Penerapan Algoritma K-Nearest Neighbor Untuk Memprediksi Kualitas Air Yang Dapat Dikonsumsi

Application of K-Nearest Neighbor Algorithm to Predict Consumable Water Quality

Hardiana Said[1], Nurhafifah Matondang[2], Helena Nurramdhani Irmanda[3]
[1,2,3]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta
E-mail: [1]hardiana@upnvj.ac.id, [2]nurhafifahmatondang@upnvj.ac.id, [3]helenairmanda@upnvj.ac.id

Abstrak

Kualitas air yang aman untuk dikonsumsi sangatlah penting bagi kesehatan masyarakat luas di setiap daerah, namun kualitas air diberbagai daerah semakin menurun terutama untuk kebutuhan manusia dalam hal air minum, dampak dari kualitas air yang tidak aman untuk dikonsumsi dapat menyebabkan penyakit seperti kolera, diare, hepatitis A dan lainnya, hal ini dikarenakan air yang memiliki sanitasi yang buruk dan zat-zat yang melebihi kadar standar. Penelitian ini dilakukan untuk mengetahui hasil evaluasi dari model yang dihasilkan untuk dapat memprediksi kualitas air yang dapat dikonsumsi atau tidaknya dengan menerapkan algoritma klasifikasi *data mining* yaitu adalah algoritma *K-Nearest Neighbor*. Algoritma ini diterapkan untuk menghitung probabilitas kualitas air yang aman atau tidak untuk dikonsumsi berdasarkan data rekaman yang diambil dari lingkungan sekitar terutama di daerah padat penduduk. Kumpulan data diperoleh dari website *kaggle*, hasil pemodelan diukur menggunakan tabel *Confusion Matrix* untuk menghitung akurasi. Setelah diuji, model ini memiliki tingkat akurasi tertinggi 85,24% dengan nilai k (tetangga terdekat) = 3.

Kata kunci: Kualitas Air, Prediksi, Klasifikasi, K-Nearest Neighbor

Abstract

The quality of water that is safe for consumption is very important for the health of the wider community in every area, but the quality of water in various areas is decreasing, especially for human needs in terms of drinking water, the impact of water quality that is not safe for consumption can cause diseases such as cholera, diarrhea, hepatitis. A and others, this is due to water that has poor sanitation and substances that exceed standard levels. This study was conducted to determine the results of the evaluation of the resulting model to be able to predict the quality of water that can be consumed or not by applying a data mining classification algorithm, namely the K-Nearest Neighbor algorithm. This algorithm is applied to calculate the probability of water quality being safe or not for consumption based on recorded data taken from the surrounding environment, especially in densely populated areas. The data collection was obtained from the kaggle website, the modeling results were measured using the Confusion Matrix table to calculate accuracy. After being tested, this model has the highest accuracy rate of 85.24% with a value of k (nearest neighbor) = 3.

Keywords: Water Quality, Prediction, Classification, K-Nearest Neighbor

1. PENDAHULUAN

Kebutuhan air bersih setiap tahunnya mengalami peningkatan namun ketersediaan air bersih malah terbatas, hal ini disebabkan pembangunan yang semakin banyak tanpa memperhatikan keseimbangan lingkungan sekitar dan sempitnya daerah resapan terutama di daerah perkotaan, hal ini menimbulkan masalah yang serius yaitu kurangnya ketersediaan sumber air bersih. Di Indonesia sendiri pencemaran air menjadi masalah utama, sumber utama

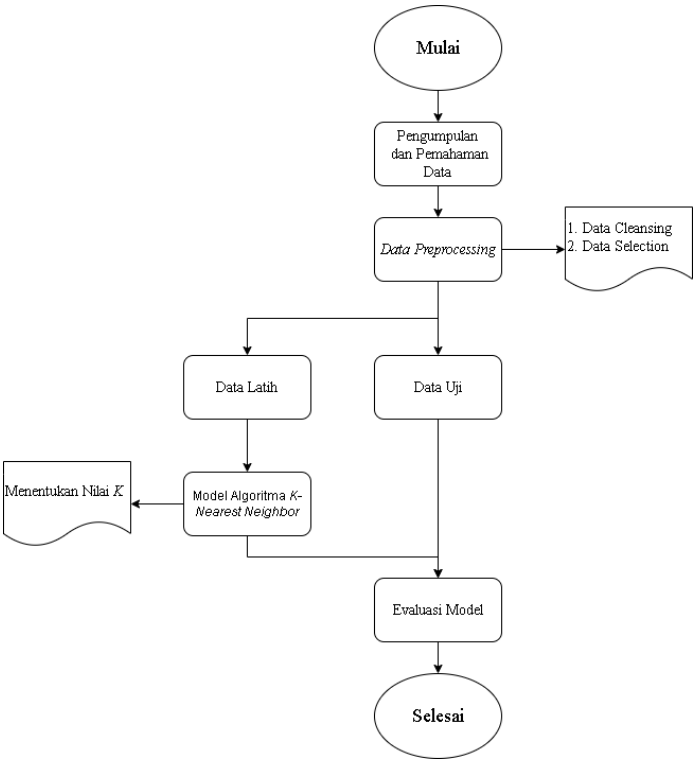
dari pencemaran air yang banyak dihasilkan dari limbah rumah tangga serta domestik, hal ini menyebabkan menurunnya kualitas air dan ketersediaan air bersih disetiap daerah. Akibat dari kualitas air yang buruk dapat menimbulkan penyakit yang cukup serius salah satunya ialah penyakit diare, disetiap tahunnya diperkirakan kurang lebih ada 800 ribu orang yang meninggal akibat diare, hal ini ditimbulkan akibat dari konsumsi air yang kurang bersih atau terkontaminasi, kurangnya kebersihan tangan saat mengkonsumsi air, dan masalah sanitasi yang ada pada air yang diminum [1]. Untuk mengetahui kualitas air yang layak dikonsumsi bagi masyarakat luas, perlu adanya identifikasi dini yang dapat dijadikan sebagai informasi awal dengan menggunakan teknik *data mining* untuk mengidentifikasi dini masalah ini, dalam penelitian ini teknik klasifikasi yang digunakan bertugas untuk melakukan prediksi suatu kategori label yang sebelumnya tidak diketahui serta membedakan objek-objek yang satu dengan lainnya, pada teknik ini juga terdapat algoritma klasifikasi yang dapat digunakan seperti prediksi dan lainnya, yang mana algoritma ini salah satunya adalah *K-Nearest Neighbor*.

Penelitian terkait pertama yang pernah dilakukan sebelumnya dengan menggunakan algoritma *K-Nearest Neighbor* yaitu dengan melakukan komparasi antara KNN dan *K-Means*, yang menghasilkan nilai akurasi yang lebih baik dengan menggunakan KNN sebesar 88% sedangkan *K-Means* sebesar 76%, selain itu data kualitas air yang digunakan pada penelitian terkait yaitu berdasarkan data dari PDAM daerah Bandarmasih, Banjarmasin [2]. Selanjutnya penelitian terkait lainnya yang mana melakukan prediksi layak atau tidak layaknya kualitas air yang ada pada PDAM daerah Surakarta, dengan menggunakan algoritma KNN menghasilkan akurasi sebesar 82,5% dan selain itu juga parameter kandungan record data kualitas air yang digunakan yaitu terdapat kandungan fisika, kimia dan biologis [3]. Penelitian terkait lainnya yaitu dengan melakukan komparasi antara algoritma *K-Nearest Neighbor* dan *Support Vector Machine* (SVM), yang mana KNN menghasilkan akurasi yang lebih tinggi sebesar 88,94% dibandingkan SVM yang hanya sebesar 87,71%, namun data masih dapat mengalami kesalahan klasifikasi dengan KNN sebesar 11,06% dan SVM sebesar 12,29% disebabkan sifat data yang masih *imbalanced* [4].

Berdasarkan penelitian terkait diatas, penelitian ini akan melakukan prediksi kualitas air yang dapat dikonsumsi atau tidaknya, data yang diambil besifat universal (didapatkan dari berbagai tempat) yang didapatkan pada *website Kaggle* dan selain itu perbedaan penelitian ini dengan yang lainnya terdapat pada data yang ada, yang mana pada bagian jenis kandungan pada record data kualitas air yang didapatkan dalam bentuk dataset ini, data kualitas air yang ada yaitu hanya dengan mengambil data dengan parameter kandungan zat kimia dan biologis yang terkandung pada data kualitas air tanpa adanya variabel-variabel dari pengaruh fisika, prediksi kualitas air dengan data yang diambil masih dapat dilakukan dikarenakan kandungan zat kimia yang ada pada record data sesungguhnya juga mempresentasikan perubahan secara fisik atau seperti perubahan warna, rasa dan bau pada air dan selain itu juga pada penelitian ini menggunakan aplikasi *machine learning* dalam membantu menerapkan proses klasifikasi dengan algoritma KNN yaitu menggunakan *jupyther notebook* dengan begitu hasil model KNN yang telah dibuat akan lebih mudah untuk diterapkan kedalam sistem prediksi atau di simpan dalam *library machine learning* untuk dipakai kembali. Kumpulan data ini bersifat sekunder yang diupload oleh MsSmartyPants pada tahun 2021 dengan judul *Water Quality (Dataset for water quality classification)*. Dengan total data berjumlah 7999 record yang mana data ini belum bersifat *balance*, maka data tersebut akan melewati tahapan proses *data preprocessing* seperti *cleansing*, *selection* dan *balancing*. Selanjutnya algoritma *K-Nearest Neighbor* akan digunakan atau dipilih dalam membantu memproses data awal untuk menjadi sebuah model yang nanti dapat dilakukan prediksi dan menghasilkan nilai evaluasi akhir dari algoritma KNN dengan menggunakan tabel *confusion matrix*, pemilihan algoritma ini didasarkan penelitian terkait diatas yang mana KNN menghasilkan akurasi tertinggi dibandingkan algoritma lainnya.

2. METODE PENELITIAN

Tahapan penelitian ini dimulai dengan pengumpulan dan pemahaman data yang didapatkan, melakukan persiapan data dengan *preprocessing* data yang mana didalamnya terdapat beberapa proses tahapan lagi seperti *cleansing* data, *selection* data dan *balancing* data, data yang telah diolah akan displit untuk selanjutnya akan dimulai proses klasifikasi dengan menggunakan algoritma *K-Nearest Neighbor*, dan terakhir dilakukannya evaluasi model untuk mengetahui potensi dari algoritma *K-Nearest Neighbor* dalam memprediksi kualitas air. Untuk lebih detailnya mengenai tahapan dari penelitian ini dapat dilihat pada Gambar 1 dibawah.



Gambar 1 Alur Penelitian

2.1 Pengumpulan dan Pemahaman Data

Pengumpulan dan pemahaman data sangat diperlukan dalam tahap awal *data mining. Data mining* adalah proses pengerukan atau pengumpulan informasi penting dari sejumlah besar data. Proses penambangan data sering menggunakan teknik statistik dan matematika untuk memanfaatkan teknologi kecerdasan buatan [5]. Dengan adanya analisis mengenai data akan banyak sekali informasi yang akan didapatkan seperti banyaknya jumlah record data, jenis data, dan banyaknya kolom data. Data kualitas air yang didapatkan sendiri bersumber dari website *www.kaggle.com* yang diupload oleh MsSmartyPants pada tahun 2021 dengan judul *Water Quality (Dataset for water quality classification)*. Variabel yang digunakan pada penelitian ini untuk memprediksi kualitas air yaitu sejumlah 21 variabel yang sudah termasuk 20 atribut dan sudah termasuk kelas/label, untuk contoh tabel dapat dilihat pada Gambar 2 dibawah.

Tabel 1 Atribut dan Kelas Pada Data Kualitas Air

No	Variabel	Rentan Nilai
1	*aluminium*	0 - 5,05
2	*ammonia*	-0.08 - 29,84
3	*arsenic*	0 - 1,05

4	*barium*	0 - 4,49
5	*cadmium*	0 - 0,13
6	*chloramine*	0 - 8,68
7	*chromium*	0 - 0,90
8	*copper*	0 - 2
9	*flouride*	0 - 1,50
10	*lead*	0 - 0,20
11	*nitrates*	0 - 19,83
12	*nitrites*	0 - 2,93
13	*mercury*	0 - 0,01
14	*perchlorate*	0 - 60,01
15	*radium*	0 - 7,99
16	*selenium*	0 - 0,100
17	*silver*	0 - 0,5
18	*uranium*	0 - 0,09
19	*bacteria*	0 - 1
20	*viruses*	0 - 1
21	*is_safe*	0 - 1

2.2 Data Preprocessing

Data Preprocessing adalah teknik penambangan data awal untuk mengubah data mentah menjadi format dan informasi yang lebih efisien dan berguna, format data mentah dari sumber yang berbeda sering kali memiliki kesalahan, nilai yang hilang, dan inkonsistensi, oleh karena itu format perlu dilakukan pembenahan agar hasil data mining tepat dan akurat [6]. Dengan pengolahan *data preprocessing* yang baik ini dapat mempengaruhi hasil dari *data mining* yang mana dampaknya akan lebih menghasilkan akurasi yang lebih baik dan memproses informasi baru yang lebih jelas, tahapan ini terdiri dari yaitu data *cleansing*, data *selection* dan data *balancing*.

2.3 Split Data

Proses pembagian data adalah salah satu metode yang dapat dipakai untuk menilai kinerja adalah dengan menambah atau mengurangi persentase data latih dan tes, sehingga pada model *machine learning* nanti akan mendapatkan hasil performa yang maksimal. Metode evaluasi model ini membagi dataset menjadi dua bagian, bagian ini akan digunakan untuk data latih dan data uji untuk proporsi tertentu. Pada penelitian ini presentase serta jumlah dari pembagian data akan dilakukan pada beberapa percobaan untuk melihat hasil akurasi tertinggi.

2.4 K-Nearest Neighbor

K-Nearest Neighbor atau KNN adalah Algoritma yang digunakan untuk mengklasifikasikan data berdasarkan data latih (*training record*) yang diperoleh dari tetangga terdekat (*nearest neighbor*), dimana *k* adalah bilangan tetangga terdekat [7]. Pada tahap pemodelan data, metode klasifikasi yang diterapkan yaitu dengan menggunakan algoritma *K-Nearest Neighbor* pada *dataset*, dimodel algoritma *K-Nearest Neighbor* memiliki beberapa tahap kerja, yaitu:

1. Pertama yaitu menentukan jumlah tetangga (*k*),
2. Menghitung jarak data baru ke jumlah tetangga yang ditentukan dengan berbagai rumus perhitungan salah satunya *euclidean distance*,
3. Mengambil *k* tetangga paling dekat untuk dibuat suatu keputusan prediksi berdasarkan hasil perhitungan jarak.

Perhitungan jarak antar objek data baru dengan data lama ini dapat diukur oleh beberapa cara pengukuran salah satunya yaitu dengan memakai rumus euclidean distance, dan rumus dari perhitungan euclidean distance dapat dilihat sebagai berikut:

$$Euclidean\ Distance = \sqrt{(a1 - b1)^2 + .. + (an - bn)^2} \quad (1)$$

Keterangan:
a = a1, a2, a3, …., an hingga nilai ke n
b = b1, b2, b3, …., bn hingga nilai ke n

2.5 Evaluasi Model

Pada tahap evaluasi model ini untuk mengetahui kinerja suatu algoritma dalam mengetahui keakuratan dari model diperlukan perhitungan *confusion matrix*. *Confusion Matrix* merupakan Salah satu teknik yang dapat digunakan untuk mengukur kinerja suatu model khusunya kasus klasifikasi (*supervised learning*) pada *machine learning* [8]. *Confusion Matrix* memiliki 4 kombinasi berbeda dari yang mana yaitu terdapat nilai prediksi dan nilai aktual, tabel evaluasi model terdapat di tabel berikut:

Tabel 2 Tabel *Confusion Matrix* untuk Evaluasi Model

Confusion Matrix		Nilai Aktual	
		Positif	**Negatif**
Nilai Prediksi	**Positif**	*True Positives*	*False Positives*
	Negatif	*False Negatives*	*True Negatives*

Keterangan 4 nilai pada tabel, yaitu sebagai berikut:

a. *True Positives* (TP): Jumlah data yang memiliki nilai positif dan diharapkan untuk bernilai positif.
b. *False Positives* (FP): Jumlah data yang memiliki nilai negatif dan diharapkan untuk bernilai positif.
c. *False Negatives* (FN): Jumlah data yang memiliki nilai positif dan diharapkan untuk bernilai negatif.
d. *True Negatives* (TN): Jumlah data yang memiliki nilai negatif dan diharapkan untuk bernilai negative

Fungsi utama *confusion matrix* sendiri yaitu untuk mewakili prediksi dan keadaan aktual (aktual) dari data yang dihasilkan oleh algoritma *machine learning* dengan menentukan nilai akurasi, presisi, recall, dan F1-score. Keempat pengukuran ini sangat berguna untuk mengukur kinerja *classifier* atau algoritma *machine learning* yang digunakan untuk membuat sebuah prediksi. Rumus untuk keempat pengukuran ini ditunjukkan pada tabel di bawah ini, berikut:

Tabel 3 Pengukuran Nilai Akurasi, Presisi, Recall, dan F1-Score

No.	Pengukuran	Rumus
1.	Akurasi	$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$
2.	Presisi	$Precision = \frac{TP}{TP + FP}$
3.	Recall	$Recall = \frac{TP}{TP + FN}$
4.	F1-Score	$F1 - Score = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision}$

3. HASIL DAN PEMBAHASAN

Dalampenelitian ini bertujuan untuk membuat sebuah model dengan menerapkan algoritma KNN, sehingga data sudah ada dan telah diolah dapat memprediksi data baru dan hasil kinerja algoritma dapat diperoleh hasil akurasi terbaik.

3.1 Proses Data Preprocessing

Tahap data *preprocessing* mencakup pengolahan data *missing value*, data yang tidak normal, *error*, tidak lengkap atau data dengan nilai berbeda dengan data lainnya/tidak konsisten sehingga adanya ketidaksesuaian data. Untuk *data preprocessing* sendiri telah dilakukan tiga tahap yaitu sebagai berikut:

a. Data Cleansing

aluminium	ammonia	arsenic	barium	cadmium	chloramin	chromium	copper
1,65	#NUM!	0,04	2,85	0,007	0,35	0,83	0,17
2,32	#NUM!	0,01	3,31	0,002	5,28	0,68	0,66
1,01	14,02	0,04	0,58	0,008	4,24	0,53	0,02
1,36	11,33	0,04	2,96	0,001	7,23	0,03	1,66
0,92	24,33	0,03	0,2	0,006	2,67	0,69	0,57
0,94	14,47	0,03	2,88	0,003	0,8	0,43	1,38
2,36	5,6	0,01	1,35	0,004	1,28	0,62	1,88
3,93	19,87	0,04	0,66	0,001	6,22	0,1	1,86

Gambar 2 *Missing Value* Pada Dataset

Pada dataset penelitian ini terdapat *missing value* yang cukup sedikit namun tersebar ke setiap fitur atau atribut yang hampir semua atribut mempunyai 3 *missing value* kecuali atribut *aluminium*, hal ini berdampak terhadap performa model yang akan digunakan nanti yang dapat menghasilkan akurasi yang berbeda. Untuk contoh data dengan *missing value* pada penelitian ini yaitu terdapat nilai = (#NUM!) atau tidak diketahui, untuk ini ini solusi yang dilakukan adalah dengan menghapus data hal ini dikarenakan data yang memiliki *missing value*. Dengan menggunakan salah satu *tools* pengolahan *data science* yaitu *jupyther notebook*. *Jupyther notebook* merupakan sebuah *software* interaktif dalam pemograman yang sering digunakan untuk baik itu teks/narasi, kode hidup, persamaan, tampilan hasil, gambar statis, dan visualisasi grafis, dalam satu file interaktif [9]. permasalahan *missing value* pada dataset dapat selesai, berikut ini perintah *python* yang digunakan untuk menghilangkan *missing value* pada dataset:

```
In [146]: df.isna().sum() #melihat data yang missing / NaN

Out[146]: aluminium      0
          ammonia        3
          arsenic        0
          barium         0
          cadmium        0
          chloramine     0
          chromium       0
          copper         0
          flouride       0
```

Gambar 3 Dataset Bersih Dari *Missing Value*

b. Data Selection

Pada tahap seleksi data ini data yang dipilih hanya data yang memiliki atribut penting yang berguna dalam kategori prediksi kualitas air yang baik untuk dikonsumsi tersebut. Pada penelitian ini semua atribut yang ada berpengaruh terhadap output yang dikeluarkan, Adapun atribut yang digunakan yang berjumlah 21 atribut dan sudah termasuk kelas/label.

c. Data Balancing

Pada tahap balancing ini, dataset yang tadi berjumlah 7996 *record* setelah melalui proses data *cleansing*, masih memiliki jumlah data yang tidak seimbang antara label nilai 0(*notsafe*) sebesar 7084 *record data* dibandingkan dengan jumlah data berlabel nilai 1(*safe*) yang hanya 912 record data. Maka melakukan balancing data dengan menggunakan salah satu *library* dari *imblearn* yang mana *balancing* dikerjakan pada *tools jupyther notebook*. Dengan smenggunakan teknik *RandomUnderSampler* dengan tingkat akurasi yang cukup tinggi serta tetap menjaga dataset agar model yang dibuat tidak *overfitting*, dengan begitu data yang dihasilkan sebanyak 1824 record data dengan label nilai 0(*notsafe*) dan 1(*safe*) memiliki *record data* yang sama banyaknya yaitu 912 record data.

3.2 Proses Split Data

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X_res,y_res,test_size=0.10, random_state=42)
```

Gambar 4 *Split Data* Menggunakan *Jupyther Notebook*

Berdasarkan Gambar 4 diatas setelah proses dari data *preprocessing* selesai, selanjutnya data akan dibagi menjadi dua yaitu data latih dan data uji untuk nantinya akan dipakai dalam pemodelan algoritma *K-Nearest Neighbor* dalam melakukan prediksi, membagi data merupakan salah satu metode yang berguna untuk mengetahui apakah model algoritma *K-Nearest Neighbor* dapat memprediksi data test dengan benar dengan mengolah data *training* dan menggunakannya di data test. Berdasarkan gambar diatas *split* data dilakukan dengan pembagian data sebesar 90% data *training* dan 10% data *testing* dengan jumlah data latih 7196 record data dan data tes sebesar 800 record data yang akan digunakan dalam penelitian ini. Untuk mengetahui performa dari setiap percobaan *split* data dapat dilihat pada evaluasi model.

3.3 Penerapan Algoritma K-Nearest Neighbor

Setelah melalui proses pembagian data (split data) sebelumnya, maka saatnya menerapkan model Algoritma *K-Nearest Neighbor*. Pada tahap ini sebelum mengetahui cara kerja dari algoritma *K-Nearest Neighbor* perlu dilakukan perhitungan manual berdasarkan urutan tahapan dari algoritma KNN dan dilanjutkan ke penerapan algoritma *K-Nearest Neighbor* akan dilakukan dengan menggunakan *jupyther notebook* untuk nanti akan dibuat sebagai model.

a. Perhitungan Manual

Perhitungan manual yang akan dilakukan akan menggunakan data berjumlah 10 data latih yang sudah dilakukan pembagian data sebelumnya, pengambilan 10 data latih ditujukan untuk menghitung jarak dari data/objek baru yang akan diprediksi, untuk lebih detail 10 data latih yang akan diambil sebagai berikut:

Tabel Horizontal	1	2	3	4	5	6	7	8	9	10
Nama	A									
aluminium	0,6100	3,4700	1,8200	0,0700	0,1700	0,0500	2,1300	1,0200	2,9100	0,7800
ammonia	2,4100	15,8400	6,8100	21,8400	13,5500	29,6800	25,1100	23,9000	19,5900	22,2000
arsenic	0,0300	0,0200	0,0100	0,0300	0,0300	0,0300	0,7600	0,7200	0,0400	0,0010
barium	0,5900	0,0600	0,8500	3,4500	3,2800	2,8600	2,6000	2,9000	1,4800	2,2500
cadmium	0,0020	0,0010	0,0060	0,0080	0,0010	0,0060	0,1200	0,1100	0,0080	0,0030
chloramine	1,9400	5,2900	2,5500	6,1000	2,3800	4,5400	1,3000	2,3200	6,6200	4,7400
chromium	0,7700	0,4700	0,2500	0,2300	0,1700	0,7200	0,5200	0,1000	0,0600	0,1500
copper	1,5400	1,0800	1,0900	0,7600	0,2500	0,1400	0,4000	1,7600	1,6900	1,3300
flouride	0,6200	1,4300	1,3500	1,1600	1,2500	1,4200	0,5900	0,7400	0,9400	0,6300
bacteria	0,2300	0,8900	0,1600	0,7900	0,7000	1,0000	0,0000	0,0000	0,0000	0,0000
viruses	0,0010	0,8900	0,0020	0,7900	0,7000	1,0000	0,6100	0,0020	0,0000	0,0000
lead	0,0170	0,0800	0,0310	0,0490	0,1290	0,0820	0,0400	0,0480	0,1980	0,1320
nitrates	1,9900	1,9100	16,9900	16,8800	15,7200	11,3000	18,6600	11,6100	10,6700	11,0400
nitrites	1,0800	1,2000	1,7000	1,4000	1,8100	1,4600	1,8800	2,0400	1,6200	1,8500
mercury	0,0070	0,0080	0,0070	0,0060	0,0030	0,0060	0,0100	0,0040	0,0080	0,0090
perchlorate	11,1600	0,1800	44,7600	59,0300	10,8800	33,0900	51,7800	15,8700	50,5500	30,9100
radium	0,9800	6,8900	1,1500	4,2900	5,4100	4,9600	0,0500	0,1000	7,9200	6,6800
selenium	0,0100	0,0600	0,0800	0,0700	0,0300	0,0000	0,0300	0,0700	0,0800	0,1000
silver	0,4700	0,1200	0,2600	0,0800	0,1100	0,1600	0,0500	0,0900	0,0900	0,1100
uranium	0,0300	0,0800	0,0800	0,0000	0,0700	0,0500	0,0200	0,0100	0,0300	0,0500
kelas										
Is_safe	1	1	1	0	0	0	1	1	0	0

Gambar 5 10 Data latih Pilihan

Lalu akan dilakukan prediksi ke data baru yang belum diketahui kelasnya, penentuan kelas bedasarkan jarak terdekat terhadap data latih diatas dan bobot dari kelas yang sama dengan untuk detail datanya sebagai berikut:

Tabel Horizontal	1
Nama	B
aluminium	2,17
ammonia	34,4
arsenic	0,05
barium	0,1
cadmium	0,002
chloramine	4,01
chromium	0,71
copper	1,4
flouride	1,6
bacteria	0
viruses	1
lead	0,25
nitrates	11
nitrites	1,55
mercury	0,007
perchlorate	57,5
radium	5,5
selenium	0,6
silver	0,1
uranium	0,4
kelas	
Is_safe	?

Gambar 6 Data Baru

Untuk perhitungan ini menggunakan langkah-langkah dari algoritma *K-Nearest Neighbor* dengan menggunakan metode perhitungan jarak *euclidean distance*, yaitu sebagai berikut:

1. Untuk langkah pertama yaitu menentukan jumlah tetangga atau parameter nilai *k*, dalam perhitungan manual ini penulis menggunakan nilai k = 3 berdasarkan nilai performa yang terbaik yang telah di lakukan evaluasi model dari algoritma *K-Nearest Neighbor*.
2. Langkah kedua ini, yaitu melakukan perhitungan jarak dengan metode *euclidean distance*, perhitungan ini dilakukan dengan memasukan data latih (a) dan data baru (b) kedalam rumus (1), untuk contoh perhitungan bisa dilihat dibawah sebagai berikut:

$$\mathrm{d1} = \sqrt{a_1 - b_1} + \ldots + (a_n - b_n)$$

$$d1 = \sqrt{(0{,}61 - 2{,}17)^2} + (2{,}41 - 34{,}4)^2 + (0{,}03 - 0{,}05)^2 + (0{,}59 - 0{,}1)^2 + (0{,}002 - 0{,}002)^2 + (1{,}94 - 4{,}01)^2 + (0{,}77 - 0{,}71)^2 + (1{,}54 - 1{,}4)^2 + (0{,}62 - 1{,}6)^2 + (0{,}23 - 0)^2 + (0{,}001 - 1)^2 + (0{,}017 - 0{,}25)^2 + (1{,}99 - 11)^2 + (1{,}08 - 1{,}55)^2 + (0{,}007 - 0{,}007)^2 + (11{,}16 - 57{,}5)^2 + (0{,}98 - 5{,}5)^2 + (0{,}01 - 0{,}6)^2 + (0{,}47 - 0{,}1)^2 + (0{,}03 - 0{,}4)^2$$

$$d1 = \sqrt{2{,}4336 + 1023{,}3601 + 0{,}0004 + 0{,}2401 + 0 + 4{,}2849 + 0{,}0036 + 0{,}0196 + 0{,}9604 + 0{,}0529 + 0{,}998001 + 0{,}054289 + 81{,}1801 + 0{,}2209 + 0 + 2147{,}3956 + 20{,}4304 + 0{,}3481 + 0{,}1396 + 0{,}1396}$$

$$d1 = \sqrt{3282{,}26219}$$

$$d1 = 57{,}29098$$

…. Hingga Perhitungan ke d10

3. Terakhir menentukan kelas dari tetangga terdekat berdasarkan nilai *k* yang telah ditentukan sebelumnya, dengan nilai *k* yang cukup besar yaitu 3 maka perlu memperhatikan juga *weighted vote* atau pemilihan bobot terbanyak untuk mendukung suatu kelas yang sama. Untuk detail penentuan dapat dilihat dibawah, sebagai berikut:

Jarak	*k* = 3	is_safe
14,76956	✓	0
14,97823	✓	1
16,88862	✓	0
25,18023	-	0
29,44636	-	0
31,35279	-	1
43,45422	-	1
51,47850	-	0
57,29098	-	1
60,98759	-	1

Gambar 7 Hasil Prediksi Perhitungan Manual

Maka dapat disimpulkan berdasarkan hasil diatas, data baru atau objek baru yang diuji memiliki kelas *is_safe* = 0(*notsafe)* dikarenakan dari ke 3 tetangga terdekat yaitu memiliki kelas *is_safe* = 0(*notsafe)* dan bobot yang lebih banyak untuk kelas *is_safe* = 0(*notsafe)*.

b. Menggunakan *Jupyther Notebook*

```
from sklearn.neighbors import KNeighborsClassifier #Melakukan import algoritma KNN
from sklearn.metrics import accuracy_score #Import Perhitungan Akurasi dari sklearn
from sklearn.metrics import confusion_matrix #Import Confusion Matrix dari sklearn
from sklearn.metrics import classification_report #Import Laporan Klasifikasi dari sklearn
```

Gambar 8 *Import Library* Yang Digunakan Untuk Menerapkan Algoritma KNN

```
KNN = KNeighborsClassifier(n_neighbors=3)
model = KNN.fit(X_res, y_res)
```

Gambar 9 Penerapan Algoritma *K-Nearest Neighbor*

Berdasarkan kedua gambar diatas nilai *k* yang digunakan yaitu 3, nilai parameter pada kode program *n_neighbors* ini dapat diubah sesuai dengan jumlah parameter yang akan digunakan atau diuji coba. Untuk mengetahui performa dari setiap nilai *k* sehingga dapat ditentukan nilai *k* = 3 dapat dilihat pada evaluasi model.

3.4 Evaluasi Model

Selanjutnya untuk mengetahui kinerja model yang diterapkan pada penelitian ini, maka hal yang dilakukan adalah mengevaluasi kemampuan prediksi menggunakan data pelatihan untuk menghasilkan nilai akurasi dari model yang telah digunakan. Pada penelitian ini dengan menggunakan *jupyther notebook* untuk menampilkan hasil evaluasi dengan menampilkan hasil akurasi, presisi, dan recall serta f1-score.

Untuk tabel *confusion matrix* yang akan digunakan untuk evaluasi model sendiri menghasilkan perbandingan sebagai berikut:

Tabel 4 Tabel *Confusion Matrix*

		Nilai Aktual	
Nilai Prediksi		Positif	Negatif
	Positif	71	12
	Negatif	15	85

Berdasarkan perbandingan diatas maka didapatkan nilai akurasi, presisi, recall serta f1-score yang dapat dilihat pada gambar dibawah dengan menggunakan *tools jupyther*:

```
print(classification_report(y_test, hasil_prediksi))
```

```
              precision    recall  f1-score   support

           0       0.85      0.88      0.86        97
           1       0.86      0.83      0.84        86

    accuracy                           0.85       183
   macro avg       0.85      0.85      0.85       183
weighted avg       0.85      0.85      0.85       183
```

Gambar 10 Hasil Performa Model Menggunakan *Confusion Matrix*

Akurasi ini juga dipengaruhi oleh beberapa hal seperti penentuan nilai *k* dan seberapa besar split data yang digunakan, untuk percobaan pertama yaitu penentuan nilai *k* yang dapat mempengaruhi performa nilai akurasi. Berikut adalah presentase pengaruh nilai *k* dengan menggunakan *jupyther*, sebegai berikut:

Tabel 5 Perbandingan Pengaruh Performa Model (*Split Data*)

No.	Pembagian Data	Hasil Akurasi
1	60% Data Training - 40% Data Testing	83.35%
2	70% Data Training - 30% Data Testing	83.47%
3	80% Data Training - 20% Data Testing	83.83%
4	90% Data Training - 10% Data Testing	85.24%

Selanjutnya pengaruh performa model juga dapat dipengaruhi oleh nilai *k* terdekat yang digunakan, untuk itu dapat nilai presentase yang dihasilkan nilai *k* dari setiap percobaan *k* = 3 hingga 7 dapat dilihat pada tabel dibawah:

Tabel 6 Perbandingan Pengaruh Performa Terhadap *Nilai k*

No.	Nilai *k*	Hasil Akurasi
1	*k* = 3	85.24%
2	*k* = 4	80.32%
3	*k* = 5	79.78%
4	*k* = 6	81.96%
5	*k* = 7	81.42%

Berdasarkan perhitungan *confusion matrix* diatas menunjukan banyak aspek yang harus diperhatikan demi mendapatkan performa akurasi yang baik untuk melakukan prediksi dan berdasarkan hasil akurasi diperoleh sebesar 85.24% dapat dikelompok kestatus sangat baik dengan nilai *k* = 3 dan *split* data sebesar 90% data latih dan 10% data uji, maka dengan penerapan model algoritma *K-Nearest Neighbor* mampu memprediksi kualitas air yang baik dikonsumsi atau tidaknya oleh masyarakat dengan sangat baik.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil perhitungan performa algoritma KNN dalam hal akurasi yang dilakukan diatas dapat disimpulkan, model algoritma *K-Nearest Neighbor* yang diterapkan mampu dalam memprediksi kualitas air dengan sangat baik berdasarkan nilai akurasi yang didapatkan yaitu sebesar 85,24%, dengan akurasi yang cukup tinggi tersebut dan hanya mengandalkan *record data* yang memiliki parameter kandungan kimia dan biologis hasil prediksi masih dapat dilakukan mengetahui kualitas air yang dapat dikonsumsi atau tidaknya sudah termasuk kedalam standar yang cukup sangat baik untuk melakukan prediksi yang meminimalisir kesalahan prediksi pada data baru nanti.

DAFTAR PUSTAKA

[1] Saraswati, Rieke, 2020, Ini 6 Penyakit Akibat Pencemaran Air Yang Perlu Diwaspadai, https://www.sehatq.com/artikel/penyakit-akibat-pencemaran-air-yang-perlu-diwaspadai,diakses tgl 2 Maret 2022.

[2] Nurmahaludin, dan Cahyano G.R., 2019, Klasifikasi Kualitas Air PDAM Menggunakan Algoritma KNN Dan K-Means, *Prosiding SNRT (Seminar Nasional Riset Terapan*), Politeknik Negeri Banjarmasin, November 7.

[3] Armono, R.A., Saptomo, W.L.Y., dan Harsadi, P. 2018. Implementasi Algoritma K-Nearest Neighbor Untuk Identifikasi Kualitas Air (Studi Kasus: PDAM Kota Surakarta). *Jurnal TIKomSIN*, No.1, Vol.6, 2338-4018.

[4] Vidiastanta, I.G., Hidayat, N., dan Dewi, R.K., 2020, Komparasi Metode K-Nearest Neighbor (K-NN) Dengan Support Vector Machine (SVM) Untuk Klasifikasi Status Kualitas Air, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, No. 1, Vol .4, 312-319.

[5] Intermedia, Beon, 2020, Data Mining: Definisi, Fungsi, Metode dan Penerapannya, https://ww.jagoanhosting.com/blog/apa-itu-data-mining/, diakses tgl 2 maret 2022.

[6] Henny, Chandra, 2021, Data Preprocessing adalah/ pengertian, Tahapan Kerja dan Manfaatnya,https://id.linkedin.com/pulse/data-preprocessing-adalah-pengertian-tahapan-kerja-dan-chandra-henny, diakses tgl 2 maret 2022.

[7] Advernesia, 2017, Pengertian dan Cara kerja Algoritma K-Nearest Neighbor (KNN), https://www.advernesia.com/blog/data-science/pengertian-dan-cara-kerja-algoritma-k-nearest-neighbours-knn/, diakses tgl 2 maret 2022.

[8] Nugroho, Kuncahyo Setyo, 2019, Confusion Matrix untuk Evaluasi Model pada Supervised Learning,https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f, diakses tgl 2 maret 2022.
[9] Prijono, Benny, 2019, Pengenalan dan panduan Jupyther Notebook untuk Pemula, https://indoml.com/2019/09/29/pengenalan-dan-panduan-jupyter-notebook-untuk-pemula/, diakses tgl 2 maret 2022.