

PREDIKSI KUALITAS UDARA MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR

Adinda Amalia¹⁾, Ati Zaidiah²⁾, Ika Nurlaili Isnainiyah³⁾

^{1, 2, 3)}Program Studi S1 Sistem Informasi, Fakultas Ilmu Komputer, UPN Veteran Jakarta

Jl.RS. Fatmawati, Pondok Labu, Jakarta

e-mail: adindaa@upnvj.ac.id¹⁾, atizaidiah@upnvj.ac.id²⁾, nurlailika@upnvj.ac.id³⁾

ABSTRAK

Dalam kehidupan sehari-hari, udara digunakan untuk bernafas oleh makhluk hidup. Udara yang bersih mengandung banyak manfaat bagi kehidupan. Namun, pada kenyataannya udara yang ada di alam tidak selalu dalam keadaan bersih, sehingga dapat menyebabkan penurunan kualitas udara. Kualitas udara seperti ini dapat memberikan dampak terhadap kesehatan manusia serta lingkungan sekitarnya. Kualitas udara di DKI Jakarta dapat diketahui melalui Indeks Standar Pencemar Udara (ISPU). Penelitian ini mempunyai tujuan memprediksi kualitas udara yang ada di DKI Jakarta berdasarkan data ISPU. Prediksi dilakukan dengan menggunakan teknik data mining dengan metode klasifikasi. Algoritma yang berfungsi dalam melakukan prediksi yaitu K-Nearest Neighbor (KNN), dimana algoritma ini adalah algoritma yang mengklasifikasikan kelas objek baru dengan didasarkan pada tetangga terdekatnya. Data yang digunakan pada penelitian berjumlah 450 data, kemudian data tersebut dibagi 2 yakni data uji dan data latih. Penelitian ini juga melakukan evaluasi model algoritma yang meliputi nilai akurasi, presisi, recall, dan f-measure untuk setiap nilai K yang diujikan. Pengukuran ini bertujuan untuk mengetahui parameter yang optimal pada dataset yang digunakan. Adapun hasil yang diperoleh dari pengujian nilai K = 3 sampai K = 9, didapatkan bahwa nilai K = 7 mempunyai performa terbaik dengan akurasi tertinggi sebanyak 96%, presisi 92%, recall 95%, dan f-measure 93%.

Kata Kunci: data mining, knn, kualitas udara, prediksi

ABSTRACT

In everyday life, air is used for breathing by living things. Clean air contains many benefits for life. However, in reality the air in nature is not always clean so that it can cause a decrease in air quality. Air quality like this can have an impact on human health and the surrounding environment. Air quality in DKI Jakarta can be known through the Air Pollutant Standard Index (ISPU). This study aims to predict the air quality in DKI Jakarta based on ISPU data. Prediction is done using data mining techniques with the classification method Algorithm that functions in making predictions, namely K-Nearest Neighbor (KNN), where this algorithm is an algorithm that classifies new object classes based on their closest neighbors. The data used in the study amounted to 450 data, then the data was divided into 2, test data and lattice data. This study also evaluates the algorithm model which includes the values of accuracy, precision, recall and f-measure for each tested K value. This measurement aims to determine the optimal parameters in the dataset used. As for the data obtained from testing the values of K = 3 to K = 9, it was found that the value of K7 has the best performance with the highest accuracy as much as 96%, precision 92%, recall 95%, and f-measure 93%.

Keywords: air quality, data mining, knn, prediction

I. PENDAHULUAN

Udara merupakan hal yang sangat penting bagi kehidupan. Udara bersih merupakan udara yang tidak bercampur dengan zat-zat atau gas-gas yang merugikan seperti debu, karbon dioksida (CO₂), nitrogen dioksida (NO₂), dan gas lainnya. Pada kenyataannya, udara yang terdapat di alam tidak selalu bersih. Hal ini dapat menimbulkan penurunan kualitas udara.

Penurunan kualitas udara dapat terjadi karena beberapa aktivitas yang dilakukan manusia seperti asap rokok, kegiatan industri, transportasi, pembakaran lahan atau hutan, dan lain-lain [1]. Faktor-faktor ini dapat menyebabkan terjadinya pencemaran udara. Pencemaran udara dapat menimbulkan penyakit bagi manusia seperti sesak napas, kanker paru-paru, penyakit jantung, infeksi saluran pernafasan, hingga kematian [2]. Tidak hanya itu, pencemaran udara juga dapat mempengaruhi pertumbuhan tanaman, pemanasan global, perubahan ekosistem, naiknya permukaan air laut, hingga penipisan lapisan ozon. Hal ini sangat berdampak buruk untuk seluruh aspek kehidupan yang ada di bumi.

Menurut *Air Quality Live Indeks* (AQLI), pada bulan April 2021, DKI Jakarta merupakan kota ke-6 (enam) dengan kualitas udara terburuk. Hal ini ditandai dengan nilai indeks AQI Jakarta memiliki nilai 156 dengan kategori tidak sehat. Polutan utama yang menyebabkan penurunan kualitas udara yaitu PM_{2.5}, di mana polutan ini jumlahnya tidak boleh melebihi 10 mikron saat berada di udara. Di DKI Jakarta, polutan ini tercatat mencapai 57 mikron per

kubik yang menandakan bahwa kualitas udara di DKI Jakarta sangat buruk [3].

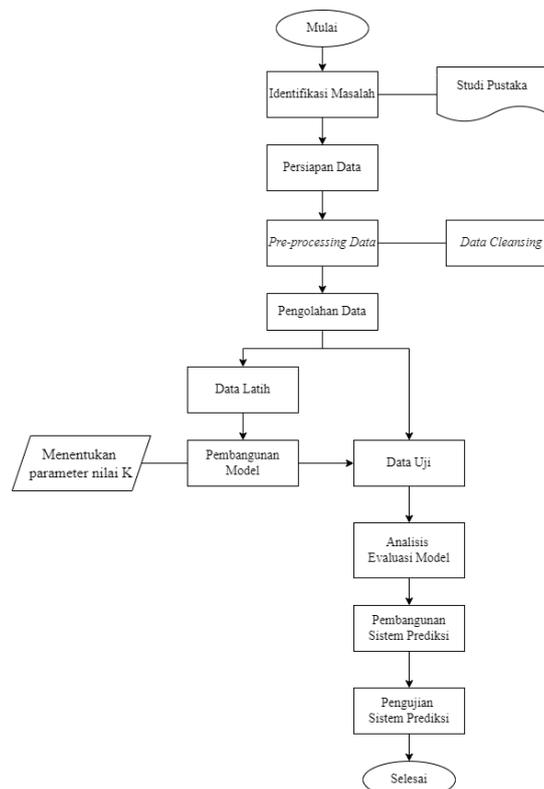
Di wilayah DKI Jakarta, kualitas udara dapat diketahui melalui pengukuran kualitas udara, pengukuran tersebut dilaksanakan oleh Dinas Lingkungan Hidup Provinsi DKI Jakarta. Pengukuran ini didokumentasikan dalam suatu indeks yang disebut Indeks Standar Pencemar Udara (ISPU). ISPU merupakan laporan dari hasil pemantauan kualitas udara yang memberikan keterangan mengenai kualitas udara yang bersih ataupun tercemar serta pengaruhnya terhadap kesehatan [4]. ISPU berisikan informasi mengenai kualitas udara yang disajikan dalam bentuk angka dan tidak memiliki satuan. ISPU memberikan informasi bagaimana keadaan kualitas udara pada lokasi tertentu [5]. Indeks ini dapat digunakan untuk memberikan informasi mengenai kualitas udara di DKI Jakarta, terutama dapat membantu dalam membuat langkah penanggulangan yang dapat mengurangi pencemaran udara di DKI Jakarta.

Data mining berfungsi untuk mendapatkan informasi yang dibutuhkan dengan cepat [6]. Salah satu metode *data mining* yang memiliki fungsi dalam memprediksi adalah klasifikasi [7]. *K-Nearest Neighbor* (KNN) yaitu suatu algoritma yang mempunyai fungsi dalam memprediksi [8]. Algoritma ini termasuk jenis algoritma *supervised learning*, di mana algoritma ini mengklasifikasikan objek data baru menurut data-data yang mempunyai jarak terdekat. Berdasarkan penelitian Susanto, dkk (2019) yang berjudul “Analisis Perbandingan Algoritma *Naïve Bayes* dan KNN Untuk Klasifikasi Multi Dataset” menghasilkan bahwa algoritma *K-Nearest Neighbor* atau KNN didapatkan nilai akurasi sebanyak 93,17% dan *Naïve Bayes* sebesar 78,38% [9]. Pada penelitian lainnya yang dilakukan oleh Wisdayani, dkk pada tahun yang sama dengan judul penelitian “Perbandingan Algoritma *K-Nearest Neighbor* dan *Naïve Bayes* Untuk Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas di Kabupaten Pati Jawa Tengah” menghasilkan bahwa algoritma *K-Nearest Neighbor* juga lebih unggul dibandingkan *Naïve Bayes* dimana nilai akurasinya sebanyak 88,82% dan *Naïve Bayes* sebesar 86,32% [10].

Maka berdasarkan pendahuluan di atas, penelitian ini akan melakukan prediksi kualitas udara di DKI Jakarta dengan algoritma *K-Nearest Neighbor*. Adapun nilai parameter yang digunakan mencakup angka 3 hingga 9. Selain itu, penelitian ini juga akan membangun sistem prediksi yang menerapkan hasil dari penerapan algoritma untuk melakukan prediksi kualitas udara DKI Jakarta.

II. METODE PENELITIAN

Berikut ini *flowchart* alur penelitian yang tersaji pada gambar 1:



Gambar. 1. Alur Penelitian

A. Identifikasi Masalah

Penulis melakukan pengidentifikasian masalah yang terjadi yaitu permasalahan mengenai penurunan kualitas udara yang terjadi di DKI Jakarta dengan membaca beberapa artikel dan jurnal terkait.

B. Persiapan Data

Penulis mendapatkan dataset penelitian ini dari *website* Open Data Jakarta dengan laman URL <https://data.jakarta.go.id/>. Dataset yang digunakan memuat informasi kualitas udara DKI Jakarta pada bulan Januari hingga Maret 2021. Dataset ini memuat total sebanyak 450 *record* data dengan 10 (sepuluh) atribut dan 1 (satu) kelas. Dataset yang diperoleh dari *website* berupa *file* berekstensi *.csv*.

C. Pre-processing Data

Tahap *pre-processing* yang akan dilakukan yaitu pembersihan data atau *data cleansing*. Pembersihan data yang akan dilakukan yaitu dengan melakukan pencarian data yang masih kosong, menghapus baris atau kolom yang kosong, hingga melakukan pengisian data pada kolom yang masih kosong. Selain itu, akan dilakukan pemilihan fitur yang akan digunakan untuk menerapkan algoritma. Pemilihan fitur ini dapat mengurangi kompleksitas atribut yang akan dipelajari oleh algoritma.

D. Pengolahan Data

Pengolahan data dilakukan dengan membagi dataset yang telah melewati tahap sebelumnya menjadi 2 (dua) yakni data uji serta data latih lalu diterapkan algoritma *K-Nearest Neighbor*. Cara kerja algoritma tersebut yakni melakukan klasifikasi terhadap objek baru dengan melakukan perhitungan jarak terdekat objek tersebut terhadap data-data yang sudah ada. Jarak yang terdapat diantara kedua objek data tersebut dapat dihitung dengan menggunakan perhitungan *euclidean distance*. Berikut ini rumus dari perhitungan *euclidean distance* [11]:

$$\text{euclidean distance (d)} = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2} \quad (1)$$

Keterangan :

a = a₁, a₂, ... a_n hingga nilai ke n

b = b₁, b₂, ... b_n hingga nilai ke n

Penggunaan dari algoritma ini memiliki tujuan untuk dapat mengklasifikasikan kelas objek data baru yang sesuai dengan atribut dan sampel data yang terdapat di data latih. Data yang digunakan untuk melakukan klasifikasi dengan menggunakan algoritma ini sangat dipengaruhi oleh nilai K yang digunakan. Untuk mendapatkan nilai K yang optimal, maka dapat dilakukan dengan melakukan perbandingan hasil dari setiap nilai K yang digunakan.

E. Analisis Evaluasi Model

Metode analisis evaluasi model yang digunakan yaitu menggunakan *confusion matrix*. Analisis performa ini melakukan perbandingan hasil akurasi, presisi, *recall*, dan *f-measure* menurut parameter nilai K yang telah digunakan. *Confusion matrix* merupakan salah satu metode evaluasi model yang dapat merepresentasikan informasi dari perbandingan hasil prediksi yang dilakukan dan kondisi yang sebenarnya. Perbandingan *confusion matrix* digambarkan dalam bentuk matriks yang memiliki 4 (empat) kombinasi nilai prediksi dengan nilai aktual seperti pada tabel 1 berikut ini [12]:

TABEL I
EVALUASI MODEL *CONFUSION MATRIX*

Kelas Prediksi	Kelas Sebenarnya	
	+	-
+	<i>True Positives</i>	<i>False Positives</i>
-	<i>False Negatives</i>	<i>True Negatives</i>

Dalam *confusion matrix*, terdapat nilai-nilai yang terdiri atas akurasi, presisi, *recall*, dan *f-measure* [13].

1) *Akurasi*

Akurasi adalah tingkatan dekatnya nilai prediksi serta nilai aktual atau nilai sesungguhnya.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

2) *Presisi*

Presisi merupakan dipilihnya rasio item relevan berdasarkan seluruh item terpilih.

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (3)$$

3) *Recall*

Recall merupakan dipilihnya rasioitem relevan berdasarkan total item relevan yang ada.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

4) *F-Measure*

F-Measure atau yang juga disebut *F1-Score* merupakan *harmonic mean* dari presisi serta *recall*.

$$F1\text{-Score} = 2 \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (5)$$

F. *Pembangunan Sistem Prediksi*

Pembuatan sistem prediksi dibangun menggunakan *framework Flask*. *Framework* ini mendukung bahasa pemrograman *Python* sehingga dapat meng-*import* model algoritma ke dalam sistem. Model algoritma yang akan digunakan dilakukan penyimpanan model ke dalam format *pickle* (.*pkl*). Kemudian dibangun sistem prediksi dengan meng-*import* file *pickle* (.*pkl*) tersebut ke dalam sistem.

G. *Pengujian Sistem Prediksi*

Pengujian sistem menggunakan metode *blackbox testing*. Tujuan dari pengujian adalah mengetahui apakah sistem dapat bekerja sesuai dengan fungsinya [14]. Teknik *blackbox testing* yang digunakan yaitu *equivalence partitioning*. Teknik ini melakukan pembagian data *input* menjadi beberapa partisi data.

III. HASIL DAN PEMBAHASAN

A. *Persiapan Data*

Data ISPU pada penelitian ini disusun oleh Dinas Lingkungan Hidup Provinsi DKI Jakarta. Secara keseluruhan, dataset ini memuat 450 *record* data dengan 10 atribut dan 1 kelas. Atribut-atribut yang terdapat dalam dataset ini antara lain tanggal, stasiun, pm10, pm25, so2, co, o3, no2, max, critical, dan kategori seperti pada tabel 2 berikut:

TABEL II
DATA CoNTOH ISPU JANUARI-MARET 2021

tanggal	stasiun	pm10	pm25	so2	co	o3	no2	max	critical	kategori
1/1/2021	DKI1 (Bundaran HI)	38	53	29	6	31	13	53	PM25	SEDANG
1/2/2021	DKI1 (Bundaran HI)	27	46	27	7	47	7	47	O3	BAIK
1/3/2021	DKI1 (Bundaran HI)	44	58	25	7	40	13	58	PM25	SEDANG
1/4/2021	DKI1 (Bundaran HI)	30	48	24	4	32	7	48	PM25	BAIK
1/5/2021	DKI1 (Bundaran HI)	38	53	24	6	31	9	53	PM25	SEDANG
1/6/2021	DKI1 (Bundaran HI)	41	58	23	13	46	13	58	PM25	SEDANG
1/7/2021	DKI1 (Bundaran HI)	35	47	22	6	39	10	47	PM25	BAIK
1/8/2021	DKI1 (Bundaran HI)	37	54	26	16	17	10	54	PM25	SEDANG
1/9/2021	DKI1 (Bundaran HI)	47	61	16	27	22	12	61	PM25	SEDANG

1/10/2021	DKI1 (Bundaran HI)	23	25	16	11	33	8	33	O3	BAIK
dst.

B. Tahap Pre-processing Data

Dataset yang digunakan dalam penelitian ini memiliki beberapa kolom *missing value*, dicatat dengan tanda strip (-), dan baris data yang tidak memiliki data pengukuran. Dalam penelitian ini, terdapat beberapa atribut yang dihilangkan antara lain tanggal, stasiun, dan critical. Fitur-fitur ini tidak diperlukan karena tidak memberikan pengaruh terhadap pengolahan data selanjutnya. Dengan kata lain, penentuan kualitas udara ditentukan berdasarkan nilai ukuran dari parameter-parameter seperti pm10, pm25, so2, co, o3, no2, dan max. Berikut ini contoh data yang perlu melewati tahap *pre-processing* pada tabel 3:

TABEL III
CONTOH DATASET *MISSING VALUE*

tanggal	stasiun	pm10	pm25	so2	co	o3	no2	max	critical	kategori
1/21/2021	DKI4 (Lubang Buaya)	63		25	21	38	7	63	PM10	SEDANG
1/22/2021	DKI4 (Lubang Buaya)	52		26	32	54	8	54	CO	SEDANG
1/23/2021	DKI4 (Lubang Buaya)	89		27	47	61	11	89	PM10	SEDANG
1/24/2021	DKI4 (Lubang Buaya)	---		---	---	---	---	0		TIDAK ADA DATA
1/25/2021	DKI4 (Lubang Buaya)	--		---	---	---	---	0		TIDAK ADA DATA
1/26/2021	DKI4 (Lubang Buaya)	57		26	34	39	7	57	PM10	SEDANG
1/27/2021	DKI4 (Lubang Buaya)	18		25	7	19	4	25	PM25	BAIK
1/28/2021	DKI4 (Lubang Buaya)	23		25	11	20	5	25	PM25	BAIK

Pembersihan data ini dilakukan dengan 2 (dua) cara, yaitu menghapus secara manual pada file *.csv* dataset dan menggunakan *Python*. Data yang diisi dengan tanda strip (-) dan baris yang tidak memiliki data pengukuran dihapus secara manual dengan menggunakan *Microsoft Excel*. Penghapusan atribut tanggal, stasiun, dan critical juga dilakukan pada tahap ini. Setelah dihapus, selanjutnya dilakukan pengisian *missing value*. Pengisian nilai ini dilakukan dengan memasukkan nilai rata-rata (*mean*) setiap kolom untuk mengisi baris yang kosong dengan menggunakan perintah *numpy* seperti pada tabel 4 berikut:

TABEL IV
KODE *PYTHON* MENGISI *MISSING VALUE*

Keterangan	Source Code
Library	import pandas as pd import numpy as np
Membaca Data	df = pd.read_csv("dataset4.csv")
Cek <i>Missing Value</i>	df.isna().sum()
Menghitung Rata-Rata	mean = df["pm25"].mean() df["pm25"] = df["pm25"].replace(np.nan, mean) mean = df["pm10"].mean() df["pm10"] = df["pm10"].replace(np.nan, mean) mean = df["so2"].mean() df["so2"] = df["so2"].replace(np.nan, mean) mean = df["co"].mean() df["co"] = df["co"].replace(np.nan, mean) mean = df["o3"].mean() df["o3"] = df["o3"].replace(np.nan, mean) mean = df["no2"].mean() df["no2"] = df["no2"].replace(np.nan, mean)

Jumlah data yang sudah melewati proses pembersihan pada tahap sebelumnya diperoleh sebanyak 445 data. Selanjutnya dataset akan dibagi menjadi 2 bagian. Algoritma akan mempelajari pola-pola yang terdapat dalam data

latih. Sedangkan data uji akan digunakan untuk melakukan pengujian prediksi dengan model yang dibangun algoritma. Pembagian data yang akan digunakan untuk membangun model algoritma pada penelitian ini menggunakan rasio sebanyak 20% pada data uji dan 80% pada data latih. Adapun fitur yang dipakai berdasarkan dataset yang sudah dilakukan tahap *pre-processing* sebanyak 7 (tujuh) kolom dan 1 (satu) kolom sebagai kelas. Pembagian data dilakukan dengan menggunakan perintah *Python* seperti pada tabel 5 berikut:

TABEL V
KODE *PYTHON* PEMBAGIAN DATA

Keterangan	Source Code
Pembagian Data	<pre> from sklearn.model_selection import train_test_split features = df.iloc[:, :7] label = df.iloc[:, 7] x_train, x_test, y_train, y_test = train_test_split(features, label, test_size=0.20, random_state=42) </pre>

C. Perhitungan Manual Algoritma *K-Nearest Neighbor*

Perhitungan manual ini dilakukan untuk mengetahui bagaimana cara kerja dari algoritma *K-Nearest Neighbor*. Adapun data latih yang digunakan merupakan data contoh sebanyak 10 data yang telah di-*split* dengan menggunakan *Python* seperti tabel 6 berikut:

TABEL VI
DATA LATIH PERHITUNGAN MANUAL

pm10	pm25	so2	co	o3	no2	max	kategori
56	82	13	29	62	15	82	SEDANG
40	52	16	11	26	12	52	SEDANG
59	89	15	19	42	7	89	SEDANG
30	42	22	10	9	16	42	BAIK
43	62	40	11	25	9	62	SEDANG
35	64,45288	28	13	38	5	38	BAIK
43	41	23	6	17	11	41	BAIK
62	83	49	12	63	19	83	SEDANG
38	67	23	7	14	10	67	SEDANG
14	64,45288	28	9	38	3	38	BAIK

Selanjutnya akan dilakukan penentuan kelas dari 1 contoh data uji yang belum diketahui kelasnya. Berikut ini data uji yang akan digunakan untuk perhitungan manual yakni pada tabel 7:

TABEL VII
DATA UJI PERHITUNGAN MANUAL

pm10	pm25	so2	co	o3	no2	max	kategori
54	80	27	17	34	18	80	n/a

Berikut ini langkah-langkah perhitungan manual algoritma *K-Nearest Neighbor* dengan rumus (1):

1. Menentukan nilai *K* sebagai parameter banyaknya jumlah tetangga terdekat dengan objek yang akan diuji. Dalam perhitungan ini, penulis akan menggunakan nilai *K* minimal yang akan digunakan untuk evaluasi model pada penelitian ini, yaitu *K* = 3.
2. Menghitung jarak antara objek yang baru terhadap semua objek data yang ada di data latih. Perhitungan jarak dilakukan pada setiap baris data dengan memasukan nilai-nilai yang ada di data latih dan data uji ke dalam

rumus seperti berikut:

$$d_1 = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2}$$

$$d_1 = \sqrt{(56 - 54)^2 + (82 - 80)^2 + (13 - 27)^2 + (29 - 17)^2 + (62 - 34)^2 + (15 - 18)^2 + (82 - 80)^2}$$

$$d_1 = \sqrt{4 + 4 + 196 + 144 + 784 + 9 + 4}$$

$$d_1 = \sqrt{1145}$$

$$d_1 = 33,837848$$

Dan seterusnya hingga perhitungan ke-10.

- Melakukan pengurutan hasil perhitungan jarak dari yang paling terkecil ke terbesar. Pengurutan jarak ini ditunjukkan pada tabel 8.
- Menentukan tetangga terdekat berdasarkan nilai K yang sudah ditentukan sebelumnya. Penentuan ini dapat dilihat pada tabel 8 berikut:

TABEL VIII
PENENTUAN JARAK TERDEKAT

Urutan ke-	d	K=3	kategori
1	22,803508	YA	SEDANG
2	33,704599	YA	SEDANG
3	33,837848	YA	SEDANG
4	34,263683	TIDAK	SEDANG
5	37,523326	TIDAK	SEDANG
6	44,955533	TIDAK	SEDANG
7	50,682571	TIDAK	BAIK
8	60,315835	TIDAK	BAIK
9	62,543688	TIDAK	BAIK
10	64,552304	TIDAK	BAIK

- Menentukan kategori dari tetangga terdekat dari objek baru yang diuji. Berdasarkan tabel 8, maka dapat disimpulkan bahwa kelas dari data uji yang digunakan termasuk dalam kategori SEDANG karena 3 (tiga) tetangga terdekatnya memiliki kelas mayoritas yaitu SEDANG.

D. Penerapan Algoritma K-Nearest Neighbor

Algoritma ini membutuhkan parameter guna menentukan jumlah tetangga terdekat dengan objek data yang baru yaitu nilai K. Adapun nilai parameter K yang digunakan pada penerapan algoritma ini menggunakan nilai K = 3 sampai K = 9. Penggunaan dari parameter ini selanjutnya akan digunakan untuk melakukan perbandingan performa algoritma dengan tujuan mencari nilai K yang optimal. Perbandingan akan dilaksanakan menggunakan nilai akurasi, presisi, *recall*, dan *f-measure* dari tiap hasil parameter yang dikenakan. Penerapan algoritma *K-Nearest Neighbor* dapat dilakukan dengan menggunakan perintah *Python* seperti pada tabel 9 berikut:

TABEL IX
KODE *PYTHON* PENERAPAN ALGORITMA KNN

Keterangan	Source Code
KNN	<pre>from sklearn.neighbors import KNeighborsClassifier from sklearn import metrics knn = KNeighborsClassifier(n_neighbors=3)</pre>
Data Latih	<pre>knn.fit(x_train, y_train)</pre>
Data Uji	<pre>y_pred = knn.predict(x_test)</pre>
Accuracy	<pre>print("Accuracy:", metrics.accuracy_score(y_test, y_pred))</pre>
Precision	<pre>print ("Precision:", metrics.precision_score(y_test, y_pred, average=None))</pre>

```

Recall          print ("Recall:",metrics.recall_score(y_test, y_pred, average=None))
F-Measure      print ("F1 score:",metrics.f1_score(y_test, y_pred, average=None))
Klasifikasi    print (classification_report(y_test, y_pred))

```

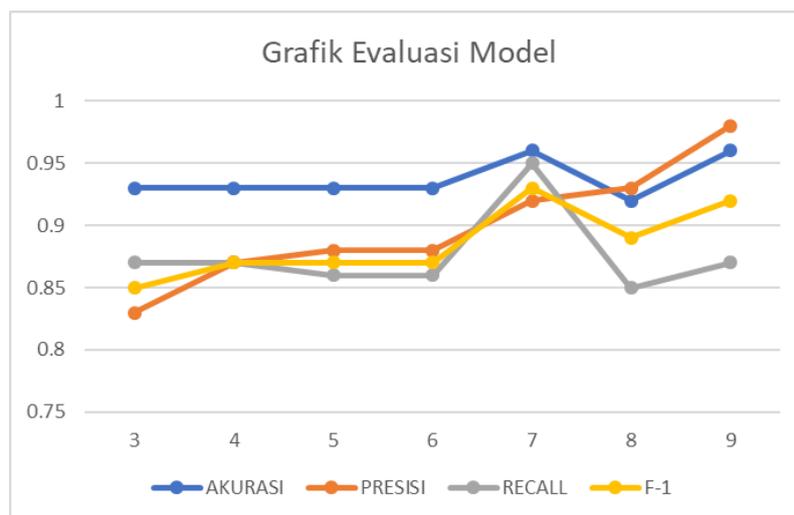
E. Evaluasi Model

Evaluasi model dilakukan dengan menggunakan *confusion matrix*. Evaluasi model ini dinilai berdasarkan pada tingkat akurasi, presisi, *recall*, dan *f-measure*. Evaluasi model ini dilaksanakan dengan membandingkan nilai K dalam mengetahui perbedaan performa dari setiap nilai K. Setiap nilai akurasi, presisi, *recall*, dan *f-measure* yang diperoleh dalam penggunaan nilai K dalam tahapan sebelumnya akan digambarkan dalam bentuk tabel. Hasil *confusion matrix* dari penerapan algoritma dengan menggunakan perintah *Python* terlihat seperti pada tabel 10 berikut:

TABEL X
EVALUASI MODEL

Nilai K	Akurasi	Presisi	<i>Recall</i>	<i>F-Measure</i>
3	0.93	0.83	0.87	0.85
4	0.93	0.87	0.87	0.87
5	0.93	0.88	0.86	0.87
6	0.93	0.88	0.86	0.87
7	0.96	0.92	0.95	0.93
8	0.92	0.93	0.85	0.89
9	0.96	0.98	0.87	0.92

Agar dapat memahami perbandingan dari performa nilai-nilai tersebut dengan mudah, maka dapat digambarkan dengan menggunakan grafik seperti gambar 2 berikut:



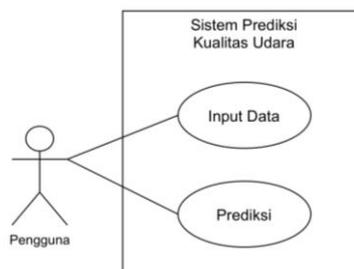
Gambar. 2. Grafik Evaluasi Model

Pada gambar 3, nilai K = 3 hingga K = 6 memiliki nilai akurasi bernilai sama yaitu 93%. Sedangkan nilai presisi dan *f-measure* mengalami peningkatan, namun untuk nilai *recall* mengalami penurunan. Pada nilai K = 7, keempat nilai ini mengalami peningkatan. Nilai akurasi yang diperoleh yaitu sebesar 96%, nilai presisi 92%, nilai *recall* 95%, dan nilai *f-measure* 93%. Namun, saat nilai K = 8, hanya nilai presisi yang mengalami peningkatan. Kemudian saat nilai K = 9, keempat nilai ini kembali mengalami peningkatan. Namun peningkatan yang dialami ini tidak berbeda jauh dengan nilai K = 7. Berdasarkan analisis evaluasi model yang sudah dilaksanakan, bisa disimpulkan bahwa nilai K yang optimum yaitu K = 7 karena tingkat akurasi, presisi, *recall*, dan *f-measure*

seluruhnya meningkat saat parameter yang digunakan adalah 7.

F. Pembangunan Sistem Prediksi

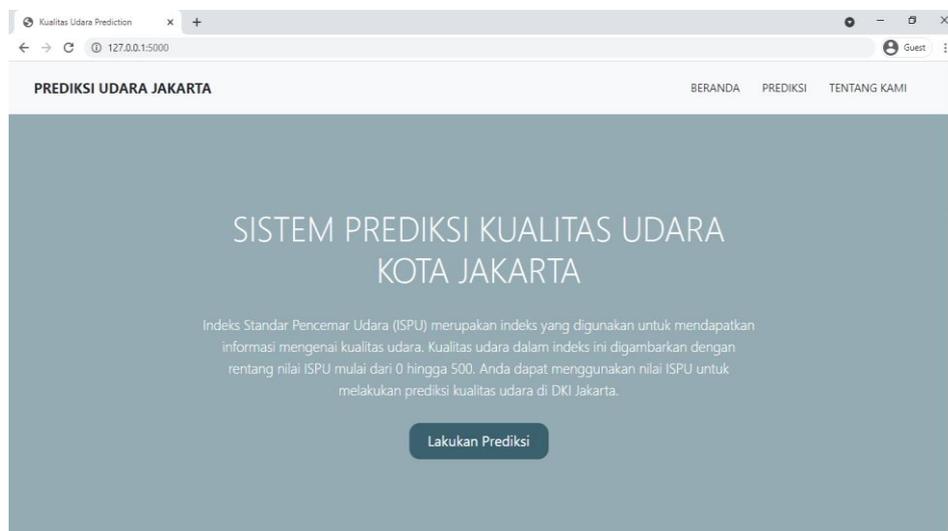
Tahap selanjutnya yaitu melakukan pembangunan sistem prediksi kualitas udara dengan mengembangkan model algoritma yang sudah dibangun. Sistem prediksi kualitas udara yang akan dibangun berbasis *website* dengan menggunakan *framework Flask* dengan bahasa pemrograman *Python*. *Flask* merupakan *web framework* yang berfungsi untuk membangun *website*, di mana *flask* menyediakan fungsi-fungsi dan kumpulan kode yang dapat digunakan untuk mendukung pembuatan *website*. *Flask* biasa dimanfaatkan untuk membuat *website* sederhana dan berskala kecil. Selain itu, *flask* juga dapat meng-*import* hasil pemodelan algoritma yang sudah dibangun. Sistem prediksi yang dibangun digambarkan pada rancangan *use case* pada gambar 3 berikut:



Gambar. 3. Rancangan *Use Case* Sistem Prediksi Kualitas Udara

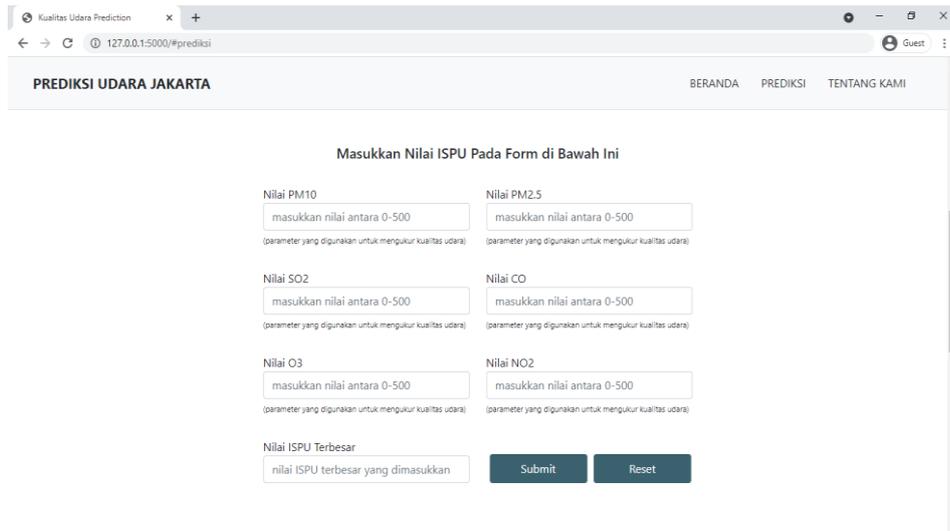
Aktivitas yang dapat dilakukan pengguna dalam sistem ini yaitu melakukan *input* data dan prediksi. Dalam sistem, pengguna dapat memasukan data-data yang digunakan untuk melakukan prediksi kualitas udara. Dalam sistem ini, diharapkan dapat menampilkan prediksi kualitas udara dengan menggunakan model algoritma yang sudah didapatkan pada tahap sebelumnya. Model algoritma dalam pembangunan model ini yakni nilai $K = 7$.

Berikut ini tampilan dari halaman awal sistem prediksi kualitas udara seperti gambar 4:



Gambar. 4. Tampilan Halaman Awal

Bagian atas halaman awal, terdapat navigasi yang mengarahkan pengguna ke halaman prediksi dan tentang kami. Halaman ini terdapat judul dari sistem prediksi kualitas udara dan informasi tambahannya. Selain itu, terdapat tombol yang bertuliskan “Lakukan Prediksi”, di mana tombol ini mengarahkan pengguna ke halaman prediksi. Pada halaman prediksi tampilannya seperti gambar 5 berikut:



The screenshot shows a web browser window with the URL '127.0.0.1:5000/#prediksi'. The page title is 'PREDIKSI UDARA JAKARTA'. Below the navigation menu, there is a heading 'Masukkan Nilai ISPU Pada Form di Bawah Ini'. The form contains several input fields for different air quality parameters, each with a range of 0-500 and a note: '(parameter yang digunakan untuk mengukur kualitas udara)'. The parameters are: Nilai PM10, Nilai PM2.5, Nilai SO2, Nilai CO, Nilai O3, and Nilai NO2. There is also a field for 'Nilai ISPU Terbesar'. At the bottom of the form are 'Submit' and 'Reset' buttons.

Gambar. 5. Tampilan Halaman Prediksi

Pada halaman prediksi, pengguna dapat memasukan nilai ISPU dari semua parameter kualitas udara yang ditampilkan. Nilai ISPU yang dimasukan dibatasi mulai dari angka 0 - 500 dengan bilangan bulat. Di halaman ini, terdapat 2 (dua) tombol. Tombol *submit* digunakan untuk menampilkan hasil prediksi berdasarkan masukan data. Tombol *reset* digunakan untuk menghapus semua masukan yang ada pada *form* prediksi. Pada halaman hasil prediksi tampilannya seperti gambar 6:



The screenshot shows the 'Prediksi Udara Jakarta' web application displaying the results of a prediction. The heading is 'Hasil Prediksi'. Below it, the text reads 'Berikut ini data yang dimasukkan untuk dilakukan prediksi level:'. A table shows the input values for various parameters and the maximum value. Below the table, the prediction level is 'SEDANG'. A recommendation is given: 'Rekomendasi untuk Anda: Orang dengan kondisi fisiknya rentan agar mengurangi aktivitas di luar ruangan dan/atau bekerja berat'. At the bottom, there is a heading 'Berikut ini penjelasan dari setiap level kualitas udara' and a table with columns 'Level ISPU' and 'Keterangan'.

PM10	PM25	SO2	CO	O3	NO2	Maksimum
56	92	25	20	16	40	92

Prediksi Kualitas Udara Berada di Level : SEDANG

Rekomendasi untuk Anda:
Orang dengan kondisi fisiknya rentan agar mengurangi aktivitas di luar ruangan dan/atau bekerja berat

Berikut ini penjelasan dari setiap level kualitas udara

Level ISPU	Keterangan

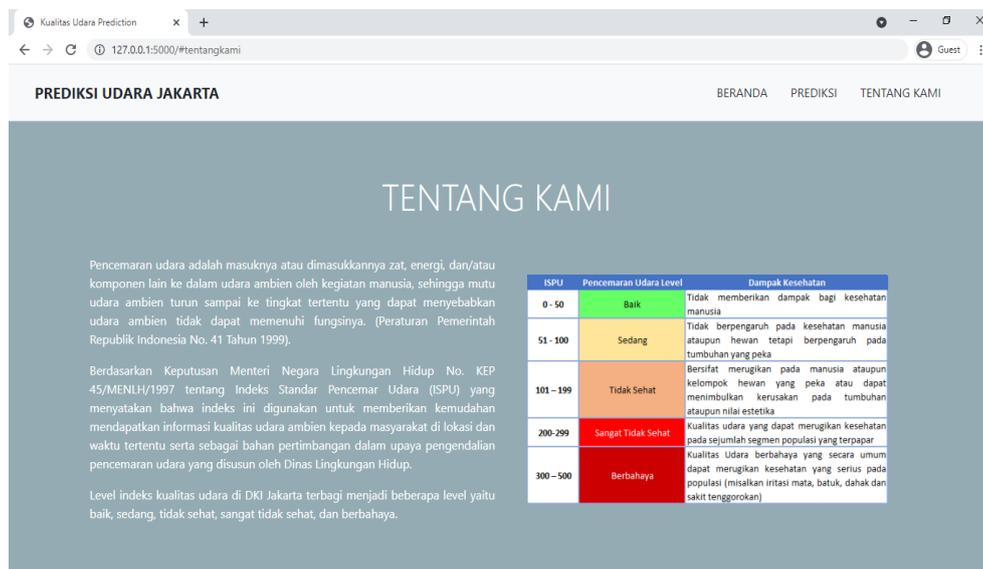
Gambar. 6. Tampilan Halaman Hasil Prediksi

Pada halaman hasil prediksi seperti gambar 6, terdapat tabel yang menampilkan informasi nilai-nilai ISPU yang telah dimasukkan pada *form* di halaman sebelumnya. Di bawah tabel, terdapat informasi yang menjelaskan prediksi level dari kualitas udara. Selain itu, sistem juga memberikan rekomendasi atau saran berdasarkan pada hasil prediksi yang dihasilkan. Berikut ini tampilan dari penjelasan setiap level kualitas udara pada gambar 7:



Gambar. 7. Tampilan Halaman Penjelasan Hasil Prediksi

Bagian ini diberikan penjelasan mengenai setiap level kualitas udara berdasarkan ISPU. Penjelasan dalam tabel ini juga bersumber pada Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia No. P.14/MENLHK/SETJEN/KUM.1/7/2020 mengenai Indeks Standar Pencemar Udara [15]. Berikut ini halaman tentang kami dengan tampilan yang tersaji dalam gambar 8:



Gambar. 8. Tampilan Halaman Tentang Kami

Pada gambar 8 yakni halaman tentang kami memuat informasi mengenai level pencemaran udara dan dampak yang diakibatkannya. Level prediksi kualitas udara yang ditampilkan pada halaman hasil prediksi sesuai dengan tabel yang ditampilkan pada halaman tentang kami ini, sehingga sebelum melakukan prediksi, pengguna dapat memahami hasil dan informasi yang diberikan pada halaman hasil prediksi.

G. Pengujian Sistem Prediksi

Sistem yang sudah dibangun selanjutnya dilakukan pengujian sistem dengan *blackbox testing*. Pengujian yang dilakukan menggunakan teknik *equivalence partitioning*. Teknik ini melakukan pembagian partisi data yang melakukan pengujian untuk setiap menu dalam sistem dengan pengklasifikasian dan pengelompokan berdasarkan fungsinya. Pengujian ini memiliki tujuan mengetahui sistem apakah dapat bekerja sesuai dengan fungsinya. Dalam hal ini, sistem harus memberikan respon yang benar dan tepat agar sistem prediksi dapat digunakan. Pengujian yang dilakukan seperti pada tabel 11 berikut:

TABEL XI
PENGUJIAN SISTEM DENGAN *BLACKBOX TESTING*

Modul	Prosedur Uji	Keluaran yang diharapkan	Hasil yang didapatkan	Status
Halaman Awal	Mengakses halaman url	Menampilkan halaman awal	Menampilkan halaman awal	Berhasil
	Klik tombol prediksi	Menampilkan halaman prediksi	Menampilkan halaman prediksi	Berhasil
	Klik tombol tentang kami	Menampilkan halaman tentang kami	Menampilkan halaman tentang kami	Berhasil
Halaman Prediksi	Klik tombol <i>submit</i>	Menampilkan halaman hasil prediksi	Menampilkan halaman hasil prediksi	Berhasil
	Klik tombol <i>reset</i>	Menghapus masukan data	Menghapus masukan data	Berhasil
Halaman Hasil Prediksi	Klik tombol lakukan prediksi	Menampilkan halaman prediksi	Menampilkan halaman prediksi	Berhasil

Melalui pengujian yang dilakukan yakni tabel 11 di atas, sebagian besar modul atau fungsi sistem memiliki status berhasil. Hal ini menandakan bahwa sistem dapat bekerja sesuai dengan yang diharapkan. Maka, sistem prediksi kualitas udara ini bisa digunakan dalam melakukan prediksi mutu udara DKI Jakarta.

IV. KESIMPULAN

Berdasarkan penelitian yang sudah dilaksanakan, maka bisa disimpulkan bahwa algoritma *K-Nearest Neighbor* bisa digunakan dalam memprediksi kualitas udara menurut Indeks Standar Pencemar Udara (ISPU). Prediksi dilakukan dengan 7 (tujuh) fitur yang merupakan parameter-parameter udara meliputi pm10, pm25, so2, co, o3, no, dan max untuk menentukan kualitas udara. Pada evaluasi model algoritma dengan menggunakan *confusion matrix* menghasilkan bahwa nilai $K = 7$ memiliki performa yang terbaik dimana nilai akurasi sebanyak 96%, presisi 92%, *recall* 95%, dan *f-measure* 93%. Selain itu, sistem prediksi kualitas udara yang dibangun dengan menggunakan *framework Flask* dapat menampilkan hasil prediksi mengenai kualitas udara di DKI Jakarta.

DAFTAR PUSTAKA

- [1] S. Nurjanah, A. M. Siregar, dan D. S. Kusumaningrum, "Penerapan Algoritma *K-Nearest Neighbor* (KNN) Untuk Klasifikasi Pencemaran Udara di Kota Jakarta", *Scientific Student Journal for Information, Technology and Science*, vol. 1, no. 2, pp. 71-76, 2020.
- [2] CNN Indonesia, 2021, Kualitas Udara Jakarta Kembali Memburuk, Peringkat 6 di Dunia [Online]. Tersedia: <https://www.cnnindonesia.com/nasional/20210421131002-20-632776/kualitas-udara-jakarta-kembali-memburuk-peringkat-6-di-dunia>.
- [3] Dinas Lingkungan Hidup, 2020, Indeks Standar Pencemaran Udara (ISPU) Selama Masa PSBB Di DKI Jakarta [Online]. Tersedia: <https://statistik.jakarta.go.id/indeks-standar-pencemaran-udara-ispu-selama-masa-psbb-di-dki-jakarta/>.
- [4] N. Aini, dkk, "Sistem Prediksi Tingkat Pencemaran Polusi Udara dengan Algoritma *Naïve Bayes* di Kota Makassar", dalam *Prosiding Seminar Nasional Komunikasi dan Informatika #3*, 2019, pp. 83-90.
- [5] Suyanto, 2017, *Data Mining Untuk Klasifikasi dan Klusterisasi Data*, Informatika Bandung.
- [6] E. Etriyanti, D. Syamsuar, dan Y. N. Kunang, "Implementasi Data Mining Menggunakan Algoritme *Naïve Bayes Classifier* dan C4.5 untuk Memprediksi Kelulusan Mahasiswa", *Telematika*, vol. 13, no. 1, pp. 56-67, 2020.
- [7] E. B. Susanto, T. A. Cahyanto, dan R. Umilasar, "Analisis Perbandingan Algoritma *Naïve Bayes* dan KNN Untuk Klasifikasi Multi Dataset", *Skripsi. Universitas Muhammadiyah Jember*, 2019.
- [8] M. R. Noviansyah, T. Rismawan, dan D. M. Midyanti, "Penerapan Data Mining Menggunakan Metode *K-Nearest Neighbor* Untuk Klasifikasi Indeks Cuaca Kebakaran Berdasarkan Data AWS (*Automatic Weather Station*) (Studi Kasus: Kabupaten Kubu Raya)", *Jurnal Coding Sistem Komputer Untan*, 6 (2), pp 48-56, 2018.
- [9] D. S. Wisdayani, I. M. Nur, dan R. Wasono, "Perbandingan Algoritma *K-Nearest Neighbor* dan *Naïve Bayes* Untuk Klasifikasi Tingkat Keparahan Korban Kecelakaan Lalu Lintas Kabupaten Pati Jawa Tengah", *Skripsi, Universitas Muhammadiyah Semarang*, 2019.
- [10] Hasran, "Klasifikasi Penyakit Jantung Menggunakan Metode *K-Nearest Neighbor*", *Indonesian Journal of Data and Science*, vol. 1, no. 1, pp. 6-10, 2020.
- [11] R. A. Arnomo, W. L. Y. Saptomo, dan P. Harsadi, "Implementasi Algoritma *K-Nearest Neighbor* Untuk Identifikasi Kualitas Air (Studi Kasus: PDAM Kota Surakarta)", *Jurnal TIKomSiN*, vol. 6, no. 1, pp. 1-5, 2018.
- [12] M. M. Baharuddin, T. Hasanuddin, dan H. Azis, "Analisis Performa Metode *K-Nearest Neighbor* Untuk Identifikasi Jenis Kaca", *ILKOM Jurnal Ilmiah*, vol. 11 no. 3, pp. 269-274, 2019.
- [13] K. Prabowo, dan B. Muslim, 2018, *Bahan Ajar Kesehatan Lingkungan Penyakit Udara*, Badan Pengembangan dan Pemberdayaan Sumber Daya Manusia Kesehatan.
- [14] R. U. Wahyudi, "Sistem Pakar *E-Tourism* Pada Dinas Pariwisata D.I.Y Menggunakan Metode *Forward Chaining*", *Jurnal Ilmiah DASi*, pp. 67-75, 2016.
- [15] Republik Indonesia. *Peraturan Menteri Lingkungan Hidup dan Kehutanan Republik Indonesia Nomor P.14/MENLHK/SETJEN/KUM.1/7/2020 tentang Indeks Standar Pencemar Udara*.