TUGAS 1
DATA PREPARATION DARI SUMBER OPEN SOURCE

disusun untuk
memenuhi tugas mata kuliah
Pembelajaran Mesin

Oleh:

Kelompok VII

Berliani Utami	(2108107010082)
Della Rahmatika	(2108107010041)
Rizky Yusmansyah	(2208107010024)
Nazwa Salsabila	(2108107010010)
Zuwi Pertiwi	(2208107010061)

JURUSAN INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH
2025

Deskripsi Tugas

Pada tugas ini, setiap kelompok akan bekerja dengan dataset dari sumber open source seperti Kaggle atau Hugging Face. Tujuan utama dari tugas ini adalah untuk memahami proses persiapan data sebelum digunakan dalam analisis atau pelatihan model machine learning.

A. Data Description

Dataset yang kelompok kami pilih yaitu "Air Quality and Pollution Assessment", yang bersumber dari Kaggle, diunggah oleh Mujtaba Matin dalam format CSV. Dataset ini berisi informasi tentang kualitas udara dan faktor-faktor yang mempengaruhi tingkat polusi di berbagai daerah. Data ini mencakup parameter lingkungan seperti suhu, kelembaban, dan konsentrasi polutan (PM2.5, PM10, NO2, SO2, CO), serta faktor demografi seperti kepadatan populasi dan kedekatan dengan kawasan industri. Tujuan dataset ini adalah untuk menilai dan memprediksi kualitas udara berdasarkan faktor-faktor tersebut. Dataset ini memiliki 5.000 sampel data dengan 10 kolom. Dimana terdapat 9 fitur dan 1 label bernama Air Quality yang menjadi variabel target untuk analisis klasifikasi.

B. Data Loading

Untuk membantu dalam memahami data, ada beberapa library yang digunakan pada tahapan awal eksplorasi data yaitu mengimport pandas, numpy, matplotlib dan seaborn. Dataset kemudian dimuat ke dalam lingkungan pemrograman melalui URL dataset yang sudah di upload ke repository GitHub.

```
# URL dataset
url = "https://raw.githubusercontent.com/rizkyyus/Kelompok_7_Tugas01_Data_Preparation/refs/heads/main/updated_pollution_dataset.csv"

# Membaca dataset langsung dari URL
df = pd.read_csv(url)
```

Gambar 1 Menambahkan dataset melalui URL ke dalam lingkungan pemrograman

C. Data Understanding

Pemahaman awal pada dataset adalah dengan menampilkan statistik dasar dataset serta menampilkan visualisasi sederhana dari dataset tersebut.

1. Menampilkan Struktur Dataset

a. Informasi umum dataset

Dataset terdiri dari 10 kolom dengan berbagai jenis data. Data yang ditampilkan mencakup informasi seperti *Temperature*, *Humidity*, *PM2.5*, *PM10*, *NO2*, *SO2*, *CO*, *Proximity_to_Industrial_Areas*, *Population_Density*, dan *Air Quality*. Dataset memiliki 5000 baris dan 10 kolom, menunjukkan ukuran data yang cukup besar untuk analisis lebih lanjut.

```
[ ] # Menampilkan data 5 baris pertama
    df.head()
```

	Temperature	Humidity	PM2.5	PM10	NO2	SO2	CO	Proximity_to_Industrial_Areas	Population_Density	Air Quality
0	29.8	59.1	5.2	17.9	18.9	9.2	1.72	6.3	319	Moderate
1	28.3	75.6	2.3	12.2	30.8	9.7	1.64	6.0	611	Moderate
2	23.1	74.7	26.7	33.8	24.4	12.6	1.63	5.2	619	Moderate
3	27.1	39.1	6.1	6.3	13.5	5.3	1.15	11.1	551	Good
4	26.5	70.7	6.9	16.0	21.9	5.6	1.01	12.7	303	Good

```
[ ] # Menampilkan jumlah baris dan kolom
    df.shape
```

```
(5000, 10)
```

Gambar 2 Informasi umum dataset

b. Informasi struktur dataset

Setiap kolom ditampilkan dengan jumlah nilai non-null dan tipe datanya. Tipe data pada dataset ini terdiri dari float64 (nilai numerik desimal), int64 (bilangan bulat), dan object (kategori teks untuk *Air Quality*). Semua kolom memiliki 5000 non-null values, artinya tidak ada data yang hilang (missing values).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 10 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Temperature                   5000 non-null   float64
 1   Humidity                      5000 non-null   float64
 2   PM2.5                         5000 non-null   float64
 3   PM10                          5000 non-null   float64
 4   NO2                           5000 non-null   float64
 5   SO2                           5000 non-null   float64
 6   CO                            5000 non-null   float64
 7   Proximity_to_Industrial_Areas 5000 non-null   float64
 8   Population_Density            5000 non-null   int64
 9   Air Quality                   5000 non-null   object
dtypes: float64(8), int64(1), object(1)
memory usage: 390.8+ KB
```

Gambar 2 Informasi tentang kolom, tipe data dan jumlah nilai non-null

c. Statistik deskriptif

Dataset ini menunjukkan adanya variasi signifikan dalam faktor lingkungan, termasuk suhu, kelembaban, dan polusi udara. Sebagian besar wilayah memiliki kualitas udara yang baik, tetapi ada beberapa area dengan tingkat polusi yang tinggi, terutama terkait dengan PM2.5 dan PM10. Faktor seperti kedekatan dengan kawasan industri dan kepadatan penduduk mungkin berkontribusi terhadap perbedaan kualitas udara di berbagai wilayah.

```
# Statistik deskriptif
print(df.describe(include="all"))
```

```
        Temperature     Humidity        PM2.5         PM10          NO2  \
count   5000.000000  5000.000000  5000.000000  5000.000000  5000.000000
unique          NaN          NaN          NaN          NaN          NaN
top             NaN          NaN          NaN          NaN          NaN
freq            NaN          NaN          NaN          NaN          NaN
mean      30.029020    70.056120    20.142140    30.218360    26.412100
std        6.720661    15.863577    24.554546    27.349199     8.895356
min       13.400000    36.000000     0.000000    -0.200000     7.400000
25%       25.100000    58.300000     4.600000    12.300000    20.100000
50%       29.000000    69.800000    12.000000    21.700000    25.300000
75%       34.000000    80.300000    26.100000    38.100000    31.900000
max       58.600000   128.100000   295.000000   315.800000    64.900000

                SO2           CO  Proximity_to_Industrial_Areas  \
count   5000.000000  5000.000000                    5000.000000
unique          NaN          NaN                            NaN
top             NaN          NaN                            NaN
freq            NaN          NaN                            NaN
mean      10.014820     1.500354                       8.425400
std        6.750303     0.546027                       3.610944
min       -6.200000     0.650000                       2.500000
25%        5.100000     1.030000                       5.400000
50%        8.000000     1.410000                       7.900000
75%       13.725000     1.840000                      11.100000
max       44.900000     3.720000                      25.800000

        Population_Density Air Quality
count          5000.000000        5000
unique                 NaN           4
top                    NaN        Good
freq                   NaN        2000
mean            497.423800         NaN
std             152.754084         NaN
min             188.000000         NaN
25%             381.000000         NaN
50%             494.000000         NaN
75%             600.000000         NaN
max             957.000000         NaN
```

Gambar 3 Statistik deskriptif

d. Mengecek missing value

Dataset tidak memiliki missing values, sehingga tidak perlu dilakukan imputasi atau penanganan khusus terhadap data yang hilang.

```
# Mengecek missing values
print(df.isnull().sum())
```

```
Temperature                    0
Humidity                       0
PM2.5                          0
PM10                           0
NO2                            0
SO2                            0
CO                             0
Proximity_to_Industrial_Areas  0
Population_Density             0
Air Quality                    0
dtype: int64
```

Gambar 4 Informasi missing value

e. Menentukan batas outlier

Data menunjukkan bahwa beberapa fitur memiliki jumlah outlier yang cukup banyak. Banyaknya outlier dalam fitur seperti PM2.5 dan PM10 menunjukkan bahwa nilai polutan ini memiliki variabilitas yang tinggi.

```
Jumlah outlier pada setiap fitur numerik:
Temperature                    72
Humidity                       19
PM2.5                         352
PM10                          324
NO2                            73
SO2                           124
CO                             45
Proximity_to_Industrial_Areas  16
Population_Density              7
dtype: int64
```

Gambar 5 Menampilkan jumlah outlier di setiap fitur numerik

f. Mengecek jumlah duplikasi

Dataset tidak memiliki data duplikat, sehingga tidak perlu dilakukan proses penghapusan duplikasi.

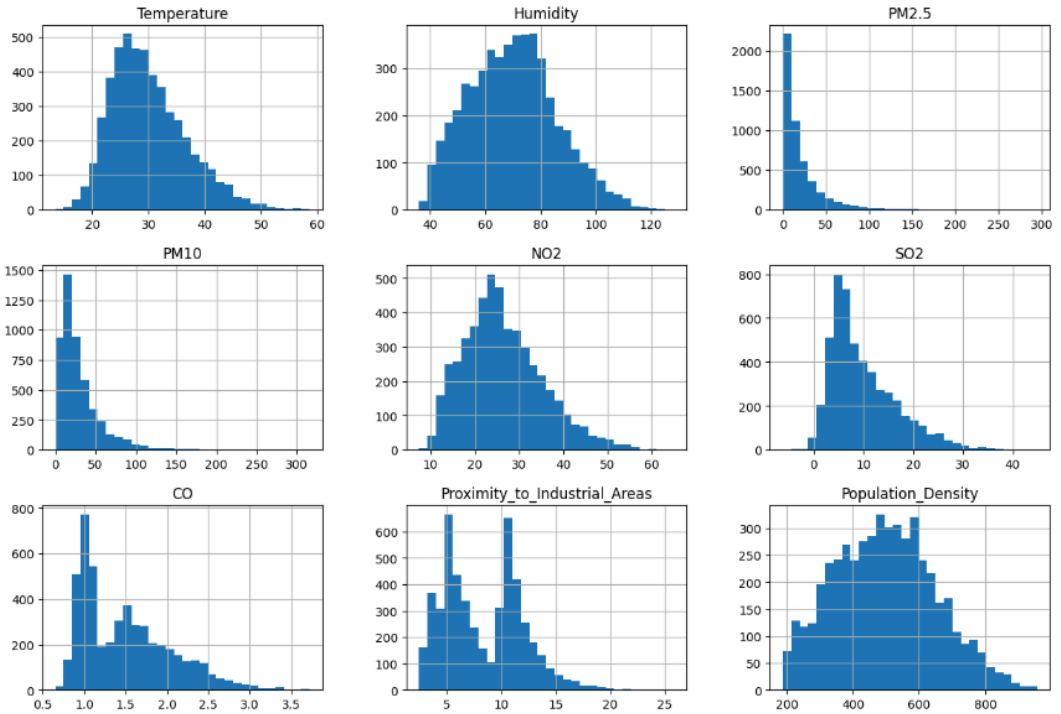
```
# Mengecek jumlah duplikasi
print(df.duplicated().sum())
```

```
0
```

Gambar 6 Informasi jumlah duplikasi

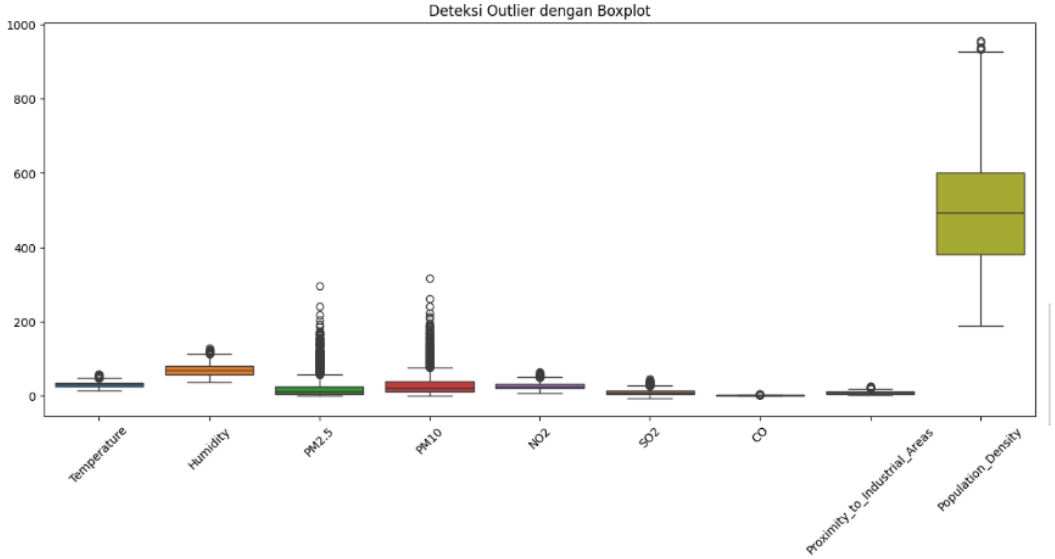
2. Menampilkan Visualisasi Dataset

a. Histogram



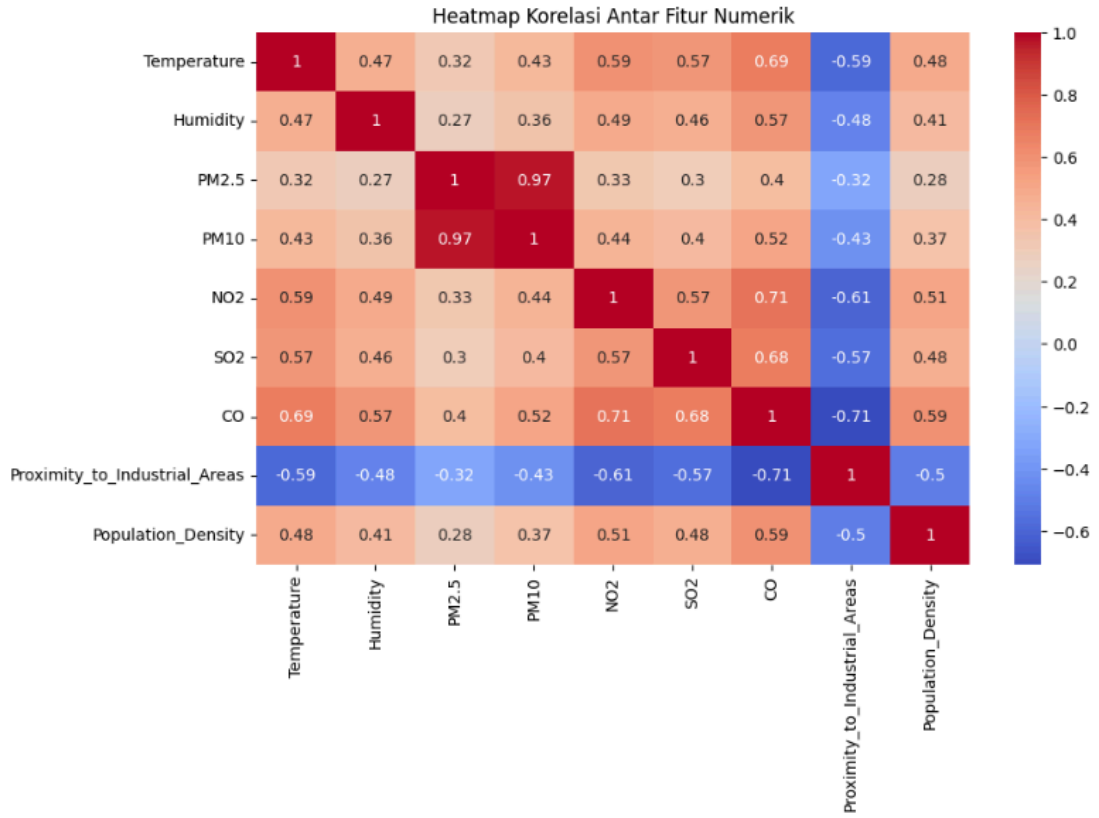
Gambar 7 Histogram untuk fitur numerik

b. Boxplot



Gambar 8 Boxplot untuk mendeteksi outlier

c. Heatmap



Gambar 9 Heatmap korelasi antar fitur numerik

Hasil visualisasi boxplot menunjukkan adanya outlier yang signifikan pada beberapa fitur polusi udara, terutama PM2.5, PM10, NO2, dan CO, yang memiliki banyak titik data jauh di atas batas normal. Hal ini mengindikasikan adanya kondisi ekstrem atau pencatatan data yang perlu diteliti lebih lanjut.

Selain itu, heatmap korelasi menunjukkan bahwa PM2.5 dan PM10 memiliki korelasi sangat tinggi (0.97), yang mengindikasikan bahwa keduanya sering muncul bersama dan mungkin redundant dalam model analisis. Fitur Proximity to Industrial Areas memiliki korelasi negatif dengan polutan, menunjukkan bahwa semakin dekat ke area industri, semakin tinggi tingkat polusi yang tercatat.

D. Data Preparation

Preprocessing bertujuan untuk memastikan data siap digunakan dalam analisis atau model machine learning. Berikut adalah langkah-langkah yang dilakukan:

1. Melakukan Fitur Encoding

Dilakukan encoding terhadap fitur Air Quality, yang awalnya berupa data kategorikal (seperti *Good*, *Moderate*, *Poor*, *Hazardous*) menjadi bentuk numerik menggunakan LabelEncoder().

```
# Mengonversi kategori Air Quality menjadi angka
encoder = LabelEncoder()
df['Air Quality'] = encoder.fit_transform(df['Air Quality'])
df['Air Quality'].head()
```

Gambar 10 encoding terhadap fitur Air Quality

Kegunaan encoding yaitu untuk mempermudah pemrosesan data dalam model Machine Learning yang hanya menerima input numerik, menghilangkan bias urutan dalam data kategorikal dibandingkan dengan metode one-hot encoding dan mengurangi dimensi dataset jika dibandingkan dengan metode encoding lainnya seperti one-hot encoding.

	Air Quality
0	2
1	2
2	2
3	0
4	0

dtype: int64

Gambar 11 Output encoding terhadap fitur Air Quality

2. Menangani Nilai Outlier

Outlier adalah nilai yang jauh dari distribusi mayoritas data, yang dapat mempengaruhi performa model machine learning. Oleh karena itu, metode capping digunakan untuk mengganti outlier dengan batas bawah atau atas yang wajar.

```
[ ] # Menangani Outlier dengan Capping
    for col in df.columns[:-1]:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df[col] = df[col].clip(lower=lower_bound, upper=upper_bound)
```

Gambar 12 Menangani nilai outlier

Dengan teknik capping berbasis IQR, nilai ekstrim telah dibatasi dalam rentang yang wajar, sehingga kualitas data lebih baik tanpa kehilangan informasi penting.

```
Jumlah outlier pada kolom Temperature: 0
Jumlah outlier pada kolom Humidity: 0
Jumlah outlier pada kolom PM2.5: 0
Jumlah outlier pada kolom PM10: 0
Jumlah outlier pada kolom NO2: 0
Jumlah outlier pada kolom SO2: 0
Jumlah outlier pada kolom CO: 0
Jumlah outlier pada kolom Proximity_to_Industrial_Areas: 0
Jumlah outlier pada kolom Population_Density: 0
Jumlah outlier pada kolom Air Quality: 0
```

Gambar 13 Output penanganan outlier

3. Menangani Nilai Negatif

Dalam konteks data kualitas udara, nilai negatif tidak memiliki makna yang valid. Misalnya, PM10, SO2, dan CO adalah konsentrasi polutan di udara yang seharusnya tidak mungkin bernilai negatif. Jika dibiarkan, nilai negatif bisa menyebabkan Kesalahan dalam analisis dan Ketidaksesuaian dengan model yang tidak bisa menangani nilai negatif.

```
[ ]  # Mengubah nilai negatif menjadi 0
     df['PM10'] = df['PM10'].apply(lambda x: max(x, 0))
     df['SO2'] = df['SO2'].apply(lambda x: max(x, 0))
     df['CO'] = df['CO'].apply(lambda x: max(x, 0))
```

Gambar 14 Menangani nilai negatif

Kode ini menggunakan fungsi apply() dengan lambda function untuk mengganti semua nilai negatif dengan nol. Teknik ini membantu mencegah error dalam analisis lebih lanjut dan memastikan data tetap masuk akal secara ilmiah.

4. Normalisasi Data

Dalam proses ini, metode MinMaxScaler digunakan untuk mengubah nilai setiap fitur ke dalam rentang 0 hingga 1. Teknik ini bekerja dengan cara menghitung nilai minimum dan maksimum dari masing-masing fitur.

```
[ ]  # Pilih kolom polutan yang ingin dinormalisasi
     polutan_cols = ['PM2.5', 'PM10', 'NO2', 'SO2', 'CO']

     # Inisialisasi MinMaxScaler
     scaler = MinMaxScaler()

     # Fit dan transform data
     df[polutan_cols] = scaler.fit_transform(df[polutan_cols])
```

Gambar 15 Melakukan Normalisasi Data

Hasil dari normalisasi ini adalah setiap nilai dalam dataset akan berada dalam rentang 0 hingga 1, yang membuat semua fitur memiliki skala yang seimbang.

5. Feature Selection

Tujuan dari feature selection adalah mengurangi dimensi data, sehingga model lebih ringan dan cepat. Meningkatkan akurasi hanya dengan menggunakan fitur yang relevan. Mencegah overfitting agar model tidak terlalu menyesuaikan diri dengan data latih.

```
[ ]  from sklearn.feature_selection import SelectKBest, f_classif

     X = df.drop(columns=['Air Quality'])  # Semua fitur kecuali label
     y = df['Air Quality']  # Label kualitas udara

     selector = SelectKBest(score_func=f_classif, k=5)
     X_new = selector.fit_transform(X, y)
```

Gambar 16 Melakukan feature selection

Dengan menggunakan metode ini, hanya fitur yang paling berpengaruh terhadap label Air Quality yang dipertahankan, sehingga model yang dibangun lebih sederhana, lebih cepat dalam pelatihan, dan lebih akurat dalam prediksi.

Kesimpulan

Proses data preparation yang dilakukan pada dataset "Air Quality and Pollution Assessment" telah melalui beberapa tahap penting untuk memastikan kualitas data yang optimal sebelum analisis lebih lanjut. Tahap pertama adalah pembersihan data, di mana tidak ditemukan missing values maupun duplikasi, sehingga tidak diperlukan proses imputasi atau penghapusan data. Selanjutnya, analisis awal menunjukkan bahwa dataset terdiri dari 5000 sampel dengan 10 fitur yang mencakup parameter lingkungan serta faktor demografi.

Dari hasil eksplorasi, ditemukan adanya outlier pada beberapa fitur seperti PM2.5, PM10, NO2, dan CO, yang kemudian ditangani menggunakan metode capping berbasis

IQR untuk menjaga kualitas data. Pada tahap transformasi, fitur kategorikal Air Quality dikonversi ke dalam bentuk numerik menggunakan Label Encoding agar dapat digunakan dalam model Machine Learning. Selain itu, analisis korelasi menunjukkan hubungan yang kuat antara PM2.5 dan PM10, serta indikasi bahwa kedekatan dengan kawasan industri berpengaruh terhadap tingkat polusi udara. Dengan berbagai tahapan ini, dataset telah diproses dengan baik sehingga siap digunakan untuk analisis lebih lanjut atau pengembangan model prediksi kualitas udara.