

# Klasifikasi Kualitas Air Layak Minum menggunakan Algoritma *Random Forest Classifier* dan *GridsearchCV*

Gusti Agung Diah Sri Ari Ningsih<sup>a1</sup>, Cokorda Pramatha<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Udayana, Bali  
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia  
<sup>1</sup>gstadiahs@gmail.com  
<sup>2</sup>cokorda@unud.ac.id

## Abstract

*Drinkable water is water that is healthy for humans to drink and does not pose significant health risks. To determine whether water has a quality that meets health standards can be determined through the substances or minerals contained in it. Conventional methods require quite a long time to evaluate and classify water quality as suitable for consumption or not. One approach that can be used to overcome this problem is to utilize machine learning. This research uses a random forest to carry out classification. Using random forest by default cannot produce optimal performance because the parameters used are not necessarily the best. Therefore, this research also uses GridsearchCV to find optimal hyperparameter values in the Random Forest Classifier. After hyperparameter tuning, an optimal model was obtained with each parameter  $n\_estimators$  100,  $max\_depth$  9,  $max\_features$  4, and  $min\_samples\_split$  2. The performance of Random Forest after hyperparameter tuning increased accuracy, which was initially 76% increase to 84%, precision which was initially 76.19% increase to 81.70%, recall which was initially 74.89% increase to 85.53%, and f1-score which was initially 75.53%, increase to 83.57%.*

**Keywords:** Classification, Drinking Water Quality, Random Forest, Optimal Hyperparameters, Hyperparameter Tuning, GridSearchCV

## 1 Pendahuluan

Pesatnya pembangunan dan pertumbuhan penduduk menyebabkan terjadinya penurunan kualitas air. Hal ini dikarenakan semakin meningkatnya aktivitas manusia dan kompleksitas industri akan mengakibatkan tingkat pencemaran semakin meningkat yang berdampak pada keseimbangan perairan, kebutuhan sanitasi yang memadai dan akses terhadap air minum sehat untuk konsumsi manusia [3]. Berdasarkan data *World Health Organization* (WHO) sebanyak 663 juta penduduk kesulitan dalam memperoleh air bersih [4]. *World Health Organization* (WHO) menjelaskan bahwa setiap tahunnya sebanyak 829 ribu orang meninggal karena air minum, sanitasi, dan kebersihan tangan yang tidak aman [5]. Setiap tahun sekitar dua juta bayi kehilangan nyawa dikarenakan kurangnya akses air minum yang bersih [6]. Perlu dilakukan suatu prediksi untuk mengetahui air yang akan dikonsumsi layak atau tidak.

Keberadaan zat-zat yang terdapat didalam air dapat menjadi indikator untuk menentukan kualitas air. Umumnya penilaian kualitas air menggunakan metode perhitungan manual seperti menghitung *Water Quality Index* (WQI) [2]. *Water Quality Index* (WQI) merupakan suatu metode untuk menilai kualitas air dengan mempertimbangkan beberapa karakteristik yang merepresentasikan kondisi air [7]. Terdapat sembilan faktor yang digunakan dalam perhitungan WQI secara tradisional meliputi PH, *Hardness*, *Solids*, *Chloramines*, *Sulfate*, *Conductivity*, *Organic\_Carbon*, *Trihalomethanes*, dan *Turbidity*. *Water Quality Index* (WQI) dibuat berdasarkan pada standar yang direkomendasikan WHO [8]. Metode konvensional bergantung pada pemahaman terhadap parameter-parameter yang telah ditetapkan untuk mengevaluasi dan mengategorikan kualitas air sebagai air yang aman atau tidak untuk dikonsumsi. Metode konvensional seringkali memerlukan waktu lama dalam perhitungannya, sehingga membutuhkan sistem yang dapat bekerja secara otomatis.

Pemanfaatan algoritma *machine learning* dapat diterapkan untuk mengatasi tantangan ini. *Machine learning* memiliki dua aplikasi utama yaitu klasifikasi dan prediksi. Klasifikasi adalah

proses pemberian label atau kategori tertentu yang sudah ditetapkan sebelumnya pada objek dalam data yang akan diklasifikasikan [9]. Dalam kasus ini klasifikasi digunakan untuk menentukan apakah air layak untuk dikonsumsi atau tidak. Penelitian sebelumnya mengenai klasifikasi kualitas air minum telah dilakukan oleh [4] dengan membandingkan berbagai algoritma klasifikasi seperti *Logistic Regression*, *Support Vector Machine*, *K-NN*, *Decision Trees*, *Naïve Bayes*, *Random Forest Classifier*, *XGBosst Classifier*, dan *Artificial Neural Network* dalam menilai apakah air dapat dikonsumsi atau tidak dapat dikonsumsi, hasil menunjukkan bahwa model klasifikasi *Random Forest Classifier* mencapai akurasi tertinggi sebesar 72,81%. Penggunaan *random forest* secara default yaitu tanpa dilakukan optimasi tidak dapat mencapai kinerja optimal dikarenakan belum tentu parameter yang digunakan adalah parameter yang paling optimal pada model [10].

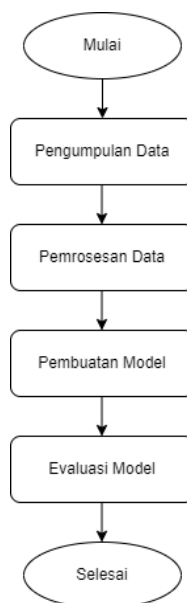
Mengatur *hyperparameter* atau *hyperparameter tuning* merupakan salah satu metode untuk meningkatkan performa model klasifikasi [11]. Hal ini dikarenakan kinerja *random forest* sangat dipengaruhi oleh pengaturan *hyperparameter* yang dipilih. Optimasi akan memberikan hasil yang lebih akurat [12]. Berdasarkan penelitian yang dilakukan oleh [13] memperoleh hasil akurasi setelah dioptimasi 94,8%, akurasi mengalami peningkatan sebesar 21,5% dari 73,31%. Selain itu penelitian oleh [10] yang memanfaatkan algoritma *random forest* untuk mengklasifikasikan malware dan melakukan penyetelan *hyperparameter* menggunakan *GridsearchCV* berhasil meningkatkan kinerja model dengan mencapai tingkat akurasi sebesar 99,23%. Mengacu pada hasil penelitian sebelumnya maka dilakukan penelitian dengan judul *Klasifikasi Kualitas Air Layak Minum dengan Algoritma Random Forest Classifier dan GridsearchCV*

## 2 Metode Penelitian

Metode penelitian merupakan serangkaian langkah yang dilakukan untuk menyelesaikan masalah selama proses penelitian dengan mematuhi aturan dan pedoman yang telah ditetapkan oleh peneliti untuk mencapai tujuan yang diinginkan serta memperoleh hasil yang optimal [14]. Penelitian ini terdiri dari empat tahapan utama meliputi

1. Pengumpulan data
2. Pemrosesan data
3. Pembuatan model
4. Evaluasi model

Tahapan penelitian ini ditunjukkan pada Gambar 1



**Gambar 1.** Alur Penelitian

Pada Gambar 1 menunjukkan alur penelitian yang menggambarkan tahapan-tahapan yang dilakukan pada penelitian ini. Tahapan penelitian pada penelitian ini terdiri dari tahap pengumpulan data, pemrosesan data, pembuatan model dan evaluasi model. Penjelasan rinci mengenai alur tersebut disajikan pada bagian berikut

## 2.1 Pengumpulan Data

Data pada penelitian ini menggunakan data sekunder yang diambil dari *Water Quality* Dataset yang diperoleh melalui website kaggle.com dengan jumlah data 3276 dan 10 atribut. Dataset ini memiliki 2 kelas output yaitu nilai 1 yang artinya air dapat dikonsumsi dan nilai 0 artinya air tidak dapat dikonsumsi. Berikut merupakan tabel atribut dan domain yang digunakan

**Tabel 1.** Atribut dan Domain Dataset

Atribut	Deskripsi	Domain
Ph	Indikator yang dipakai untuk menilai derajat keasaman atau kebasaaan larutan.	0 – 14
Hardness	Konsentrasi garam-garam mineral tertentu, terutama kalsium dan magnesium yang larut dalam air.	47,432 – 323,124
Solids	Jumlah garam anorganik dan kandungan bahan organik yang terlarut dalam air.	320,94 – 61227,19
Chloramines	Kandungan kimia yang dihasilkan dari interaksi klorin dalam air dan amonia atau senyawa nitrogen lainnya.	0,352 – 13,127
Sulfate	Bahan alami yang terdapat dalam mineral, tanah, dan batuan.	129 – 481.03
Conductivity	Kadar ion positif (kation) dan ion negatif (anion) dalam air.	181,4 – 753,3
Organic Carbon	Materi yang berasal dari senyawa yang terurai dalam air.	2,2 – 28,3
Trihalomethanes	Senyawa yang terbentuk dalam air minum sebagai hasil dari penambahan klorin ke zat organik.	0,738 – 124
Turbidity	Ukuran kekeruhan atau kekeruhan relatif air yang disebabkan oleh partikel-paditersuspensi yang terdapat dalam air.	1,45 – 6,739
Potability	Indikator apakah air tersebut layak untuk diminum atau tidak.	0 – 1

Pada tabel 1 menjelaskan 10 atribut yang ada didalam dataset beserta deskripsi dan rentang nilainya. Atribut-atribut tersebut antara lain pH, *hardness*, *solids*, *chloramines*, *sulfate*, *conductivity*, *organic carbon*, *trihalomethanes*, *turbidity*, dan *potability*. Deskripsi menjelaskan setiap atribut, sementara kolom domain menunjukkan rentang nilai yang ada pada setiap atribut

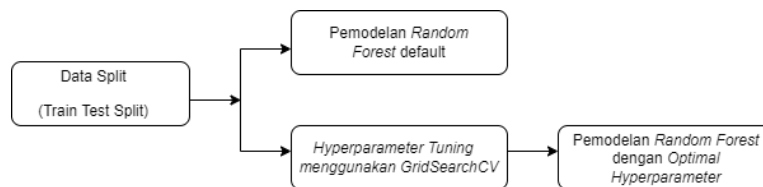
## 2.2 Pemrosesan Data

Tahapan pemrosesan data meliputi penanganan data yang memiliki nilai null atau kosong dengan cara menghapus nilai null tersebut. Pemrosesan data selanjutnya adalah *feature scaling* untuk mengubah rentang atau skala nilai dari fitur-fitur dalam dataset ke rentang nilai baru dengan menerapkan suatu metode atau pendekatan tertentu sehingga memiliki karakteristik yang serupa antar fitur. Metode untuk melakukan *feature scaling* pada penelitian ini adalah standarisasi. Standarisasi mengubah nilai-nilai fitur sehingga memiliki rata-rata nol dan deviasi standar satu. Selain itu tahapan pemrosesan data yang dilakukan yaitu *resampling* untuk menyeimbangkan kelas data sehingga menghasilkan dataset baru yang memiliki jumlah sampel yang seimbang antara kelas mayoritas dan minoritas. Proses *resampling* pada penelitian ini diawali dengan pengidentifikasian kelas minoritas yang umumnya memiliki jumlah sampel yang lebih sedikit. Kemudian, dengan menggunakan teknik *resampling*, sampel-sampel dari kelas minoritas dipilih secara acak. Penggunaan *resampling* dengan penggantian atau *replacement* memungkinkan

sampel yang sama untuk dipilih beberapa kali, sehingga meningkatkan jumlah sampel kelas minoritas.

### 2.3 Pembuatan Model

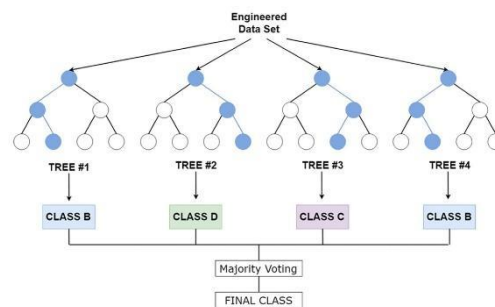
Pada penelitian ini model dibangun dengan dua kondisi. Model pertama yaitu *Random Forest* sebelum *hyperparameter tuning* atau menggunakan nilai default. Model kedua yaitu *Random Forest* dengan menggunakan *GridSearchCV*. Model kedua yakni *Random Forest* dengan optimal parameter yang ditemukan melalui *hyperparameter tuning* menggunakan *GridSearchCV*. Pada model kedua, terlebih dahulu dilakukan *hyperparameter tuning* menggunakan *GridSearchCV* terlebih dahulu, selanjutnya *Random Forest* dibangun dengan hyperparameter optimal yang sudah ditemukan pada proses *hyperparameter tuning*



**Gambar 2.** Tahapan pembuatan model

Pada Gambar 2 menunjukkan tahapan pembuatan dua model yaitu model pertama yang menggunakan parameter default dan model kedua yang menggunakan parameter optimal setelah proses *hyperparameter tuning*. Tahapan ini dilakukan untuk membandingkan performa model sebelum dan sesudah hyperparameter tuning, sehingga dapat melihat peningkatan yang dihasilkan. Berikut penjelasan untuk setiap tahap pembuatan model yang ditunjukkan pada gambar 2

- a. Split data (Train test split)  
 Dalam tahap pembagian data, dataset dibagi menjadi data pelatihan dan data pengujian. Pembagian ini dilakukan dengan rasio 80% untuk data pelatihan dan 20% untuk data pengujian (80:20).
- b. Pemodelan *Random Forest* default  
*Random Forest* (RF) adalah algoritma klasifikasi *ensemble* yang menggunakan banyak pohon keputusan (*Decision Trees*). Setiap pohon keputusan memberikan prediksi, dan prediksi akhir untuk suatu contoh pengujian dihasilkan dari gabungan prediksi semua pohon individu [15]. Proses klasifikasi dalam *random forest* dimulai dengan membagi data, lalu secara acak mengambil sampel untuk dianalisis dalam setiap pohon keputusan. Setelah pohon-pohon terbentuk, setiap kelas dari sampel data dipertimbangkan, kemudian dilakukan pemungutan suara untuk menentukan kelas yang paling sering muncul sebagai prediksi akhir [16].



**Gambar 3.** Diagram alur *Random Forest* [16]

Pada Gambar 3 menunjukkan diagram alur *Random Forest* yang terdiri dari beberapa *decision tree* yang dibuat dari subset data yang berbeda. Prediksi akhir ditentukan melalui *majority voting*. *Majority voting* adalah metode dalam *Random Forest* di mana keputusan akhir untuk klasifikasi didasarkan pada suara terbanyak dari semua *decision*

*tree* [15]. Jika mayoritas *decision tree* memprediksi kelas tertentu, maka kelas tersebut akan menjadi hasil prediksi akhir

Berdasarkan *scikit-learn* nilai parameter default *random forest* yang dimaksud adalah jumlah *n\_estimators* 100, *max\_features* menggunakan *sqrt*, *max\_depth* menggunakan *None* dan jumlah *min\_samples\_split* 2. *n\_estimators* yaitu jumlah pohon keputusan. *max\_feature* yaitu jumlah fitur acak yang diambil ketika mencari simpul pada *tree*. *min\_samples\_split* yaitu parameter yang menentukan jumlah minimum sampel yang diperlukan di sebuah node agar node tersebut dapat dibagi selama proses pembangunan pohon.

- c. Pemodelan *Random Forest* menggunakan *Optimal Hyperparameter*  
*Optimal hyperparameter* ditentukan melalui proses *hyperparameter tuning*. *Hyperparameter tuning* merupakan pendekatan yang dapat digunakan dengan mencoba berbagai kombinasi parameter untuk mengevaluasi kinerja masing-masing modelnya [14]. *Grid search* adalah sebuah proses memilih kombinasi nilai dari *hyperparameter* yang digunakan dengan mencoba berbagai kombinasi *hyperparameter* dan mengevaluasi masing-masing kombinasi. *Grid search* digunakan untuk menemukan kombinasi terbaik dari *hyperparameter* untuk suatu model[17].

*Grid search cross-validation* adalah proses pengujian beberapa kombinasi nilai *hyperparameter* secara simultan dengan menggunakan proses validasi silang [18]. *K-fold cross-validation* adalah salah satu metode *cross-validation* yang memungkinkan data pelatihan dan pengujian diulang hingga k iterasi menggunakan 1/k bagian dari kumpulan data sebagai data pengujian [19]. Algoritma *Random Forest* memiliki banyak *hyperparameter* yang dapat dioptimasi. Penelitian ini berfokus pada beberapa *hyperparameter* dari *random forest* seperti *n\_estimators*, *max\_features*, *max\_depth*, dan *min\_samples\_split*. Tabel 2 menampilkan daftar parameter yang diuji menggunakan *GridSearchCV* [20] [21] [22]

**Tabel 2.** Daftar Parameter Model

Parameter	Nilai
<i>n_estimators</i>	50, 100, 300
<i>max_features</i>	2, 3, 4
<i>max_depth</i>	3, 5, 7, 9
<i>min_samples_split</i>	2, 5, 8

Pada Tabel 2 menunjukkan rentang parameter yang diuji. Parameter-parameter ini akan digunakan dalam proses pengujian untuk menentukan konfigurasi terbaik pada model. *Hyperparameter tuning* untuk model *Random Forest* dilakukan dengan menggunakan *Grid Search Cross-Validation* yang menggunakan k-fold sebanyak 5. Setelah menemukan *n\_estimators*, *max\_features*, *max\_depth*, dan *min\_samples\_split* yang optimal untuk model *Random Forest*, langkah berikutnya adalah menggunakan *hyperparameter* tersebut untuk membangun model klasifikasi *Random Forest*.

## 2.4 Evaluasi Model

*Confusion matrix* digunakan sebagai metode evaluasi untuk mengukur kinerja proses klasifikasi dari model dengan membandingkan prediksi yang dihasilkan oleh model dengan data yang sebenarnya dari data yang diuji.

**Tabel 3. Confusion Matrix**

	Nilai Prediksi	
Nilai Aktual	Positif(1)	Negatif(0)
Positif (1)	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Negatif(0)	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

Pada tabel 3 menampilkan *confusion matrix* yang menunjukkan bagaimana prediksi model sesuai dengan nilai aktual. *True Positive* (TP) yaitu banyak data yang secara benar terklasifikasi sebagai positif. *False Positif* (FP) yaitu banyak data yang sebenarnya negatif tetapi terklasifikasi sebagai positif. *False Negative* (FN) yaitu banyak data yang sebenarnya positif tetapi terklasifikasi sebagai negatif. *True Negative* (TN) yaitu banyak data yang sebenarnya negatif yang secara benar terklasifikasi sebagai negatif. Dalam mengevaluasi kinerja model, performa yang digunakan diantaranya *Accuracy*, *Precision*, *Recal* dan *F1-Score*.

a. *Accuracy*

Akurasi adalah persentase data yang diklasifikasikan dengan benar setelah diuji. Nilai *Accuracy* dapat diperoleh dengan menggunakan persamaan

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (1)$$

b. *Precision*

*Precision* adalah persentase dari data positif yang diprediksi dengan benar dari total data yang diprediksi sebagai positif. Nilai *Precision* dapat diperoleh dengan menggunakan persamaan

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

c. *Recall*

*Recall* merupakan persentase dari data positif yang diprediksi dengan benar dari total data positif yang sebenarnya. Nilai *Recall* dapat diperoleh dengan menggunakan persamaan

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

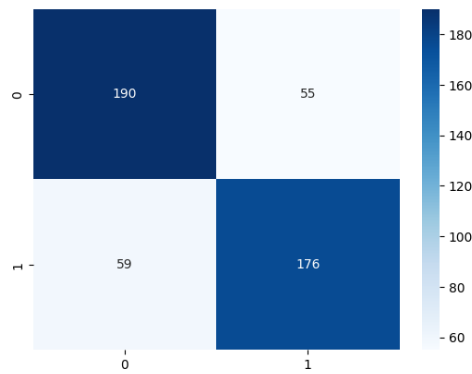
d. *F1-Score*

*F1-score* adalah metrik yang mengevaluasi hasil dari perhitungan *precision* dan *recall* dengan menghitung rata-rata harmonik dari keduanya. Nilai *F1-Score* dapat diperoleh dengan menggunakan persamaan

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision+ Recall} \times 100\% \quad (4)$$

### 3 Hasil dan Diskusi

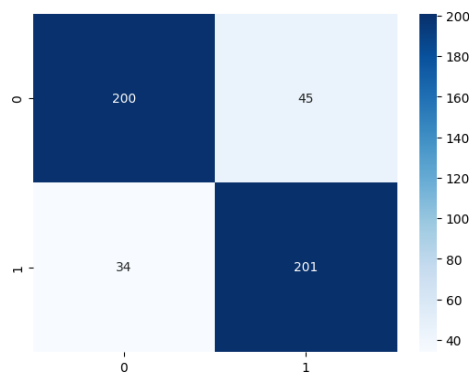
Pada model pertama, *random forest* digunakan dengan konfigurasi default sehingga tidak memerlukan *hyperparameter tuning*. Gambar 4 menggambarkan *confusion matrix* dari proses klasifikasi menggunakan *Random Forest* tanpa melalui proses *hyperparameter tuning*.



**Gambar 4.** *Confusion Matrix* tanpa *Hyperparameter Tuning*

Pada gambar 4 menunjukkan hasil *confusion matrix* dari model tanpa *hyperparameter tuning*. Data yang benar terklasifikasi sebagai positif berjumlah 190, sedangkan data negatif yang salah terklasifikasi sebagai positif berjumlah 59. Sebaliknya, data positif yang salah terklasifikasi sebagai negatif berjumlah 55, dan data yang benar terklasifikasi sebagai negative berjumlah 176.

Pada model kedua, *random forest* mengalami *hyperparameter tuning* dengan menggunakan *GridSearchCV*. *Hyperparameter tuning* menggunakan *GridSearch Cross-Validation* dengan jumlah *k-fold* bernilai 5. Dalam model *Random Forest* yang telah melalui proses *hyperparameter tuning* ditemukan bahwa model yang paling optimal menggunakan 100 pada *n\_estimators*, 9 pada *max\_depth*, 4 pada *max\_features*, dan 2 pada *min\_samples\_split*. Gambar 5 merupakan *confusion matrix* dari model klasifikasi *Random Forest* yang telah mengalami proses *hyperparameter tuning* menggunakan *GridsearchCV*.



**Gambar 5.** *Confusion Matrix* dengan *Hyperparameter Tunning GridSearchCV*

Pada gambar 5 menunjukkan hasil *confusion matrix* dari model tanpa *hyperparameter tuning*. Data yang benar terklasifikasi sebagai positif berjumlah 200, sedangkan data negatif yang salah terklasifikasi sebagai positif berjumlah 34. Sebaliknya, data positif yang salah terklasifikasi sebagai negatif berjumlah 45, dan data yang benar terklasifikasi sebagai negatif berjumlah 201.

Evaluasi model meliputi akurasi, presisi, *recall*, dan *f1-score* dihitung berdasarkan data yang terdapat dalam *confusion matrix*. Perbandingan antara *random forest* tanpa *hyperparameter tuning* atau dengan *hyperparameter default* dan *random forest* dengan menggunakan *GridSearchCV* untuk *hyperparameter tuning* ditunjukkan pada Tabel 4

**Tabel 4.** Hasil Perbandingan Performa *Random Forest*

Metric	Default	Hyperparameter tuning
Accuracy	76,00%	84,00%
Precision	76,19%	81,70%
Recall	74,89%	85,53%
F1-Score	75,53%	83,57%

Pada Tabel 4 menunjukkan hasil perbandingan performa *Random Forest* sebelum dan sesudah *hyperparameter tuning*. Hasilnya menunjukkan bahwa performa model meningkat setelah *hyperparameter tuning*. Akurasi yang sebelumnya 76% meningkat menjadi 84%, presisi yang sebelumnya 76,19% meningkat menjadi 81,70%, *recall* yang sebelumnya 74,89% meningkat menjadi 85,53%, dan nilai *f1-score* yang sebelumnya 75,53% meningkat menjadi 83,57%.

#### 4 Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa melalui proses *hyperparameter tuning* menggunakan *GridsearchCV*, kinerja model meningkat dibandingkan dengan model yang tidak melalui proses *hyperparameter tuning*. Sebelum dilakukan *hyperparameter tuning*, performa algoritma *Random Forest* telah menghasilkan *accuracy* sebesar 76%, *precision* sebesar 76,19%, *recall* sebesar 74,89%, dan *f1-score* sebesar 75,53%. Namun, setelah dilakukan *hyperparameter tuning* dengan parameter *n\_estimators* yaitu 100, *max\_depth* yaitu 9, *max\_features* yaitu 4, dan *min\_samples\_split* yaitu 2 ditemukan model yang optimal. Hasilnya menunjukkan peningkatan dalam performa model, di mana akurasi meningkat menjadi 84,00%, *precision* meningkat menjadi 81,70%, *recall* meningkat menjadi 85,53%, dan *f1-score* meningkat menjadi 83,57%. Hal ini menunjukkan bahwa pengaturan parameter yang tepat dapat secara substansial meningkatkan kemampuan prediktif algoritma *Random Forest* dalam mengklasifikasikan kualitas air layak minum.

#### Referensi

- [1] Y. B. Riyanto, E. Setiawan, M. Hannats, and H. Ichsan, "Sistem Klasifikasi Kelayakan Air Minum pada Depot Air Minum Isi Ulang (DAMIU) berdasarkan pH, Total Dissolved Solids, dan Kekeuhan menggunakan Metode Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 7, pp. 3525-3536, 2023.
- [2] G. L. Pritalia, "Analisis Komparatif Algoritma Machine Learning pada Klasifikasi Kualitas Air Layak Minum," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 2, no. 1, pp. 43-55, 2022.
- [3] A. Kustanto, "Water quality in Indonesia: The role of socioeconomic indicators," *Jurnal Ekonomi Pembangunan*, vol. 18, no. 1, pp.47-62, 2020, doi: 10.29259/jep.v18i1.11509.
- [4] P. A. Riyantoko, T. M. Fahrudin, K. M. Hindrayani, S. Data, and J. Timur, "Analisis Sederhana Pada Kualitas Air Minum Berdasarkan Akurasi Model Klasifikasi Dengan Menggunakan Lucifer Machine Learning," *Seminar Nasional Sains Data*, vol. 1, no. 01, pp. 12–18, 2021.
- [5] A. Sudin, M. Salmin, M. Fhadli, and AM Mamonto, "Klasifikasi Kelayakan Air Minum Bagi Tubuh Manusia Menggunakan Metode Support Vektor Machine Dengan Backward Elimination," *Jurnal Jaringan dan Teknologi Informasi*, vol. 3, no. 01, pp. 87–95, 2023
- [6] A. M. Graboski, J. Martinazzo, S. C. Ballen, J. Steffens, and C. Steffens, "Nanosensors for water quality control," in *Nanotechnology in the Beverage Industry: Fundamentals and Applications*, 2020. doi: 10.1016/B978-0-12-819941-1.00004-3.
- [7] S. S. Maulina Putri, M. Arhami, and H. Hendrawaty, "Penerapan Metode SVM pada Klasifikasi Kualitas Air," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 3, no. 2, pp. 94-101, 2023, doi: 10.30811/jaise.v3i2.4630.



- [8] A. S. Handayani, S. Soim, T. E. Agusdi, Rumiasih, and A. Nurdin, "Klasifikasi Kualitas Udara Dengan Metode Support Vector Machine," *JIRE (Jurnal Informatika & Rekayasa Elektronika)*, vol. 3, no. 2, pp. 187-199, 2020.
- [9] R. J. Alfirdausy and S. Bahri, "Implementasi Algoritma K-Nearest Neighbor untuk Klasifikasi Diagnosis Penyakit Alzheimer," *Techno.Com*, vol. 22, no. 3, pp. 635-642, 2023, doi: 10.33633/tc.v22i3.8393.
- [10] I. Muhamad Malik Matin, "Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware," *MULTINETICS*, vol. 9, no. 1, pp. 43-50, 2023, doi: 10.32722/multinetics.v9i1.5578.
- [11] C. G. Siji George and B. Sumathi, "Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, 2020, doi: 10.14569/IJACSA.2020.0110920.
- [12] I.P.G.H. Suputra and C.R.A. Pramatha, "Rekomendasi Rute Perjalanan Wisata Berbasis Web Menggunakan Algoritma Genetika," *Jurnal Ilmu Komputer*, vol. 13, no. 1, 2020, doi: 10.24843/jik.2020.v13.i01.p03.
- [13] A. Toha, P. Purwono, and W. Gata, "Model Prediksi Kualitas Udara dengan Support Vector Machines dengan Optimasi Hyperparameter GridSearch CV," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 4, no. 1, pp. 12-21, 2022, doi: 10.12928/biste.v4i1.6079.
- [14] M. A. A. Jihad, A. Adiwijaya, and W. Astuti, "Analisis Sentimen Terhadap Ulasan Film Menggunakan Algoritma Random Forest," *e-Proceeding of Engineering*, vol. 8, no. 5, pp. 1-13, 2021.
- [15] M. Azhar and H. F. Pardede, "Klasifikasi Dialek Pengujar Bahasa Inggris Menggunakan Random Forest," *Jurnal Media Informatika Budidarma*, vol. 5, no. 2, pp.439-446, 2021, doi: 10.30865/mib.v5i2.2754.
- [16] J. D. Muthohhar and A. Prihanto, "Analisis Perbandingan Algoritma Klasifikasi untuk Penyakit Jantung," *Journal of Informatics and Computer Science (JINACS)*, vol.4, no.3, pp. 298-304.2023, doi: 10.26740/jinacs.v4n03.
- [17] D. N. Triwibowo, P. I. A. Ashari, A. S. Sandi, and Y. F. Rahman, "Enkripsi Pesan Menggunakan Algoritma Linear Congruential Generator (LCG) dan Konversi Kode Morse," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 3, no. 3, pp. 194-201, doi: 10.12928/biste.v3i3.5546.
- [18] Fatmawati and N. A. K. Rifai, "Klasifikasi Penyakit Diabetes Retinopati Menggunakan Support Vector Machine dengan Algoritma Grid Search Cross-validation," *Jurnal Riset Statistika*, vol. 3, no. 1, pp.79-86, 2023, doi: 10.29313/jrs.v3i1.1945.
- [19] K. R. Singh, K. P. Neethu, K. Madhurekaa, A. Harita, and P. Mohan, "Parallel SVM model for forest fire prediction," *Soft Computing Letters*, vol. 3, 2021, doi: 10.1016/j.socl.2021.100014.
- [20] G. S. K. Ranjan, A. Kumar Verma, and S. Radhika, "K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries," in *2019 IEEE 5th International Conference for Convergence in Technology, I2CT 2019*, 2019. doi: 10.1109/I2CT45611.2019.9033691.
- [21] S. P. Tamba and Edric, "Prediksi Penyakit Gagal Jantung dengan Menggunakan Random Forest," *Jurnal Sistem Informasi dan Ilmu Komputer Prima(JUSIKOM PRIMA)*, vol. 5, no. 2, pp. 176-181, 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v5i2.2445.
- [22] Pirjatullah, D. Kartini, D. T. Nugrahadi, Muliadi, and A. Farmadi, "Hyperparameter Tuning using GridsearchCV on the Comparison of the Activation Function of the ELM Method to the Classification of Pneumonia in Toddlers," in *Proceedings - 2021 4th International Conference on Computer and Informatics Engineering: IT-Based Digital Industrial Innovation for the Welfare of Society, IC2IE 2021*, 2021. doi: 10.1109/IC2IE53219.2021.9649207.

*This page is intentionally left blank.*