dari dua kelas selalu dapat dipisahkan dengan *hyperplane* tersebut. Teknik ini berusaha untuk menemukan fungsi pemisah (klasifier) terbaik untuk membedakan dua kelas yang berbeda. Ini berusaha untuk menemukan fungsi pemisah (*hyperplane*) terbaik dari berbagai fungsi yang tersedia. *Hyperplane* yang terletak di tengah-tengah antara dua set objek dari dua kelas adalah yang terbaik[8].

### E. Random forest

Metode *Random forest* digunakan untuk membangun pohon keputusan yang terdiri dari *node root*, *inner*, dan *leaf* dengan mengambil atribut dan data acak sesuai layout yang sesuai. Metode ini dapat meningkatkan hasil akurasi karena pembangkitan node anak untuk setiap node dilakukan secara acak[9]. Simpul teratas atau akar dari pohon keputusan disebut simpul akar. Node terakhir, leaf node atau terminal node memiliki satu input dan satu output. Sedangkan node dalam adalah node percabangan dengan minimal dua input dan satu output[9]. Dimulai dengan menghitung nilai entropy, pohon keputusan menentukan tingkat impuritas atribut dan nilai informasi yang diperoleh. [9].

### F. Adaptive Boosting (AdaBoost)

Boosting diklasifikasikan sebagai *ensemble learning* dan merupakan teknik untuk menggabungkan beberapa pengklasifikasi yang lemah untuk membuat prediksi yang sangat akurat. Lebih fleksibel dan tidak terlalu sensitive terhadap data dapat dicapai dengan melatih beberapa model individu secara berurutan. Setiap model mengambil pelajaran dari kesalahan model sebelumnya.

Algoritma ini menggunakan konsep pengajaran yang diawasi untuk melakukan klasifikasi. Algoritma AdaBoost meningkatkan akurasi prediksi yang dibuat melalui penggunaan fungsi *exponential loss*. Tujuan dari algoritma ini adalah untuk membuat algoritma berkonsentrasi pada pengklasifikasi yang lemah dengan memberikan prioritas lebih besar pada pengklasifikasi yang lemah. Memberikan prioritas lebih besar pada pengklasifikasi yang lemah. Pengklasifikasi lemah adalah klasifikasi yang menghasilkan prediksi yang sedikit lebih buruk daripada tebakan acak[10].

### G. Evaluasi

*Confusion matrix* adalah tabel yang membandingkan label kelas sebenarnya untuk setiap kelas dengan label kelas yang diharapkan untuk semua data sampel[11]. Baris matriks konfusi mencerminkan tampilan kelas yang sebenarnya, sedangkan kolom menunjukkan contoh kelas yang diharapkan. Matriks ini menggeneralisasikan contoh data positif, data yang diidentifikasi dengan benar oleh algoritme sebagai positif disebut *true positive* (TP), sedangkan data yang salah diklasifikasikan sebagai negatif disebut *negative false* (FN). Di sisi lain, negatif yang diidentifikasi dengan benar sebagai negatif sejati disebut negatif sejati (TN), sedangkan negatif yang diprediksi salah sebagai positif disebut positif palsu atau *false positive* (FP) [12].

TABEL 1  
Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Setelah mengetahui pengelompokan berdasarkan tabel Confusion Matrix, dapat dilakukan perhitungan untuk mendapatkan akurasi, presisi, *recall*, dan *F1-score*.

$$\text{Accuracy} = \frac{TP+TN}{\text{Total Data}} \quad (2)$$

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

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

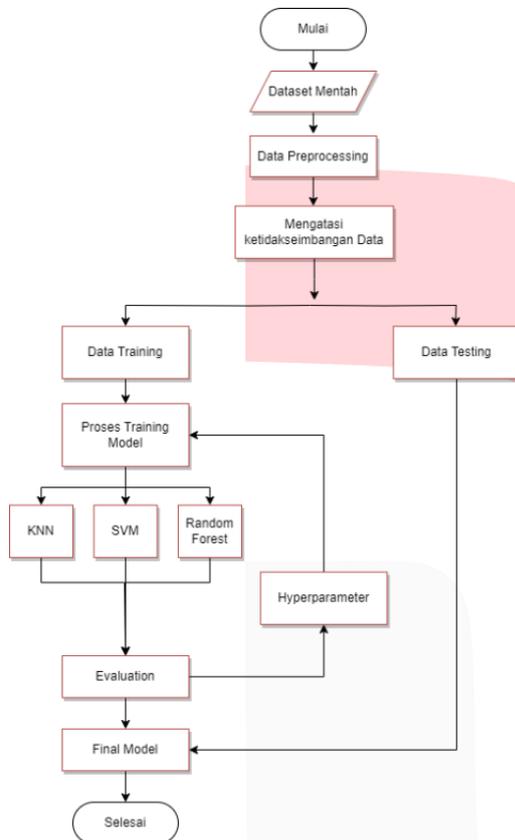
$$\text{F1 score} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Menurut formula di atas, akurasi adalah rasio jumlah sampel yang diprediksi dengan benar terhadap total sampel input. Presisi menunjukkan seberapa baik kinerja pengklasifikasi dalam memprediksi sampel positif. *Recall* adalah metrik penting untuk set data yang tidak seimbang, yang menunjukkan rasio sampel positif yang diprediksi dengan semua sampel yang relevan, yang menunjukkan sampel yang benar-benar positif. *F1-Score* menunjukkan

keseimbangan antara *recall* dan ketepatan pengklasifikasi. Semakin dekat nilainya dengan satu, semakin seimbang mengingat dan presisi[13].

### III. METODE PENELITIAN

Secara keseluruhan, penelitian ini akan dilakukan sesuai dengan diagram yang terdapat dibawah ini.



GAMBAR 1  
Flowchart Penelitian

#### A. Pengumpulan Data

Dataset untuk penelitian ini disediakan oleh Dinas Lingkungan Hidup Provinsi Jawa Barat. Parameter kualitas air sungai terdapat dalam dataset. Dataset ini memuat informasi pengukuran kualitas air Sungai Citarum dari tahun 2018 hingga 2022. Terdapat delapan faktor antara lain pH, TSS, COD, DO, BOD, Nitrat, fecal coliform dan fosfat, yang dapat digunakan untuk menilai kualitas air sungai. Pengambilan data dilakukan di 8 lokasi di sepanjang Sungai Citarum dengan total 2500 data.

TABEL 2  
Parameter pada Dataset

Parameter	Satuan	Tipe Data
pH	-	float
TSS	mg/L	float
COD	mg/L	float
DO	mg/L	float
BOD	mg/L	float

Nitrat	mg/L	float
Total Fosfat	mg/L	float
Fecal coliform	Jlm./100 L	int

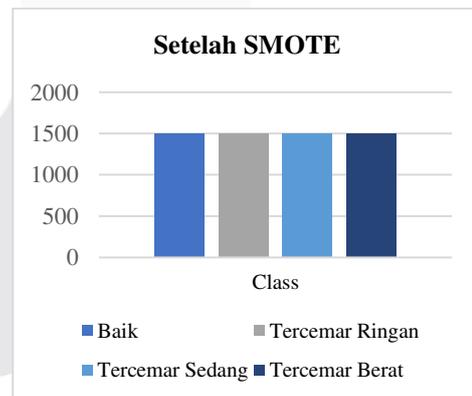
Pada implementasi *machine learning*, delapan parameter diatas akan menjadi variabel bebas atau fiturnya. Sedangkan yang menjadi variabel terikat adalah label atau kategori kelas seperti pada tabel 3.

TABEL 3  
Keterangan Label Kelas

Kualitas	Label Kelas
Kondisi Baik	1
Tercemar Ringan	2
Tercemar Sedang	3
Tecemar Berat	4

### IV. HASIL DAN PEMBAHASAN

Preprocessing data menunjukkan ketidakseimbangan kelas klasifikasi. Ketidakseimbangan data dapat mengakibatkan kelas minoritas cenderung salah diklasifikasikan karena model lebih banyak belajar dengan kelas mayoritas. Akibatnya, langkah-langkah harus diambil untuk menyeimbangkan jumlah data untuk setiap kelas. SMOTE dipilih untuk menyelesaikan masalah ini. Sebelum *oversampling*, ada 2500 data, dengan 948 data untuk kelas 1 "Kondisi Baik", 1496 data untuk kelas 2 "Tercemar Ringan", 39 data untuk kelas 3 "Tercemar Sedang", dan 17 data untuk kelas 4 "Tercemar Berat". Setelah *oversampling*, total data menjadi 5984 data, dengan persebaran data setiap kelas berjumlah 1496 data. Gambar 2 menunjukkan jumlah masing-masing kelas setelah SMOTE.



GAMBAR 2  
Diagram Batang Data Dalam Keadaan Seimbang

Setelah jumlah setiap kelas dalam keadaan seimbang, dilanjutkan dengan proses pemodelan algoritma. Untuk semua algoritma, baik KNN, SVM, dan *random forest* dilakukan pembagian data pelatihan dan pengujian dengan perbandingan 80:20 yang berjumlah 4787 data untuk data pelatihan dan 1197 data untuk data uji.

Berikut adalah paparan mengenai hasil penelitian dari setiap percobaan algoritma yang dilakukan.

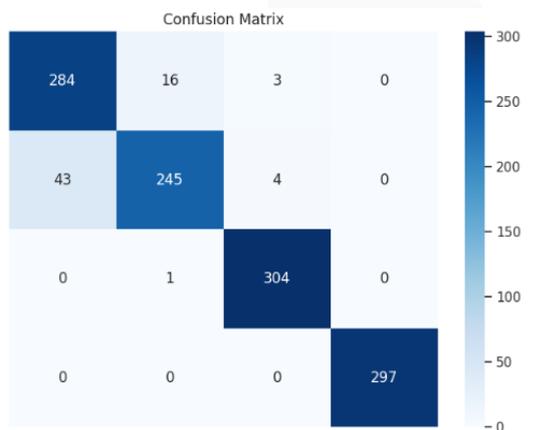
1. *K-Nearest Neighbors (KNN)*

Berikut adalah hasil evaluasi pada model KNN dengan penggunaan nilai  $K=15$  dan matiks yang digunakan adalah Manhattan.

TABEL 4  
Classification Report KNN

KNN Akurasi : 94.40%				
	Presisi	Recall	F1-Score	Support
Kelas 1	0.87	0.94	0.90	303
Kelas 2	0.94	0.84	0.88	292
Kelas 3	0.98	1.00	0.99	305
Kelas 4	1.00	1.00	1.00	297

Tabel *Classification report* diatas dengan nilai presisi, *recall*, *F1-Score* sama dengan 1 pada kelas 4 menunjukkan bahwa model KNN ini sangat baik dalam mengklasifikasikan kelas 4, namun masih kurang baik dalam mengklasifikasikan kelas 1, kelas 2, dan kelas 3.



GAMBAR 3  
Confusion Matrix Model KNN

Pada gambar *confusion matrix* KNN diatas, dapat terlihat bahwa 303 data test untuk kelas 1 diklasifikasikan sebanyak 284 data terklasifikasi sebagai kelas 1, 16 data terklasifikasi sebagai kelas 2, dan 3 data terklasifikasi sebagai kelas 3. Kemudian, 292 data uji untuk kelas 2 diklasifikasikan sebanyak 43 data terklasifikasi sebagai kelas 1, 245 data terklasifikasi sebagai kelas 2, dan 4 data terklasifikasi sebagai kelas 3. Kemudian, 305 data uji untuk kelas 3 diklasifikasikan 1 data untuk kelas 2 dan 304 data terklasifikasi sebagai kelas 3. Terakhir pada kelas 4 terdapat total data uji 297 dan semua data benar terklasifikasi pada kelas 4.

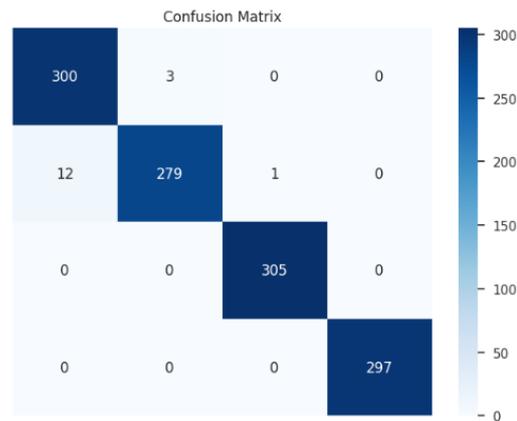
2. *Support Vector Machine (SVM)*

Berikut adalah hasil evaluasi pada model SVM dengan penggunaan nilai  $C=10$ ,  $\text{Gamma}=0.1$ , dan kernel yang digunakan adalah Linear.

TABEL 5  
Classification Report SVM

SVM Akurasi : 98.66%				
	Presisi	Recall	F1-Score	Support
Kelas 1	0.96	0.99	0.98	303
Kelas 2	0.99	0.96	0.97	292
Kelas 3	1.00	1.00	1.00	305
Kelas 4	1.00	1.00	1.00	297

Tabel *Classification report* diatas dengan nilai presisi, *recall*, *F1-Score* sama dengan 1 pada kelas 3 dan kelas 4 menunjukkan bahwa model SVM ini sangat baik dalam mengklasifikasikan kelas 3 dan kelas 4, namun masih kurang baik dalam mengklasifikasikan kelas 1 dan kelas 2.



GAMBAR 4  
Confusion Matrix Model SVM

Pada gambar *confusion matrix* SVM diatas, dapat terlihat bahwa 303 data test untuk kelas 1 diklasifikasikan sebanyak 300 data terklasifikasi sebagai kelas 1 dan 3 data terklasifikasi sebagai kelas 2. Kemudian, 292 data uji untuk kelas 2 diklasifikasikan sebanyak 12 data terklasifikasi sebagai kelas 1, 279 data terklasifikasi sebagai kelas 2, dan 1 data terklasifikasi sebagai kelas 3. Kemudian, 305 data uji untuk kelas 3 diklasifikasikan 305 data terklasifikasi sebagai kelas 3. Terakhir pada kelas 4 terdapat total data uji 297 dan semua data benar terklasifikasikan pada kelas 4.

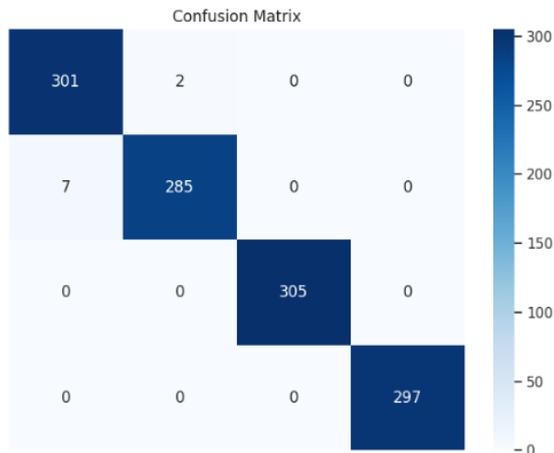
3. *Random forest*

Berikut adalah hasil evaluasi pada model *random forest* dengan penggunaan nilai  $n\_estimators=100$ ,  $max\_depth=7$ , dan *criterion* yang digunakan adalah Gini.

TABEL 6  
*Classification Report random forest*

<b>Random forest Akurasi : 99.24%</b>				
	Presisi	Recall	F1-Score	Support
Kelas 1	0.98	0.99	0.99	303
Kelas 2	0.99	0.99	0.99	292
Kelas 3	1.00	1.00	1.00	305
Kelas 4	1.00	1.00	1.00	297

Tabel *Classification report* diatas dengan nilai presisi, *recall*, *F1-Score* sama dengan 1 pada kelas 3 dan kelas 4 menunjukkan bahwa model *random forest* ini sangat baik dalam mengklasifikasikan kelas 3 dan kelas 4, namun masih kurang baik dalam mengklasifikasikan kelas 1 dan kelas 2.



GAMBAR 5  
Confusion Matrix Model random forest

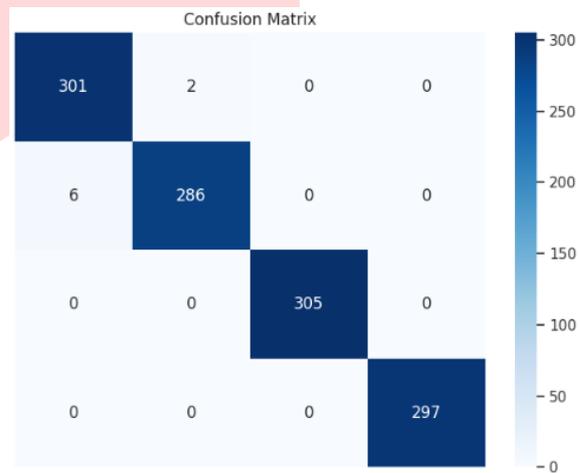
Pada gambar *confusion matrix random forest* diatas, dapat terlihat bahwa 303 data test untuk kelas 1 diklasifikasikan sebanyak 301 data terklasifikasi sebagai kelas 1 dan 2 data terklasifikasi sebagai kelas 2. Kemudian, 292 data uji untuk kelas 2 diklasifikasikan sebanyak 7 data terklasifikasi sebagai kelas 1 dan 285 data terklasifikasi sebagai kelas 2. Kemudian, 305 data uji untuk kelas 3 diklasifikasikan 305 data terklasifikasi sebagai kelas 3. Terakhir pada kelas 4 terdapat total data uji 297 dan semua data benar terklasifikasi pada kelas 4.

Selanjutnya dilakukan percobaan dengan menambahkan metode Adaboost untuk meningkatkan akurasi *random forest*. Berikut adalah tabel *Classification Report*.

TABEL 7  
*Classification Report random forest + Adaboost*

<b>Random forest Akurasi : 99.33%</b>				
	Presisi	Recall	F1-Score	Support
Kelas 1	0.98	0.99	0.99	303
Kelas 2	0.99	0.98	0.99	292
Kelas 3	1.00	1.00	1.00	305
Kelas 4	1.00	1.00	1.00	297

Tabel *Classification report* diatas dengan nilai presisi, *recall*, *F1-Score* sama dengan 1 pada kelas 3 dan kelas 4 menunjukkan bahwa model *random forest* ditambah AdaBoost ini sangat baik dalam mengklasifikasikan kelas 3 dan kelas 4, namun masih kurang baik dalam mengklasifikasikan kelas 1 dan kelas 2.



GAMBAR 6  
Confusion Matrix Model random forest + AdaBoost

Pada *Confusion matrix* model *random forest + AdaBoost* terlihat sama dengan *Confusion Matrix random forest*. Perbedaannya hanya terletak di peningkatan kondisi klasifikasi benar pada kelas 2 yang semula berjumlah 285 data menjadi 286 data. Dengan 303 data test untuk kelas 1 diklasifikasikan sebanyak 301 data terklasifikasi sebagai kelas 1 dan 2 data terklasifikasi sebagai kelas 2. Kemudian, 292 data uji untuk kelas 2 diklasifikasikan sebanyak 6 data terklasifikasi sebagai kelas 1 dan 286 data terklasifikasi sebagai kelas 2. Kemudian, 305 data uji untuk kelas 3 diklasifikasikan 305 data terklasifikasi sebagai kelas 3. Terakhir pada kelas 4 terdapat total data uji 297 dan semua data benar terklasifikasi pada kelas 4.

Model klasifikasi yang telah dibuat kemudian dibandingkan untuk melihat nilai akurasi yang didapatkan dari ketiga model yang telah dilakukan percobaan. Berikut adalah perbandingan hasil dari ketiga model klasifikasi.

TABEL 8  
Perbandingan Akurasi Model Klasifikasi

Model	Akurasi
<i>K-nearest Neighbors</i>	94.40 %
<i>Support Vector Machine</i>	98.66 %
<i>Random forest</i>	99.24 %
<i>Random forest + AdaBoost</i>	99.33 %

## V. KESIMPULAN

Perbandingan algoritma *machine learning* yang telah dilakukan pada dataset kualitas air sungai Citarum menghasilkan nilai akurasi yang cukup akurat yaitu pada metode KNN didapatkan hasil akurasi sebesar 94.40 %, SVM mendapatkan akurasi sebesar 98.66%, dan *random forest* mendapatkan akurasi sebesar 99.24 %. Penelitian ini menunjukkan bahwa penggunaan algoritma *random forest* memiliki akurasi yang lebih tinggi dibandingkan KNN dan SVM. Jika metode *random forest* ditambahkan dengan metode AdaBoost maka didapatkan akurasi menjadi 99.33%. Sehingga, penggunaan metode AdaBoost pada penelitian ini terbukti dapat meningkatkan akurasi model *random forest*.

## REFERENSI

- [1] A. Danades, D. Pratama, D. Anggraini, and D. Anggriani, "Comparison of accuracy level K-Nearest Neighbor algorithm and support vector machine algorithm in classification water quality status," in *Proceedings of the 2016 6th International Conference on System Engineering and Technology, ICSET 2016*, Institute of Electrical and Electronics Engineers Inc., Feb. 2017, pp. 137–141. doi: 10.1109/FIT.2016.7857553.
- [2] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python," *Bina Insani ICT Journal*, vol. 7, no. 2, pp. 156–165, 2020.
- [3] D. Atika, A. Ari Aldino, S. Informasi, J. Pagar Alam No, L. Ratu, and K. Kedaton, "TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY SUPPORT VECTOR MACHINE UNTUK ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP TEKANAN MENTAL PADA MEDIA SOSIAL TWITTER," 2022. [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>
- [4] R. Siringoringo, "Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan K-Nearest Neighbor," *Jurnal ISD*, vol. 3, no. 1, 2018.
- [5] J. T. Kumalasari and A. Merdekawati, "Analisis Sentimen Terhadap Program Kampus Merdeka Pada Twitter Menggunakan Metode Naive Bayes, Union dan Synthetic Minority Sampling Technique (SMOTE)," *SATIN-Sains dan Teknologi Informasi*, vol. 9, no. 1, pp. 01–12, 2023, doi: 10.33372.
- [6] R. Kini M G and R. Bhandarkar, "Quality Assessment of Seed Using Supervised Machine Learning Technique," *Journal of The Institution of Engineers (India): Series B*, 2023, doi: 10.1007/s40031-023-00902-2.
- [7] J. Verma, M. Nath, P. Tripathi, and K. K. Saini, "Analysis and identification of kidney stone using Kth nearest neighbour (KNN) and support vector machine (SVM) classification techniques," *Pattern Recognition and Image Analysis*, vol. 27, no. 3, pp. 574–580, Jul. 2017, doi: 10.1134/S1054661817030294.
- [8] A. S. Ritonga and E. S. Purwaningsih, "PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) DALAM KLASIFIKASI KUALITAS PENGELASAN SMAW (SHIELD METAL ARC WELDING)," 2018.
- [9] Y. Sulisty Nugroho and dan Nova Emiliyawati, "Sistem Klasifikasi Variabel Tingkat Penerimaan Konsumen Terhadap Mobil Menggunakan Metode Random Forest." [Online]. Available: <http://archive.ics.uci.edu/ml/>
- [10] J. Elektronik and I. Komputer Udayana, "Penerapan Metode Adaboost Untuk Multi-Label Classification Pada Dokumen Teks".
- [11] Á. A. Cabrera *et al.*, "Zeno: An Interactive Framework for Behavioral Evaluation of Machine Learning," in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, Apr. 2023. doi: 10.1145/3544548.3581268.
- [12] D. Chicco, N. Tötsch, and G. Jurman, "The matthews correlation coefficient (Mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation," *BioData Min*, vol. 14, pp. 1–22, 2021, doi: 10.1186/s13040-021-00244-z.
- [13] S. Hussain, S. S. Oyelere, J. H. Joloudari, A. Marefat, J. H. Joloudari, and M. A. Nematollahi, "Effective Class-Imbalance learning based on SMOTE and Convolutional Neural Networks." [Online]. Available: <https://www.researchgate.net/publication/363269818>