Klasifikasi Kelayakan Air Minum berdasarkan kinerja K-Nearest Neighbor, Decision Tree, Naïve Bayes, dan Support Vector Machine

Imamul Khair[1)*], Agung Fatwanto[2)]

[1)] Prodi Teknik Industri, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga Yogyakarta

[2)] Prodi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga Yogyakarta

Received: 26 Juni 2026 Accepted: 12 November 2026 Published: 1 December 2026

Email penulis: agung.fatwanto@uin-suka.ac.id

Kata Kunci: Klasifikasi kelayakan air minum, K-Nearest Neighbor, Support Vector Machine, Perbandingan algoritma, Data leakage

DSI: Jurnal Data Science Indonesia is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Abstrak: Penelitian ini membandingkan kinerja K-Nearest Neighbor (KNN), Decision Tree, Naïve Bayes, dan Support Vector Machine (SVM) dalam klasifikasi kelayakan air minum serta mengevaluasi prapemrosesan RapidMiner. Dataset Drinking Water Potability terdiri atas 3.276 observasi, yang berkurang menjadi 2.011 observasi lengkap setelah penghapusan baris dengan nilai kosong. Audit proses awal menunjukkan normalisasi Z-transformation dilakukan sebelum pembagian data sehingga berpotensi menyebabkan data leakage. Pada pengujian ulang, data dibagi menjadi 80% data latih dan 20% data uji, dengan parameter normalisasi dihitung hanya dari data latih. Hasil pengujian terhadap 402 observasi menunjukkan akurasi KNN, Decision Tree, Naïve Bayes, dan SVM masing-masing sebesar 64,18%, 59,95%, 59,45%, dan 59,70%, dengan recall kelas layak sebesar 21,60%, 0,62%, 19,14%, dan 0%. KNN menghasilkan akurasi tertinggi secara deskriptif, tetapi seluruh model masih memiliki keterbatasan dalam mengidentifikasi kelas layak. Reanalisis Python independen menggunakan normalisasi berbasis data latih dan validasi silang menghasilkan akurasi SVM kernel RBF sebesar 69,40% dengan recall 33,33%. Hasil tersebut bukan replikasi identik eksperimen RapidMiner. Dengan demikian, model belum memadai untuk menentukan keamanan air minum tanpa pengujian laboratorium dan validasi eksternal.

PENDAHULUAN

Kelayakan air untuk dikonsumsi berkaitan erat dengan perlindungan kesehatan masyarakat. Parameter fisik dan kimia, antara lain pH, kandungan padatan terlarut, sulfat, serta kekeruhan, dapat digunakan sebagai atribut untuk mempelajari pola kualitas air. Pemanfaatan machine learning berpotensi membantu proses penyaringan awal data pengujian, tetapi hasil klasifikasinya tidak menggantikan pemeriksaan laboratorium dan pemenuhan standar kualitas air yang berlaku [1], [2]. Penelitian klasifikasi kualitas air menunjukkan hasil yang bervariasi meskipun menggunakan algoritma yang lazim, seperti K-Nearest Neighbor (KNN), Decision Tree, Naïve Bayes, dan Support Vector Machine (SVM). Perbedaan jumlah sampel, pemrosesan nilai kosong, transformasi skala, pemilihan fitur, komposisi kelas, dan protokol evaluasi dapat memengaruhi besaran metrik yang dilaporkan [3], [4], [5]–[6]. Oleh sebab itu, pengukuran akurasi perlu dilengkapi dengan evaluasi tiap kelas untuk mengetahui apakah suatu model hanya mengikuti kelas yang paling banyak muncul.

Kesenjangan penelitian ini berkaitan dengan pelaporan hasil klasifikasi yang berpusat pada akurasi, tanpa pemeriksaan kesalahan kelas minoritas maupun dokumentasi prapemrosesan yang dapat diuji ulang. Penelitian bertujuan (1) mengaudit data dan confusion matrix RapidMiner historis; (2) memeriksa empat

proyek RapidMiner yang menempatkan normalisasi setelah pembagian data dan melaporkan keluaran eksekusi ulangnya; (3) membandingkan accuracy, precision, recall, F1-score, dan balanced accuracy; (4) melakukan reanalisis independen dengan protokol yang terdokumentasi; dan (5) membahas risiko salah klasifikasi pada keamanan air minum. Perbedaan hasil historis, RapidMiner revisi, dan eksperimen Python dilaporkan secara eksplisit.

TINJAUAN LITERATUR

Klasifikasi potabilitas dan evaluasi berbasis kelas

Dataset Drinking Water Potability memuat sembilan atribut numerik, yakni pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, dan Turbidity. Target Potability membedakan air tidak layak (0) dan layak (1). Kelas yang tidak seimbang dapat menghasilkan akurasi yang tampak memadai ketika model hanya menebak kelas dominan. Karena itu, recall kelas layak, precision kelas layak, F1-score, serta balanced accuracy diperlukan untuk membaca performa secara lebih lengkap. Kesalahan memprediksi air yang sebenarnya tidak layak sebagai layak (false positive terhadap kelas layak) juga perlu dinilai secara khusus.

Prinsip dan konteks empat algoritma

KNN menetapkan kelas berdasarkan label tetangga terdekat di ruang fitur. Kinerjanya dipengaruhi nilai K dan skala numerik karena keduanya menentukan struktur kedekatan; studi Jansen dkk. menerapkan normalisasi dan mengeksplorasi beberapa nilai K pada klasifikasi kualitas air [3]. Decision Tree melakukan pemisahan atribut secara bertahap menjadi aturan keputusan yang mudah diinterpretasikan, tetapi pilihan kriteria splitting, batas kedalaman, dan ukuran daun dapat memengaruhi kemampuan menangkap kelas minoritas [7].

Naïve Bayes mengestimasi probabilitas kelas berdasarkan atribut dengan asumsi independensi bersyarat, sehingga kecocokan asumsi distribusi menentukan performanya [8], [9]. SVM menyusun batas pemisah antarkelas; jenis kernel, regularisasi, dan penetapan bobot kelas dapat mengubah hasil klasifikasi. Dalam studi potabilitas air, SVM dengan kernel RBF menunjukkan performa tinggi pada konfigurasi tertentu, tetapi hasil tersebut tidak dapat digeneralisasi begitu saja ke protokol pengujian lain [4], [10].

Penelitian terdahulu dan kesenjangan penelitian

Jansen dkk. (2025) melaporkan akurasi KNN 78% pada 100.000 sampel dan recall kelas layak 91% setelah normalisasi dan pengaturan K [3]. Ulum dkk. (2023) membandingkan KNN dan SVM untuk kelayakan air minum dan melaporkan akurasi berturut-turut 65,341% dan 69,764% [4]. Sementara itu, Zaky dkk. (2023) memperoleh rerata akurasi Decision Tree sekitar 54,33% menggunakan validasi silang lima lipatan pada dataset water quality and potability [7]. Kajian Tangkelayuk dan Mailoa juga membandingkan KNN, Naïve Bayes, dan Decision Tree pada domain kualitas air [11]. Variasi temuan menguatkan pentingnya mencatat protokol pengolahan dan evaluasi.

Pada metode probabilistik, Sari (2021) melaporkan akurasi Naïve Bayes 50,6% dalam klasifikasi kualitas air Jakarta [8], sedangkan Salam dkk. (2023) memperoleh 62,8% melalui pendekatan CRISP-DM yang mengimplementasikan Naïve Bayes dengan RapidMiner dan Google Colab [9]. Nuraeni (2025) melaporkan akurasi SVM hingga 95% pada skenario pengujian kualitas air minum menggunakan kernel RBF [10]. Perbedaan data, proses pembersihan, jumlah fitur, skenario split, dan optimasi menghalangi pembacaan angka antarpublikasi sebagai perbandingan langsung yang setara.

Penelitian potabilitas mutakhir memperlihatkan perbedaan hasil antarmodel. Sofiah dkk. [12] melaporkan akurasi neural network 85%, KNN 81%, SVM 73%, dan Naïve Bayes 63%; Koh dkk. [13] melaporkan akurasi terbaik multilayer perceptron sebesar 66,0% pada data potabilitas Kaggle. Sinap [14] menguji 3.276 observasi dan menunjukkan pentingnya penanganan nilai kosong dan outlier, sedangkan Chatterjee dkk. [15] mengkaji reduksi dimensi dan variasi pengklasifikasi. Perbedaan pengolahan, pembagian data, dan parameter membuat skor antarpublikasi tidak dapat dibandingkan langsung. Kajian Zhu dkk. [5], Lowe dkk. [16], dan Karim dkk. [6] memperkuat kebutuhan dokumentasi kualitas data, validasi, dan interpretabilitas model.

Juna dkk. [17] membandingkan penghapusan nilai kosong dengan KNN imputation, tetapi pada dataset

Kaggle berukuran 935 observasi, bukan dataset 3.276 observasi dalam penelitian ini. Karena itu, temuan tersebut dipakai sebagai justifikasi untuk mengevaluasi pilihan prapemrosesan, bukan sebagai hasil pembanding yang identik. Kesenjangan penelitian ini terletak pada kurangnya audit leakage, pelaporan seluruh sel confusion matrix, dan pembahasan risiko salah klasifikasi pada kelas Layak.

Tabel 1. Ringkasan penelitian relevan dan implikasi

Studi	Metode / data	Temuan dilaporkan	Implikasi metodologis
Jansen dkk. [3]	KNN; 100.000 sampel	Akurasi 78%; recall layak 91%	Normalisasi, pengaturan K, dan karakteristik dataset perlu dicatat.
Ulum dkk. [4]	KNN dan SVM; potabilitas air	65,341% (KNN); 69,764% (SVM)	Urutan performa dapat berubah menurut eksperimen.
Zaky dkk. [7]	Decision Tree; 5-fold CV	Rerata akurasi ~54,33%	Validasi silang memberi estimasi selain satu split.
Sari [8]; Salam dkk. [9]	Naïve Bayes; data dan prosedur berbeda	50,6%; 62,8%	Metode dan data berbeda membatasi perbandingan akurasi.
Nuraeni [10]	SVM; kernel RBF	Akurasi terbaik 95%	Kernel dan preprocessing perlu didokumentasikan.

Kontribusi penelitian mencakup verifikasi asal 2.011 observasi, pemeriksaan seluruh sel confusion matrix historis dan revisi, perbaikan alur normalisasi pada tingkat proyek RapidMiner, perbandingan dengan baseline mayoritas, serta reanalisis independen dengan validasi terkontrol. Keluaran RapidMiner revisi dan hasil Python merupakan eksperimen yang berbeda dan tidak dianggap saling menggantikan.

METODE PENELITIAN

Desain, sumber data, dan definisi kelas

Penelitian ini adalah studi kuantitatif komparatif tiga komponen: audit hasil historis RapidMiner Studio, pemeriksaan hasil running ulang dari empat proyek RapidMiner revisi, dan reanalisis independen menggunakan Python. Dataset Water Potability yang dipublikasikan Kadiwal berisi 3.276 observasi dengan sembilan fitur numerik dan label Potability (0 = Tidak layak; 1 = Layak). Hasil audit terhadap CSV yang disertakan peneliti menghasilkan 2.011 complete cases, persis sama dengan 2.011 rekaman pada sheet pengujian Excel KNN, Decision Tree, Naïve Bayes, dan SVM. Pencocokan nilai seluruh sembilan fitur dan label antarkedua sumber menunjukkan tidak ada selisih. Satu contoh tambahan bernama "Data Baru" dalam sheet latihan KNN dikecualikan karena bukan observasi dataset. Bukti ini mendukung penggunaan complete-case analysis pada penyusunan input 2.011 observasi; dalam proyek RapidMiner tidak ditemukan operator imputasi; sumber Excel telah berisi hanya 2.011 baris lengkap. Audit ini tidak dapat membuktikan langkah pembersihan manual sebelum file Excel dibuat. Sembilan fiturnya adalah pH, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, dan Turbidity.

Audit data, missing values, dan prapemrosesan

Audit langsung terhadap berkas CSV menunjukkan nilai kosong pada pH sebanyak 491 (14,99%), Sulfate 781 (23,84%), dan Trihalomethanes 162 (4,95%) dari total 3.276 rekaman; variabel lain dan Potability lengkap. Jumlah sel kosong antarkolom tidak dapat dijumlahkan untuk menghitung baris terhapus karena beberapa rekaman mempunyai lebih dari satu nilai kosong. Sebanyak 1.265 baris (38,61%) mempunyai sedikitnya satu nilai kosong dan dikeluarkan pada pembentukan complete cases sehingga diperoleh 2.011 rekaman. Pada sumber, label Tidak layak = 1.998 dan Layak = 1.278, sedangkan setelah complete-case filtering masing-masing 1.200 dan 811. Pencocokan semua 2.011 data dengan file Excel pengujian mendukung bahwa input analisis setara dengan hasil penghapusan baris tidak lengkap. Strategi ini mengurangi ukuran sampel dan berpotensi menimbulkan bias seleksi apabila pola ketidaklengkapan data tidak acak; audit ini tidak dapat membuktikan asumsi missing completely at random.

Lima proyek RapidMiner historis (.rmp; versi 7.1.001) menggunakan Read Excel → Normalize (Z-

transformation) → Set Role → Split Data 80:20 → algoritma → Apply Model → Performance. Karena normalisasi dilakukan sebelum split, statistik data uji memengaruhi parameter standardisasi (data leakage). Pengaturan split adalah sampling_type=automatic; use_local_random_seed=false, seed proses=2001; identitas data uji tidak tersimpan dalam .rmp. Dampak kuantitatif kebocoran tidak dapat dipastikan dari berkas proses saja.

Dataset lengkap terdiri atas 1.200 kelas Tidak layak (59,67%) dan 811 kelas Layak (40,33%), menunjukkan ketidakseimbangan kelas moderat. Hasil RapidMiner historis melaporkan 402 data uji: 240 Tidak layak dan 162 Layak. Kelima proyek tidak memuat operator SMOTE, undersampling, atau pembobotan kelas. Secara khusus, pada SVM parameter balance_cost=false dan L_pos=L_neg=1. Ketiadaan penyeimbangan serta normalisasi sebelum split merupakan karakteristik eksperimen historis yang harus dibedakan dari prosedur reanalisis Python dengan scaling dan SMOTE/class weighting pada data latih saja.

Konfigurasi model dan reprodusibilitas

Tabel 2. Parameter historis yang terverifikasi melalui lima berkas proyek RapidMiner (.rmp)

Model / file sumber	Parameter terverifikasi dari berkas .rmp	Catatan reproduksibilitas
KNN (1.rmp, 4.rmp)	K=45;weighted_vote=false; mixed_measure=MixedEuclideanDistance; numerical_measure=EuclideanDistance.	1.rmp merujuk Pengujian Data KNN.xlsx; 4.rmp merujuk Pengujian Data KNN, Naive Bayes, Decision Tree.xlsx.
Decision Tree (5.rmp)	criterion=gain_ratio; maximal_depth=20; apply_pruning=true; confidence=0,25; apply_prepruning=true; minimal_gain=0,1; minimal_leaf_size=2; minimal_size_for_split=4; number_of_prepruning_alternatives=3.	Konfigurasi ini berasal dari operator parallel_decision_tree, bukan Gini tree Python.
Naïve Bayes (3.rmp)	Operator naive_bayes; laplace_correction=true.	Pengolahan atribut numerik mengikuti implementasi RapidMiner; tidak identik dengan GaussianNB Python.
SVM (2.rmp)	kernel_type=dot (linear); C=0,0; convergence_epsilon=0,001; max_iterations=100000; scale=true; calculate_weights=true; balance_cost=false; L_pos=L_neg=1.	Nilai C=0,0 dicatat sebagaimana tersimpan, tanpa menyamakannya dengan pengaturan C pada pustaka lain.
Seluruh model	Normalize: Z-transformation sebelum Split Data; 80:20; sampling_type=automatic; random_seed proses=2001; use_local_random_seed=false.	Tidak ada operator imputasi/SMOTE/class weighting; data dibaca dari Excel yang sudah berisi 2.011 baris lengkap.

Audit alur dan keluaran running ulang empat proyek RapidMiner revisi

Empat proyek revisi (knn rev.rmp, decsion tree rev.rmp, bayes rev.rmp, svm rev.rmp) diperiksa pada struktur XML. Urutannya Read Excel → Set Role → Split Data 80:20 → Normalize data latih → pelatihan model, sementara model normalisasi diterapkan pada data uji sebelum klasifikasi. Screenshot Performance setelah running ulang mendukung metrik yang dilaporkan, tetapi tidak memuat identitas baris uji atau artefak model terlatih; oleh sebab itu, kesamaan sampel antarmodel dan besarnya dampak kausal koreksi normalisasi belum dapat dibuktikan dari screenshot saja.

Parameter utama revisi mengikuti historis: KNN K=45/Euclidean; Decision Tree gain ratio, depth 20 dan pruning; Naïve Bayes Laplace correction; SVM linear (dot), C=0,0, tanpa class balancing. Split menggunakan automatic sampling tanpa local seed. Setiap screenshot menunjukkan 402 data uji (240 Tidak layak dan 162 Layak), tetapi tidak menyediakan ID observasi. Eksperimen RapidMiner utama tidak menerapkan SMOTE, undersampling, atau class weighting. Sebaliknya, reanalisis Python memeriksa SMOTE dan pembobotan kelas hanya pada data latih/fold validasi. Tabel 2 merangkum parameter proyek historis,

sementara Tabel 4 berasal dari keluaran Performance RapidMiner revisi. Karena prediksi individual per observasi tidak tersedia, uji McNemar tidak diterapkan pada hasil RapidMiner.

Protokol reanalisis independen (Python; bukan replikasi konfigurasi RapidMiner)

Reanalisis menggunakan 2.011 complete cases, target biner Potability, dan sembilan fitur asli. Seluruh model diuji dengan pembagian identik train–test berstrata (1.609:402 observasi; random_state=42). Baseline KNN dalam reanalisis Python memakai K=45, bobot uniform, dan StandardScaler; Decision Tree memakai kriteria Gini dengan random_state=42 tanpa pembatasan kedalaman; Naïve Bayes memakai GaussianNB dengan var_smoothing=10^{-9}; SVM memakai kernel RBF, C=1, gamma=scale, dan StandardScaler. StandardScaler hanya di-fit pada data latih dan diterapkan pada data uji. Nilai parameter baseline dipilih sebagai eksperimen pembanding yang eksplisit, bukan klaim terhadap parameter RapidMiner historis.

Pemilihan konfigurasi reanalisis menggunakan GridSearchCV dengan stratified 5-fold cross-validation pada 1.609 data latih, pengacakan seed 42, dan objective F1-macro. Grid KNN: K ∈ {9, 21, 45}, weights ∈ {uniform, distance}, tanpa/dengan SMOTE (k_neighbors=5); Decision Tree: max_depth ∈ {4, 8, None}, min_samples_leaf ∈ {1, 5, 15}, class_weight ∈ {None, balanced}, criterion=Gini; GaussianNB: var_smoothing ∈ {10^{-9}, 10^{-7}}, tanpa/dengan SMOTE; SVM: C ∈ {0,1; 1; 10}, gamma ∈ {scale; 0,1}, class_weight ∈ {None, balanced}, kernel=RBF. Pemilihan parameter hanya menggunakan fold training; set uji tetap tertutup hingga evaluasi akhir. Imbalanced-learn Pipeline memastikan SMOTE diterapkan pada fold training saja.

Uji exact McNemar digunakan pada prediksi berpasangan 402 sampel Python: empat perbandingan baseline-versus-tuning dan enam pasangan baseline dengan koreksi Holm (α=0,05). Karena hanya satu holdout dan tidak ada validasi eksternal, hasil bersifat eksploratif. Prediksi individual RapidMiner tidak tersedia sehingga uji ini tidak dapat diterapkan pada keluaran RapidMiner.

Metrik evaluasi dan baseline

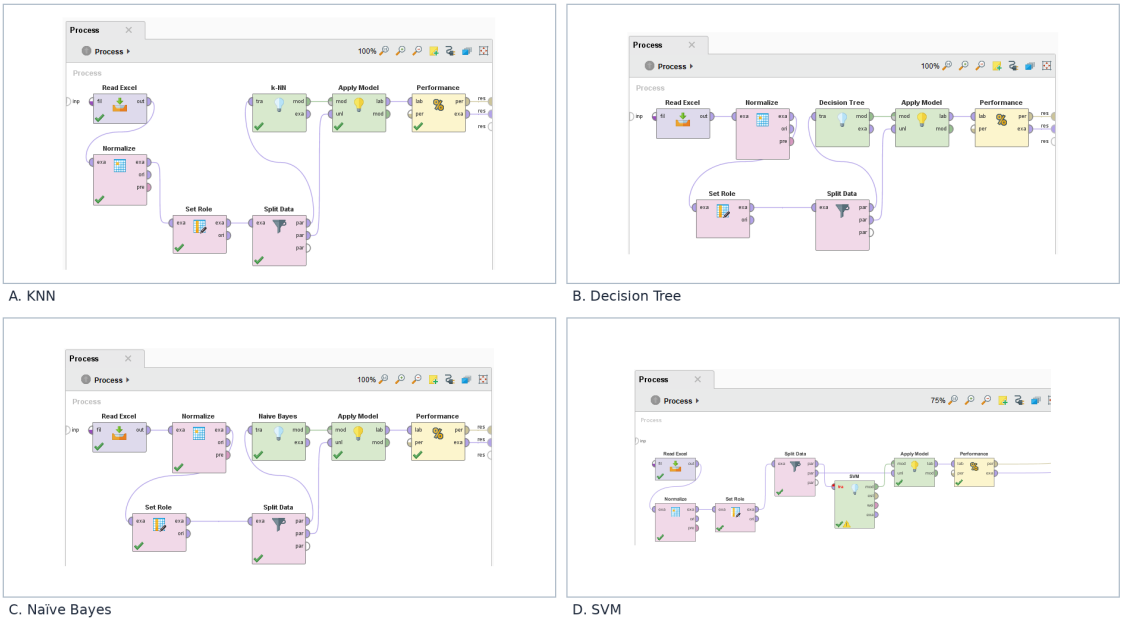
Confusion matrix disajikan dengan kelas positif = Layak (1). True positive (TP) adalah air layak yang diprediksi layak; true negative (TN) adalah air tidak layak yang diprediksi tidak layak; false positive (FP) adalah air tidak layak yang diprediksi layak; dan false negative (FN) adalah air layak yang diprediksi tidak layak. Formula evaluasi: Accuracy = (TP + TN)/(TP + TN + FP + FN); Precision (Layak) = TP/(TP + FP); Recall (Layak) = TP/(TP + FN); F1 (Layak) = 2 × Precision × Recall/(Precision + Recall); Balanced Accuracy = [Recall (Layak) + Recall (Tidak layak)]/2. Jika tidak terdapat prediksi positif, precision ditetapkan 0 untuk pelaporan, meskipun pecahan TP/(TP + FP) secara matematis tidak terdefinisi.

Baseline mayoritas (semua sampel diprediksi Tidak layak) menghasilkan akurasi 240/402 = 59,70% dan recall Layak = 0%. Hasil masing-masing model perlu dibandingkan dengan baseline ini, tidak hanya dengan model lain.

HASIL PENELITIAN

Distribusi sampel uji dan hasil empat model

Gambar 1 mendokumentasikan empat alur historis RapidMiner. Gambar 2 menyajikan ulang angka empat confusion matrix historis dengan orientasi yang eksplisit agar seluruh sel dapat diverifikasi; nilainya bersumber dari keluaran model awal, bukan pengujian ulang. Tabel 3 mencantumkan metrik per kelas.



Gambar 1. Diagram proses historis KNN, Decision Tree, Naïve Bayes, dan SVM.

A. KNN — Akurasi 63,93%

Prediksi / Aktual	Tidak layak (0)	Layak (1)
Tidak layak (0)	222	127
Layak (1)	18	35
Recall (%)	92,50%	21,60%

B. Decision Tree — Akurasi 59,95%

Prediksi / Aktual	Tidak layak (0)	Layak (1)
Tidak layak (0)	240	161
Layak (1)	0	1
Recall (%)	100,00%	0,62%

C. Naïve Bayes — Akurasi 59,45%

Prediksi / Aktual	Tidak layak (0)	Layak (1)
Tidak layak (0)	208	131
Layak (1)	32	31
Recall (%)	86,67%	19,14%

D. SVM — Akurasi 59,70%

Prediksi / Aktual	Tidak layak (0)	Layak (1)
Tidak layak (0)	240	162
Layak (1)	0	0
Recall (%)	100,00%	0,00%

Gambar 2. Rekonstruksi numerik confusion matrix historis (A) KNN, (B) Decision Tree, (C) Naïve Bayes, dan (D) SVM; baris = prediksi, kolom = aktual.

Hasil historis menunjukkan KNN mengidentifikasi 35 dari 162 sampel Layak, Decision Tree hanya 1, Naïve Bayes 31, dan SVM tidak satu pun. Rekonstruksi pada Gambar 2 mempertahankan seluruh sel TN, FP, FN, dan TP; Gambar 1 tetap memuat tangkapan layar alur proses.

Perbandingan metrik secara berdampingan

Sel confusion matrix historis (TN/FP/FN/TP): KNN 222/18/127/35; Decision Tree 240/0/161/1; Naïve Bayes 208/32/131/31; SVM 240/0/162/0.

Tabel 3. Performa empat model historis RapidMiner pada kelas Layak (1)

Algoritma	Accuracy	Precision (1)	Recall (1)	F1 (1)	Bal. Acc.
KNN	63,93%	66,04%	21,60%	32,56%	57,05%
Decision Tree	59,95%	100,00%*	0,62%	1,23%	50,31%
Naïve Bayes	59,45%	49,21%	19,14%	27,56%	52,90%
SVM	59,70%	0,00%**	0,00%	0,00%	50,00%
Baseline kelas mayoritas	59,70%	0,00%**	0,00%	0,00%	50,00%

*Precision 100% untuk Decision Tree dihitung dari hanya satu prediksi kelas layak (TP = 1, FP = 0), sehingga sangat tidak stabil. **Precision = 0 dilaporkan dengan konvensi zero-division karena tidak ada prediksi kelas layak. Semua nilai dihitung ulang dari confusion matrix yang terdapat dalam naskah awal. Pada data historis, KNN memperoleh akurasi 63,93% dan balanced accuracy 57,05%; selisihnya dari tiga algoritma lain hanya 3,98–4,48 poin persentase, sehingga tidak membuktikan superioritas algoritma secara umum.

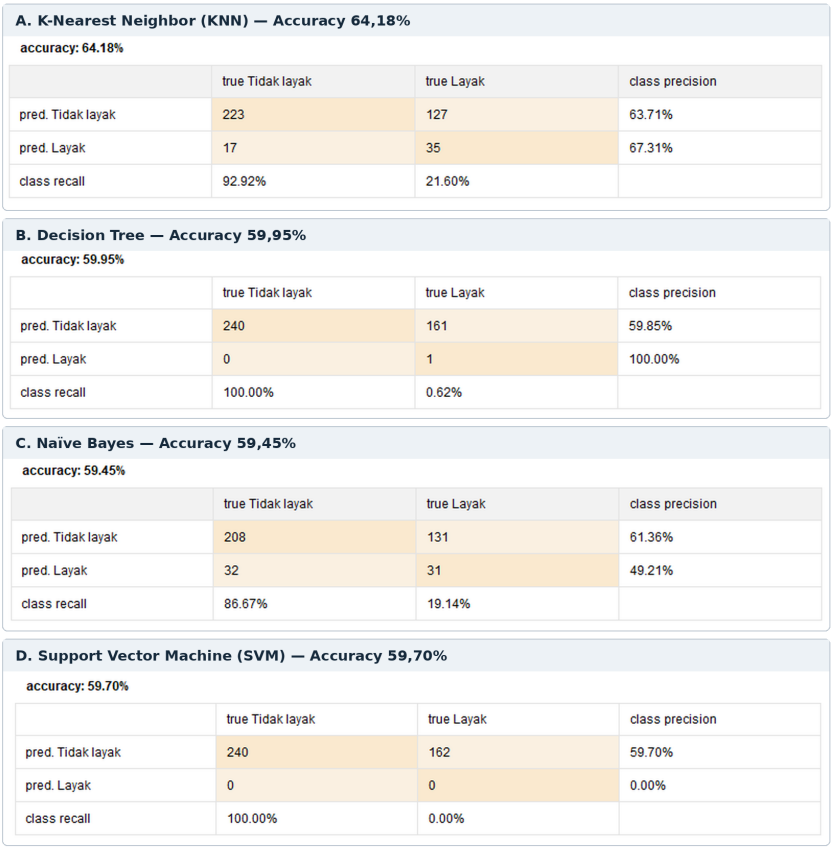
Hasil running ulang RapidMiner pada alur normalisasi terkoreksi

Berdasarkan empat screenshot Performance yang disampaikan setelah proses dijalankan ulang, setiap model memiliki 402 observasi uji dengan jumlah label aktual 240 Tidak layak dan 162 Layak. Pada tabel ini kelas positif adalah Layak (1). TN: aktual 0 diprediksi 0; FP: aktual 0 diprediksi 1; FN: aktual 1 diprediksi 0; TP: aktual 1 diprediksi 1. Seluruh metrik dihitung ulang dari empat sel confusion matrix yang terbaca pada screenshot. Kesesuaian identitas baris uji antarmodel belum dapat dibuktikan tanpa ekspor ExampleSet.

Tabel 4. Hasil running ulang empat model RapidMiner setelah koreksi alur normalisasi

Algoritma	TN / FP / FN / TP	Akurasi	Precision (1)	Recall (1)	F1 (1)	Bal. Acc.
KNN	223 / 17 / 127 / 35	64,18%	67,31%	21,60%	32,71%	57,26%
Decision Tree	240 / 0 / 161 / 1	59,95%	100,00%*	0,62%	1,23%	50,31%
Naïve Bayes	208 / 32 / 131 / 31	59,45%	49,21%	19,14%	27,56%	52,90%
SVM	240 / 0 / 162 / 0	59,70%	0,00%**	0,00%	0,00%	50,00%

*Precision 100% pada Decision Tree didasarkan pada satu prediksi Layak yang benar. **Precision 0% pada SVM menggunakan konvensi zero-division karena tidak ada prediksi Layak. Metrik dibulatkan menjadi dua desimal; sumber: empat screenshot Performance RapidMiner setelah running ulang.



A. K-Nearest Neighbor (KNN) — Accuracy 64,18%

accuracy: 64.18%

	true Tidak layak	true Layak	class precision
pred. Tidak layak	223	127	63.71%
pred. Layak	17	35	67.31%
class recall	92.92%	21.60%	

B. Decision Tree — Accuracy 59,95%

accuracy: 59.95%

	true Tidak layak	true Layak	class precision
pred. Tidak layak	240	161	59.85%
pred. Layak	0	1	100.00%
class recall	100.00%	0.62%	

C. Naïve Bayes — Accuracy 59,45%

accuracy: 59.45%

	true Tidak layak	true Layak	class precision
pred. Tidak layak	208	131	61.36%
pred. Layak	32	31	49.21%
class recall	86.67%	19.14%	

D. Support Vector Machine (SVM) — Accuracy 59,70%

accuracy: 59.70%

	true Tidak layak	true Layak	class precision
pred. Tidak layak	240	162	59.70%
pred. Layak	0	0	0.00%
class recall	100.00%	0.00%	

Gambar 3. Confusion matrix running ulang RapidMiner: KNN, Decision Tree, Naïve Bayes, dan SVM.

Dibandingkan hasil RapidMiner historis, akurasi KNN berubah dari 63,93% (TN=222, FP=18, FN=127, TP=35) menjadi 64,18% (TN=223, FP=17, FN=127, TP=35), selisih +0,25 poin persentase. Nilai Decision Tree (59,95%), Naïve Bayes (59,45%), dan SVM (59,70%) beserta seluruh sel confusion matrix tetap sama dengan

laporan historis. KNN masih hanya mengenali 35 dari 162 data Layak; Decision Tree 1 dari 162; Naïve Bayes 31 dari 162; SVM 0 dari 162. Selisih skor KNN tidak boleh diklaim disebabkan oleh perubahan urutan normalisasi tanpa bukti bahwa data uji dan kondisi pengujian lain benar-benar identik.

Reanalisis Python independen menggunakan 402 observasi uji yang identik pada empat model (240 Tidak layak; 162 Layak). Konfigurasi baseline dan hasil validasi silang lima lipatan diringkas dalam Tabel 5. Uji McNemar berpasangan mengevaluasi perbedaan akurasi pada data uji Python, bukan pada keluaran RapidMiner.

Tabel 5. Hasil reanalisis independen pada set uji yang sama

Model	Skenario	Akurasi	Recall (1)	F1-macro	FP (0→1)
KNN	Baseline	64,18%	16,05%	51,42%	8
KNN	Terpilih CV	58,71%	59,88%	58,25%	101
Decision Tree	Baseline	59,95%	53,70%	58,81%	86
Decision Tree	Terpilih CV	58,71%	45,06%	56,53%	77
Naive Bayes	Baseline	63,68%	24,07%	54,82%	23
Naive Bayes	Terpilih CV	57,96%	58,64%	57,47%	102
SVM	Baseline	69,40%	33,33%	62,64%	15
SVM	Terpilih CV	67,66%	61,73%	66,59%	68

Uji exact McNemar berpasangan pada 402 data uji Python, dengan koreksi Holm untuk enam pasangan baseline, menghasilkan p-Holm: SVM vs KNN = 0,0064; SVM vs Decision Tree = 0,0077; SVM vs Naïve Bayes = 0,0423. Hasil bersifat eksploratif dan bukan uji atas RapidMiner.

PEMBAHASAN

Kinerja relatif terhadap baseline dan ketidakseimbangan kelas

Perbandingan hasil RapidMiner revisi dengan baseline kelas mayoritas (59,70%) memperlihatkan bahwa KNN memperoleh akurasi 64,18% (+4,48 poin persentase), Decision Tree 59,95% (+0,25 poin), Naïve Bayes 59,45% (−0,25 poin), dan SVM 59,70% (setara baseline). Meskipun KNN berada pada urutan pertama secara numerik, recall kelas Layak hanya 21,60% (35 dari 162). Decision Tree mendeteksi satu dari 162 sampel Layak dan SVM tidak mendeteksi satu pun. Dengan demikian, perbandingan akurasi saja tidak memberikan dasar bagi klaim superioritas KNN maupun kesiapan model untuk penetapan keamanan air. Dalam reanalisis Python independen, SVM RBF baseline mencapai akurasi 69,40%; perbandingan ini tidak boleh disamakan dengan hasil RapidMiner karena implementasi, kernel dan prosedur pembagian data berbeda.

Mengapa SVM dan Decision Tree gagal mengenali kelas Layak?

Kegagalan SVM pada eksperimen historis RapidMiner terlihat dari 402 prediksi yang semuanya masuk kelas Tidak layak. Audit proyek 2.rmp menunjukkan kernel_type=dot (linear), C=0,0, balance_cost=false, serta penggunaan Z-transformation sebelum pemisahan train–test. Decision Tree dalam 5.rmp menggunakan gain ratio, maksimal depth 20, dan pruning/prepruning aktif, serta hanya menghasilkan satu prediksi Layak yang benar pada hasil historis. Konfigurasi tersebut memberi konteks faktual untuk kegagalan mendeteksi kelas minoritas, tetapi tidak membuktikan hubungan sebab-akibat tunggal antara suatu parameter dan recall rendah. Reanalisis Python menggunakan kernel RBF serta prosedur data latih yang berbeda; karena itu selisih hasil tidak boleh ditafsirkan sebagai peningkatan pada konfigurasi RapidMiner yang identik.

Kondisi historis ini menunjukkan mengapa precision 100% pada Decision Tree menyesatkan bila dilepaskan dari recall 0,62%: hanya satu prediksi Layak yang dibuat dan benar, sedangkan 161 kasus Layak terlewat. Pada hasil historis SVM, tidak ada satu pun prediksi Layak. Sementara itu, hasil eksperimen independen menunjukkan performa berbeda ketika prosedur dan parameter model dijelaskan secara eksplisit. Reanalisis menunjukkan bahwa tuning KNN dengan SMOTE meningkatkan recall kelas Layak dari 16,05% menjadi 59,88%, tetapi akurasi berkurang dari 64,18% menjadi 58,71% dan FP bertambah dari 8 menjadi 101. SVM dengan pembobotan kelas dan tuning meningkatkan recall kelas Layak dari 33,33% menjadi 61,73%, sedangkan akurasi turun dari 69,40% menjadi 67,66% dan FP meningkat dari 15 menjadi 68. F1-macro SVM berubah dari 62,64% menjadi 66,59%. Exact McNemar baseline versus tuning tidak menunjukkan perbedaan akurasi yang meyakinkan pada taraf 5% (KNN p=0,1007; Decision Tree p=0,7185; Naïve Bayes p=0,0598; SVM p=0,5584). Hasil ini menegaskan perlunya mempertimbangkan sekaligus recall kelas Layak dan risiko salah melabeli air Tidak layak sebagai Layak. Pada perbandingan antarmodel baseline reanalisis, SVM juga berbeda secara statistik dari Decision Tree (p-Holm=0,0077) dan Naïve Bayes (p-Holm=0,0423). Hasil tersebut hanya menguji akurasi; risiko kesalahan menerima air Tidak layak sebagai Layak tetap harus dibahas terpisah.

Prapemrosesan dan perbandingan dengan penelitian sebelumnya

Tabel 6. Perbedaan protokol penelitian terdahulu dan hasil historis

Algoritma / studi	Studi terdahulu	RapidMiner historis	Keterangan
KNN; Jansen dkk. [3]	100.000 sampel; normalisasi; variasi K; akurasi 78%.	2.011 data lengkap; Z-transform pra-split; K=45; akurasi 63,93%.	Ukuran data dan protokol berbeda; tidak sebanding secara langsung.
Decision Tree; Zaky dkk. [5]	Validasi silang 5 lipatan; rerata akurasi ±54,33%.	Split 80:20; gain ratio; depth 20; pruning; akurasi 59,95%.	Evaluasi 5-fold berbeda dengan satu holdout.
Naïve Bayes; Salam dkk. [7]	CRISP-DM; RapidMiner/Colab; akurasi 62,8%.	Split 80:20; Laplace correction aktif; akurasi 59,45%.	Fitur dan prosedur optimasi belum setara.
SVM; Nuraeni [8]	Menguji beberapa kernel; akurasi RBF tertinggi 95%.	Split 80:20; kernel linear; C=0,0; tanpa balancing; akurasi 59,70%.	Kernel dan prosedur pengujian berbeda.

Tabel 6 membandingkan protokol penelitian terdahulu dengan eksperimen RapidMiner historis secara deskriptif. Audit proyek awal menunjukkan Z-transformation sebelum Split Data yang mengakibatkan kebocoran statistik data uji. Empat berkas proses revisi memperlihatkan urutan Split Data → Normalize data latih → Apply Model normalisasi pada data uji. Keluaran eksekusi yang dilaporkan tercantum pada Tabel 4 dan Gambar 3: hanya KNN yang menunjukkan perubahan kecil pada akurasi (+0,25 poin), sedangkan tiga algoritma lainnya tidak berubah. Tanpa catatan identitas tiap baris test dan versi lengkap lingkungan eksekusi, perubahan skor tidak dapat diatribusikan secara kausal hanya kepada urutan normalisasi. Reanalisis Python menjadi pembanding independen, bukan replikasi identik RapidMiner.

Analisis sensitivitas nilai kosong dengan imputasi pada data penuh untuk menilai dampak pilihan complete-case filtering, eksperimen sensitivitas terpisah memakai seluruh 3.276 rekaman sumber dengan label 1.998 Tidak layak dan 1.278 Layak. Data dibagi 80:20 secara berstrata dengan random_state=42 (2.620 latih; 656 uji, terdiri dari 400 Tidak layak dan 256 Layak). Nilai kosong pada fitur diimputasi menggunakan median yang hanya dihitung dari data latih (SimpleImputer dalam Pipeline), dengan penskalaan train-only pada KNN dan SVM. Keempat baseline memakai konfigurasi Python yang telah didefinisikan; tidak dilakukan tuning pada set uji ini. Hasil dilaporkan pada Tabel 7 sebagai pemeriksaan sensitivitas, bukan pengulangan RapidMiner atau perbandingan langsung dengan skor complete-case pada sampel uji yang berbeda.

Tabel 7 Eksperimen sensitivitas imputasi median pada 656 data uji

Model baseline	Akurasi	Recall Layak	F1-macro	FP (0→1)	TP Layak
KNN	64,02%	15,23%	50,60%	19	39
Decision Tree	59,60%	45,31%	57,08%	125	116
Naive Bayes	61,13%	18,36%	50,23%	46	47

SVM	67,07%	26,95%	58,22%	29	69

Dari 656 data uji pada eksperimen sensitivitas, baseline SVM RBF menghasilkan akurasi 67,07% dengan recall Layak 26,95% (69 dari 256) dan 29 FP, sedangkan KNN K=45 mencapai akurasi 64,02% dengan recall Layak 15,23% (39 dari 256) dan 19 FP. Perbedaan angka dengan eksperimen complete-case tidak dapat diatribusikan hanya kepada metode imputasi karena kedua eksperimen menggunakan populasi observasi dan partisi uji yang berbeda. Eksperimen pembanding yang adil akan memerlukan evaluasi kedua strategi pada observasi uji dan prosedur validasi yang diselaraskan.

Implikasi praktis dan batasan penelitian

Pada hasil RapidMiner revisi, akurasi 59,45%–64,18% dan recall kelas Layak 0%–21,60% tidak memadai untuk keputusan keamanan air minum. KNN menghasilkan 17 false positive kelas Layak dan Naïve Bayes 32, yakni sampel Tidak Layak yang keliru dinyatakan Layak; kesalahan ini dapat memberikan rasa aman yang keliru. SVM menyamai baseline mayoritas (59,70%) karena selalu memprediksi Tidak Layak. Reanalisis Python menunjukkan trade-off: SVM berbobot kelas meningkatkan recall Layak menjadi 61,73%, tetapi menghasilkan 68 false positive. Karena belum ada validasi eksternal dan kesalahan berisiko masih tinggi, tidak ada model dalam studi ini yang layak dipakai sebagai dasar tunggal keputusan konsumsi air. Estimasi performa juga rentan terhadap pemilihan complete cases, variasi holdout, dan pergeseran kondisi pengambilan sampel.

Audit CSV dan Excel memverifikasi 2.011 complete cases; berkas .rmp historis menunjukkan normalisasi sebelum pembagian data, sedangkan rancangan .rmp revisi menerapkan train-only scaling. Walaupun confusion matrix running ulang telah tersedia, prediksi individual RapidMiner belum diekspor sehingga identitas holdout tiap model, dampak murni perubahan normalisasi, serta McNemar antarmodel RapidMiner belum dapat diverifikasi. Semua hasil Python adalah reanalisis independen, bukan replikasi identik RapidMiner. Evaluasi lanjutan memerlukan ekspor label aktual, prediksi, dan ID observasi dari set uji yang sama serta validasi eksternal.

KESIMPULAN

Dataset 3.276 observasi menghasilkan 2.011 complete cases (1.200 Tidak layak dan 811 Layak). Setelah alur RapidMiner direvisi agar normalisasi dipelajari dari data latih, keluaran running ulang pada 402 data uji melaporkan akurasi KNN 64,18%, Decision Tree 59,95%, Naïve Bayes 59,45%, dan SVM 59,70%. Recall kelas Layak berturut-turut 21,60%, 0,62%, 19,14%, dan 0%. KNN memiliki akurasi tertinggi pada rangkaian pengujian ini, tetapi seluruh model menunjukkan keterbatasan deteksi kelas Layak. Hasil historis tetap dilaporkan terpisah sebagai pembanding.

Decision Tree hanya mengenali 1 dari 162 data Layak, sedangkan SVM tidak mengenali satu pun dan setara baseline mayoritas. Dibandingkan hasil RapidMiner historis, akurasi KNN meningkat 0,25 poin, sementara tiga model lain tidak berubah. Peningkatan kecil tersebut bukan bukti superioritas KNN atau dampak kausal koreksi preprocessing karena identitas data uji tidak terdokumentasi. Reanalisis Python terpisah menghasilkan akurasi SVM RBF 69,40% (recall Layak 33,33%) dan, setelah tuning serta pembobotan kelas, 67,66% (recall Layak 61,73%); pengujian ini bukan replikasi identik RapidMiner.

Empat proyek RapidMiner revisi telah diverifikasi memiliki normalisasi train-only dalam struktur operator dan empat keluaran confusion matrix hasil running ulang telah dilaporkan. Reanalisis Python tambahan menggunakan scaling dan SMOTE pada fold training saja, validasi silang lima lipatan, serta uji McNemar berpasangan dengan koreksi Holm pada set uji reanalisis yang sama (SVM vs KNN baseline p-Holm=0,0064). Uji tersebut tidak berlaku untuk perbandingan model RapidMiner karena prediksi individual tidak tersedia. Penelitian selanjutnya sebaiknya mengekspor ID observasi uji, label aktual dan prediksi tiap model untuk pemeriksaan kesamaan holdout dan uji berpasangan, menerapkan repeated stratified validation, membandingkan imputasi secara adil pada partisi seragam, serta menggunakan data eksternal. Integrasi IoT tetap diposisikan sebagai prospek lanjutan, bukan hasil penerapan model yang sudah siap digunakan.

REFERENCE

[1] M. G. Uddin, S. Nash, M. T. Mahammad Diganta, A. Rahman, and A. I. Olbert, "Robust machine learning algorithms for predicting coastal water quality index," *J. Environ. Manage.*, vol. 321, no. August, p. 115923, 2022, doi: 10.1016/j.jenvman.2022.115923.

[2] M. H. Al-Adhaileh and F. W. Alsaade, "Modelling and prediction of water quality by using artificial intelligence," *Sustain.*, vol. 13, no. 8, pp. 1–18, 2021, doi: 10.3390/su13084259.

[3] J. Jansen, C. Tanova, D. Dariel, M. Marciano, and A. Maulana, "Penerapan Algoritma K-Nearest Neighbors untuk Klasifikasi Kualitas Air Minum," *JDMIS J. Data Min. Inf. Syst.*, vol. 3, no. 2, pp. 86–93, 2025, doi: 10.54259/jdmis.v3i2.4513.

[4] Sopiatul Ulum, R. F. Alifa, P. Rizkika, and C. Rozikin, "Perbandingan Performa Algoritma KNN dan SVM dalam Klasifikasi Kelayakan Air Minum," *Gener. J.*, vol. 7, no. 2, pp. 141–146, 2023, doi: 10.29407/gj.v7i2.20270.

[5] M. Zhu *et al.*, "Eco-Environment & Health A review of the application of machine learning in water quality evaluation," *Eco-Environment Heal.*, vol. 1, no. 2, pp. 107–116, 2022, doi: 10.1016/j.eehl.2022.06.001.

[6] M. Akmal *et al.*, "Water Quality Monitoring and Assessment Using Machine Learning : A Review of Formulation , Modeling Approaches , and Explainable Artificial Intelligence," pp. 1–20, 2026.

[7] U. Zaky, A. Naswin, and A. W. Murdiyanto, "Performance Analysis of the Decision Tree Classification Algorithm on the Water Quality and Potability Dataset," vol. 4, no. 3, pp. 145–150, 2023.

[8] Y. S. Sari, "Penerapan Metode Naïve Bayes Untuk Mengetahui Kualitas Air Di Jakarta," *J. Ilm. FIFO*, vol. 13, no. 2, p. 222, 2021, doi: 10.22441/fifo.2021.v13i2.010.

[9] I. Alfitra Salam, K. Wahyudi Putra, S. Yuliatina, and B. Nurina Sari, "Application of Naïve Bayes for Classification of Criteria for Potable Water with the CRISP-DM Method," *Paradigma*, vol. 25, no. 1, pp. 1–5, 2023, [Online]. Available: http://jurnal.bsi.ac.id/index.php/paradigma/

[10] N. Nuraeni, "Klasifikasi dengan metode Support Vector Machine (SVM) dalam menentukan kualitas air minum," vol. 3, pp. 2248–2258, 2025.

[11] A. Tangkelayuk and E. Mailoa, "Klasifikasi Kualitas Air Menggunakan Metode KNN , Naïve Bayes Dan Decision Tree," vol. 9, no. 2, pp. 1109–1119, 2022.

[12] N. A. Sofiah, F. Olivia, and M. I. Jambak, "Predictive Analytics for Water Safety: Data Mining and Supervised Learning in Potability Classification," vol. 13, no. 4, pp. 5066–5077, 2024.

[13] J. Feng, A. Mustapha, Z. Yang, R. Zhen, and H. Yap, "Water Potability : A Comparison between Machine Learning Models," vol. 15, no. 6, pp. 1993–1998, 2025.

[14] V. Sinap, "Comparative analysis of machine learning techniques for detecting potability of water," pp. 135–161, 2024.

[15] D. Chatterjee, P. G. Id, A. Banerjee, S. Shekhar, and D. Id, "Optimizing machine learning for water safety : A comparative analysis with dimensionality reduction and classifier performance in potability prediction," no. 2023, pp. 1–25, 2024, doi: 10.1371/journal.pwat.0000259.

[16] M. Lowe and R. Qin, "A Review on Machine Learning , Artificial Intelligence , and Smart Technology in Water Treatment and Monitoring," pp. 1–28, 2022.

[17] A. Juna *et al.*, "Water Quality Prediction Using KNN Imputer and Multilayer Perceptron," pp. 1–19, 2022.