ISSN : 2614-0985 (Cetak)
ISSN : 2614-0977 (Online)
KOMPUTEK

url : http://studentjournal.umpo.ac.id/index.php/komputek

PREDIKSI KUALITAS AIR MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN RANDOM FOREST

Yefta Christian*,Jacky, Putra Agung Winata, Ricky, Nicholas Jeonanto, Ricky

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam

E-mail Korespondensi : yefta@uib.ac.id

History Artikel

Diterima : 18 Juli 2022 Disetujui : 26 September 2022 Dipublikasikan : 14 Oktober 2022

Abstract

Water is very important for the living of all organism including human, plant, or animal. Good quality of water is very important, the pollution of water can pose a dangerous risk. Currently to detect the quality of water, we need to use laboratory test. This test will need a complex analysis that cause need longer time to detect the water quality. Hence, the researcher will conduct research on water quality prediction by looking at some parameters that contain the information of the water. The dataset of water quality comes from Kaggle. The data mining method used is Random Forest algorithm and Naïve Bayes algorithm. These two algorithms are classification algorithm that can help us to predict the water quality. With using these two algorithms we obtained 79% accuracy for Random Forest algorithm and 55% accuracy for Naïve Bayes algorithm. After that we implement these two algorithms to a simple website with flask framework. Further research can try to use other algorithm such as ANN, K-NN, or Decision Tree.

Keywords: *Water Quality, Data Mining, Naïve Bayes, Random Forest*

Abstrak

Air sangat penting bagi kehidupan semua organisme termasuk manusia, tumbuhan, atau hewan. Kualitas air yang baik sangat penting, pencemaran air dapat menimbulkan risiko yang berbahaya. Saat ini untuk mendeteksi kualitas air perlu dilakukan uji laboratorium. Pengujian ini membutuhkan analisis yang kompleks sehingga membutuhkan waktu yang lebih lama untuk mendeteksi kualitas air. Oleh karena itu, peneliti akan melakukan penelitian tentang prediksi kualitas air dengan melihat beberapa parameter yang mengandung informasi air tersebut. Dataset kualitas air berasal dari Kaggle. Metode data mining yang digunakan adalah algoritma Random Forest dan algoritma Naïve Bayes. Kedua algoritma ini merupakan algoritma klasifikasi yang dapat membantu kita untuk memprediksi kualitas air. Dengan menggunakan kedua algoritma tersebut didapatkan akurasi 79% untuk algoritma Random Forest dan akurasi 55% untuk algoritma Naïve Bayes. Setelah itu kami mengimplementasikan kedua algoritma tersebut ke website sederhana dengan framework flask. Penelitian selanjutnya dapat mencoba menggunakan algoritma lain seperti ANN, K-NN, atau Decision Tree.

Kata Kunci: *Kualitas Air, Data mining, Naïve Bayes, Random Forest*

How to Cite: Jacky (2022). Prediksi Kualitas Air Menggunakan Algoritma Naïve Bayes Dan Random Forest. KOMPUTEK : Jurnal Teknik Universitas Muhammadiyah Ponorogo Vol 6 (2): Halaman 42-48

ISSN 2614-0985 (Print)
ISSN 2614-0977 (Online)

I. PENDAHULUAN

Air adalah zat kimia anorganik, transparan, dan tidak berwarna yang diperlukan untuk kelangsungan hidup sebagian besar organisme di dunia, baik manusia, hewan, maupun tumbuhan. Air dengan kualitas yang cukup menjadi hal yang penting bagi kelangsungan makhluk hidup. Polusi pada air tidak boleh melalui batas-batas tertentu, polusi yang melebihi batas-batas tertentu akan mengakibatkan pengguna air dalam keadaan berbahaya [1]. Air merupakan salah satu media yang paling mudah dalam menularkan suatu penyakit dengan jarak jangkau yang jauh. Kualitas air yang buruk diketahui menjadi salah satu factor utama penyebab penyakit berat [2].

Saat ini untuk mengetahui kualitas air diperlukannya uji laboratorium. Uji laboratorium ini memiliki analisis yang cukup rumit yang mengakibatkan diperlukannya waktu yang cukup lama untuk menentukan kualitas dan kelayakan air [3]. Berdasarkan masalah ini maka diperlukan suatu cara untuk membantu melakukan prediksi kualitas air dengan cepat dan mudah

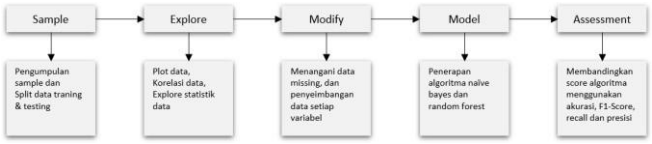
Penelitian sejenis juga telah pernah dilakukan oleh Jefferson dan Mia pada penelitian yang berjudul "*Pattern Extraction of Water Quality Prediction Using Machine Learning Algorithm of Water Reservoir*" [4]. Data yang digunakan merupakan data yang diperoleh dari hasil pemantauan kualitas air di Dunai Laguna di Filipina. Berdasarkan penelitian tersebut didapatkan hasil akurasi sebesar 87,69% menggunakan algoritma *Decision Tree*, 72,31% menggunakan algoritma *Naïve Bayes*, 78,46% menggunakan algoritma *Random Forest*, 73,8% menggunakan algoritma *Gradient Boost*, dan 72,3% menggunakan algoritma *Deep Learning*.

Penelitian ini akan menggunakan dataset *Water Quality Dataset* pada *Kaggle* yang telah digunakan juga pada penelitian yang dilakukan oleh Sai Sreeja Kurra, dkk. Penelitian tersebut menerapkan algoritma Decision Tree dan K-NN dengan pembagian data 80% sebagai *data training* dan 20% sebagai data testing. Penelitian tersebut mendapatkan akurasi sebesar 58.5% untuk algoritma Decision Tree dan 61.7% untuk algoritma K-NN [5].

Penelitian yang akan dilakukan penulis menggunakan dua jenis algoritma yang berbeda dari penelitian sebelumnya. Algoritma yang akan digunakan adalah *Naïve Bayes* dan *Random Fores*. Proses *data mining* dengan kedua algoritma ini akan dibantu oleh bahasa pemrograman Pyhon. Kemudian model yang dihasilkan pada penelitian ini juga akan diimplementasikan kedalam bentuk aplikasi berbasis web sederhana menggunakan *framework flask.*

II. METODOLOGI PENELITIAN

Penelitian ini akan menggunakan metode *Sample*, *Explore*, *Modify*, *Model*, dan *Assesment* (SEMMA). Metode SEMMA adalah metode *data mining* yang dikeluarkan oleh institusi SAS [6]. Institusi SAS mendefinisikan metode SEMMA sebagai metode pemilihan, eksplorasi, dan pemodelan data dalam jumlah besar untuk menemukan suatu pola bisnis yang belum diketahui. Gambar 1 merupakan alur dari penerapan metode SEMMA.



Gambar 1. Alur penerapan SEMMA

Berikut ini adalah penjelasan setiap tahapan SEMMA:

A. *Sample*

Pada tahap ini akan dilakukan proses pengumpulan data. *Dataset* penelitian ini akan menggunakan *dataset water quality* dari *Kaggle*. *Dataset* ini berisikan 3726 data yang terdiri dari 9 buah variable prediksi dan 1 buah variabel target (1 yang berarti layak minum dan 0 berarti tidak layak minum). Semua variabel berisikan data angka. Pada penelitian ini akan dilakukan pembagian antara *data training* dan *data testing* dengan proposi 70:30.

B. *Explore*

Tahap ini merupakan tahap akan dilakukannya eksplorasi data. Eksplorasi data akan dilakukan dalam bentuk grafik, *plot*, dan data statistik. Tahap ini juga berfungsi untuk mendeteksi data-data bermasalah dan data-data *null*. Melalui hasil ekplorasi data ini akan ditentukan proses yang akan dilakukan terhadap data yang tersedia [7].

C. *Modify*

Pada tahap ini akan dilakukan proses *data cleaning*. *Data Cleaning* adalah suatu proses untuk melakukan pembersihan data [8]. *Data cleaning* yang dilakukan pada penelitian ini adalah dengan mengganti data missing menggunakan nilai rata-rata dari variable. Pada *dataset* yang digunakan terdapat 3 buah variable yang memiliki data missing yaitu *PH*, *Sulfate*, dan *Trihalomethanes*. Selain *data cleaning*, juga dilakukan penyeimbangan data untuk setiap kelas target sehingga memiliki jumlah data yang sama. Hal ini dilakukan untuk meningkatkan kualitas dari data tersebut [9].

D. *Model*

Setelah semua data telah melalui tahap data cleaning, kemudian akan dilakukan penerapan algoritma prediksi. *Dataset water quality* merupakan data klasifikasi. Pada penelitian ini akan digunakan algoritma *naïve bayes* dan *random forest* yang merupakan algoritma yang dapat digunakan pada data klasifikasi. Klasifikasi adalah salah satu metode dalam data mining. Klasifikasi merupakn suatu proses untuk mengelompokkan objek-objek dalam kelas-kelas tertentu [10].

Algoritma *naïve bayes* merupakan salah satu algoritma klasifikasi yang dapat digunakan untuk melakukan prediksi probabilitas keanggotaan suatu kelas. Algoritma ini didasarkan pada teorema Bayes [11].

Algoritma *random forest* adalah algoritma *data mining* yang dilakukan melalui penggabungan pohon (*tree*) dengan melakukan *training* pada *sample* data yang dimiliki. Algoritma *random forest* ini cocok untuk digunakan bila memiliki jumlah data yang cukup besar [12].

E. Assesment

Pada tahap ini akan dilakukan proses pengujian hasil kedua algoritma yang digunakan. Pengujian akan dilakukan berdasarkan hasil dari *confusion matrix*. *Confusion matrix* adalah suatu matrix yang berisikan informasi mengenai benar atau salahnya suatu prediksi dan keadaan sebenarnya [13]. Tabel 1 merupakan contoh bentuk dari *confusion matrix*.

Tabel I. Confusion Matrix

	Positive Predicted	**Negative Predicted**
Positive	True Positive	True Negative
Negative	False Positive	False Negative

- *True Positive*: merupakan total dari kasus saat kondisi sebenarnya adalah benar dan hasil prediksinya juga benar. Pada penelitian ini adalah hasil prediksi dan aktual adalah air layak minum.
- *False Negative*: merupakan total dari kasus saat kondisi sebenarnya adalah salah, namun di prediksi benar. Pada penelitian ini adalah kondisi tidak layak minum yang di prediksi sebagai layak minum.
- *False Positive*: merupakan total dari kasus saat kondisi sebenarnya adalah benar, namun di prediksi salah. Pada penelitian ini adalah kondisi layak minum yang di prediksi sebagai tidak layak minum.
- *True Negative*: merupakan total dari kasus saat kondisi sebenarnya adalah salah dan hasil prediksinya juga salah. Pada penelitian ini adalah kondisi tidak layak minum dan juga di prediksi sebagai tidak layak minum.

Berdasarkan *confusion matrix* ini akan dilakukan perhitungan *F1-Score*, akurasi, *recall*, dan *precision*. Rumus 1 merupakan rumus yang digunakan untuk melakukan perhitungan *precision*, rumus 2 merupakan rumus yang digunakan untuk melakukan perhitungan *recall*, rumus 3 merupakan rumus yang digunakan untuk melakukan perhitungan akurasi, sedangkan rumus 4 digunakan untuk melakukan perhitungan *F1-Score* [14].

$$precision = \frac{TP}{TP \$ FP} \quad (1)$$

$$recall = \frac{TP}{TP \$ F\&} \quad (2)$$

$$akurasi = \frac{TP \$ T\&}{TP \$ T\& \$ FP \$ F\&} \quad (3)$$

$$F1\ Score = \frac{2TP}{2TP\$FP\$F\&} \quad (4)$$

Kemudian model yang dihasilkan akan diimplementasikan kedalam aplikasi web sederhana menggunakan *framework flask*. *Framwork flask* adalah *framework* pemrograman yang menggunakan Bahasa pemprograman Python. Framework ini digunakan untuk membantu proses pembuatan aplikasi berbasis web [15].

III. HASIL DAN PEMBAHASAN

A. Sample

Dataset pada penelitian ini diambil dari *Kaggle*. Seluruh data berjumlah 3276 data. Berikut ini merupakan contoh data yang akan digunakan pada penelitian ini.

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0
...	...	...	...	...	...	...	...	...	...	...
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	13.894419	66.687695	4.435821	1
3272	7.808856	193.553212	17329.802160	8.061362	NaN	392.449580	19.903225	NaN	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	NaN	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	NaN	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	NaN	327.459760	16.140368	78.698446	2.309149	1

3276 rows × 10 columns

Gambar 2. Contoh Dataset

Dari 3276 data ini, peneliti akan membagi 70% data sebagai *data training* dan 30% sebagai *data testing*.

B. Explore

Eksplorasi data yang dilakukan untuk mengetahui jenis tipe data dan kondisi setiap variabel. Berdasarkan gambar 3 kita dapat mengetahui bahwa terdapat 9 variabel yang bertipe data float64 dan 1 variabel bertipe data int64. Dari gambar ini juga kita dapat melihat bahwa terdapat beberapa variable yang memiliki *data null* yaitu *PH*, *Sulfate*, dan *Trihalomethanes*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   ph               2785 non-null   float64
 1   Hardness         3276 non-null   float64
 2   Solids           3276 non-null   float64
 3   Chloramines      3276 non-null   float64
 4   Sulfate          2495 non-null   float64
 5   Conductivity     3276 non-null   float64
 6   Organic_carbon   3276 non-null   float64
 7   Trihalomethanes  3114 non-null   float64
 8   Turbidity        3276 non-null   float64
 9   Potability       3276 non-null   int64
dtypes: float64(9), int64(1)
memory usage: 256.1 KB
```

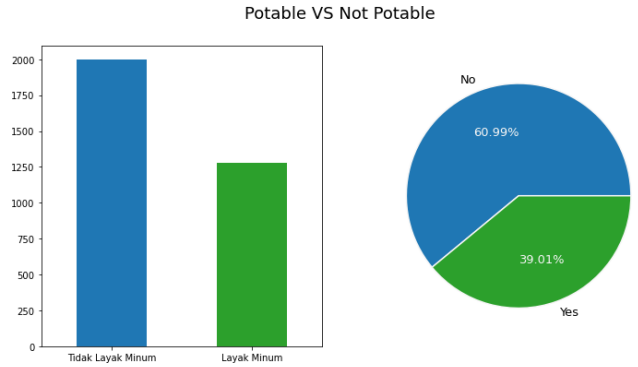
Gambar 3. Informasi tipe data setiap variable

Pada gambar 4, kita dapat melihat informasi mengenai rata-rata, *median*, dan nilai maksimum setiap variabel.

	count	mean	std	min	25%	50%	75%	max
ph	2785.0	7.080795	1.594320	0.000000	6.093092	7.036752	8.062066	14.000000
Hardness	3276.0	196.369496	32.879761	47.432000	176.850538	196.967627	216.667456	323.124000
Solids	3276.0	22014.092526	8768.570828	320.942611	15666.690297	20927.833607	27332.762127	61227.196008
Chloramines	3276.0	7.122277	1.583085	0.352000	6.127421	7.130299	8.114887	13.127000
Sulfate	2495.0	333.775777	41.416840	129.000000	307.699498	333.073546	359.950170	481.030642
Conductivity	3276.0	426.205111	80.824064	181.483754	365.734414	421.884968	481.792304	753.342620
Organic_carbon	3276.0	14.284970	3.308162	2.200000	12.065801	14.218338	16.557652	28.300000
Trihalomethanes	3114.0	66.396293	16.175008	0.738000	55.844536	66.622485	77.337473	124.000000
Turbidity	3276.0	3.966786	0.780382	1.450000	3.439711	3.955028	4.500320	6.739000
Potability	3276.0	0.390110	0.487849	0.000000	0.000000	0.000000	1.000000	1.000000

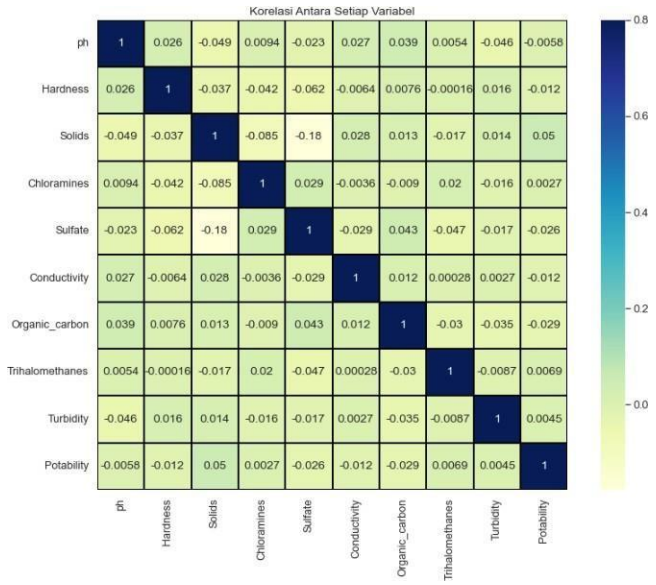
Gambar 4. Deskripsi data setiap variable

Gambar 5 merupakan pembagian data untuk setiap kelas pada *dataset*. Terdapat 1998 data (60.99%) yang memiliki kelas tidak layak minum dan 1278 data (39.01%) yang layak minum.



Gambar 5. Pembagian data setiap kelas

Gambar 6 merupakan tabel pembagian korelasi antara setiap variable.



Gambar 6. Korelasi data

C. *Modify*

Pada tahap ini akan dilakukan proses *data cleaning* yang bertujuan untuk mempersiapkan data sebelum dilakukannya tahap penerapan algoritma. Gambar 7 merupakan pembagian hasil pembagian data yang memiliki data missing.

```
ph                491
Hardness            0
Solids              0
Chloramines         0
Sulfate           781
Conductivity        0
Organic_carbon      0
Trihalomethanes   162
Turbidity           0
Potability          0
dtype: int64
```

Gambar 7. Data missing

Dari gambar tersebut diketahui baha variable PH memiliki 491 *data missing*, variabel *sulfate* memiliki 781 *data missing*, dan variabel *trihalomethanes* memiliki 162 *data missing*. Proses *data cleaning* ini akan dilakukan dengan menggunakan python. Gambar 5 merupakan *coding* pada python untuk mengganti semua *data missing* dengan nilai rata-rata setiap variabel.

```
In [6]: df.fillna(df.mean(),inplace=True)
        df.isnull().sum()

Out[6]: ph              0
        Hardness        0
        Solids          0
        Chloramines     0
        Sulfate         0
        Conductivity    0
        Organic_carbon  0
        Trihalomethanes 0
        Turbidity       0
        Potability      0
        dtype: int64
```

Gambar 8. *Data Cleaning*

Setelah *data cleaning*, kemudian akan dilakukan penyeimbangan data antara setiap kelas. Kelas "tidak layak minum" memiliki lebih banyak data daripada kelas "layak minum". Dikarenakan hal ini dilakukanlah proses *resample* untuk melakukan penyeimbangan data. Gambar 9 merupakan proses *resample* yang dilakukan menggunakan python.

```
In [9]: notpotable = df[df['Potability']==0]
        potable = df[df['Potability']==1]

        df_minority_upsampled = resample(potable, replace = True, n_samples = 1998)

        df = pd.concat([notpotable, df_minority_upsampled])
        df = shuffle(df)
        df.Potability.value_counts()

Out[9]: 1    1998
        0    1998
        Name: Potability, dtype: int64
```

Gambar 9. Data resample

D. *Model*

Sebelum melakukan penerapan model, diperlukan untuk membagi data menjadi 2 bagian, yaitu *data training* dan *data testing*. Gambar 10 menunjukan proses pembagian data

dengan rasio 70% sebagai *data training* dan 30% sebagai *data testing*.

```
In [13]: X_train, X_test, Y_train, Y_test = train_test_split(x,y, test_size = 0.3)
```

Gambar 10. Split dataset

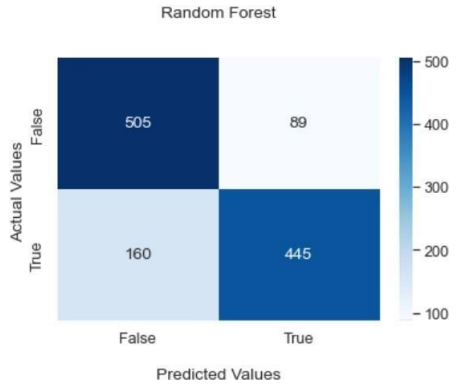
Setelah pembagian data *sample*, proses selanjutnya adalah menerapkan algoritma kedalam data tersebut. Algoritma yang digunakan pada penelitian ini adalah algoritma *naïve bayes* dan *random forest*. Gambar 11 menunjukan proses yang dilakukan untuk penerapan algoritma *naïve bayes* dan *random forest*.

```
In [14]: rf = RandomForestClassifier()
         GNB = GaussianNB()

In [15]: models = [ ('Random Forest', rf), ('Naive Bayes', GNB)]


         for model_name, model in models:
             model.fit(X_train, Y_train)
             y_pred = model.predict(X_test)
             accuracy = accuracy_score(Y_test,y_pred)
             print('{:s} : {:.2f}'.format(model_name, accuracy))
             print(classification_report(Y_test,y_pred))
             print('\n')
```

Gambar 11. Implementasi Algoritma

E. Assesment

Pada tahap ini akan dilakukan pengujian terhadap kedua algoritma yang digunakan. Pengujian akan dilakukan dengan *confusion matrix*. Gambar 12 merupakan hasil dari *confusion matrix* algoritma *random forest*.



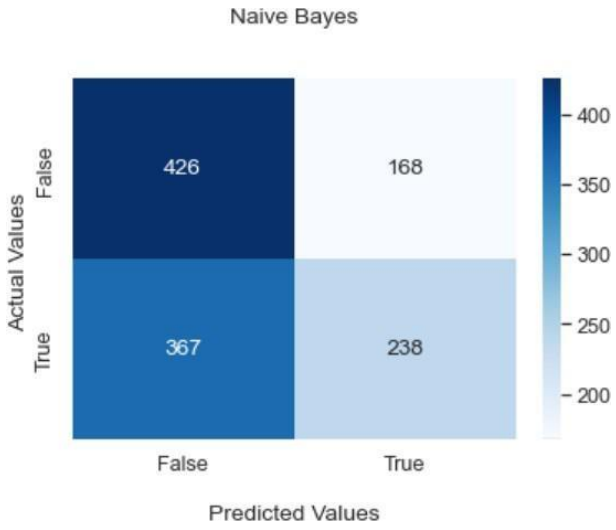
Gambar 12. Confusion matrix random forest

Berdasarkan *confusion matrix* ini akan dilakukan perhitungan akurasi, *recall*, *precision*, dan *f1-score*. Berdasarkan gambar 13 dapat dilihat bahwa algoritma *random forest* memiliki akurasi sebesar 79%, *precision* sebesar 83%, *recall* sebesar 74%, dan *f1-score* sebesar 78%.

	precision	recall	f1-score	support
0	0.76	0.85	0.80	594
1	0.83	0.74	0.78	605
accuracy			0.79	1199
macro avg	0.80	0.79	0.79	1199
weighted avg	0.80	0.79	0.79	1199

Gambar 13. Hasil pengujian random forest

Sedangkan gambar 14 merupakan *confusion matrix* untuk algoritma *naïve bayes*.



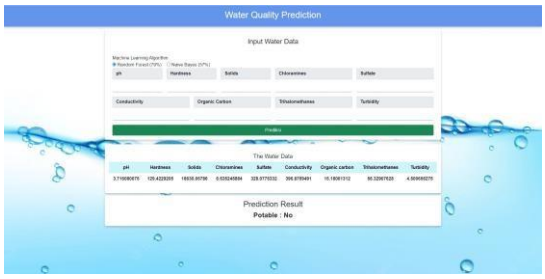
Gambar 14. Confusion matrix naïve bayes

Berdasarkan confusion matrix ini juga akan dilakukan perhitungan akurasi, *recall*, *precision*, dan *f1-score*. Berdasarkan hasil perhitungan yang ditunjukkan oleh gambar 15, diketahui bahwa algoritma *naïve bayes* untuk kasus prediksi kualitas air memiliki akurasi sebesar 55% dengan *precision* sebesar 59%, *recall* sebesar 39% dan *f1-score* sebesar 57%.

	precision	recall	f1-score	support
0	0.54	0.72	0.61	594
1	0.59	0.39	0.47	605
accuracy			0.55	1199
macro avg	0.56	0.56	0.54	1199
weighted avg	0.56	0.55	0.54	1199

Gambar 15. Hasil pengujian random forest

Kedua model algoritma ini akan diimplementasikan ke dalam aplikasi web sederhana. Aplikasi ini dibuat menggunakan *framework flask*. Gambar 16 merupakan hasil akhir aplikasi prediksi sederhana berbasis web. Pada aplikasi ini user akan diminta untuk memasukan informasi-informasi mengenai air dan jenis algoritma yang ingin digunakan. Setelah user menekan tombol prediksi, aplikasi ini akan langsung menampilkan hasil prediksi. Hasil prediksi "*No*" menandakan kualitas air tidak layak minum dan sebaliknya hasil prediksi "*Yes*" menandakan kualitas air layak minum.



Gambar 16. Hasil pengujian random forest

V. KESIMPULAN

Penelitian ini bertujuan untuk melakukan pengujian algoritma *naïve bayes* dan *random forest* dalam melakukan prediksi kualitas air. *Dataset* yang digunakan pada penelitian ini diambil dari *dataset water quality* pada *Kaggle*. Berdasarkan hasil pengujian kedua jenis algoritma menggunakan metode *confusion matrix*, didapatkan hasil akurasi sebesar 79%, *precision* sebesar 83%, *recall* sebesar 74%, dan *f1-score* sebesar 78% untuk algoritma *random forest*. Sedangkan algorima *naïve bayes* memiliki akurasi sebesar 55% dengan *precision* sebesar 59%, *recall* sebesar 39% dan *f1-score* sebesar 57%. Algoritma *random forest* memiliki akurasi yang lebih besar daripada algoritma *naïve bayes*, sehingga algoritma *random forest* ini lebih cocok untuk digunakan dalam proses prediksi kualitas air.

DAFTAR PUSTAKA

[1] Md. M. Hassan *et al.*, "Efficient Prediction of Water Quality Index (WQI) Using Machine Learning Algorithms," *Atlantis Press International B.V.*, 2021.

[2] U. Ahmed, R. Mumtaz, H. Anwar, A. A. Shah, R. Irfan, and J. García-Nieto, "Efficient Water Quality Prediction Using Supervised Machine Learning," *MDPI*, vol. 11, pp. 1–14, Oct. 2019.

[3] R. Rachmat, Y. H. Chrisnanto, and F. R. Umbara, "Sistem Prediksi Mutu Air di Perusahaan Daerah Air Minum Tirta Raharja Menggunakan K-Nearest Neighbors (K-NN)," in *Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK) ke 4*, 2020, pp. 189–193.

[4] J. L. Lerios and M. v Villarica, "Pattern Extraction of Water Quality Prediction Using Machine Learning Algorithms of Water Reservoir," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 6, 2019.

[5] S. S. Kurra, S. G. Naidu, S. Chowdala, S. C. Yellanki, and B. E. Sunanda, "Water Quality Prediction Using Machine Learning," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 5, pp. 692–696, 2022, [Online]. Available: www.irjmets.com

[6] R. F. R. Forradellas, S. L. N. Alonso, M. L. Rodriguez, and J. Jorge-Vazquez, "Applied Machine Learning in Social Sciences: Neural Networks and Crime Prediction," *MDPI*, vol. 10, no. 1, pp. 1–20, Jan. 2021, doi: 10.3390/socsci10010004.

[7] H. Sabita, Fitria, and R. Herwanto, "Analisa dan Prediksi Iklan Lowongan Kerja Palsu Dengan Metode Natural Language Programming dan Machine Learning," *Jurnal Informatika*, vol. 21, no. 1, pp. 14–22, Jun. 2021.

[8] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, "Big Data Cleaning Based on Mobile Edge Computing in Industrial Sensor-Cloud," *Journal of Latex Class Files*, vol. 16, no. 2, pp. 1321–1329, Feb. 2020.

[9] J. Duncan, R. Kapoor, A. Agarwal, C. Singh, and B. Yu, "Veridical Flow: a Python Package for Building Trustworthy Data Science Pipelines With PCS," *Journal of Open Source Software*, Jan. 2022.

[10] Y. Christian, "Application of K-Means Algorithm for Clustering the Quality of Lecturer Learning at Batam International University," *International Journal of Information System & Technology*, vol. 3, no. 2, pp. 191–199, 2020.

[11] H. Annur, "KLASIFIKASI MASYARAKAT MISKIN MENGGUNAKAN METODE NAÏVE BAYES," *ILKOM*, vol. 10, no. 2, pp. 160–165, 2018.

[12] L. J. Muhammad, A. A. Haruna, I. A. Mohammed, M. Abubakar, B. G. Badamasi, and J. M. Amshi, "Performance Evaluation of Classification Data Mining Algorithms on Coronary Artery Disease Dataset," in *9th International Conference on Computer and Knowledge Engineering (ICCKE 2019)*, 2019, pp. 1–5.

[13] S. Turgut, M. Dagtekin, and T. Ensari, "Microarray Breast Cancer Data Classification Using Machine Learning Methods," *IEEE*, 2018.

[14] I. Markoulidakis, I. Rallis, I. Georguolas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *MDPI*, pp. 1–22, Nov. 2021.

[15] M. Singh, A. Verma, A. Parasher, N. Chauhan, and G. Budhiraja, "Implementation of Database Using Python Flask Framework," *International Journal of Engineering and Computer Science*, vol. 8, no. 12, pp. 24890–24893, Dec. 2019.