

BAB II

TINJAUAN PUSTAKA

2.1. Teori Terkait

Kajian terhadap penelitian terdahulu perlu dilakukan untuk mengetahui metode, teknologi, serta algoritma *machine learning* yang telah diterapkan pada sistem monitoring dan klasifikasi kualitas air. Selain itu, kajian ini bertujuan untuk mengidentifikasi kelebihan, keterbatasan, serta celah penelitian yang masih dapat dikembangkan sebagai dasar dalam menentukan pendekatan yang digunakan pada penelitian ini. Jurnal terdahulu bisa dilihat pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

No	Judul	Metode	Hasil Evaluasi	Keterbatasan
1.	<i>IoT based real-time water quality monitoring system in water treatment plants (WTPs)</i> [11].	Sistem monitoring berbasis IoT menggunakan sensor pH, DO, TDS, dan suhu dengan kontrol PLC dan cloud monitoring.	Tingkat error sensor rendah ($\pm 0,1$ hingga $0,2$); sistem mampu monitoring real-time dan logging historis.	Tidak menerapkan <i>machine learning</i> untuk prediksi atau klasifikasi kualitas air. Fokus pada lingkungan WTP skala industri.

No	Judul	Metode	Hasil Evaluasi	Keterbatasan
2.	<i>A Study of Water Quality Monitoring System With Internet of Things and Machine learning Regression Techniques</i> [28]	Pengujian 10 model regresi <i>machine learning</i> menggunakan 4.762 data sensor kualitas air (pH, TDS, kekeruhan, suhu) secara <i>real-time</i> .	<i>Random Forest Regressor</i> (RFR) menghasilkan performa terbaik dengan nilai 0,9963 dan RMSE 0,0304.	Penelitian berfokus pada pendekatan regresi dan belum menggunakan klasifikasi berbasis DNN.
3.	<i>A Review of Machine learning and Internet-of-Things on the Water Quality Assessment</i> [24]	Kajian literatur penerapan IoT dan berbagai algoritma <i>machine learning</i> (SVM, RF, ANN, XGBoost).	Beberapa studi menunjukkan akurasi klasifikasi >85 sampai dengan 95% tergantung <i>dataset</i> dan algoritma.	Bersifat review, tidak mengimplemen tasikan sistem secara langsung pada prototipe nyata.
4.	<i>Smart Water Quality Monitoring</i>	Sistem IoT dengan algoritma	Akurasi klasifikasi tinggi (di atas	Parameter sensor terbatas dan belum

No	Judul	Metode	Hasil Evaluasi	Keterbatasan
	<i>with IoT and AI: An XGBoost Approach in Household Water Feasibility [18]</i>	XGBoost untuk klasifikasi kelayakan air berbasis data sensor.	90%), evaluasi menggunakan accuracy dan <i>Confusion Matrix</i> .	membahas pra-pemrosesan data secara rinci.
5.	<i>Real Time Water Quality Analysis Using Machine learning [29]</i>	Klasifikasi data dari sensor IoT (pH, TDS, <i>Turbidity</i>) menggunakan algoritma <i>Decision Tree</i> , SVM, dan <i>Random Forest</i> .	Akurasi klasifikasi secara keseluruhan mencapai >90%, di mana algoritma <i>Random Forest</i> memberikan hasil paling optimal.	Evaluasi belum memaparkan metrik kinerja yang spesifik (seperti <i>precision</i> , <i>recall</i> , <i>F1-score</i>) maupun penjabaran <i>Confusion Matrix</i> .

2.2. Landasan Teori

2.2.1. Kualitas Air

Kualitas air merupakan karakteristik fisik, kimia, dan biologis yang menentukan tingkat kelayakan air untuk digunakan sesuai kebutuhan tertentu. Penilaian kualitas air dilakukan dengan mengukur beberapa parameter seperti pH, *Total Dissolved Solids* (TDS), suhu, kekeruhan, dan

parameter lainnya yang dibandingkan dengan standar kualitas air yang berlaku [30].

2.2.2. pH

pH (*Potential of Hydrogen*) merupakan parameter yang digunakan untuk menunjukkan tingkat keasaman atau kebasaan suatu larutan berdasarkan konsentrasi ion hidrogen (H^+) di dalam air. Nilai pH memiliki peran penting dalam menentukan karakteristik kimia air karena perubahan nilai pH dapat memengaruhi kandungan oksigen terlarut, ketersediaan nutrisi, aktivitas mikroorganisme, tingkat kekeruhan, serta kualitas air secara keseluruhan. Oleh karena itu, pengukuran pH menjadi salah satu indikator penting dalam proses penilaian kualitas air [31].

2.2.3. Total Dissolved Solids (TDS)

Zat padat terlarut total atau *Total Dissolved Solids* (TDS) merupakan jumlah zat padat terlarut dalam air yang terdiri atas mineral, garam, logam, dan senyawa organik lainnya. Nilai TDS digunakan sebagai salah satu indikator kualitas air karena tingginya kandungan zat terlarut dapat memengaruhi rasa, kejernihan, dan tingkat kelayakan air untuk digunakan [32].

2.2.4. Turbidity

Tingkat kekeruhan atau *turbidity* merupakan parameter yang menunjukkan tingkat kejernihan air akibat adanya partikel tersuspensi

seperti lumpur, bahan organik, maupun sedimen di dalam air. Semakin tinggi nilai turbidity, maka tingkat kekeruhan air semakin meningkat sehingga dapat memengaruhi kualitas air dan proses pengolahan air [33].

2.2.5. *Preprocessing Data*

Pra-pemrosesan (*preprocessing*) data merupakan tahapan awal dalam *machine learning* yang bertujuan untuk meningkatkan kualitas data sebelum digunakan pada proses pelatihan model. Tahapan *preprocessing* dapat meliputi pembersihan data, penanganan data yang tidak seimbang, normalisasi, dan transformasi data sehingga model yang dibangun dapat menghasilkan performa yang lebih baik [34].

2.2.6. *Machine learning dan Deep Learning*

1. *Machine learning* merupakan metode analisis data yang mengotomatisasi pembangunan model analitik, dimana komputer mampu belajar dan meningkatkan kemampuannya berdasarkan pengalaman tanpa perlu diprogram secara eksplisit. Teknologi ini bekerja dengan menggunakan algoritma statistik untuk mempelajari pola dalam data dengan tujuan meningkatkan akurasi serta efisiensi dalam setiap keputusan prediksi yang dihasilkan [35].
2. *Deep Learning* merupakan salah satu cabang dari *Machine learning* yang memanfaatkan jaringan saraf tiruan (*Artificial Neural Network*) dengan banyak lapisan (*hidden layer*) untuk mempelajari pola dan

hubungan kompleks dari data. Metode ini mampu melakukan proses pembelajaran secara otomatis melalui optimasi bobot pada setiap lapisan jaringan sehingga dapat menghasilkan performa yang baik pada berbagai permasalahan klasifikasi, prediksi, maupun pengenalan pola. *Deep Learning* telah banyak diterapkan dalam berbagai bidang karena kemampuannya dalam mengekstraksi fitur dan merepresentasikan data secara lebih mendalam dibandingkan metode *Machine learning* konvensional [36].

a. *Deep Neural Network* (DNN)

Deep Neural Network (DNN) atau sering juga disebut *Deep Learning* adalah bagian spesifik dari *machine learning* yang mengeksplorasi arsitektur jaringan yang lebih dalam. DNN berbeda dari teknik *machine learning* konvensional karena algoritma ini memiliki keunggulan komputasi untuk memproses data langsung dalam bentuk mentahnya. Berbeda dengan metode konvensional yang mengharuskan perancangan ekstraktor fitur secara manual untuk mengubah data mentah menjadi vektor fitur, sistem DNN mampu belajar secara otomatis dari data tanpa mengharuskan manipulasi atau ekstraksi fitur ekstensif sebelumnya. Komposisi struktural dari DNN yang memiliki banyak lapisan tersembunyi ini memungkinkan algoritma tersebut

untuk memodelkan dan mempelajari fungsi-fungsi matematis yang sangat kompleks. [27].

2.2.7. *Library Python*

1. Pandas, merupakan *library* Python yang digunakan untuk manipulasi dan analisis data terstruktur melalui struktur data *DataFrame* dan *Series*. *Library* ini menyediakan berbagai fungsi untuk membaca, membersihkan, menggabungkan, serta mengolah *dataset* secara efisien. Dalam penelitian ini, Pandas digunakan untuk memuat *dataset* kualitas air dari berbagai sumber, melakukan proses penggabungan *dataset*, serta menyiapkan data sebelum memasuki tahap *preprocessing* dan pelatihan model. Pandas memudahkan proses manipulasi dan analisis data sehingga banyak digunakan dalam bidang data science [37].
2. NumPy (*Numerical Python*), merupakan *library* fundamental dalam komputasi numerik berbasis Python yang menyediakan objek array multidimensi serta berbagai fungsi matematika dan statistik. NumPy berperan penting dalam pemrosesan data ilmiah karena mampu melakukan operasi numerik secara efisien pada data berukuran besar. Dalam penelitian ini, NumPy digunakan untuk manipulasi array data, pengaturan random seed, serta mendukung proses komputasi selama pelatihan model *Deep Neural Network*. NumPy menjadi fondasi

utama berbagai *library* ilmiah Python seperti Pandas, Scikit-learn, dan *TensorFlow* [38].

3. Scikit-learn, merupakan *library machine learning* berbasis Python yang menyediakan berbagai fungsi untuk *preprocessing* data, pembagian *dataset*, serta evaluasi model. *Library* ini mendukung implementasi berbagai metode *machine learning* melalui antarmuka yang sederhana dan mudah digunakan. Pada penelitian ini, Scikit-learn digunakan untuk melakukan pembagian data menggunakan *Train-Test-Split*, normalisasi menggunakan *StandardScaler*, serta evaluasi model menggunakan *Confusion Matrix* dan *Classification Report*. Scikit-learn memberikan kemudahan dalam pengembangan *machine learning* baik untuk *supervised* maupun *unsupervised learning* [39].
4. Matplotlib, merupakan *library* visualisasi data pada Python yang digunakan untuk membuat berbagai jenis grafik seperti *line chart*, *bar chart*, *scatter plot*, dan histogram. Dalam penelitian ini, Matplotlib digunakan untuk memvisualisasikan proses pelatihan model melalui grafik *accuracy* dan *loss* serta menampilkan hasil evaluasi model.
5. Seaborn, merupakan *library* visualisasi statistik berbasis Python yang dibangun di atas Matplotlib dan terintegrasi dengan struktur data Pandas. *Library* ini menyediakan antarmuka tingkat tinggi untuk menghasilkan visualisasi data yang informatif dan berkualitas

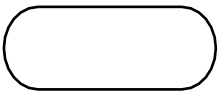
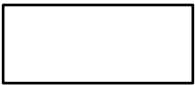
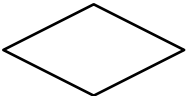
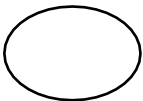
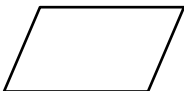
publikasi. Dalam penelitian ini, Seaborn digunakan untuk memvisualisasikan *Confusion Matrix* hasil evaluasi model DNN. Seaborn memudahkan eksplorasi data dan pembuatan visualisasi statistik secara efektif [40].

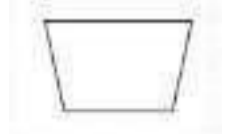
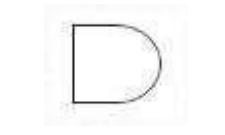
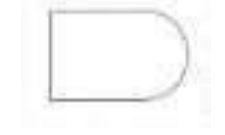
6. TensorFlow, merupakan salah satu library open source untuk pemrograman berbasis data flow (aliran data). Mempunyai banyak library untuk matematika dan sering digunakan untuk pengaplikasian machine learning seperti neural networks. Tensorflow sendiri dikembangkan oleh tim Google Brain untuk kegunaan *internal* dan pertama kali diperkenalkan pada akhir 2015. Pada tahun 2017 baru versi pertama dari Tensorflow muncul. Tensorflow bahkan telah digunakan untuk mencegah kebutaan. Caranya, dengan membantu dokter melakukan filter terhadap retinopati diabetes [41].

2.2.8. *Flowchart*

Diagram alir (*flowchart*) merupakan representasi visual terstruktur dari urutan proses, aliran data, dan logika suatu sistem. Diagram ini banyak digunakan di berbagai bidang untuk mempermudah pemahaman tahapan kerja secara sistematis. Penggunaan *flowchart* membantu menggambarkan hubungan antarproses sehingga alur kerja sistem dapat dianalisis dan dipahami dengan lebih mudah [42]. Adapun simbol-simbol *flowchart* program dapat dilihat pada Tabel 2.2 *Flowchart*.

Tabel 2. 2 *Flowchart*

No	Simbol	Nama	Fungsi
1.		Terminator	Simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan.
2.		<i>Process</i>	Simbol menunjukkan proses atau langkah kerja.
3.		<i>Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
4.		<i>Connector</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar atau halaman yang sama.
5.		<i>Flow Direction</i>	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connection line</i> .
6.		<i>Input - Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
7.		<i>Predefined Process</i>	Merupakan simbol yang digunakan untuk menunjukkan sub-proses yang belum dijelaskan secara rinci dan akan diuraikan pada bagian lain
8.		<i>Connector (offpage)</i>	Merupakan simbol yang digunakan untuk menghubungkan simbol pada halaman yang berbeda.
9.		<i>Preparation Symbol</i>	Merupakan simbol untuk menunjukkan proses penyimpanan data ke dalam media penyimpanan.

No	Simbol	Nama	Fungsi
10.		<i>Manual Input Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan proses <i>input</i> data secara manual menggunakan <i>keyboard</i> atau perangkat <i>input</i> .
11.		<i>Manual Operation Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan kegiatan atau proses yang dilakukan komputer atau sistem. Simbol ini biasanya menggambarkan aktivitas.
12.		<i>Display Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan penggunaan perangkat <i>output</i> , seperti monitor, printer, atau perangkat lainnya.
13.		<i>Delay Symbol</i>	Merupakan simbol yang digunakan untuk menunjukkan proses penundaan (<i>delay</i>) atau menunggu dalam suatu alur sistem.