

# Data Preparation

## Kelompok 6

Iffatun Nisa Nasrullah (2208107010009)

Muhammad Bintang Indra Hidayat (2208107010023)

Qandila Ahmara (2208107010039)

Alhusna Hanifah (2208107010060)

Farhanul Khair (2208107010076)

# Data Description



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

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

# Data Loading

Proses memuat data

```
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
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

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

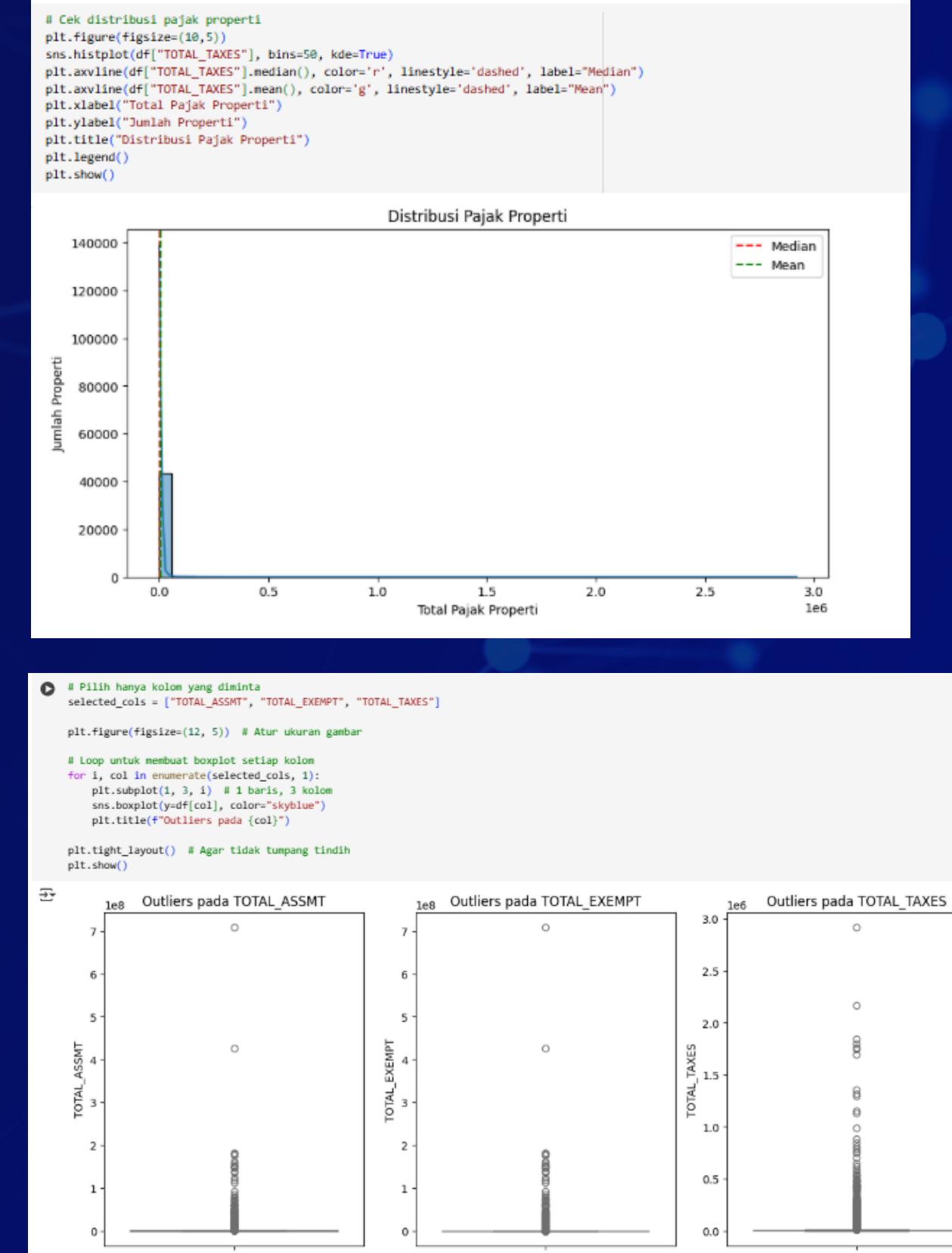
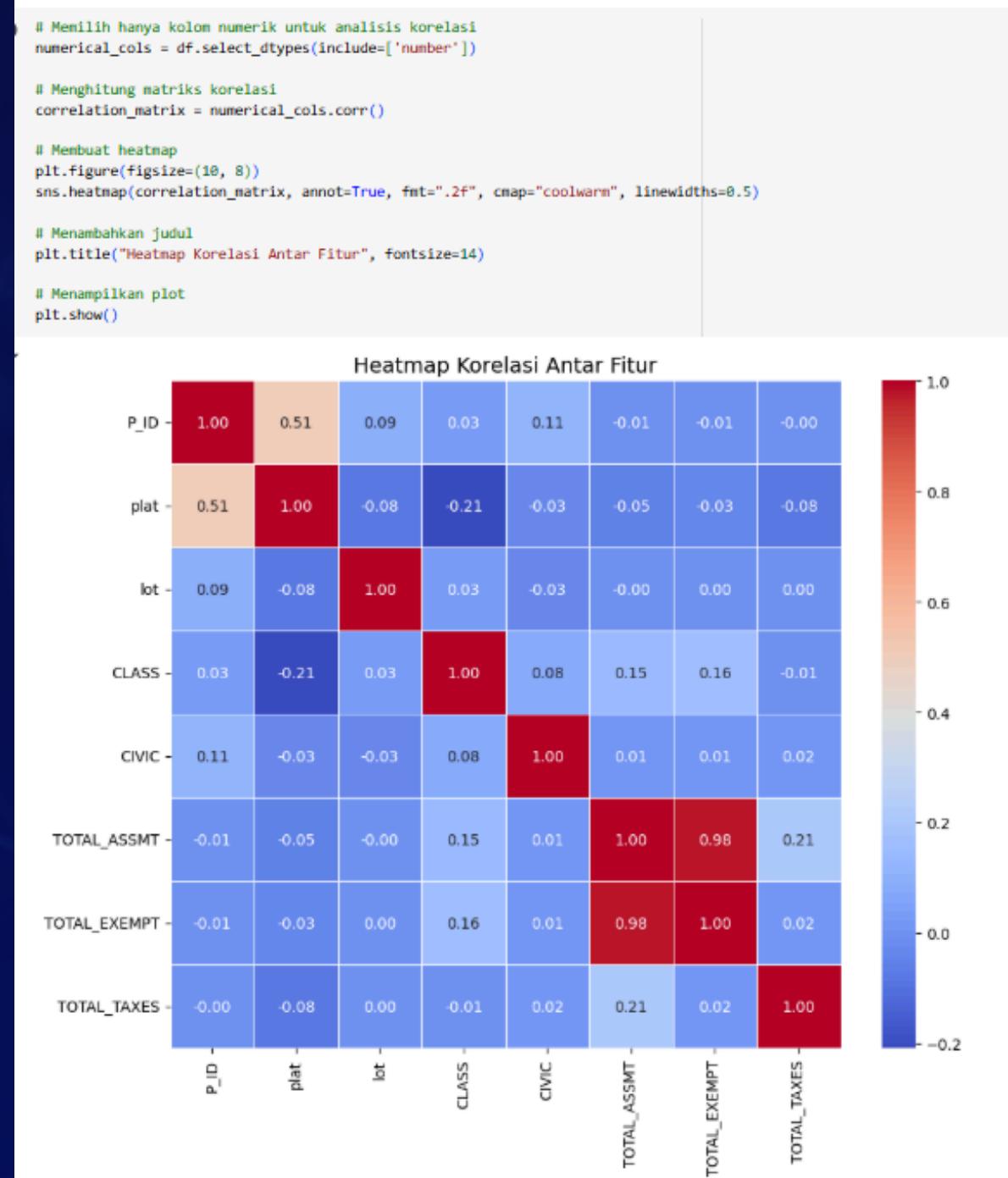
```
# Menampilkan jumlah baris dan kolom
print(f"Jumlah baris: {df.shape[0]}")
print(f"Jumlah kolom: {df.shape[1]}")
```

```
df.head()
```

```
Jumlah baris: 44034
Jumlah kolom: 30
```

# Data Understanding

# Menampilkan jumlah missing value
df.isnull().sum()
0
P_ID 1
TAX_MAP 1
plat 1
lot 1
unit 1
CLASS 1
SHORT_DESC 1
LEVY_CODE_1 1
SHORT_DESC1 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
ZIP_POSTAL 1 38
TOTAL_ASSMT 1
TOTAL_EXEMPT 1
TOTAL_TAXES 1
Property_Location 0



- Melakukan pengecekan missing value
- data duplikat
- distribusi data
- outlier

Terdapat beberapa missing values yang harus diselesaikan. Berdasarkan Distribusi data disamping, terlihat distribusi data miring ke kanan. Ini menunjukkan bahwa sebagian besar properti dikenakan pajak rendah, sementara hanya sedikit properti yang membayar pajak sangat tinggi.

# Data Preparation

## Mengatasi Missing Values

```
# 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 kategorikal dengan 'Unknown'  
categorical_cols = ['SUFFIX', 'S_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 inputasi  
print(df.isnull().sum())
```

```
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  
CIVIC 1        0  
STREET 1       0  
S_SUFFIX       0  
CITY 1         0  
STATE          0  
ZIP_POSTAL 1  0  
TOTAL_ASSMT   0  
TOTAL_EXEMPT  0  
TOTAL_TAXES   0  
Property_Location 0  
dtype: int64
```

## Mengatasi Outlier

```
# 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)

## Standarisasi Data Kategorikal

```
[ ] # 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
```

## Memisahkan Latitude dan Longitude

```
# 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 \((([-\d.]+) ([-\d.]+)\))')  
  
# Menampilkan hasil  
print("\nData setelah ekstraksi Longitude dan Latitude:")  
print(df[["Longitude", "Latitude"]].head())
```

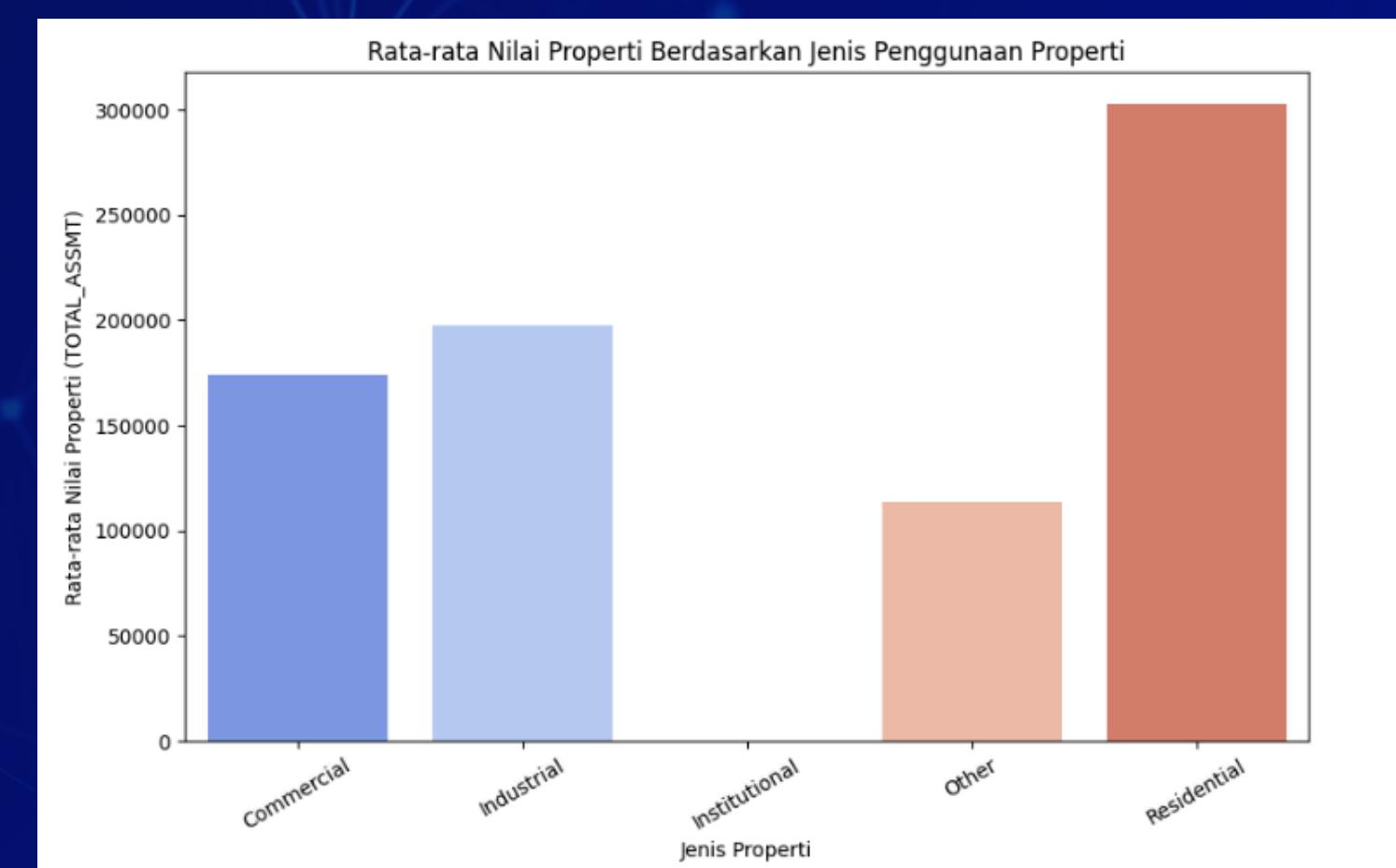
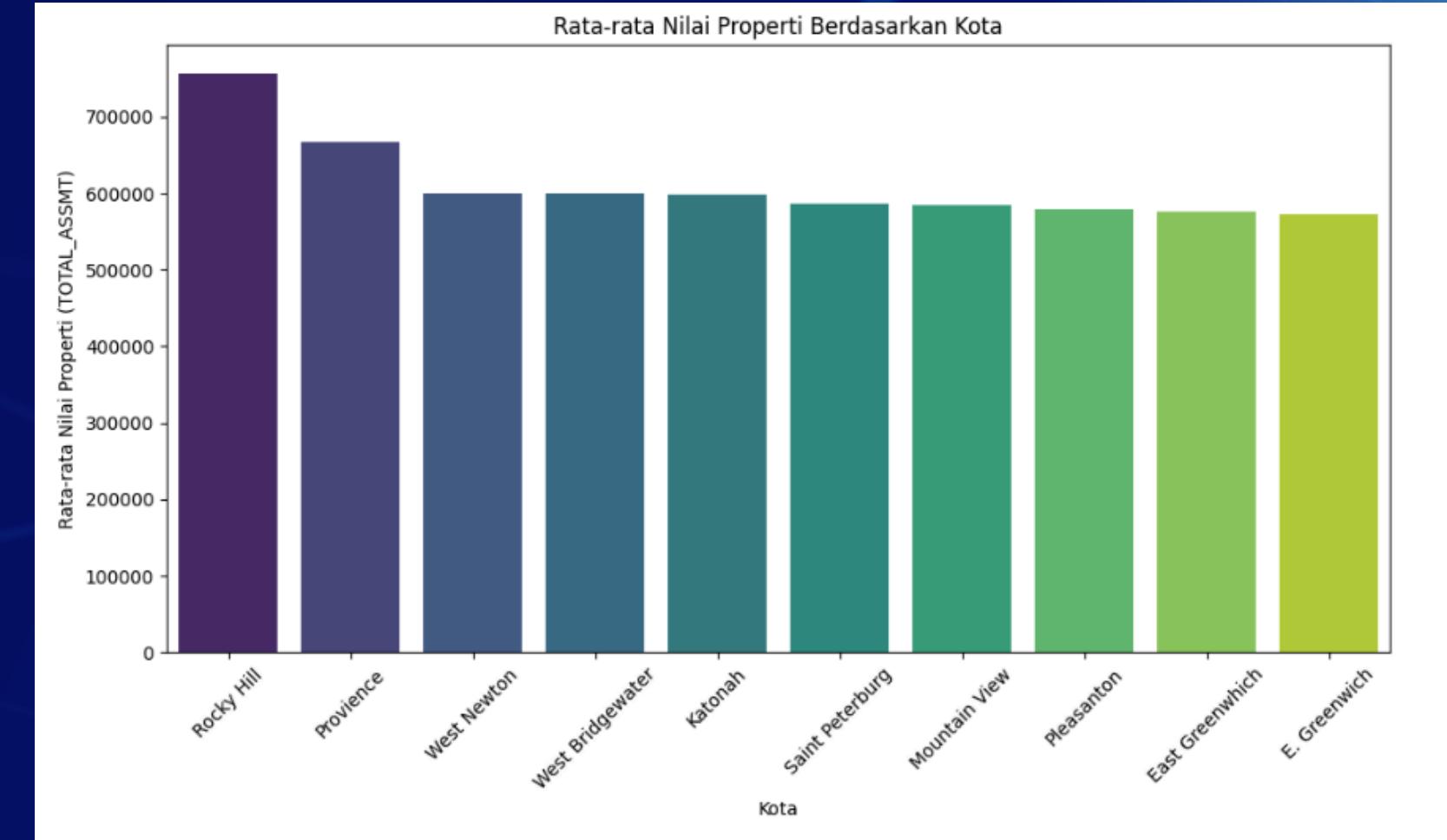
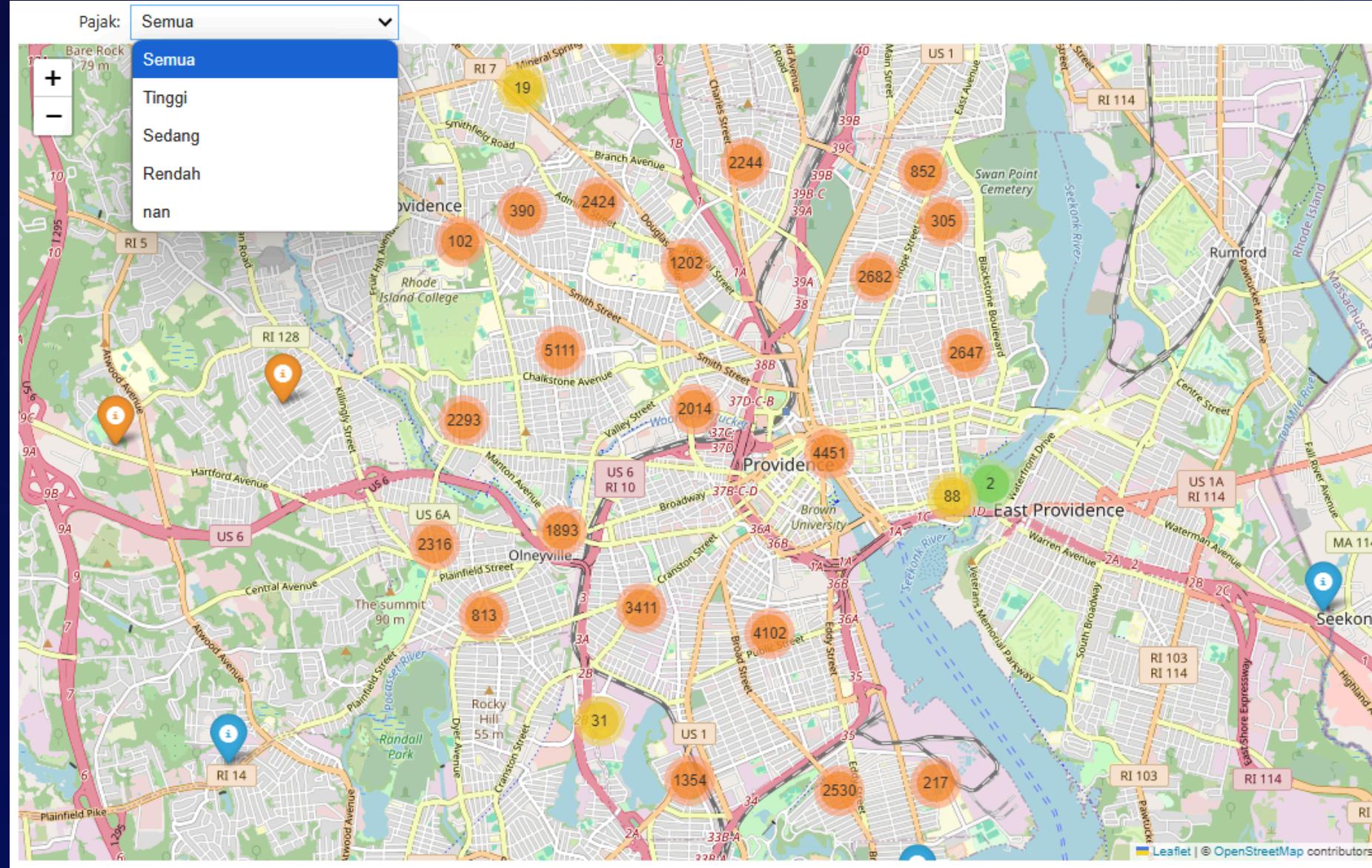
→ Contoh data Property\_Location:

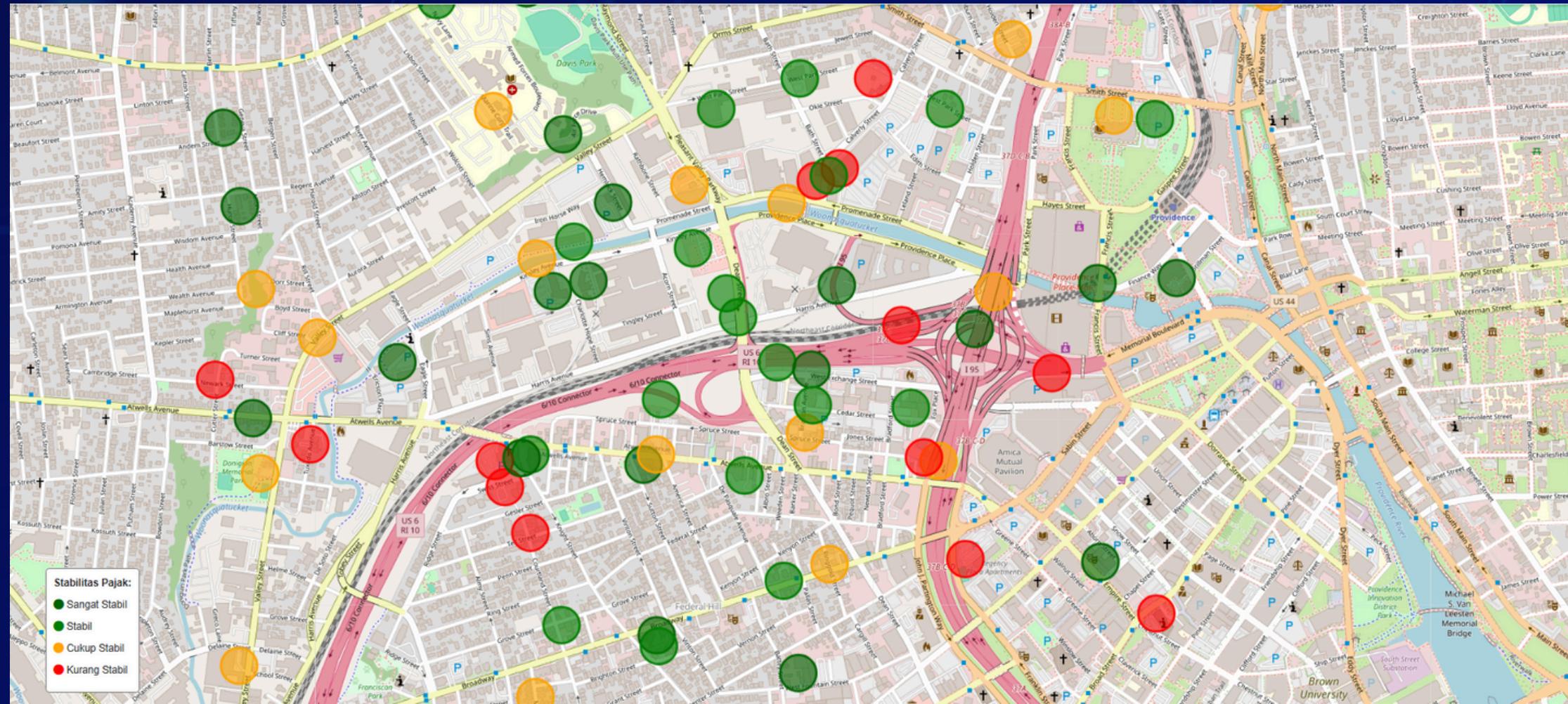
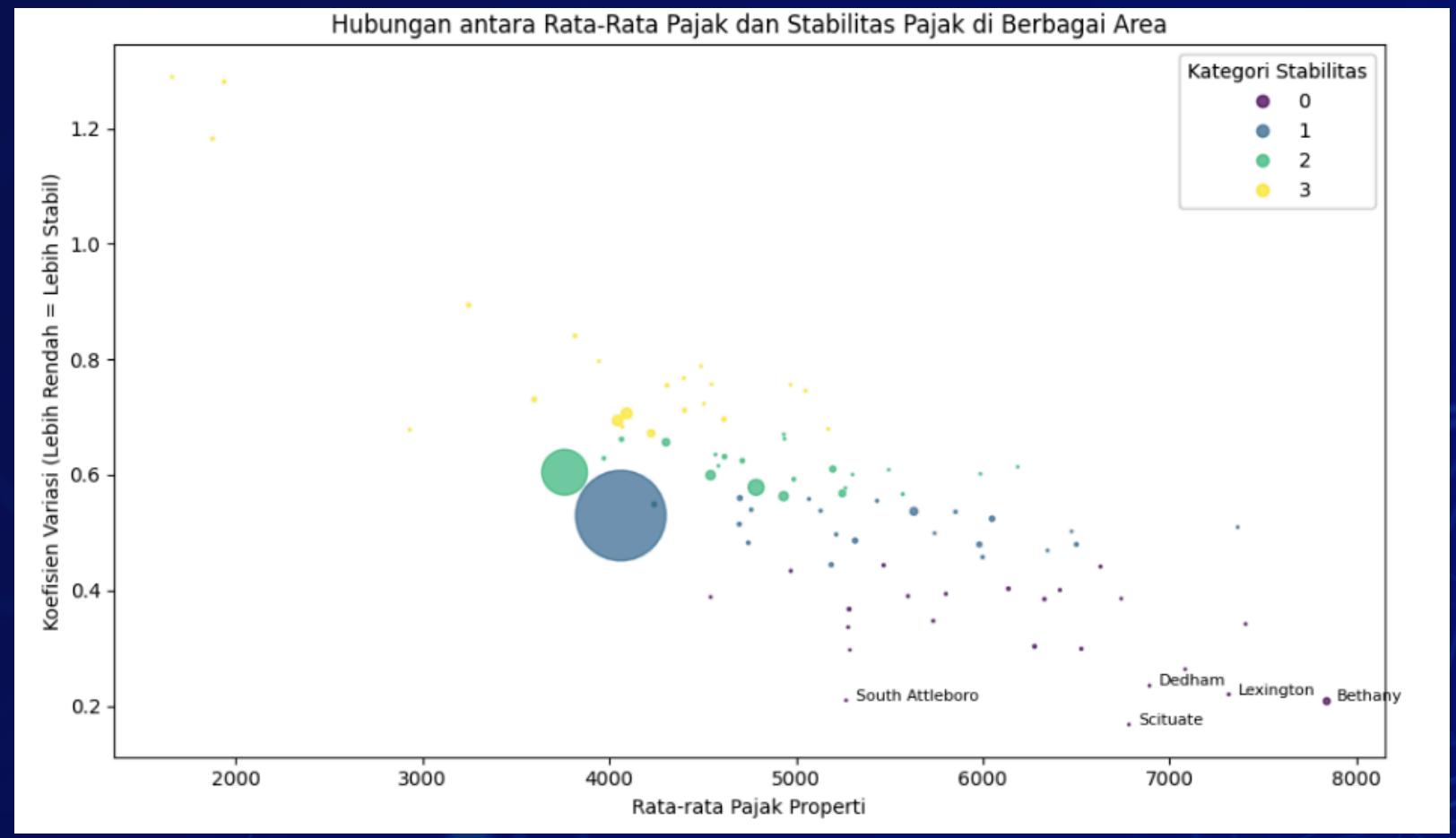
```
2 POINT (-71.415484981 41.858835988)  
4 POINT (-71.400355981 41.821832998)  
5 POINT (-71.394502982 41.820428818)  
10 POINT (-71.424641037 41.810907019)  
11 POINT (-71.430050969 41.807336999)  
Name: Property_Location, dtype: object
```

Data setelah ekstraksi Longitude dan Latitude:

	Longitude	Latitude
2	-71.415484981	41.858835988
4	-71.400355981	41.821832998
5	-71.394502982	41.820428818
10	-71.424641037	41.810907019
11	-71.430050969	41.807336999

# Data Visualization







Thank You

