

TUGAS 1 DATA PREPARATION

Disusun untuk memenuhi

Tugas Mata Kuliah Pembelajaran Mesin

Oleh :

Iffatun Nisa Nasrullah	(2208107010009)
Muhammad Bintang Indra Hidayat	(2208107010023)
Qandila Ahmara	(2208107010039)
Alhusna Hanifah	(2208107010060)
Farhanul Khair	(2208107010076)



JURUSAN INFORMATIKA

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SYIAH KUALA
DARUSSALAM, BANDA ACEH**

2025

1. Data Description

Nama Dataset: 2024_Property_Tax_Roll

Sumber:

<https://www.kaggle.com/datasets/aniket0712/2024-property-tax-roll/data>

Deskripsi singkat:

Dataset "**2024_Property_Tax_Roll**" adalah kumpulan data yang berisi informasi terperinci tentang penilaian properti dan perpajakan. Dataset ini mencakup berbagai atribut yang mendukung analisis properti, seperti identifikasi properti, peta pajak, klasifikasi, deskripsi properti, kode retribusi, serta alamat sipil.

Dataset ini mengkategorikan properti berdasarkan penggunaannya, termasuk:

1. Rumah Tunggal (Single-Family Residential)
2. Rumah Banyak (Multi-Family Residential)
3. Komersial (Commercial)
4. Industri (Industrial)
5. Properti yang Dikecualikan dari Pajak (Exempted Properties)

Selain itu, dataset ini juga menyediakan informasi mengenai nilai retribusi properti, besaran pajak yang dikenakan, dan klasifikasi properti untuk tujuan perpajakan serta penilaian.

Jumlah Data:

- Total Entri: 44.034 properti
- Total Atribut: 30 kolom

Dataset ini berisi informasi mengenai properti, termasuk identifikasi, klasifikasi, alamat, pemilik, serta detail pajak.

- Identifikasi Properti: Setiap properti memiliki P_ID unik, serta atribut seperti TAX_MAP, plat, lot, dan unit untuk mengidentifikasi lokasi dalam sistem perpajakan.
- Klasifikasi Properti: Properti dikategorikan berdasarkan CLASS (kode jenis penggunaan) dan SHORT_DESC (deskripsi singkat).
- Kode Pajak: LEVY_CODE_1 menunjukkan kode pajak yang berlaku, dengan deskripsi tambahan pada SHORT_DESC 1.
- Alamat Properti: Terdiri dari CIVIC (nomor rumah), STREET, SUFFIX, dan FORMATED_ADDRESS untuk alamat lengkap, serta informasi kota (CITY) dan kode pos (ZIP_POSTAL).
- Informasi Pemilik: Terdapat data nama pemilik (FIRST_NAME, LAST_NAME) atau perusahaan (COMPANY), namun banyak nilai yang kosong.
- Alamat Alternatif: Beberapa properti memiliki alamat tambahan pada FREE_LINE_2, CIVIC 1, STREET 1, S_SUFFIX, CITY 1, STATE, dan ZIP_POSTAL 1.
- Informasi Pajak: Properti memiliki TOTAL_ASSMT (nilai penilaian pajak), TOTAL_EXEMPT (pengecualian pajak), dan TOTAL_TAXES (total pajak yang dikenakan).
- Lokasi Properti: Informasi lokasi dalam bentuk teks pada Property_Location

2. Data Loading

Memuat dataset ke dalam lingkungan pemrograman Python dapat dilakukan menggunakan berbagai library seperti Pandas, NumPy, atau library lain yang relevan tergantung pada format data. Berikut adalah penjelasan dan contoh kode.

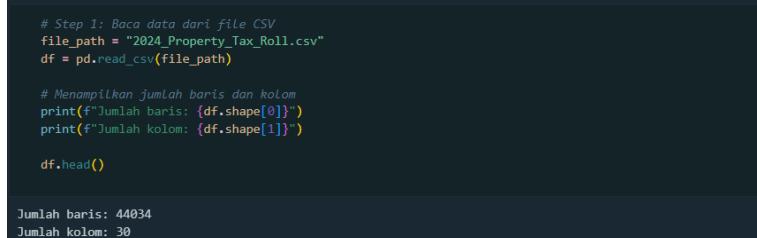
sebelumnya kita harus mengimport terlebih dahulu library - library yang akan akan kita gunakan



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import folium
from folium.plugins import MarkerCluster
import ipywidgets as widgets
from IPython.display import display, clear_output
import threading
import numpy as np
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

Cara Memuat Dataset di Python

Pandas merupakan library yang paling umum digunakan untuk memuat dataset, terutama dalam format **CSV**, **Excel**, **JSON**, atau **SQL**. Berikut adalah menunjukkan bagaimana cara memuat dataset dari file CSV dan melakukan eksplorasi awal:



```
# Step 1: Baca data dari file CSV
file_path = "2024_Property_Tax_Roll.csv"
df = pd.read_csv(file_path)

# Menampilkan jumlah baris dan kolom
print(f"Jumlah baris: {df.shape[0]}")
print(f"Jumlah kolom: {df.shape[1]}")

df.head()

Jumlah baris: 44034
Jumlah kolom: 30
```

penjelasan:

- file_path = "2024_Property_Tax_Roll.csv" Variabel file_path menyimpan nama atau lokasi file CSV yang akan dibaca. Dalam hal ini, file yang dimuat bernama "2024_Property_Tax_Roll.csv".
- pd.read_csv(file_path) adalah fungsi dari **Pandas** yang digunakan untuk membaca file CSV dan mengubahnya menjadi **DataFrame**, yaitu struktur data berbentuk tabel di Python.
- Hasil dari pembacaan ini disimpan dalam variabel df, yang akan digunakan untuk analisis lebih lanjut.
- print(f"Jumlah baris: {df.shape[0]}") .Menggunakan **f-string** untuk mencetak jumlah baris dalam format yang lebih mudah dibaca.
- print(f"Jumlah kolom: {df.shape[1]}"). Sama seperti sebelumnya, tetapi untuk jumlah kolom.

3. Data Understanding

1. Statistik Dasar Dataset

Dalam proses eksplorasi awal, kita perlu memahami struktur dan karakteristik dataset. Beberapa langkah yang dilakukan meliputi:

1. Menampilkan Jumlah Missing Value

Kode berikut digunakan untuk melihat jumlah nilai yang hilang di setiap kolom:

```
# Menampilkan jumlah missing values per kolom
df.isnull().sum()
```

output:

```
P_ID           1
TAX_MAP        1
plat          1
lot            1
unit           1
CLASS          1
SHORT_DESC     1
LEVY_CODE_1    1
SHORT_DESC_1   1
CIVIC          96
STREET         3
SUFFIX         1592
FORMATED_ADDRESS 3
CITY           35
ZIP_POSTAL     1436
FIRST_NAME     9738
LAST_NAME      9674
COMPANY        34902
FREE_LINE_2    3
CIVIC_1        1596
STREET_1       1058
S_SUFFIX       3523
UNIT           39476
CITY_1          7
STATE          14
...
TOTAL_ASSMT    1
TOTAL_EXEMPT   1
TOTAL_TAXES    1
Property_Location 0
dtype: int64
```

Data yang hilang dapat berdampak pada analisis, sehingga perlu ditangani dengan metode tertentu seperti penghapusan atau imputasi.

2. Mengecek Data Duplikat

Kode berikut digunakan untuk mengecek jumlah data yang duplikat:

```
# Mengecek jumlah data duplikat
df.duplicated().sum()
```

Dataset ini tidak memiliki data duplikat (jumlah duplikat = 0), sehingga tidak perlu dilakukan pembersihan duplikasi.

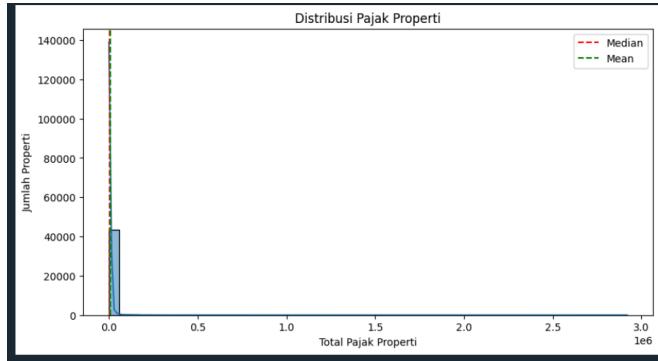
2. Visualisasi Data

Visualisasi digunakan untuk memahami distribusi data, mendeteksi anomali, dan mengevaluasi hubungan antar variabel.

1. Distribusi Pajak Properti

Histrogram digunakan untuk melihat distribusi kolom TOTAL_TAXES:

```
# Cek distribusi pajak properti
plt.figure(figsize=(10,5))
sns.histplot(df["TOTAL_TAXES"], bins=50, kde=True)
plt.axvline(df["TOTAL_TAXES"].median(), color='r', linestyle='dashed', label="Median")
plt.axvline(df["TOTAL_TAXES"].mean(), color='g', linestyle='dashed', label="Mean")
plt.xlabel("Total Pajak Properti")
plt.ylabel("Jumlah Properti")
plt.title("Distribusi Pajak Properti")
plt.legend()
plt.show()
```



- Distribusi data pajak properti sangat **condong ke kanan**, menunjukkan bahwa sebagian besar properti memiliki pajak yang relatif kecil, tetapi ada beberapa properti dengan pajak sangat tinggi.
- Rata-rata (mean) lebih besar dari median, menunjukkan adanya outlier dalam dataset.

2. Korelasi Antar Fitur

Menganalisis korelasi antara fitur numerik menggunakan heatmap:

```
# Memilih hanya kolom numerik untuk analisis korelasi
numerical_cols = df.select_dtypes(include='number')

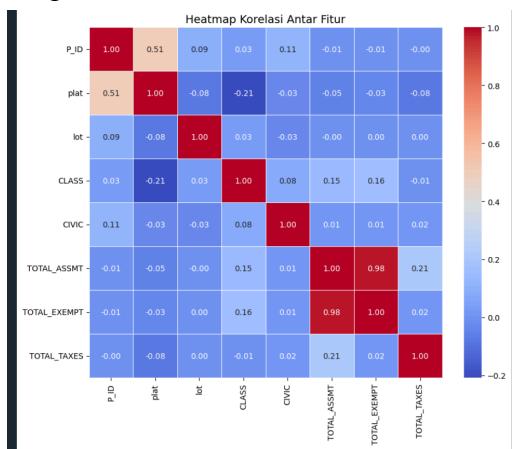
# Menghitung matriks korelasi
correlation_matrix = numerical_cols.corr()

# Membuat heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", linewidths=0.5)

# Menambahkan judul
plt.title("Heatmap Korelasi Antar Fitur", fontsize=14)

# Menampilkan plot
plt.show()
```

Output:



- Korelasi antar beberapa fitur menunjukkan adanya kemungkinan redundansi atau hubungan linier.
- Pajak properti (TOTAL_TAXES) sangat dipengaruhi oleh TOTAL_ASSMT dan TOTAL_EXEMPT.
- Beberapa fitur seperti CLASS dan CIVIC memiliki korelasi sangat rendah dengan pajak, sehingga mungkin tidak terlalu penting dalam model prediksi pajak.

- Korelasi negatif kecil antara beberapa variabel menunjukkan bahwa mereka tidak saling berpengaruh signifikan.

3. Deteksi Outlier

Outlier dideteksi menggunakan metode Interquartile Range (IQR):

```
# Fungsi untuk mendeteksi outliers menggunakan IQR
def detect_outliers_iqr(df, col):
    Q1 = df[col].quantile(0.25) # Kuartil pertama
    Q3 = df[col].quantile(0.75) # Kuartil ketiga
    IQR = Q3 - Q1 # Interquartile Range
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
    return outliers

# Menampilkan jumlah outlier di setiap kolom numerik
for col in df.select_dtypes(include=['float64', 'int64']).columns:
    num_outliers = detect_outliers_iqr(df, col).shape[0]
    print(f"Jumlah outliers di kolom '{col}': {num_outliers}")
```

Output:

```
Jumlah outliers di kolom 'P_ID': 0
Jumlah outliers di kolom 'plat': 0
Jumlah outliers di kolom 'lot': 395
Jumlah outliers di kolom 'CLASS': 2589
Jumlah outliers di kolom 'CIVIC': 4581
Jumlah outliers di kolom 'TOTAL_ASSMT': 4124
Jumlah outliers di kolom 'TOTAL_EXEMPT': 7639
Jumlah outliers di kolom 'TOTAL_TAXES': 3597
```

Outlier ini bisa berdampak pada analisis statistik dan model prediktif, sehingga perlu dipertimbangkan apakah akan dihapus atau distandarisasi.

4. Analisis Outlier pada Fitur Pajak Properti

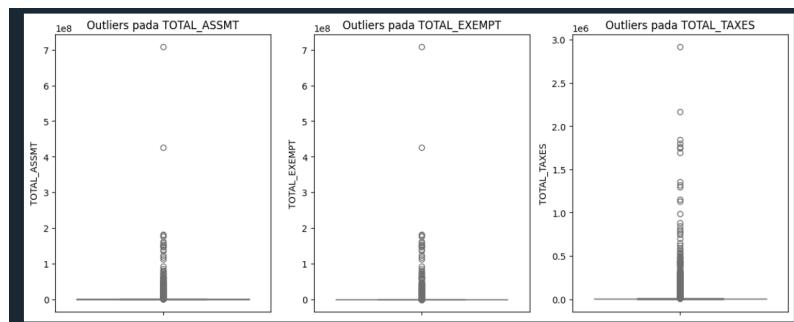
```
# Pilih hanya kolom yang diminta
selected_cols = ["TOTAL_ASSMT", "TOTAL_EXEMPT", "TOTAL_TAXES"]

plt.figure(figsize=(12, 5)) # Atur ukuran gambar

# Loop untuk membuat boxplot setiap kolom
for i, col in enumerate(selected_cols, 1):
    plt.subplot(1, 3, i) # 1 baris, 3 kolom
    sns.boxplot(y=df[col], color="skyblue")
    plt.title(f"Outliers pada {col}")

plt.tight_layout() # Agar tidak tumpang tindih
plt.show()
```

output:



Gambar di atas menampilkan **boxplot** untuk tiga fitur pajak properti.

Ada properti tertentu dengan penilaian pajak (TOTAL_ASSMT) dan pengecualian pajak (TOTAL_EXEMPT) yang sangat tinggi dibandingkan properti lainnya. Properti dengan pajak tinggi kemungkinan berasosiasi dengan

4. Data Preparation

1. Mengatasi Missing Value

Pada tahap data preparation, missing values diatasi dengan menghapus baris pada kolom penting, mengisi kolom kategorikal dengan 'Unknown', dan mengganti

nilai kosong pada kolom numerik dengan 0. Kolom UNIT dihapus karena memiliki lebih dari 90% missing values. Setelah itu, dilakukan pengecekan ulang untuk memastikan data bersih dan siap digunakan.

```
# 1. Drop baris jika kolom penting numerik (TOTAL_ASSMT, TOTAL_TAXES, dll.) kosong
df.dropna(subset=['P_ID', 'TAX_MAP', 'plat', 'lot', 'CLASS', 'TOTAL_ASSMT', 'TOTAL_EXEMPT', 'TOTAL_TAXES'], inplace=True)

# 2. Isi kolom katagorial dengan 'Unknown'
categorical_cols = ['SUFFIX', 'COMPANY', 'STATE', 'FIRST_NAME', 'LAST_NAME', 'CITY', 'CITY_1', 'STREET', 'STREET_1', 'FORMATED_ADDRESS', 'FREE_LINE_2']
df[categorical_cols] = df[categorical_cols].fillna('Unknown')

# 3. Isi dengan 0 untuk kolom berikut
fill_cols = ['CIVIC_1', 'CIVIC', 'ZIP_POSTAL', 'ZIP_POSTAL_1']
for col in fill_cols:
    df[col].fillna(0, inplace=True)

# 4. Menghapus kolom UNIT karena 90% missing values
df.drop(columns=['UNIT'], inplace=True)

# 5. Cek hasil setelah imputasi
print(df.isnull().sum())

```

	0
P_ID	0
TAX_MAP	0
plat	0
lot	0
UNIT	0
CLASS	0
SHORT_DESC	0
LEVY_CODE_1	0
SHORT_DESC_1	0
CIVIC	0
STREET	0
SUFFIX	0
FORMATED_ADDRESS	0
CITY	0
ZIP_POSTAL	0
FIRST_NAME	0
LAST_NAME	0
COMPANY	0
FREE_LINE_2	0
Others	0

2. Mengatasi Outlier

Pada tahap data preparation, outlier dihapus menggunakan metode Interquartile Range (IQR) pada kolom numerik tertentu. Nilai Q1 (kuartil 25%) dan Q3 (kuartil 75%) dihitung, lalu batas bawah dan atas ditentukan dengan rumus $Q1 - 1.5 * IQR$ dan $Q3 + 1.5 * IQR$. Data yang berada di luar batas ini dianggap outlier dan dihapus. Setelah proses ini, dataset berisi 31.605 baris dan 29 kolom.

```
# Daftar kolom yang akan diperiksa
cols = ['TOTAL_ASSMT', 'TOTAL_EXEMPT', 'TOTAL_TAXES']

# Menghapus outlier dengan IQR
for col in cols:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1

    # Menentukan batas bawah dan atas
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Filter hanya data dalam rentang normal
    df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

print(f'Data setelah menghapus outlier: {df.shape}')

Data setelah menghapus outlier: (31605, 29)
```

3. Standarisasi Data Katagori

Pada tahap data preparation, kolom SHORT_DESC dikonversi ke huruf kecil menggunakan `.str.lower()` untuk memastikan konsistensi format teks. Setelah konversi, ditampilkan 5 baris pertama dari kolom tersebut untuk memverifikasi perubahan.

```

# Mengubah teks menjadi huruf kecil agar konsisten
df["SHORT_DESC"] = df["SHORT_DESC"].str.lower()

# Menampilkan 5 baris pertama setelah konversi
print(df[["SHORT_DESC"]].head())

      SHORT_DESC
2   single family
4   single family
5    2 - 5 family
10   2 - 5 family
11   2 - 5 family

```

4. Memisahkan Longitude dan Latitude

Kode ini melakukan pembersihan nama kolom dari spasi ekstra, kemudian menampilkan beberapa contoh data untuk memastikan formatnya benar. Setelah itu, kode mengekstrak nilai Longitude dan Latitude dari format teks "POINT (Longitude Latitude)" menggunakan ekspresi reguler. Hasil ekstraksi ditampilkan untuk memastikan data telah diproses dengan benar.

```

# Membersihkan nama kolom dari spasi ekstra
df.columns = df.columns.str.strip()

# Menampilkan beberapa contoh data untuk memastikan formatnya benar
print("Contoh data Property_Location:")
print(df["Property_Location"].head())

# Mengekstrak Longitude dan Latitude dari format "POINT (longitude latitude)"
df[["Longitude", "Latitude"]] = df["Property_Location"].str.extract(r'POINT \((([-\d.]+) ([-\d.]+))\)')
print("\nData setelah ekstraksi Longitude dan Latitude:")
print(df[["Longitude", "Latitude"]].head())

Contoh data Property_Location:
2 POINT (-71.415484981 41.858035988)
4 POINT (-71.400355981 41.821832998)
5 POINT (-71.394502982 41.820420018)
10 POINT (-71.424641037 41.810907019)
11 POINT (-71.430050969 41.807336999)
Name: Property_Location, dtype: object

Data setelah ekstraksi longitude dan Latitude:
   Longitude      Latitude
2 -71.415484981  41.858035988
4 -71.400355981  41.821832998
5 -71.394502982  41.820420018
10 -71.424641037  41.810907019
11 -71.430050969  41.807336999

```

Data Visualisasi

