JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)
Vol. 5 No. 1, Agustus 2021 E-ISSN : 2580-2879

PREDIKSI WATER QUALITY INDEX (WQI) MENGGUNAKAN ALGORITMA REGRESSI DENGAN HYPER-PARAMETER TUNING

Muhammad Radhi*[1], Amalia[2], Stiven Hamonangan Sinurat[3], Daniel Ryan Hamonangan Sitompul[4], Evta Indra[5]
[1,2,3,4,5] Sistem Informasi, Fakultas Teknologi dan Ilmu Komputer, Universitas Prima Indonesia
Jl. Sampul No.4, Medan Petisah, Kota Medan
E-mail : *muhammad_radhi05@yahoo.com

ABSTRAK- Air merupakan salah satu sumber daya alam esensial untuk kelangsungan hidup seluruh makhluk hidup di dunia ini. Kita memerlukan air untuk kebutuhan kita sehari-hari, begitu juga dengan tanaman dan hewan yang memerlukan air untuk kelangsungan hidupnya. Indeks Kualitas Air (*Water Quality Index/*WQI) merupakan satuan untuk mengetahui apakah air dapat dinyatakan layak minum (*Potable)* atau tidak. Pada penelitian ini, untuk membuat prediksi nilai WQI lebih akurat dan memiliki tingkat akurasi model yang lebih tinggi, dirancang sebuah algoritma regresi yang kemudian akan dikonfigurasi kembali dengan *tuning* algoritma. Model *machine learning* yang telah dibuat memiliki nilai yang berbeda, namun pada penelitian ini telah ditentukan bahwasanya model *Linear Regression* yang dipakai sebagai model utama karena memiliki nilai R2 lebih tinggi (0.9965 / 96,5%). Sesuai dengan hasil plotting dari Linear Regression, data diprediksi dengan baik dan persebaran data masih berdekatan dengan prediksi model (*robust).*

Kata kunci : Machine Learning, Hyper-Parameter Tuning, Water Potability, Regression Algorithm.

1. PENDAHULUAN

Air merupakan salah satu sumber daya alam esensial untuk kelangsungan hidup seluruh makhluk hidup di dunia ini. Kita memerlukan air untuk kebutuhan kita sehari-hari, begitu juga dengan tanaman dan hewan yang memerlukan air untuk kelangsungan hidupnya. Menurut data dari *World Health Organization* (WHO), diperkirakan sebanyak 1.1 Miliar orang tidak memiliki akses untuk mendapatkan air yang layak minum dan 2.6 Miliar lainnya tidak mendapatkan fasilitas sanitasi yang baik [1]. Indeks Kualitas Air (*Water Quality Index/*WQI) merupakan satuan untuk mengetahui apakah air dapat dinyatakan layak minum (*Potable)* atau tidak. WQI dihitung berdasarkan kadar kimia yang terkandung dalam air, seperti tingkat pH, oksigen terlarut dalam air (*Dissolved Oxygen/*DO), kebutuhan oksigen biologis (*Biochemical Oxygen Demand/*BDO), tingkat konduktivitas air(*Water Conductivity/WCO),* dll [2].

Dapat dilakukan banyak cara untuk melakukan prediksi WQI, salah satunya dengan metode *Machine Learning*. Dalam *Machine Learning*, terdapat dua algoritma yang umumnya digunakan, yaitu algoritma regresi dan klasifikasi. Penelitian sebelumnya yang melakukan prediksi WQI umumnya menggunakan algoritma klasifikasi [3-8]. Penelitian [3] menyajikan sebuah model klasifikasi *Machine Learning* menggunakan algoritma C5.0, Random Forest Classifier dan K-Nearest Neighbor. Hasil akurasi model dari penelitian ini mendapat nilai tertinggi 96% yaitu dengan menggunakan algoritma C5.0. Penelitian [6] menyajikan sebuah model regresi *Machine Learning* menggunakan algoritma *Linear Regression, Random Forest Regressor* dan *Gradient Boosting Regressor.* Hasil akurasi model dari penelitian ini mendapat nilai tertinggi 74% yaitu dengan menggunakan algoritma *Gradient Boosting Regressor*.

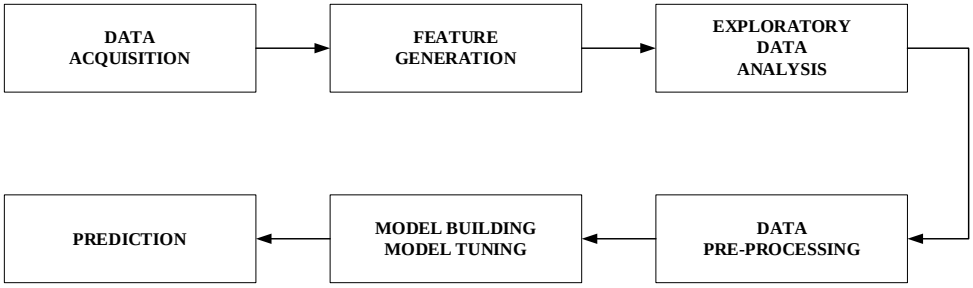
Pada penelitian ini, untuk membuat prediksi nilai WQI lebih akurat dan memiliki tingkat akurasi model yang lebih tinggi, dirancang sebuah algoritma regresi yang kemudian akan dikonfigurasi kembali dengan *tuning* algoritma. Perbedaan antara penelitian ini dan penelitian [6] adalah pada penggunaan *Supervised-Machine Learning*. Dataset yang dipakai terlebih dahulu akan dilakukan pembobotan pada tiap variabel sesuai dengan ketentuan WHO. Kemudian data yang telah diberi bobot akan dilakukan train menggunakan algoritma regresi yang dapat memberikan akurasi model diatas 90%. Algoritma regresi yang dipakai sama seperti yang terdapat pada penelitian [6], hanya saja akan dilakukan *Hyper-Parameter Tuning*.

2. METODOLOGI PENELITIAN

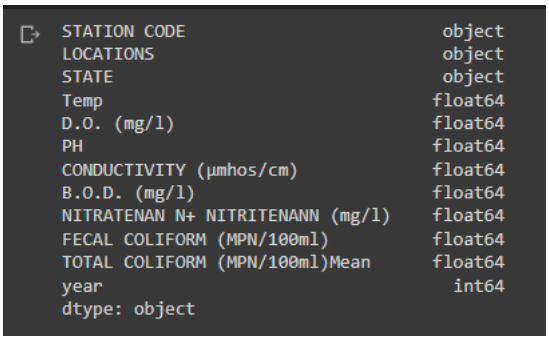
Penelitian ini dilakukan menggunakan bantuan Google Colaboratory untuk membuat model. Tahapan penelitian dapat dilihat pada Gambar 1. Penelitian ini dimulai dengan pengambilan dataset (*Data Acquisition*); pengambilan fitur lebih banyak dari satu variabel (*Feature Generation);* Melakukan eksplorasi data (*Exploratory Data Analysis),* Tahapan *preprocessing* yang didalamnya terdapat *Feature Selection, Feature Scaling* dan *Dataset Split;* Pembuatan Model dan yang terakhir visualisasi hasil.

2.1 Data Acquisition

Dataset yang dipakai pada penelitian ini berisikan 1991 baris dan 12 kolom. 12 kolom ini merupakan kandungan kimia yang terdapat pada air, dimana data diambil dan tahun pengambilan data. Detail Dataset dapat dilihat pada Gambar 2.
.



Gambar 1. Flow Kerja Project

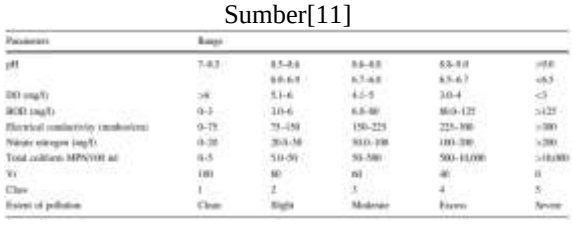


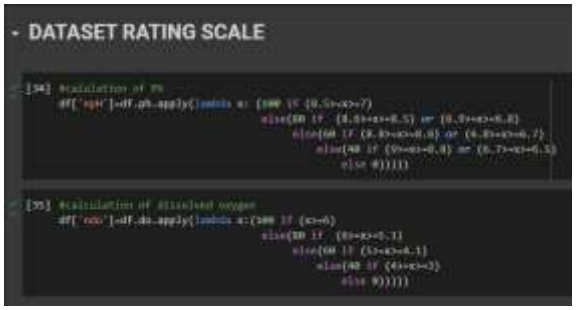
Gambar 2. Detail Dataset

2.2 Feature Generation

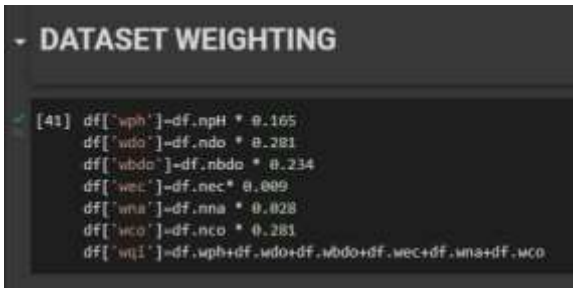
Feature Generation bertujuan untuk menghasilkan fitur lebih banyak dari suatu variabel [9,10]. Pada penelitian ini, variabel dasar sepergi *pH* sampai *Total Coliform* akan diberikan *rating scale* dan pembobotan berdasarkan perhitungan penelitian [11]. Pemberian *rating scale* dilakukan dengan fungsi *concatenate* dan pembobotan dilakukan dengan cara melakukan perkalian dari nilai *rating scale* dengan bobot. Tabel perhitungan nilai disajikan pada Tabel 1, Proses pemberian *rating scale* dan bobot dapat dilihat pada Gambar 3 dan 4. Hasil dari kedua proses dapat dilihat pada Gambar 6.

Tabel 1. Rating Scale Variabel
Sumber[11]





Gambar 3. Proses Penerapan Rating Scale

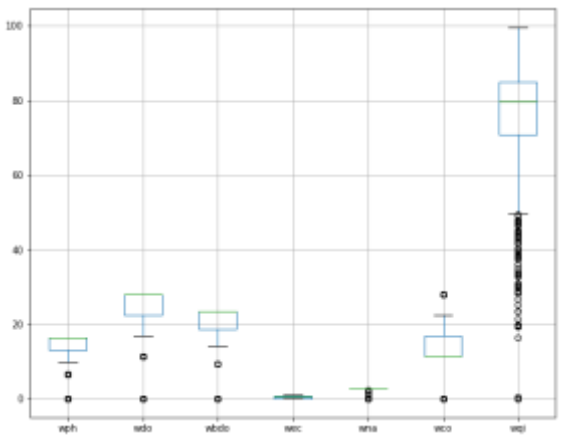


Gambar 4. Proses Pembobotan Variabel

2.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) adalah proses eksplorasi data yang bertujuan untuk memahami isi dan komponen penyusun data [12,13]. Pada penelitian ini, EDA dilakukan pada dataset untuk melihat boxplot data, korelasi data, persebaran data.

Visualisasi boxplot dipakai untuk melihat deskripsi dari dataset dalam bentuk visual, sehingga kita dapat lebih mengerti bagaiman pembagian kuartil hingga outlier pada dataset. Visualisasi boxplot pada penelitian ini dapat dilihat pada Gambar 5.



Gambar 5. Visualisasi Boxplot Dataset

Visualisasi korelasi data dipakai untuk melihat tingkat korelasi tiap variabel. Pada penelitian ini, nilai korelasi antar variabel tidak memiliki nilai yang tinggi, sehingga tidak diperlukan penghapusan variabel. Visualisasi korelasi data dapat dilihat pada Gambar 7. .

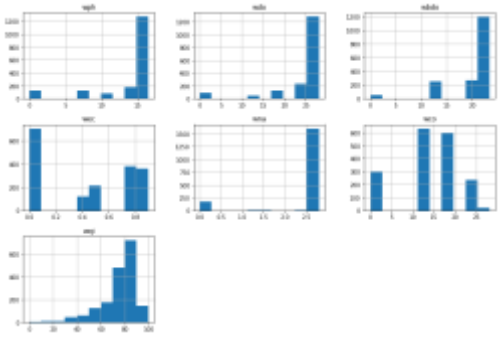
```
df.head()
```

	n	state	temp	do	ph	co	bod	na	tc	year	npH	ndo	nco	nbdo	nec	nna	wph	wdo	wbdo	wec	wna	wco	wqi
JI		GOA	29.5	6.3	6.9	179.0	1.7	0.1	5330.0	2014	80	100	40	100	60	100	13.2	28.10	23.40	0.54	2.8	11.24	79.28
E		GOA	29.7	5.8	6.9	64.0	3.8	0.5	8443.0	2014	80	80	40	80	100	100	13.2	22.48	18.72	0.90	2.8	11.24	69.34
Y		GOA	29.5	5.8	7.3	83.0	1.9	0.4	5500.0	2014	100	80	40	100	80	100	16.5	22.48	23.40	0.72	2.8	11.24	77.14
D A		GOA	30.0	5.5	7.4	81.0	1.5	0.1	4049.0	2014	100	80	40	100	80	100	16.5	22.48	23.40	0.72	2.8	11.24	77.14
A		GOA	29.2	6.1	6.7	308.0	1.4	0.3	5672.0	2014	60	100	40	100	0	100	9.9	28.10	23.40	0.00	2.8	11.24	75.44

Gambar 6. Dataset Setelah Proses Rating Scale dan Pembobotan

Gambar 7. Visualisasi Korelasi Variabel

Visualisasi persebaran data diperlukan untuk melihat apakah data tersebar secara merata atau tidak (bias). Visualisasi persebaran data dapat dilihat dengan histogram pada Gambar 8.



Gambar 8. Visualisasi Persebaran Data

2.4 Preprocessing

2.4.1 Feature Selection

Feature Selection diperlukan untuk mendapatkan tingkat pentingnya suatu variabel dalam pembuatan model kelak [14,15]. Pada penelitian ini *Feature Selection* digunakan dengan bantuan fungsi *Extra Tree Regressor* yang merupakan bagian dari algoritma *Random Forest Regressor*. Proses *Feature Selection* ditampilkan pada Gambar 9.

```
[51] from sklearn.ensemble import ExtraTreesRegressor
     selector = ExtraTreesRegressor()

[52] selector.fit(X,y)

     ExtraTreesRegressor()

[53] feature_imp = selector.feature_importances_

[54] for index, val in enumerate(feature_imp) :
       print(index, round((val * 100), 2))

     0 8.88
     1 41.15
     2 17.47
     3 0.1
     4 0.68
     5 32.52
```

Gambar 9. Proses Feature Selection

Dapat dilihat bahwasanya variabel kolom ke 3 dan 4 memiliki nilai relevansi yang kecil, maka dari itu penulis melakukan drop column terhadap kolom WEC dan WNA pada dataset dan hanya memilih kolom WPH, WDO, WBDO dan WCO dari dataset

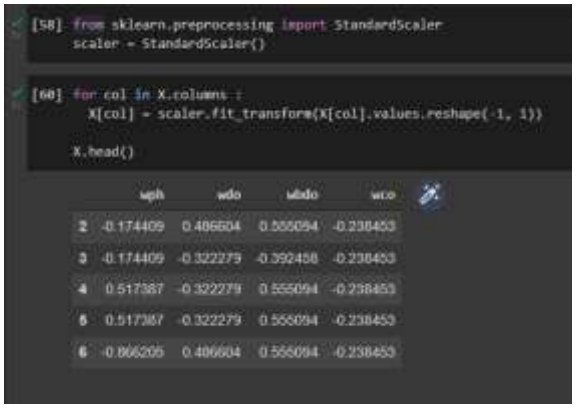
```
X.drop(['wec', 'wna'], axis=1, inplace=True)
X
```

	wph	wdo	wbdo	wco
2	13.2	28.10	23.40	11.24
3	13.2	22.48	18.72	11.24
4	16.5	22.48	23.40	11.24
5	16.5	22.48	23.40	11.24
6	9.9	28.10	23.40	11.24
...	...	...	...	...
1774	0.0	16.86	14.04	22.48
1775	16.5	28.10	23.40	11.24

Gambar 10. Dataset Setelah Proses Feature Selection

2.4.2 Feature Scaling

Feature Scaling diperlukan untuk menghindari bias, untuk mencegah adanya suatu variabel yang mendominasi dan menjaga agar nilai variabel terdapat pada rentange (*range)* yang sama [15,16]. Pada penelitian ini, *Feature Scaling* digunakan dengan fungsi *StandardScaler* dari *Sci-kit Learn*. Rentang yang ditentukan dari fungsi ini dimulai dari -1 hingga 1. Proses *Feature Scaling* dapat dilihat pada Gambar 11.



Gambar 11. Proses Feature Scaling

2.4.3 Dataset Split

Dataset split bertujuan untuk membagi dataset menjadi data training dan testing yang akan digunakan untuk pembuatan model. Umumnya dipakai konfigurasi split 80:20 atau 80:10:10 [18,19]. Pada penelitian ini, dipakai konfigurasi split 80:20, dimana 80% data dari *dataset* akan dipakai mesin untuk mengetahui pola dan 20% akan dijadikan data untuk melakukan prediksi. Proses *Dataset Split* dapat dilihat pada Gambar .

Gambar 12. Proses Dataset Split

2.5 Model Building

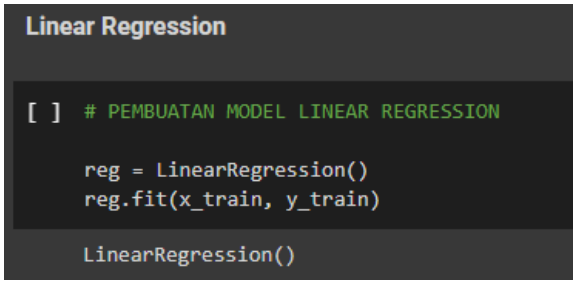
Pada penelitian ini, akan dibuat tiga model regressi untuk melakukan prediksi persentase kelayakan minum dari air. Umumnya, algoritma regresi yang dipakai adalah *Linear Regression*, *Random Forest Regression* dan *Gradient Boost Regression*.

2.5.1 Linear Regression

Algoritma *Linear Regression* ini menjelaskan relasi antara dua variabel dengan cara menyesuaikan garis regresi satu data dependent pada suatu data independent [20,21]. Pada penelitian ini, algoritma *Linear Regression* dibuat menggunakan konfigurasi *default*. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (1) dan pembuatan algoritma dapat dilihat pada Gambar 13.

$$Y = a * X + b \quad \text{.....(1)}$$

Dimana Y adalah variabel tetap, a adalah slope, X adalah varibel tidak tetap dan b adalah intercept.



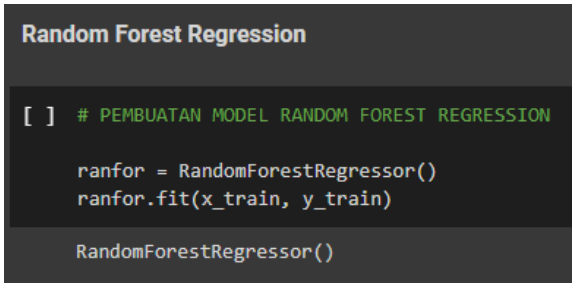
Gambar 13. Pembuatan Model Linear Regression

2.5.2 Random Forest Regression

Random Forest (RF) menghasilkan ratusan atau bahkan ribuan decision tree, yang akan menjadi fungsi regressi dan output yang dihasilkan dari RF adalah rata-rata dari tiap output dari decision tree [22,23]. Pada penelitian ini, algoritma RF dibuat menggunakan konfigurasi *default*. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (2) dan pembuatan algoritma dapat dilihat pada Gambar 14.

$$n_{ij} = w_l C_j - w_{left(j)} C_{left(j)} - w_{right(j)} - C_{right(j)} \quad \text{.....(2)}$$

n_{ij} = The importance of node j
w_l = Weighted number of samples reaching node j
C_j = The impurity value of node j
$left(j)$ = Child node from left split on node j
$right(j)$ = Child node from right split on node j [15]



Gambar 14. Pembuatan Model Random Forest

2.5.3 Gradient Boost Regressor

Gradient Boost Regression (GBR) adalah sekumpulan decision tree untuk melakukan klasifikasi dan regresi. Cara kerja umum dari gradient boost adalah membuat urutan pohon, dimana tiap pohon berfokus pada sisa prediksi pohon-pohon sebelumnya [24,25]. Pada penelitian

ini, GBR dibuat menggunakan konfigurasi *default*. Persamaan untuk pembuatan algoritma ini dapat dilihat pada (3).

$$\frac{1}{n}\sum_i(y_i' - y_i) \ldots\ldots\ldots\ldots\ldots\ldots(3)$$

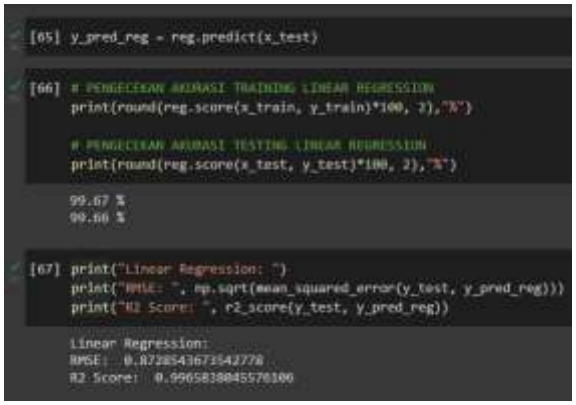
y_i' = Hasil Prediksi
y_i = Nilai Observasi
n = Banyaknya sampel pada y [25]

3. KESIMPULAN

3.1 Hasil Pembuatan Model

3.1.1 Linear Regression

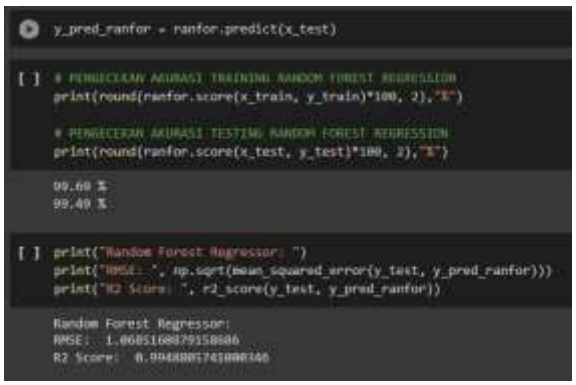
Hasil dari training model yang telah dilakukan dengan *Linear Regression* mendapatakan nilai akurasi training 99.67% dan testing 99,66%, RMSE sebesar 0.87 dan nilai R2 sebesar 0.9965. Detail Proses training model dapat dilihat pada Gambar 15.



Gambar 15. Hasil Model Linear Regression

3.1.2 Random Forest Regressor

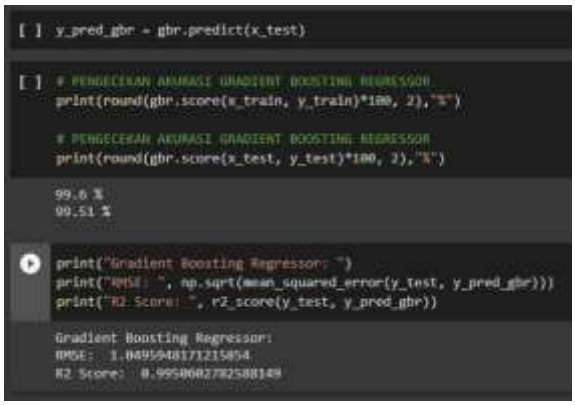
Hasil dari training model yang telah dilakukan dengan *Random Forest Regressor* mendapatkan nilai akurasi training 99,69% dan testing 99,49%, RMSE sebesar 1.065 dan R2 sebesar 0.9948. Detail Proses training model dapat dilihat pada Gambar 16.



Gambar 16. Hasil Model Random Forest

3.1.3 Gradient Boosting Regressor

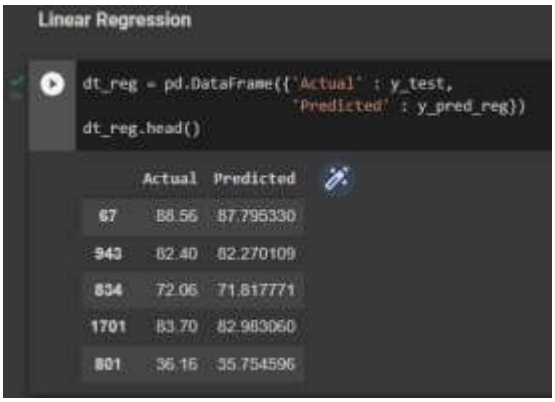
Hasil dari training model yang telah dilakukan dengan *Gradient Boosting Regressor* mendapatkan nilai akurasi training 99,6% dan testing 99,51%, RMSE sebesar 1.049 dan R2 sebesar 0.995. Detail Proses training model dapat dilihat pada Gambar .



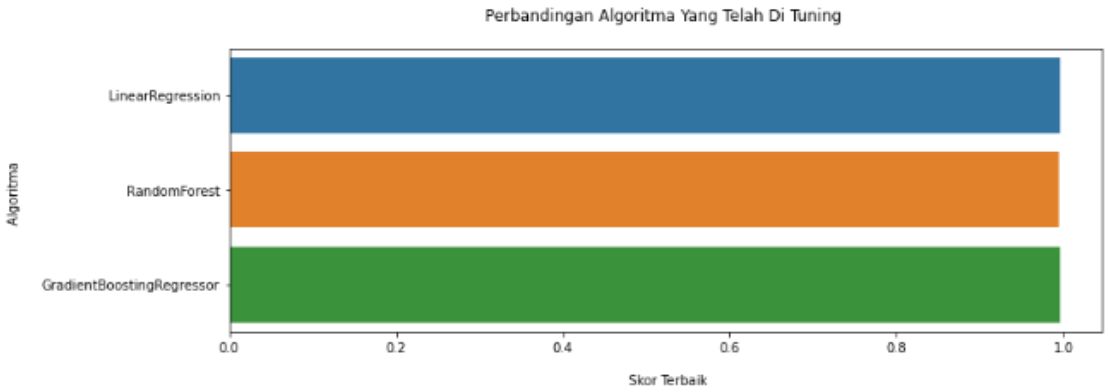
Gambar 17. Hasil Model Gradient Boosting

3.2 Plotting Hasil

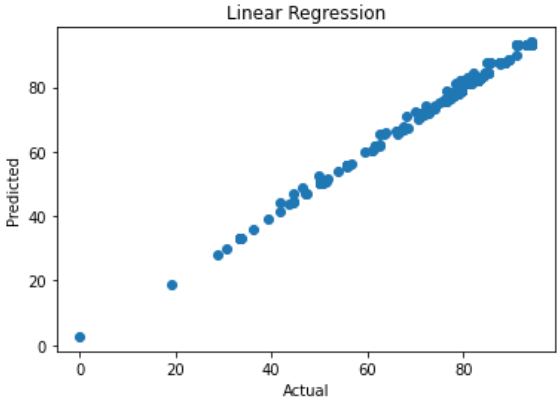
Berdasarkan hasil training yang telah dijabarkan pada sub-bab sebelumnya, maka plotting hanya ditujukan untuk menampilan hasil dari model regresi yang memiliki tingkat akurasi tertinggi yaitu *Linear Regression*. Pada Gambar 18 akan dijabarkan nilai yang diprediksi model dengan nilai asli yang terdapat pada dataset testing dan pada Gambar 20, ditampilkan plotting prediksi data secara linear.



Gambar 18. Hasil Prediksi Model Linear Regression



Gambar 19. Komparasi Ketiga Model Setelah Hyper Parameter Tuning



Gambar 20. Plotting Prediksi Secara Linear

3.3 Hasil Hyper Parameter Tuning

Hasil dari tuning model uang didapat dari ketiga model adalah Linear Regression tidak mendapat perubahan akurasi, Random Forest mendapat perubahan sebesar 0,02% menjadi 99,5% dari nilai akurasi sebelumnya dan Gradient Boost tidak mendapat perubahan akurasi. Komparsi ketiga model ini akan disajikan dalam bentuk diagram blok sebagaimana pada Gambar 19.

4. PENUTUP

Pengaruh teknologi yang mulai menyebar secara masif ke seluruh bidang memang dapat membantu siapapun. Prediksi WQI yang telah dibuat dalam penelitian ini juga dapat membantu untuk pencegahan masyarakat untuk tidak sembarangan meminum air guna mencegah penyebaran penyakit. Model *machine learning* yang telah dibuat memiliki nilai yang berbeda, namun pada penelitian ini telah ditentukan bahwasanya model *Linear Regression* yang dipakai sebagai model utama karena memiliki nilai R2 lebih tinggi (0.9965 / 96,5%) daripada model lainnya. *Hyper-Parameter Tuning* yang diharapkan dapat meningkatkan akurasi model ternyata tidak memberikan efek berlebih sehingga pada penelitian ini dinyatakan *tuning* model tidak diperlukan. Sesuai dengan hasil plotting dari Linear Regression, data diprediksi dengan baik dan persebaran data masih berdekatan dengan prediksi model (*robust).*

DAFTAR PUSTAKA

[1] World Health Organization, "Meeting the MDG drinking water and sanitation target: the urban and rural challenge of the decade", Geneva, 2006.

[2] N. M. Gazzaz, M. K. Yusoff, A. Z. Aris, H. Juahir, and M. F. Ramli, “Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors,” Marine Pollution Bulletin, vol. 64, no. 11, pp. 2409–2420, 2012.

[3] Kamaraj, K & Mohammad, Sameer. (2019). A Novel Approach on Various Machine Learning Algorithms for Predicting Ground Water Quality. SSRN Electronic Journal. 6. 37-40.

[4] P. Riyantoko, T. Fahrudin and K. Hindrayani, "Analisis Sederhana Pada Kualitas Air Minum Berdasarkan Akurasi Model Klasifikasi Dengan Menggunakan Lucifer Machine Learning", Seminar Nasional Sains Data 2021 (SENADA 2021), pp. 12-18, 2021.

[5] N. Radhakrishnan and A. S. Pillai, “Comparison of water quality classification models using machine learning,” 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020.

[6] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, “Efficient water quality prediction using supervised machine learning,” Water, vol. 11, no. 11, p. 2210, 2019.

[7] M. Hmoud Al-Adhaileh and F. Waselallah Alsaade, “Modelling and prediction of water quality by using artificial intelligence,” Sustainability, vol. 13, no. 8, p. 4259, 2021.

[8] S. B. Asadollah, A. Sharafati, D. Motta, and Z. M. Yaseen, “River Water Quality Index Prediction and Uncertainty Analysis: A Comparative Study of Machine Learning Models,” Journal of Environmental Chemical Engineering, vol. 9, no. 1, p. 104599, 2021.

[9] B. Liu, R. Tang, Y. Chen, J. Yu, H. Guo, and Y. Zhang, "Feature generation by convolutional neural network for click-through rate prediction," The World Wide Web Conference on - WWW '19, 2019.
[10] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, "An efficient feature generation approach based on Deep Learning and feature selection techniques for traffic classification," Computer Networks, vol. 132, pp. 81–98, 2018.
[11] K. A. Shah and G. S. Joshi, "Evaluation of water quality index for River Sabarmati, Gujarat, India," Applied Water Science, vol. 7, no. 3, pp. 1349–1358, 2015.
[12] A. Kumar, Learning predictive analytics with python: Gain practical insights into predictive modelling by implementing predictive analytics algorithms on public datasets with pyton. Packt Publishing, 2016.
[13] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and Deep Learning," Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.
[14] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," IEEE Access, vol. 8, pp. 125076–125096, 2020.
[15] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in Machine Learning: A new perspective," Neurocomputing, vol. 300, pp. 70–79, 2018.
[16] T. D.K., P. S. B.G, and F. Xiong, "Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques," Pattern Recognition Letters, vol. 128, pp. 544–550, 2019.
[17] X. Wan, "Influence of feature scaling on convergence of gradient iterative algorithm," Journal of Physics: Conference Series, vol. 1213, no. 3, p. 032021, 2019.
[18] A. Rácz, D. Bajusz, and K. Héberger, "Effect of dataset size and train/test split ratios in QSAR/QSPR multiclass classification," Molecules, vol. 26, no. 4, p. 1111, 2021.
[19] K. Pawluszek-Filipiak and A. Borkowski, "On the importance of train–test split ratio of datasets in automatic landslide detection by supervised classification," Remote Sensing, vol. 12, no. 18, p. 3054, 2020.
[20] B. Sravani and M. M. Bala, "Prediction of student performance using linear regression," 2020 International Conference for Emerging Technology (INCET), 2020.
[21] S. Liu, M. Lu, H. Li, and Y. Zuo, "Prediction of gene expression patterns with generalized linear regression model," Frontiers in Genetics, vol. 10, 2019.
[22] Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J. C.-W. Chan, P. van den Bossche, J. Van Mierlo, and N. Omar, "Random Forest regression for online capacity estimation of lithium-ion batteries," Applied Energy, vol. 232, pp. 197–210, 2018.
[23] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A comparative analysis of logistic regression, Random Forest and KNN models for the text classification," Augmented Human Research, vol. 5, no. 1, 2020.
[24] S. H. Samadi, B. Ghobadian, and M. Nosrati, "Prediction of higher heating value of biomass materials based on proximate analysis using gradient boosted regression trees method," Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, vol. 43, no. 6, pp. 672–681, 2019.
[25] F.-K. Wang and T. Mamo, "Gradient boosted regression model for the degradation analysis of prismatic cells," Computers & Industrial Engineering, vol. 144, p. 106494, 2020.