Mutiara: Jurnal Penelitian dan Karya Ilmiah
Vol. 1 No. 6 Desember 2023
P-ISSN: 2988-313X, E-ISSN: 2988-3148, Hal 347-362
DOI: https://doi.org/10.59059/mutiara.v1i6.702

Penerapan Metode Naïve Bayes dalam Peramalan Polusi Udara di Kota Jakarta

Sandy Andika Maulana
Universitas Negeri Medan
Korespodensi penulis: samsandi134@gmail.com

Shabrina Husna Batubara
Universitas Negeri Medan
Email: shabrinahusna1@gmail.com

Wahyu Kurnia Rahman
Universitas Negeri Medan
Email: wahyukurniarahman59@gmail.com

Jl. Williem Iskandar Ps. V Medan Estate, Percut Sei Tuan, Deli Serdang

Abstract. *This research aims to analyze and predict the level of air pollution in Jakarta City using ISPU data of DKI Province, adopting the Naïve Bayes method. The test results show that the Naïve Bayes algorithm has excellent performance, with 93% accuracy, 98% precision, 100% recall, and 99% f1-score. The implication is that this model can be effectively used for air pollution forecasting in Jakarta City, assisting authorities in making decisions related to air quality and environmental improvement efforts.*

Keywords: *Air Pollution, ISPU, Naïve Bayes, Data mining, Confusion Matrix.*

Abstrak. Penelitian ini bertujuan untuk menganalisis dan memprediksi tingkat polusi udara di Kota Jakarta dengan menggunakan data ISPU Provinsi DKI, mengadopsi metode Naïve Bayes. Hasil pengujian menunjukkan bahwa algoritma Naïve Bayes memiliki kinerja yang sangat baik, dengan akurasi 93%, presisi 98%, recall 100%, dan f1-score 99%. Implikasinya adalah bahwa model ini dapat efektif digunakan untuk peramalan polusi udara di Kota Jakarta, membantu pihak berwenang dalam mengambil keputusan terkait kualitas udara dan upaya-upaya perbaikan lingkungan.

Kata kunci: Polusi Udara, ISPU, Naïve Bayes, Data Mining, Confusion Matrix.

LATAR BELAKANG

Pencemaran udara merujuk pada kehadiran satu atau lebih substansi fisik, kimia, atau biologis dalam atmosfer. Keberadaan zat-zat tersebut dalam konsentrasi tertentu dapat menimbulkan risiko bagi kesehatan manusia, hewan, dan tanaman. Selain itu, pencemaran udara juga dapat mengganggu aspek estetika dan kenyamanan, serta dapat merusak property (Chaniago, Dasrul. Annisa Zahara, Annisa.Suci Suci R, 2020). Pencemaran udara di Kota Jakarta telah menjadi perhatian serius selama beberapa tahun terakhir, Ahmad Safrudin menyatakan bahwa Jakarta dan sekitarnya telah mengalami kondisi yang sangat buruk selama tiga dekade terakhir, sejak Pengprograman Lingkungan

Received Oktober 21, 2023; Revised Oktober 30, 2023; Accepted November 24, 2023
*Sandy Andika Maulana, samsandi134@gmail.com

Perserikatan Bangsa-Bangsa (UNEP) pertama kali mengumumkan bahwa kualitas udara di Jakarta melampaui batas yang ditetapkan oleh Organisasi Kesehatan Dunia (WHO) (Indonesia, 2023).

Menurut Keputusan Badan Pengendalian Dampak Lingkungan (Bapedal) Nomor KEP-107/Kabapedal/11/1997, Pemerintah Indonesia menetapkan Indeks Standar Pencemar Udara sebagai acuan untuk menilai kualitas udara di suatu wilayah. Dampak dari menghirup udara dengan tingkat ISPU yang semakin tinggi dapat berpengaruh terhadap kesehatan. Semakin tinggi level ISPU, semakin buruk kualitas udara yang terhirup oleh tubuh. ISPU terdiri dari lima tingkatan, yaitu Baik, Sedang, Tidak Sehat, Sangat Tidak Sehat, dan Berbahaya (Kepala Bapedal, 1997).

Dalam konteks ini, penelitian ini menggunakan Indeks Standar Pencemar Udara (ISPU) di Jakarta sebagai sumber data kunci. ISPU adalah parameter penting yang mencakup sejumlah pencemar udara utama seperti PM2.5, PM10, SO2, CO, NO2, dan O3. Data yang digunakan dalam penelitian ini berasal dari stasiun pemantauan udara yang tersebar pada 5 lokasi di seluruh Kota Jakarta dan wilayah Provinsi DKI. Dataset ISPU memberikan gambaran yang komprehensif tentang kualitas udara di wilayah ini.

Dalam rangka meningkatkan akurasi peramalan polusi udara, penelitian ini memilih menggunakan metode Naïve Bayes. Metode ini dipilih karena fleksibilitasnya (Jain, 2021). dalam mengolah data multivariabel, yang mencakup data historis ISPU, faktor cuaca, kondisi lalu lintas, dan variabel lingkungan lainnya yang berdampak pada polusi udara. Metode Naïve Bayes dapat menghasilkan prediksi yang andal berdasarkan pemahaman yang lebih dalam tentang hubungan antarvariabel ini.

Penelitian ini menghadapi dua permasalahan utama. Pertama, penelitian memerinci proses analisis data ISPU untuk melakukan peramalan tingkat polusi di kota Jakarta. Kedua, penelitian ini mengungkap secara efektif setiap langkah penerapan metode Naïve Bayes dalam meramalkan tingkat polusi udara di Kota Jakarta dengan memanfaatkan data ISPU Provinsi DKI, menekankan signifikansi eksplorasi metodologi terstruktur untuk memastikan keakuratan model.

KAJIAN TEORITIS

Udara

Kehadiran udara memiliki peran yang sangat berpengaruh dalam menjaga kelangsungan hidup semua makhluk hidup, terutama dalam penyediaan oksigen. Oksigen dihasilkan melalui proses fotosintesis oleh tumbuhan dan alga, yang menyerap karbon dioksida (CO2). Bagi makhluk hidup oksigen berfungsi sebagai zat yang diperlukan dalam proses pernafasan. Setiap hari, kita perlu menghirup udara sebagai suatu kebutuhan. Secara kategoris, udara dapat dibedakan menjadi dua kategori, yaitu udara yang bersih dan udara yang tidak bersih (Kirono et al., 2022).

Indeks Standar Pencemaran Udara (ISPU)

Indeks Standar Pencemaran Udara (ISPU) merupakan nilai numerik yang tidak memiliki satuan, digunakan untuk mencerminkan kondisi kualitas udara di suatu lokasi ambien, dengan dasar dampaknya terhadap kesehatan manusia, nilai estetika, dan makhluk hidup lainnya.

Data Mining

Aktivitas data mining merupakan komponen penting dari proses Knowledge Discovery in Database (KDD), yang mencakup serangkaian langkah, seperti pemilihan data, pra-pemrosesan, transformasi, pelaksanaan data mining, dan evaluasi hasil. Proses KDD juga kerap disebut sebagai penemuan pengetahuan dalam basis data (Decy Arwini, 2020).

Machine Learning

Machine learning adalah suatu bentuk pembelajaran mesin yang sangat berguna dalam mengatasi masalah dan secara signifikan menyederhanakan eksekusi tugas. Machine learning dimulai dengan pertimbangan manusia mengenai bagaimana komputer dapat mempelajari dari pengalaman atau menyimpan informasi yang baru saja diolah oleh komputer tersebut. Tujuan dari pengembangan machine learning adalah untuk memberikan bantuan yang efektif kepada manusia dalam menyelesaikan masalah tanpa memerlukan instalasi ulang yang berulang, sehingga penggunaan machine learning menjadi lebih nyaman dan efisien (Kirono et al., 2022)

Classification (Klasifikasi)

Klasifikasi adalah langkah dalam menemukan fungsi yang menggambarkan data class, dengan maksud untuk memprediksi class dari objek atau data yang tidak memiliki label atau nilai yang diketahui. Untuk sampai ke tujuan ini, diperlukan suatu proses klasifikasi. Klasifikasi merujuk pada sebuah model atau algoritma yang bertujuan untuk memisahkan data ke kategori-kategori yang berbeda berdasarkan peran khusus. Model tersebut dapat berupa peraturan "jika-maka," seperti decision tree, atau rumus matematika (Kirono et al., 2022).

Naïve Bayes

Naïve Bayes adalah salah satu algoritma dalam klasifikasi yang berdasarkan probabilitas dan statistik. Algoritma ini pertama kali dikembangkan oleh Thomas Bayes. Pendekatan ini memiliki tujuan untuk membuat prediksi tentang kemungkinan kejadian di masa depan berdasarkan pengalaman di masa lalu, sehingga sering disebut sebagai Teorema Bayes. Fungsi Naïve Bayes digunakan untuk membangun model atau menghitung atribut data yang memiliki karakteristik berkelanjutan. Perhitungan yang menggambarkan metode Naïve Bayes adalah (Kirono et al., 2022).

$$P\,(xi \mid y\,) = \frac{1}{\sigma^2 y}\,\exp\left(-\frac{(xi - \mu y)^2}{2\sigma^2 y}\right)$$

Keterangan:

P(Xi|y): Likelihood (kemungkinan hasil kejadian)

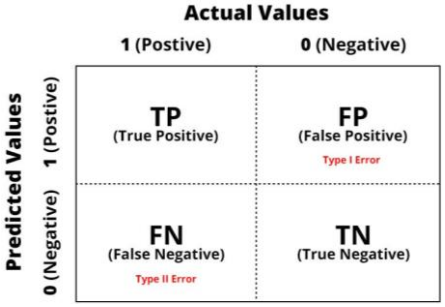
σ : Standar Deviasi dari atribut

μ : Mean dari atribut

Confusion Matrix

Confusion matrix adalah table matrix yang digunakan untuk menghitung kinerja suatu model data atau algoritma (Irkham Widhi Saputro, 2019). Tiap baris dalam matriks mencerminkan kelas data yang sesungguhnya, sementara setiap kolom mencerminkan kelas prediksi (atau sebaliknya).

Tabel 1. Confusion Matrix



Sumber: ksnugroho.medium.com (2019).

Dengan mengacu pada data ini, kita dapat menghimpun informasi yang berguna untuk mengevaluasi kinerja suatu model, termasuk:

1. Akurasi menilai sejauh mana kemampuan model dalam mengklasifikasikan data secara benar, perbandingan antara jumlah data yang diklasifikasikan dengan benar terhadap total data. Perhitungan akurasi diuraikan sebagai berikut:

$$\frac{TP + TN}{TP + FP + FN + TN}$$

2. Presisi merupakan ukuran tingkat ketepatan prediksi model. Ini mengukur seberapa sering prediksi positif model benar ketika memprediksi data sebagai positif. Perhitungan untuk presisi dapat dijelaskan sebagai berikut:

$$\frac{TP}{TP + FP}$$

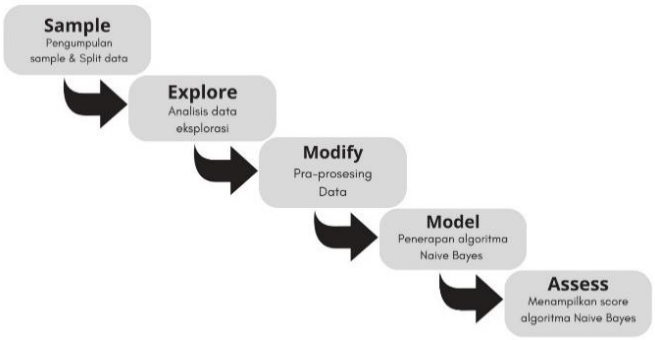
3. Recall mengukur sejauh mana model mampu mengidentifikasi semua data aktual yang bernilai positif. Persamaan recall adalah:

$$\frac{TP}{TP + FN}$$

4. F1-Score adalah metrik yang mencerminkan perbandingan rata-rata antara presisi dan recall, dengan memberikan bobot pada keduanya. Persamaan F1-Score adalah:

$$\frac{2(Recall * Precision)}{(Recall + Precision)}$$

METODE PENELITIAN



Gambar 1. Proses SEMMA

Sample

Ditahap ini, dilakukan proses pengumpulan data. Dataset yang akan digunakan dalam penelitian ini adalah dataset ISPU DKI Jakarta tahun 2021. Data tersebut diperoleh melalui portal Data Terbuka Pemerintah Provinsi DKI Jakarta melalui pranala: https://data.jakarta.go.id/dataset/indeks-standar-pencemaran-udara-ispu-tahun-2021. Dataset ini terdiri dari 365 data dan 11 tuple. Tuple-tuple tersebut yaitu: Categori, Critical, Max, NO2, O3, CO, SO2, PM2.5, PM 10, Stasiun, dan Tanggal

Variabel target (categori) ini memiliki 3 nilai, yaitu baik, sedang, dan tidak sehat. Semua variabel dalam dataset ini berisi data numerikal dan ketegorikal. Dalam penelitian ini, data akan dibagi (split) menjadi dua data yaitu, data latihan (training) dan data uji (testing) dengan proporsi 80:20.

Explore

Langkah ini merupakan fase eksplorasi data yang melibatkan analisis grafik, plot, dan statistik data. Tahap eksplorasi memiliki tujuan untuk mengidentifikasi potensi masalah atau nilai nol dalam data. Hasil dari eksplorasi data ini akan menjadi dasar untuk menentukan tindakan yang diperlukan terhadap data yang sedang dianalisis (H. Sabita, Fitria, 2021).

Modify

Di tahapan ini, Pra-prosessing digunakan untuk memproses data mentah sebelum data tersebut digunakan dalam proses analisis atau pemodelan. Pra-prosessing data memainkan peran penting dalam mempersiapkan data yang berkualitas untuk analisis dan

pemodelan yang lebih akurat dan andal. Tujuan dari pra-pemrosesan data adalah untuk membersihkan, mengubah format, dan mengatur data agar lebih siap digunakan.

Model

Setelah seluruh data menjalani proses pra-pemrosesan, langkah berikutnya adalah menerapkan algoritma prediksi. Dataset indeks standar pencemaran udara menjadi data yang akan diprediksi. Pada penelitian ini, algoritma yang akan diterapkan adalah algoritma naïve bayes, sebuah algoritma yang cocok digunakan pada data prediksi.

Assessment

Tahap ini melibatkan pengujian hasil dari algoritma yang digunakan. Pengujian ini dilakukan dengan menggunakan confusion matrix 3x3, yang berfungsi untuk memberikan informasi tentang kebenaran atau kesalahan Predicted serta keadaan Actual (S. Turgut, M. Dagtekin, 2018). Dengan menggunakan confusion matrix ini, akan dilakukan perhitungan F1-Score, presisi, recall, dan akurasi.

HASIL DAN PEMBAHASAN

Sample

Data yang digunakan pada penelitian ini diperoleh dari platform data.jakarta, dengan total 365 entri data. Di bawah ini adalah dataset yang akan menjadi fokus penelitian ini. Dari total data, akan dialokasikan 80% dari data ini untuk melatih model (data pelatihan) dan 20% sisanya akan diperuntukkan sebagai data yang digunakan untuk menguji model (data pengujian).

	tanggal	pm10	pm25	so2	co	o3	no2	max	critical	categori	location
0	1/1/2021	43	58	37	14	65	13	65	O3	SEDANG	DKI2
1	1/2/2021	58	86	38	38	80	12	86	PM25	SEDANG	DKI3
2	1/3/2021	64	93	37	20	86	13	93	PM25	SEDANG	DKI3
3	1/4/2021	50	67	36	16	77	7	77	O3	SEDANG	DKI2
4	1/5/2021	59	89	36	19	77	9	89	PM25	SEDANG	DKI3
...	...	...	...	...	...	...	...	...	...	...	...
360	12/27/2021	75	121	61	23	40	47	121	PM25	TIDAK SEHAT	DKI4
361	12/28/2021	59	89	53	16	34	33	89	PM25	SEDANG	DKI4
362	12/29/2021	61	98	54	15	37	29	98	PM25	SEDANG	DKI4
363	12/30/2021	60	102	53	17	38	44	102	PM25	TIDAK SEHAT	DKI4
364	12/31/2021	64	90	52	44	37	53	90	PM25	SEDANG	DKI4

365 rows × 11 columns

Gambar 2. Data Set

Explore

Pada langkah eksplorasi data ini, tujuannya adalah untuk mengidentifikasi jenis dan keadaan variabel-variabel dalam dataset. Berdasarkan visualisasi pada Gambar 3, dapat diidentifikasi bahwa terdapat 7 variabel dengan tipe data int64 dan 4 variabel dengan tipe data object.

```
dataForISPU.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 365 entries, 0 to 364
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   tanggal   365 non-null    object
 1   pm10      365 non-null    int64
 2   pm25      365 non-null    int64
 3   so2       365 non-null    int64
 4   co        365 non-null    int64
 5   o3        365 non-null    int64
 6   no2       365 non-null    int64
 7   max       365 non-null    int64
 8   critical  365 non-null    object
 9   categori  365 non-null    object
 10  location  365 non-null    object
dtypes: int64(7), object(4)
memory usage: 31.5+ KB
```

Gambar 3. Informasi tipe data dari masing-masing variable

Dalam ilustrasi pada Gambar 4 dan Gambar 5, dapat diperoleh data tentang mean, median, serta nilai maksimum dari setiap variable.

```
dataForISPU.describe()
```

	pm10	pm25	so2	co	o3	no2	max
count	365.000000	365.000000	365.000000	365.000000	365.000000	365.000000	365.000000
mean	60.506849	92.613699	49.189041	15.068493	52.460274	28.904110	94.030137
std	15.155896	24.912734	9.978818	5.683813	13.927074	10.639658	24.408647
min	19.000000	33.000000	19.000000	7.000000	27.000000	7.000000	45.000000
25%	53.000000	76.000000	43.000000	11.000000	42.000000	21.000000	77.000000
50%	62.000000	93.000000	51.000000	14.000000	51.000000	29.000000	93.000000
75%	68.000000	107.000000	54.000000	18.000000	59.000000	36.000000	108.000000
max	179.000000	174.000000	82.000000	47.000000	151.000000	65.000000	179.000000

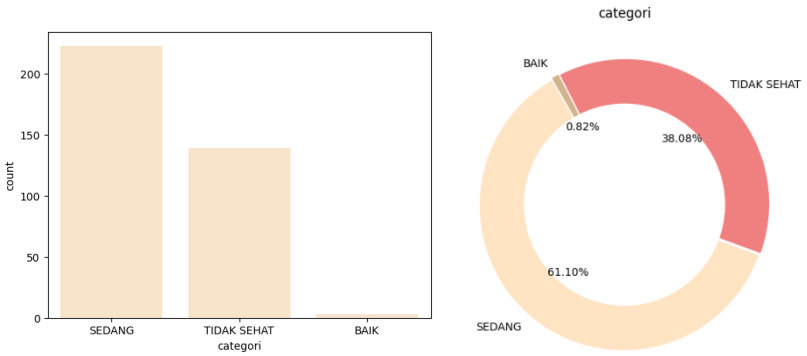
Gambar 4. Deskripsi variable tipe data angka (int64)

```
dataForISPU.describe(include=['object'])
```

	tanggal	critical	categori	location
count	365	365	365	365
unique	365	5	3	5
top	1/1/2021	PM25	SEDANG	DKI4
freq	1	336	223	226

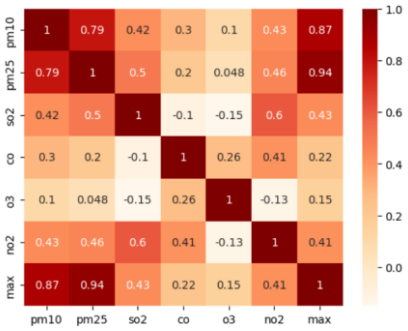
Gambar 5. Deskripsi variable tipe data object

Diagram pada Gambar 6 memperlihatkan pembagian data berdasarkan kelas di dalam dataset. Terdapat 223 entri data (sekitar 61.10%) yang termasuk dalam kategori "sedang", sementara 139 entri data (sekitar 38.08%) masuk dalam kategori "tidak sehat", dan sisanya 3 entri data (sekitar 0.82%) termasuk dalam kategori "baik".



Gambar 6. Pengelompokan data dalam setiap kategori

Gambar 7 menunjukkan tabel yang menggambarkan hubungan korelasi antara masing-masing variabel.



Gambar 7. Korelasi data

Modify

Pada langkah ini, akan dilakukan proses pembersihan data dengan tujuan mempersiapkan data sebelum langkah penerapan algoritma. Pada dataset ini, tuple tanggal tidak diperlukan, sehingga bisa dihapus tuple tesebut seperti pada gambar 8. Begitu juga untuk tuple max, tuple tersebut merupakan nilai maximum dari setiap baris, sehingga tidak diperlukannya dan bisa dihapus seperti pada gambar 8.

```
dfNew = dataForISPU.drop('tanggal', axis=1).drop('max', axis=1)
dfNew.head()
```

	pm10	pm25	so2	co	o3	no2	critical	categori	location
0	43	58	37	14	65	13	O3	SEDANG	DKI2
1	58	86	38	38	80	12	PM25	SEDANG	DKI3
2	64	93	37	20	86	13	PM25	SEDANG	DKI3
3	50	67	36	16	77	7	O3	SEDANG	DKI2
4	59	89	36	19	77	9	PM25	SEDANG	DKI3

Gambar 8. Data cleaning (dropping)

Gambar 9 menunjukkan proses memeriksa apakah data memiliki nilai yang hilang (missing value) atau tidak, dan ternyata tidak memilikinya.

```
dfNew.isnull().sum()
```

```
pm10        0
pm25        0
so2         0
co          0
o3          0
no2         0
critical    0
categori    0
location    0
dtype: int64
```

Gambar 9. Data cleaning (missing value)

Setelah tahap data cleaning, langkah selanjutnya adalah mengubah data bertipe object menjadi data dalam bentuk angka. Gambar 10 menunjukkan proses mengubah nilai 'TIDAK SEHAT' menjadi 0, 'SEDANG' menjadi 1, 'BAIK' menjadi 2, lalu digabung ke dalam dataframe dan tidak lupa diubah menjadi tipe data integer.

```
dictForClasses = {k: v for k, v in zip(['TIDAK SEHAT', 'SEDANG', 'BAIK'], list(range(len(['TIDAK SEHAT', 'SEDANG', 'BAIK']))))
print(dictForClasses)

for i in range(365):
    dfNew.iloc[i, 7] = dictForClasses[dfNew.iloc[i, 7]]

dfNew['categori']  = dfNew['categori'].astype('int')
dfNew.head()
```

```
{'TIDAK SEHAT': 0, 'SEDANG': 1, 'BAIK': 2}
```

	pm10	pm25	so2	co	o3	no2	critical	categori	location
0	43	58	37	14	65	13	O3	1	DKI2
1	58	86	38	38	80	12	PM25	1	DKI3
2	64	93	37	20	86	13	PM25	1	DKI3
3	50	67	36	16	77	7	O3	1	DKI2
4	59	89	36	19	77	9	PM25	1	DKI3

Gambar 10. Transformasi data categori

Gambar 11 menunjukkan proses mengubah data critical dan location menjadi sebuah vektor biner dengan nilai 1 pada kategori yang sesuai dan 0 untuk kategori lainnya.

```
X = dfNew.loc[:, ['critical', 'location']]
X = OneHotEncoder(cols=['critical', 'location']).fit_transform(X)
X.astype('int')

dfNew2 = pd.concat([dfNew.iloc[:, 0:6], X, dfNew['categori']], axis=1)
dfNew2
```

	pm10	pm25	so2	co	o3	no2	critical_1	critical_2	critical_3	critical_4	location_1	location_2	location_3	location_4	location_5	categori
0	43	58	37	14	65	13	1	0	0	0	1	0	0	0	0	1
1	58	86	38	38	80	12	0	1	0	0	0	1	0	0	0	1
2	64	93	37	20	86	13	0	1	0	0	0	1	0	0	0	1
3	50	67	36	16	77	7	1	0	0	0	1	0	0	0	0	1
4	59	89	36	19	77	9	0	1	0	0	0	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
360	75	121	61	23	40	47	0	1	0	0	0	0	0	1	0	0
361	59	89	53	16	34	33	0	1	0	0	0	0	0	1	0	1
362	61	98	54	15	37	29	0	1	0	0	0	0	0	1	0	1
363	60	102	53	17	38	44	0	1	0	0	0	0	0	1	0	0
364	64	90	52	44	37	53	0	1	0	0	0	0	0	1	0	1

Gambar 11. Transformasi data critical dan location

Setelah tahap transformasi data, langkah selanjutnya adalah menyeimbangkan data antara kelas-kelas yang ada. Kelas "SEDANG" memiliki jumlah data yang lebih besar daripada kelas "TIDAKSEHAT" dan kelas "BAIK". Oleh karena itu, diperlukan proses resampling untuk mencapai keseimbangan data tersebut. Gambar 12 menggambarkan proses resampling yang dilakukan menggunakan bahasa pemrograman Python.

```
from sklearn.utils import resample
tidaksehat = dfNew2[dfNew2['categori']==0]
sedang = dfNew2[dfNew2['categori']==1]
baik = dfNew2[dfNew2['categori']==2]

df_minority_1 = resample(tidaksehat, replace = True, n_samples = 223)
df_minority_2 = resample(baik, replace = True, n_samples = 223)

dflagi = pd.concat([df_minority_1, sedang, df_minority_2])
from sklearn.utils import shuffle
dflagi = shuffle(dflagi)
dflagi.categori.value_counts()
```

```
1    223
2    223
0    223
Name: categori, dtype: int64
```

Gambar 12. Data resample

Model

Sebelum melaksanakan model, langkah awal yang penting adalah mendefinisikan variable bebas (x) dan variable terikat (y). Gambar 13 memvisualisasikan proses deklarasi variable x dan y.

```
X = np.array(dflagi.drop(['categori'], axis=1))
y = np.asarray(dflagi.iloc[:, 15]).astype('int64')
```

Gambar 13. Variabel x dan y

Selanjutnya diperlukan langkah untuk memecah dataset menjadi dua bagian: data pelatihan dan data pengujian. Gambar 14 memvisualisasikan proses pembagian data, di mana 80% digunakan untuk melatih model dan 20% untuk menguji model.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.2)
```

Gambar 14. Split dataset

Setelah data sudah dibagi (split), langkah berikutnya adalah menerapkan algoritma pada data tersebut. Dalam penelitian ini, algoritma yang digunakan adalah Naïve Bayes. Gambar 15 mengilustrasikan langkah-langkah yang dilakukan dalam proses penerapan algoritma Naïve Bayes.

```
gaussian_nb = GaussianNB()
gaussian_nb.fit(X_train, y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
gaussian_predict = gaussian_nb.predict(X_test)
rill = y_test
predicted = gaussian_predict
accuracy = accuracy_score(predicted, rill)
print('Data actual    :', rill)
print('Data predicted :', predicted)
print('accuracy :', accuracy)
```

```
Data actual    : [2 1 1 1 2 1 0 2 2 0 0 0 0 0 2 0 1 1 2 2 0 2 1 2 0 0 0 2 1 2 2 2 2 1 0 1 0
 0 0 2 2 2 0 1 0 1 1 0 1 0 0 1 1 0 2 0 1 2 2 0 2 2 1 1 1 2 1 1 1 0 2 1 0 0
 1 0 2 2 2 2 1 2 1 0 0 1 0 1 1 1 1 2 2 0 1 1 1 0 1 0 0 2 0 2 0 1 1 2 0 2 0
 1 1 2 0 0 1 1 1 0 1 2 0 0 0 1 0 1 2 2 2 2 2 1]
Data predicted : [2 0 1 0 2 0 0 2 2 0 0 0 0 0 0 2 0 0 1 2 2 0 2 1 2 0 0 0 2 1 2 2 2 2 1 0 1 0
 0 0 2 2 2 2 0 0 0 0 0 0 1 0 0 0 0 0 2 0 1 2 2 0 2 2 2 0 1 0 2 1 0 1 0 2 1 0 0
 1 0 2 2 2 2 2 0 2 1 0 0 0 0 1 0 0 1 2 2 2 0 0 0 0 0 0 0 0 0 0 2 0 2 0 1 0 2 0 2 0
 1 1 2 0 0 0 0 0 0 0 1 2 0 0 0 1 0 0 2 2 2 2 2 2 0]
accuracy : 0.8059701492537313
```

Gambar 15. Implementasi model

Dari gambar di atas, dapat dilihat bahwa telah diperoleh hasil akurasi sebesar 80%. Karena hasil akurasi yang diperoleh belum tinggi atau belum mencapai lebih dari 90%, maka langkah selanjutnya yaitu mengembangkan model untuk meningkatkan akurasi dari klasifikasi naïve bayes dengan melakukan Tuning Parameter menggunakan GridSearch. Gambar 16 memperlihatkan proses melakukan Tuning Parameter pada Naïve Bayes Menggunakan GridSearch.

```
param_grid = {'var_smoothing': np.logspace(0,-9, num=100)}
grid = GridSearchCV(GaussianNB(), param_grid, cv=cv_method, verbose=1, scoring='accuracy')
grid.fit(X_train, y_train)
```

```
Fitting 15 folds for each of 100 candidates, totalling 1500 fits
```

```
▸ GridSearchCV
▸ estimator: GaussianNB
    ▸ GaussianNB
```

Gambar 16. Proses tuning parameter

Setelah itu, langkah berikutnya adalah Membuat ulang model NB sesuai dengan GridSearch. Gambar 17 memperlihatkan tahapan yang dilakukan dalam proses penerapan algoritma Naïve Bayes dengan menggunakan hasil estimator terbaik.

```
GaussianNB_2 = grid.best_estimator_
GaussianNB_2.fit(X_train, y_train)
```

```
GaussianNB(var_smoothing=0.0023101297000831605)
```

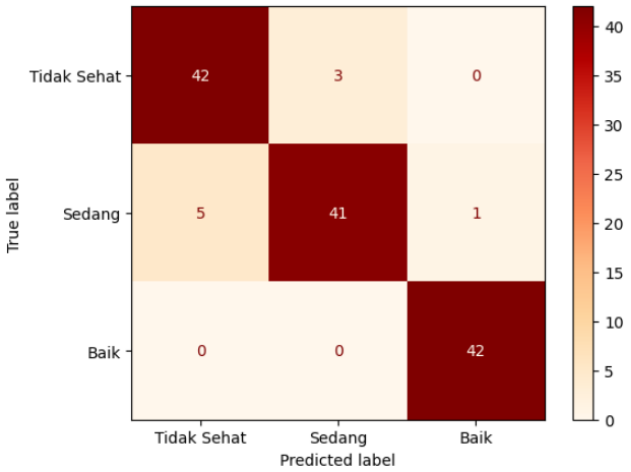
```
rill_2 = y_test
predicted_2 = GaussianNB_2.predict(X_test)
accuracy_2 = accuracy_score(predicted_2, rill_2)
print('Data actual    :', rill_2)
print('Data predicted :', predicted_2)
print('accuracy :', accuracy_2)
```

```
Data actual    : [2 1 1 1 2 1 0 2 2 0 0 0 0 0 2 0 1 1 2 2 0 2 1 2 0 0 0 2 1 2 2 2 2 1 0 1 0
 0 0 2 2 2 0 1 0 1 1 0 1 0 0 1 1 0 2 0 1 2 2 0 2 2 1 1 1 2 1 1 1 0 2 1 0 0
 1 0 2 2 2 2 1 2 1 0 0 1 0 1 1 1 1 2 2 0 1 1 1 0 1 0 0 2 0 2 0 1 1 2 0 2 0
 1 1 2 0 0 1 1 1 0 1 2 0 0 0 1 0 1 2 2 2 2 2 1]
Data predicted : [2 1 1 1 2 1 0 2 2 0 0 0 0 0 2 0 1 1 2 2 0 2 1 2 0 0 1 2 1 2 2 2 2 1 0 1 0
 1 0 2 2 2 0 1 0 0 1 0 1 0 0 1 1 0 2 0 0 2 2 0 2 2 1 0 1 2 1 1 1 0 2 1 0 1
 1 0 2 2 2 2 0 2 1 0 0 1 0 1 1 1 1 2 2 0 1 1 1 0 1 0 0 2 0 2 0 1 1 2 0 2 0
 2 1 2 0 0 0 1 1 0 1 2 0 0 0 1 0 1 2 2 2 2 2 1]
accuracy : 0.9328358208955224
```

Gambar 17. Implementasi model terbaru

Assessment

Pada tahap ini, dilakukan pengujian performa algoritma yang telah diimplementasikan dengan memanfaatkan matriks kebingungan (confusion matrix). Gambar 18 menampilkan hasil matriks kebingungan untuk algoritma "naïve bayes".



Gambar 18. Confusion matrix naïve bayes

Dengan menggunakan matriks kebingungan ini, kita dapat melakukan perhitungan untuk mengukur beberapa metrik kinerja seperti akurasi, recall, presisi, dan nilai F1. Dari Gambar 19, tampak bahwa algoritma " naïve bayes " mencapai akurasi sekitar 93%, presisi sekitar 98%, recall sekitar 100%, dan nilai F1 sekitar 99%.

	precision	recall	f1-score	support
0	0.89	0.93	0.91	45
1	0.93	0.87	0.90	47
2	0.98	1.00	0.99	42
accuracy			0.93	134
macro avg	0.93	0.94	0.93	134
weighted avg	0.93	0.93	0.93	134

Gambar 19. Hasil pengujian naïve bayes

Model algoritma ini akan diaplikasikan ke dalam sebuah aplikasi web sederhana yang dibangun menggunakan Streamlit. Gambar 20 adalah hasil dari aplikasi sederhana berbasis web yang berjudul peramalan polusi udara. Pengguna akan diminta untuk memasukkan informasi seputar kondisi udara melalui sidebar. Setelah pengguna mengisi informasi tersebut, aplikasi secara langsung akan menampilkan hasil prediksi. Jika hasil prediksi menunjukkan "Baik", itu berarti kualitas udara sangat baik untuk dihirup. Sebaliknya, jika prediksi adalah "Sedang", itu menandakan bahwa udara masih cukup layak untuk dihirup. Namun, jika hasil prediksi adalah "Tidak Sehat", itu mengindikasikan bahwa kualitas udara sangat merugikan jika dihirup.

Gambar 20. Hasil pengujian naïve bayes pada aplikasi

KESIMPULAN DAN SARAN

Tujuan dari penelitian ini adalah untuk menguji efektivitas algoritma naïve bayes dalam memprediksi tingkat polusi udara di kota Jakarta. Berdasarkan hasil pengujian, dapat disimpulkan bahwa algoritma naïve bayes efektif dalam memprediksi tingkat polusi udara di kota Jakarta menggunakan dataset ISPU yang tersedia di yang tersedia di Data

Terbuka Pemerintah Provinsi DKI Jakarta. Dengan menggunakan metode confusion matrix, pengujian menunjukkan bahwa algoritma naïve bayes mencapai tingkat akurasi sebesar 93%, precision sebesar 98%, recall sebesar 100%, dan f1-score sebesar 99%. Hasil ini menunjukkan bahwa algoritma naïve bayes memiliki kinerja yang sangat baik dalam memprediksi tingkat polusi udara di kota Jakarta, dengan tingkat akurasi yang tinggi, presisi yang tinggi, serta kemampuan untuk mengenali dan mengambil kembali data yang relevan dengan sempurna.

Saran yang dapat diberikan berdasarkan penelitian ini adalah untuk mempertimbangkan penggunaan algoritma Naïve Bayes dalam pengawasan dan pemantauan tingkat polusi udara di kota Jakarta. Algoritma ini dapat memberikan kontribusi dalam mengidentifikasi potensi pencemaran udara. Selain itu, disarankan untuk melakukan penelitian lebih lanjut guna membandingkan performa algoritma Naïve Bayes dengan algoritma lainnya, serta menguji model ini dengan dataset yang lebih luas untuk mendapatkan hasil yang lebih komprehensif dan valid.

DAFTAR REFERENSI

Chaniago, Dasrul. Annisa Zahara, Annisa.Suci Suci R, I. (2020). *Indeks Standar Pencemar Udara (Ispu) sebagai Informasi Mutu Udara Ambien di Indonesia*. https://ditppu.menlhk.go.id/portal/read/indeks-standar-pencemar-udara-ispu-sebagai-informasi-mutu-udara-ambien-di-indonesia

Decy Arwini, N. P. (2020). Dampak Pencemaran Udara Terhadap Kualitas Udara Di Provinsi Bali. *Jurnal Ilmiah Vastuwidya*, *2*(2), 20–30. https://doi.org/10.47532/jiv.v2i2.86

H. Sabita, Fitria, and R. H. (2021). Analisa dan Prediksi Iklan Lowongan Kerja Palsu Dengan Metode Natural Language Programming dan Machine Learning. *Jurnal Informatika*, *21*, *n*, 14.

Indonesia, B. (2023). *Polusi udara: Mengapa Jakarta disebut 'sudah kiamat' dan apa solusi agar kualitas udara membaik?* https://www.bbc.com/indonesia/indonesia-66514776

Irkham Widhi Saputro, B. W. S. (2019). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Citec Journal*, *6 No. 1*(2460–4259).

Jain, K. (2021). *Why use Naïve Bayes?* https://medium.com/analytics-vidhya/why-use-naive-bayes-a56cbae55181

Kepala Bapedal. (1997). *Pedoman Teknis Perhitungan dan Pelaporan serta Informasi Indeks Standar Pencemar Udara* https://123dok.com/document/q5w50rwq-pedoman-teknis-perhitungan-pelaporan-informasi-indeks-standar-pencemar.html#:~:text=Pedoman Teknis Perhitungan dan Pelaporan serta Informasi Indeks,Bupati%2FWalikotamadya kepala daerah tingkat II terkait%3B Pasal 2

Kirono, A. A. H., Asror, I., & ... (2022). Klasifikasi Tingkat Kualitas Udara Dki Jakara Dengan Algoritma Naïve Bayes. *EProceedings …*, *Vol. 9*, *No*(3), 1962–1969. https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/18002%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/18002/17631

Nugroho, K. S. (2019). *Confusion Matrix untuk Evaluasi Model pada Supervised Learning* https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine-learning-bc4b1ae9ae3f

S. Turgut, M. Dagtekin, and T. E. (2018). *Microarray Breast Cancer Data Classification Using Machine Learning Methods*.