2108799 Ade Mulyana

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
df = pd.read_csv(r"C:\Users\ACER\Documents\UPI\Semester3\Data_mining_WH\UAS\dataset\pmi.csv")
```

In [3]:
```
df.head(3)
```

Out[3]:

	prov	tahun	kategori_pmi
0	ACEH	2022	TINGGI
1	SUMATERA UTARA	2022	TINGGI
2	SUMATERA BARAT	2022	TINGGI

In [4]:
```
df.shape
```

Out[4]: (442, 3)

Untuk menghitung kepadatan penduduk

In [5]:
```
dfh1 = pd.read_csv(r"C:\Users\ACER\Documents\UPI\Semester3\Data_mining_WH\UAS\dataset\kepadatan_penduduk.csv")
```

In [6]:
```
dfh1.head()
```

Out[6]:

	prov	tahun	kepadatan_penduduk
0	ACEH	2021	92
1	SUMATERA UTARA	2021	205
2	SUMATERA BARAT	2021	133
3	RIAU	2021	75
4	JAMBI	2021	72

Untuk melihat persentase penduduk melek huruf di atas 15

In [7]:
```
dfh2 = pd.read_csv(r"C:\Users\ACER\Documents\UPI\Semester3\Data_mining_WH\UAS\dataset\melek_huruf_diatas15.csv")
```

In [8]:
```
dfh2.head()
```

Out[8]:

	prov	tahun	melek_huruf_diatas15
0	ACEH	2022	98.25
1	SUMATERA UTARA	2022	99.11
2	SUMATERA BARAT	2022	99.29
3	RIAU	2022	99.18
4	JAMBI	2022	98.1

Penggabungan data kepadatan penduduk dengan data melek_huruf_diatas15

In [9]:
```
dfm = pd.merge(dfh1, dfh2, how='left',on=["prov", "tahun"])
```

In [10]:
```
dfm.head(5)
```

Out[10]:

	prov	tahun	kepadatan_penduduk	melek_huruf_diatas15
0	ACEH	2021	92	98.24
1	SUMATERA UTARA	2021	205	99.19
2	SUMATERA BARAT	2021	133	99.26
3	RIAU	2021	75	99.2
4	JAMBI	2021	72	98.08

In [11]:
```
dfm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 280 entries, 0 to 279
Data columns (total 4 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   prov                  280 non-null    object
 1   tahun                 280 non-null    int64 
 2   kepadatan_penduduk    280 non-null    object
 3   melek_huruf_diatas15  210 non-null    object
dtypes: int64(1), object(3)
memory usage: 10.9+ KB
```

In [12]:
```
dfm.describe()
```

Out[12]:

	tahun
count	280.000000
mean	2016.000000
std	3.541864
min	2010.000000
25%	2013.750000
50%	2015.500000
75%	2019.250000
max	2021.000000

In [13]:
```
dfm.isna().sum()
```

Out[13]:
```
prov                   0
tahun                  0
kepadatan_penduduk     0
melek_huruf_diatas15    70
dtype: int64
```

In [14]:
```
dfm.dtypes
```

Out[14]:
```
prov                   object
tahun                   int64
kepadatan_penduduk     object
melek_huruf_diatas15    object
dtype: object
```

In [15]:
```
dfh3 = pd.read_csv(r"C:\Users\ACER\Documents\UPI\Semester3\Data_mining_WH\UAS\dataset\persen_rumah_menyewa.csv")
```

In [16]:
```
dfh3.head()
```

Out[16]:

	tahun	persen_rumah_menyewa	prov
0	2021	6.86	ACEH
1	2021	14.13	SUMATERA UTARA
2	2021	11.37	SUMATERA BARAT
3	2021	11.82	RIAU
4	2021	5.95	JAMBI

In [17]:
```
dfm2 = pd.merge(dfh3, dfm, how='left',on=["tahun"])
```

In [18]:
```
dfm2.head()
```

Out[18]:

	tahun	persen_rumah_menyewa	prov_x	prov_y	kepadatan_penduduk	melek_huruf_diatas15
0	2021	6.86	ACEH	ACEH	92	98.24
1	2021	6.86	ACEH	SUMATERA UTARA	205	99.19
2	2021	6.86	ACEH	SUMATERA BARAT	133	99.26
3	2021	6.86	ACEH	RIAU	75	99.2
4	2021	6.86	ACEH	JAMBI	72	98.08

In [19]:
```
dfm2.isna().sum()
```

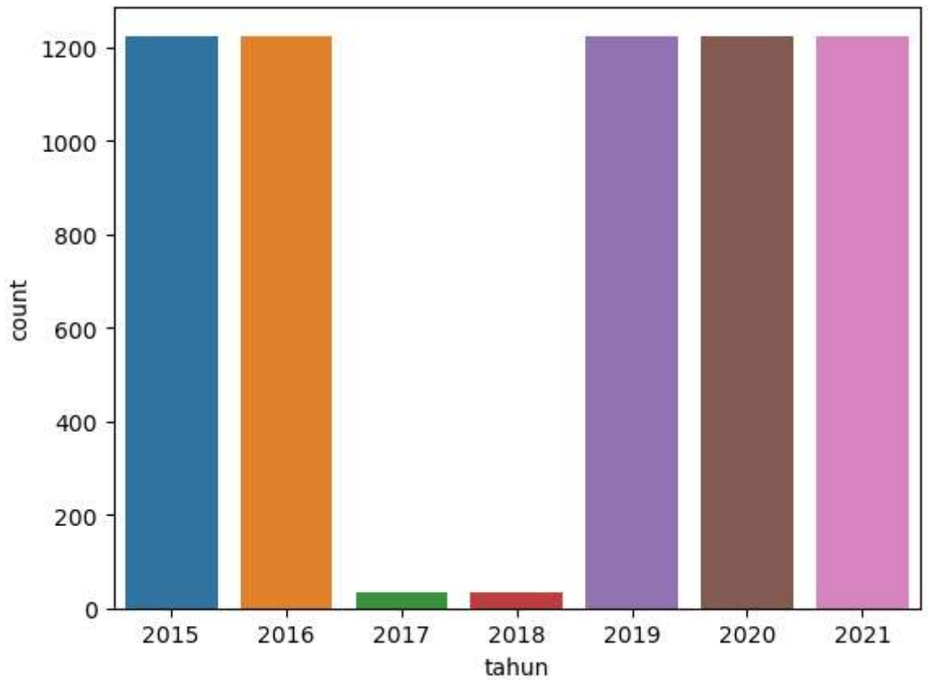
Out[19]:
```
tahun                  0
persen_rumah_menyewa   0
prov_x                 0
prov_y                70
kepadatan_penduduk    70
melek_huruf_diatas15  70
dtype: int64
```

In [20]:
```
dfm2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 6195 entries, 0 to 6194
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   tahun                 6195 non-null   int64
 1   persen_rumah_menyewa  6195 non-null   float64
 2   prov_x                6195 non-null   object
 3   prov_y                6125 non-null   object
 4   kepadatan_penduduk    6125 non-null   object
 5   melek_huruf_diatas15  6125 non-null   object
dtypes: float64(1), int64(1), object(4)
memory usage: 338.8+ KB
```

In [21]:
```
sns.countplot(x="tahun", data=dfm2)
```

Out[21]:
```
<AxesSubplot:xlabel='tahun', ylabel='count'>
```



In [22]:
```
dfm2[['kepadatan_penduduk','melek_huruf_diatas15']].describe()
```

Out[22]:

	kepadatan_penduduk	melek_huruf_diatas15
count	6125	6125
unique	131	144
top	9	98.01
freq	245	140

import dataset persen penduduk trampil tik

In [23]:
```
dfh4 = pd.read_csv(r"C:\Users\ACER\Documents\UPI\Semester3\Data_mining_WH\UAS\dataset\persen_penduduk_trampil_tik.csv")
```

In [24]:
```
dfh4.isna().sum()
```

Out[24]:
```
tahun                          0
persen_penduduk_trampil_tik    0
prov                           0
dtype: int64
```

In [25]:
```
dfm3 = pd.merge(dfh4, dfm2, how='left',on=["tahun"])
```

In [26]:
```
dfm3.head()
```

Out[26]:

	tahun	persen_penduduk_trampil_tik	prov	persen_rumah_menyewa	prov_x	prov_y	kepadatan_penduduk	melek_huruf_diatas15
0	2021	60.21	ACEH	6.86	ACEH	ACEH	92	98.24
1	2021	60.21	ACEH	6.86	ACEH	SUMATERA UTARA	205	99.19
2	2021	60.21	ACEH	6.86	ACEH	SUMATERA BARAT	133	99.26
3	2021	60.21	ACEH	6.86	ACEH	RIAU	75	99.2
4	2021	60.21	ACEH	6.86	ACEH	JAMBI	72	98.08

Isi nilai null / kosong dengan median

In [27]:
```
dfm3['kepadatan_penduduk'] = dfm3['kepadatan_penduduk'].fillna(dfm3['kepadatan_penduduk'].median())
dfm3['melek_huruf_diatas15'] = dfm3['melek_huruf_diatas15'].fillna(dfm3['melek_huruf_diatas15'].median())
```

Drop kolom prov yang merupakan anomali

In [28]:
```
dfm3 = dfm3.drop(['prov'],axis=1)
```

In [29]:
```
dfm3.head()
```

Out[29]:

	tahun	persen_penduduk_trampil_tik	persen_rumah_menyewa	prov_x	prov_y	kepadatan_penduduk	melek_huruf_diatas15
0	2021	60.21	6.86	ACEH	ACEH	92	98.24
1	2021	60.21	6.86	ACEH	SUMATERA UTARA	205	99.19
2	2021	60.21	6.86	ACEH	SUMATERA BARAT	133	99.26
3	2021	60.21	6.86	ACEH	RIAU	75	99.2
4	2021	60.21	6.86	ACEH	JAMBI	72	98.08

In [30]:
```
df_final = pd.merge(dfm3, df, how='left',on=["tahun"])
```

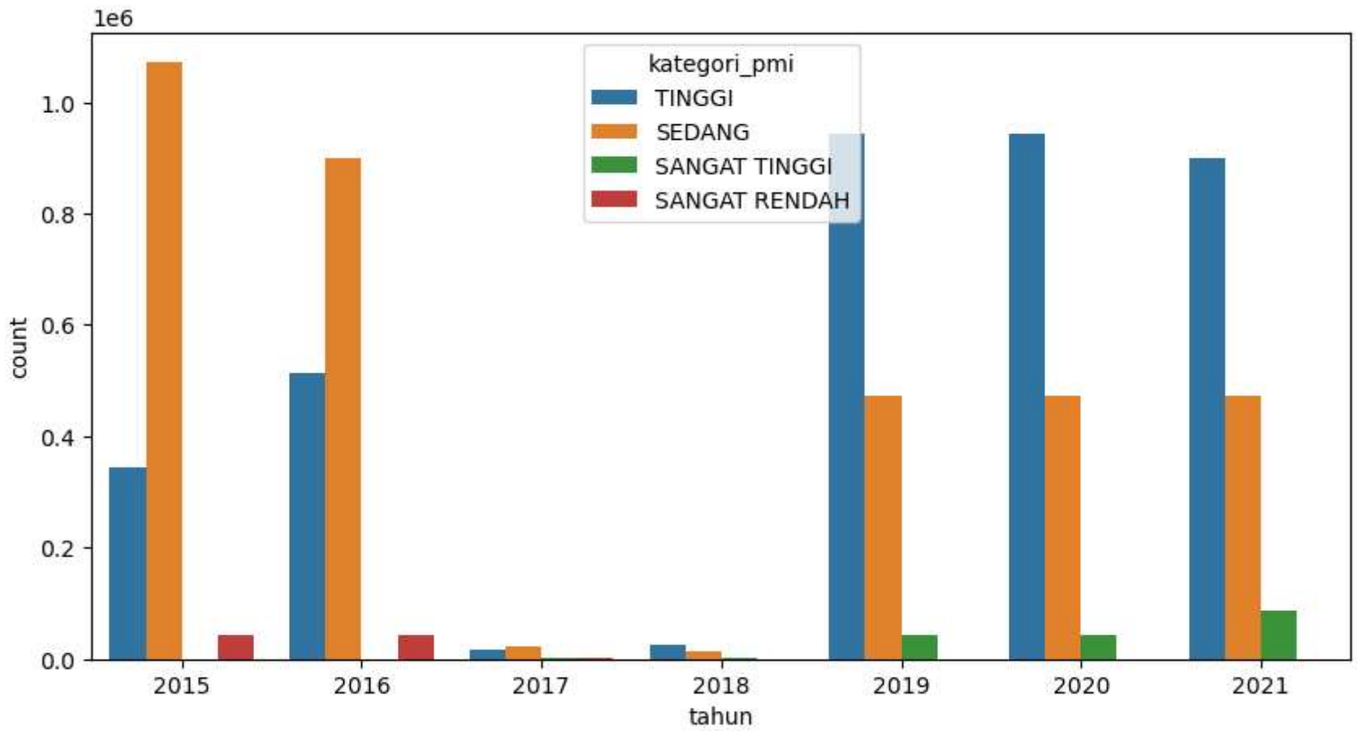
In [31]:
```
df_final.head()
```

Out[31]:

	tahun	persen_penduduk_trampil_tik	persen_rumah_menyewa	prov_x	prov_y	kepadatan_penduduk	melek_huruf_diatas15	prov	kategori_pmi
0	2021	60.21	6.86	ACEH	ACEH	92	98.24	ACEH	TINGGI
1	2021	60.21	6.86	ACEH	ACEH	92	98.24	SUMATERA UTARA	TINGGI
2	2021	60.21	6.86	ACEH	ACEH	92	98.24	SUMATERA BARAT	TINGGI
3	2021	60.21	6.86	ACEH	ACEH	92	98.24	RIAU	TINGGI
4	2021	60.21	6.86	ACEH	ACEH	92	98.24	JAMBI	TINGGI

In [32]:
```
fig, ax = plt.subplots(figsize=(10, 5)) # atur ukuran chart
sns.countplot(ax=ax, x="tahun", hue="kategori_pmi", data=df_final)
```

Out[32]: <AxesSubplot:xlabel='tahun', ylabel='count'>



Melihat persentase pmi di tiap tahunnya

In [33]:
```
# df_final = df_final.drop(['prov_y'],axis=1
df_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7372050 entries, 0 to 7372049
Data columns (total 9 columns):
 #   Column                        Dtype
---  ------                        -----
 0   tahun                         int64
 1   persen_penduduk_trampil_tik   float64
 2   persen_rumah_menyewa          float64
 3   prov_x                        object
 4   prov_y                        object
 5   kepadatan_penduduk            object
 6   melek_huruf_diatas15          object
 7   prov                          object
 8   kategori_pmi                  object
dtypes: float64(2), int64(1), object(6)
memory usage: 562.4+ MB
```

In [34]:
```
df_final['kategori_pmi'] = df_final['kategori_pmi'].astype("category")
df_final['prov'] = df_final['prov'].astype("category")
df_final['tahun'] = df_final['tahun'].astype("category")
```

In [35]:
```
df_final.isna().sum()
```

Out[35]:
```
tahun                             0
persen_penduduk_trampil_tik       0
persen_rumah_menyewa              0
prov_x                            0
prov_y                        83300
kepadatan_penduduk                0
melek_huruf_diatas15              0
prov                              0
kategori_pmi                      0
dtype: int64
```

In [36]:
```
df_final2 = df_final.copy()
```

In [37]:
```
df_final2['kepadatan_penduduk'] = df_final2['kepadatan_penduduk'].fillna(df_final2['kepadatan_penduduk'].median())
df_final2['melek_huruf_diatas15'] = df_final2['melek_huruf_diatas15'].fillna(df_final2['melek_huruf_diatas15'].median())
```

In [38]:
```
df_final2.isna().sum()
```

Out[38]:
```
tahun                            0
persen_penduduk_trampil_tik      0
persen_rumah_menyewa             0
prov_x                           0
prov_y                       83300
kepadatan_penduduk               0
melek_huruf_diatas15             0
prov                             0
kategori_pmi                     0
dtype: int64
```

Penghapusan kolom tahun dengan prov

In [39]:
```
df_final2 = df_final2.drop(['tahun', 'prov', 'prov_x', 'prov_y'],axis=1)
```

In [40]:
```
df_final2.head()
```

Out[40]:

	persen_penduduk_trampil_tik	persen_rumah_menyewa	kepadatan_penduduk	melek_huruf_diatas15	kategori_pmi
0	60.21	6.86	92	98.24	TINGGI
1	60.21	6.86	92	98.24	TINGGI
2	60.21	6.86	92	98.24	TINGGI
3	60.21	6.86	92	98.24	TINGGI
4	60.21	6.86	92	98.24	TINGGI

In [41]:
```
df_final2.isna().sum()
```

Out[41]:
```
persen_penduduk_trampil_tik    0
persen_rumah_menyewa           0
kepadatan_penduduk             0
melek_huruf_diatas15           0
kategori_pmi                   0
dtype: int64
```

In [43]:
```
df_final2.head()
```

Out[43]:

	persen_penduduk_trampil_tik	persen_rumah_menyewa	kepadatan_penduduk	melek_huruf_diatas15	kategori_pmi
0	60.21	6.86	92	98.24	TINGGI
1	60.21	6.86	92	98.24	TINGGI
2	60.21	6.86	92	98.24	TINGGI
3	60.21	6.86	92	98.24	TINGGI
4	60.21	6.86	92	98.24	TINGGI

Kelas target

In [46]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(df_final2.kategori_pmi)
Y = le.transform(df_final2.kategori_pmi)
X = df_final2.drop("kategori_pmi",axis=1)
```

In [48]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=123)
```

Lakukan Learning dengan Naive Bayes

In [49]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
clf = GaussianNB()
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
acc = accuracy_score(Y_test, Y_pred)
print("Akurasi {}".format(acc))
print(classification_report(Y_test, Y_pred))
```

```
Akurasi 0.6203722166832835

C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

              precision    recall  f1-score   support

           0       0.00      0.00      0.00     17439
           1       0.00      0.00      0.00     34589
           2       0.63      0.56      0.59    685678
           3       0.61      0.72      0.66    736704

    accuracy                           0.62   1474410
   macro avg       0.31      0.32      0.31   1474410
weighted avg       0.60      0.62      0.61   1474410


C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

DECISION TREE

In [50]:
```
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
acc = accuracy_score(Y_test, Y_pred)
print("Akurasi {}".format(acc))
print(classification_report(Y_test, Y_pred))
```

```
Akurasi 0.6512367658927978

C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

              precision    recall  f1-score   support

           0       0.00      0.00      0.00     17439
           1       0.00      0.00      0.00     34589
           2       0.67      0.58      0.62    685678
           3       0.64      0.76      0.69    736704

    accuracy                           0.65   1474410
   macro avg       0.33      0.34      0.33   1474410
weighted avg       0.63      0.65      0.64   1474410
```

```
C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

RANDOM FOREST

In [51]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=10, random_state=123)
clf.fit(X_train, Y_train)
Y_pred = clf.predict(X_test)
acc = accuracy_score(Y_test, Y_pred)
print("Akurasi {}".format(acc))
print(classification_report(Y_test, Y_pred))
```

```
Akurasi 0.6493933166486934

C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

              precision    recall  f1-score   support

           0       0.00      0.00      0.00     17439
           1       0.00      0.00      0.00     34589
           2       0.67      0.58      0.62    685678
           3       0.63      0.76      0.69    736704

    accuracy                           0.65   1474410
   macro avg       0.33      0.33      0.33   1474410
weighted avg       0.63      0.65      0.63   1474410


C:\Users\ACER\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with
no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```