```
##header library

%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df1 = pd.read_csv("D:\\aaa\\Data Mining\\UAS\\2006836_Atta-Arrezie-
Kurnia_UAS_DatMin\\dataset_uas_datamining_jan_23\\pmi.csv")

df1[:]
```

	prov	tahun	kategori_pmi
0	ACEH	2022	TINGGI
1	SUMATERA UTARA	2022	TINGGI
2	SUMATERA BARAT	2022	TINGGI
3	RIAU	2022	TINGGI
4	JAMBI	2022	TINGGI
..	...	...	...
437	SULAWESI BARAT	2010	SANGAT RENDAH
438	MALUKU	2010	SEDANG
439	MALUKU UTARA	2010	SEDANG
440	PAPUA BARAT	2010	SANGAT RENDAH
441	PAPUA	2010	SANGAT RENDAH

[442 rows x 3 columns]

Proses EDA

dataset yang digunakan merupakan data ketidak cukupan pangan dalam provinsi dan data proporsirt dengan akses pelayanan dasar dalam provinsi data tersebut digunakan karena dalam mengukur keberhasilan dalam upaya membangun kualitas hidup manusia (masyarakat/penduduk), kecukupan pangan dan kemudahan akses untuk pelayanan dasar sangat diperlukan.

```
df2 = pd.read_csv("D:\\aaa\\Data Mining\\UAS\\2006836_Atta-Arrezie-
Kurnia_UAS_DatMin\\dataset_uas_datamining_jan_23\\
ketidak_cukupan_pangan_prov.csv")
df3 = pd.read_csv("D:\\aaa\\Data Mining\\UAS\\2006836_Atta-Arrezie-
Kurnia_UAS_DatMin\\dataset_uas_datamining_jan_23\\
proporsi_rt_akses_layanan_dasar.csv")

df2[:]
```

	tahun	pct_tdk_cukup_pangan	prov
0	2021	6.90	ACEH
1	2021	6.33	SUMATERA UTARA
2	2021	6.02	SUMATERA BARAT
3	2021	10.61	RIAU
4	2021	9.25	JAMBI

```
..     ...                   ...             ...
170   2017                 26.57          MALUKU
171   2017                 34.05    MALUKU UTARA
172   2017                 27.22     PAPUA BARAT
173   2017                 34.27           PAPUA
174   2017                  8.23       INDONESIA

[175 rows x 3 columns]
```

```
df3[:]
```

```
     tahun  proporsi_rt_akses_layanan_dasar            prov
0     2021                            69.31            ACEH
1     2021                            68.22  SUMATERA UTARA
2     2021                            82.09  SUMATERA BARAT
3     2021                            74.93            RIAU
4     2021                            73.41           JAMBI
..     ...                              ...             ...
100   2019                            75.60          MALUKU
101   2019                            72.81    MALUKU UTARA
102   2019                            69.47     PAPUA BARAT
103   2019                            35.55           PAPUA
104   2019                            76.07       INDONESIA

[105 rows x 3 columns]
```

```
df_merge = pd.merge(df2, df3, how="left", on=["tahun", "prov"])
df_merge[:]
```

```
     tahun  pct_tdk_cukup_pangan            prov  \
0     2021                  6.90            ACEH
1     2021                  6.33  SUMATERA UTARA
2     2021                  6.02  SUMATERA BARAT
3     2021                 10.61            RIAU
4     2021                  9.25           JAMBI
..     ...                   ...             ...
170   2017                 26.57          MALUKU
171   2017                 34.05    MALUKU UTARA
172   2017                 27.22     PAPUA BARAT
173   2017                 34.27           PAPUA
174   2017                  8.23       INDONESIA

     proporsi_rt_akses_layanan_dasar
0                              69.31
1                              68.22
2                              82.09
3                              74.93
4                              73.41
..                               ...
170                            NaN
171                            NaN
```

```
172                                NaN
173                                NaN
174                                NaN

[175 rows x 4 columns]
```

```
df_merge["pct_tdk_cukup_pangan"].value_counts(dropna = False)
```

```
8.71     3
10.18    2
8.58     2
7.47     2
9.16     2
        ..
7.39     1
11.80    1
9.77     1
30.75    1
9.25     1
Name: pct_tdk_cukup_pangan, Length: 166, dtype: int64
```

```
df_merge["tahun"].value_counts(dropna = False)
```

```
2017    35
2018    35
2019    35
2020    35
2021    35
Name: tahun, dtype: int64
```

data ketidak cukupan pangan berada dalam rentang 2017 sampai 2021

```
df_merge.isnull().sum()
```

```
tahun                                0
pct_tdk_cukup_pangan                 0
prov                                 0
proporsi_rt_akses_layanan_dasar     70
dtype: int64
```

ternyata ada yang kosong di proporsi rt, cek menggunakan loc untuk setiap tahun. dicurigai bahwa data proporsi tidak ada dari tahun 2017 sampai 2018

```
df_merge.loc[df_merge.tahun.isin([2017,
2018]),"proporsi_rt_akses_layanan_dasar"]
```

```
105   NaN
106   NaN
107   NaN
108   NaN
109   NaN
       ..
```

```
170   NaN
171   NaN
172   NaN
173   NaN
174   NaN
Name: proporsi_rt_akses_layanan_dasar, Length: 70, dtype: float64
```

setelah di teliti ternyata memang tidak ada data untuk proprosi rt dari tahun 2017 sampai 2018, maka kita hanya menggunakan data dari tahun 2019 keatas

```
df_merge = df_merge.drop(df_merge[df_merge.tahun < 2019].index)
```

```
df_merge[:]
```

```
     tahun  pct_tdk_cukup_pangan            prov  \
0     2021                  6.90            ACEH
1     2021                  6.33  SUMATERA UTARA
2     2021                  6.02  SUMATERA BARAT
3     2021                 10.61            RIAU
4     2021                  9.25           JAMBI
..     ...                   ...             ...
100   2019                 34.12          MALUKU
101   2019                 35.81    MALUKU UTARA
102   2019                 19.22     PAPUA BARAT
103   2019                 38.21           PAPUA
104   2019                  7.63       INDONESIA

     proporsi_rt_akses_layanan_dasar
0                              69.31
1                              68.22
2                              82.09
3                              74.93
4                              73.41
..                               ...
100                            75.60
101                            72.81
102                            69.47
103                            35.55
104                            76.07

[105 rows x 4 columns]
```

```
df_merge["pct_tdk_cukup_pangan"].value_counts(dropna = False)
```

```
9.16     2
8.58     2
35.48    1
9.41     1
23.09    1
        ..
6.91     1
```

```
5.86     1
11.17    1
12.56    1
9.25     1
Name: pct_tdk_cukup_pangan, Length: 103, dtype: int64
```

```
df_merge = pd.merge(df1, df_merge, how="left", on=["tahun", "prov"])
df_merge[:]
```

```
               prov  tahun kategori_pmi_x kategori_pmi_y  \
0              ACEH   2022         TINGGI         TINGGI
1    SUMATERA UTARA   2022         TINGGI         TINGGI
2    SUMATERA BARAT   2022         TINGGI         TINGGI
3              RIAU   2022         TINGGI         TINGGI
4             JAMBI   2022         TINGGI         TINGGI
..              ...    ...            ...            ...
437  SULAWESI BARAT   2010  SANGAT RENDAH  SANGAT RENDAH
438          MALUKU   2010         SEDANG         SEDANG
439    MALUKU UTARA   2010         SEDANG         SEDANG
440     PAPUA BARAT   2010  SANGAT RENDAH  SANGAT RENDAH
441           PAPUA   2010  SANGAT RENDAH  SANGAT RENDAH

     pct_tdk_cukup_pangan  proporsi_rt_akses_layanan_dasar
0                     NaN                              NaN
1                     NaN                              NaN
2                     NaN                              NaN
3                     NaN                              NaN
4                     NaN                              NaN
..                    ...                              ...
437                   NaN                              NaN
438                   NaN                              NaN
439                   NaN                              NaN
440                   NaN                              NaN
441                   NaN                              NaN

[442 rows x 6 columns]
```

karena kita hanya mengklasifikasi data dari tahun 2019 sampai 2021, maka hapus data diluar dari tahun tersebut

```
df_merge = df_merge.drop(df_merge[(df_merge.tahun < 2019)].index)
df_merge = df_merge.drop(df_merge[(df_merge.tahun > 2021)].index)
```

```
df_merge[:]
```

```
               prov  tahun kategori_pmi_x kategori_pmi_y  \
34             ACEH   2021         TINGGI         TINGGI
35   SUMATERA UTARA   2021         TINGGI         TINGGI
36   SUMATERA BARAT   2021         TINGGI         TINGGI
37             RIAU   2021         TINGGI         TINGGI
38            JAMBI   2021         TINGGI         TINGGI
```

```
..              ...    ...           ...           ...
131  SULAWESI BARAT   2019         SEDANG         SEDANG
132          MALUKU   2019         SEDANG         SEDANG
133    MALUKU UTARA   2019         SEDANG         SEDANG
134     PAPUA BARAT   2019         SEDANG         SEDANG
135           PAPUA   2019         SEDANG         SEDANG

     pct_tdk_cukup_pangan  proporsi_rt_akses_layanan_dasar
34                   6.90                            69.31
35                   6.33                            68.22
36                   6.02                            82.09
37                  10.61                            74.93
38                   9.25                            73.41
..                    ...                              ...
131                  6.79                            80.77
132                 34.12                            75.60
133                 35.81                            72.81
134                 19.22                            69.47
135                 38.21                            35.55

[102 rows x 6 columns]
```

```
df_merge = df_merge.drop(['kategori_pmi_y'],axis=1) #buang atribut
kategori_pmi_y karena duplikat
```

```
---------------------------------------------------------------------------
-----
KeyError                                  Traceback (most recent call
last)
<ipython-input-120-f27102a9cc16> in <module>
----> 1 df_merge = df_merge.drop(['kategori_pmi_y'],axis=1) #buang
atribut kategori_pmi_y karena duplikat

~\anaconda3\lib\site-packages\pandas\core\frame.py in drop(self,
labels, axis, index, columns, level, inplace, errors)
   4306                 weight  1.0     0.8
   4307         """
-> 4308         return super().drop(
   4309             labels=labels,
   4310             axis=axis,

~\anaconda3\lib\site-packages\pandas\core\generic.py in drop(self,
labels, axis, index, columns, level, inplace, errors)
   4151         for axis, labels in axes.items():
   4152             if labels is not None:
-> 4153                 obj = obj._drop_axis(labels, axis,
level=level, errors=errors)
   4154
   4155         if inplace:
```

```
~\anaconda3\lib\site-packages\pandas\core\generic.py in
_drop_axis(self, labels, axis, level, errors)
   4186                 new_axis = axis.drop(labels, level=level,
errors=errors)
   4187             else:
-> 4188                 new_axis = axis.drop(labels, errors=errors)
   4189             result = self.reindex(**{axis_name: new_axis})
   4190

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in
drop(self, labels, errors)
   5589         if mask.any():
   5590             if errors != "ignore":
-> 5591                 raise KeyError(f"{labels[mask]} not found in
axis")
   5592             indexer = indexer[~mask]
   5593         return self.delete(indexer)

KeyError: "['kategori_pmi_y'] not found in axis"
```

```
df_merge.rename(columns = {'kategori_pmi_x':'kategori_pmi'}, inplace =
True)
```

```
df_merge[:]
```

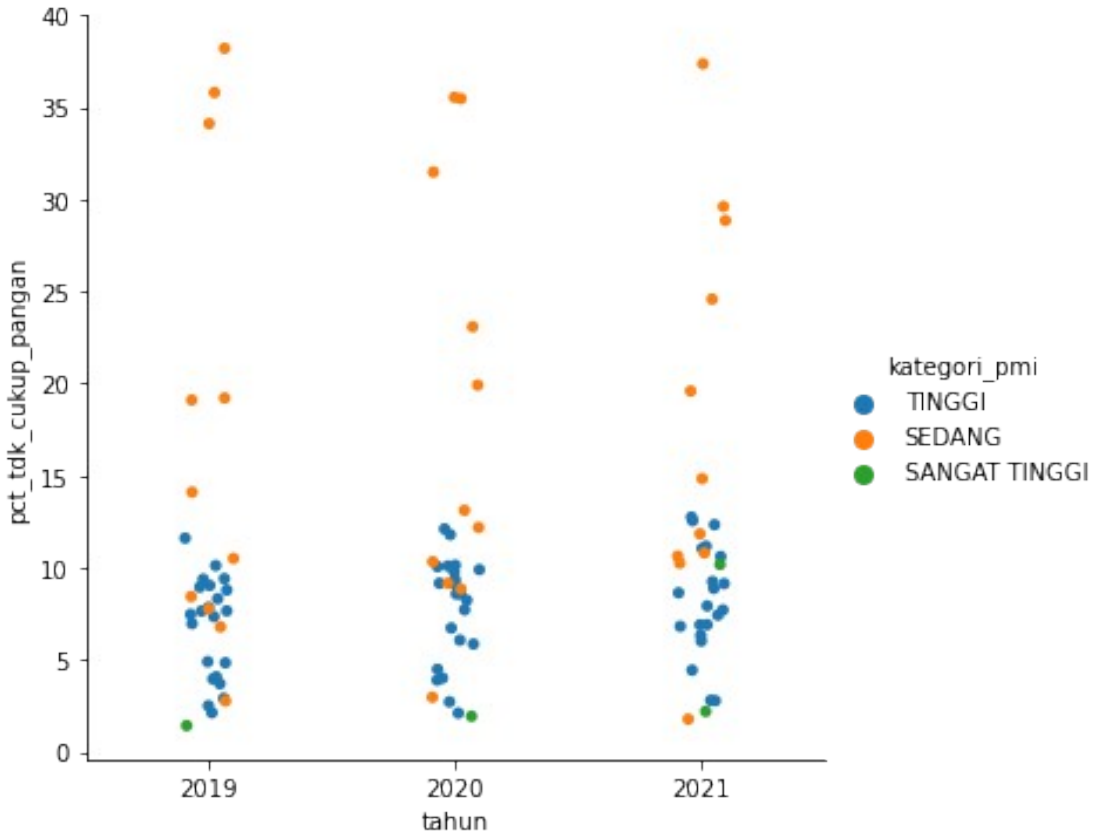
```
               prov  tahun kategori_pmi  pct_tdk_cukup_pangan  \
34             ACEH   2021       TINGGI                  6.90
35   SUMATERA UTARA   2021       TINGGI                  6.33
36   SUMATERA BARAT   2021       TINGGI                  6.02
37             RIAU   2021       TINGGI                 10.61
38            JAMBI   2021       TINGGI                  9.25
..              ...    ...          ...                   ...
131  SULAWESI BARAT   2019       SEDANG                  6.79
132          MALUKU   2019       SEDANG                 34.12
133    MALUKU UTARA   2019       SEDANG                 35.81
134     PAPUA BARAT   2019       SEDANG                 19.22
135           PAPUA   2019       SEDANG                 38.21

     proporsi_rt_akses_layanan_dasar
34                             69.31
35                             68.22
36                             82.09
37                             74.93
38                             73.41
..                               ...
131                            80.77
132                            75.60
133                            72.81
134                            69.47
135                            35.55
```

```
[102 rows x 5 columns]
```

```
##melihat rentang kategori pmi pada setiap tahun untuk persentase
ketidak cukupan pangan
sns.catplot(x="tahun", y="pct_tdk_cukup_pangan", hue="kategori_pmi",
data=df_merge)
```

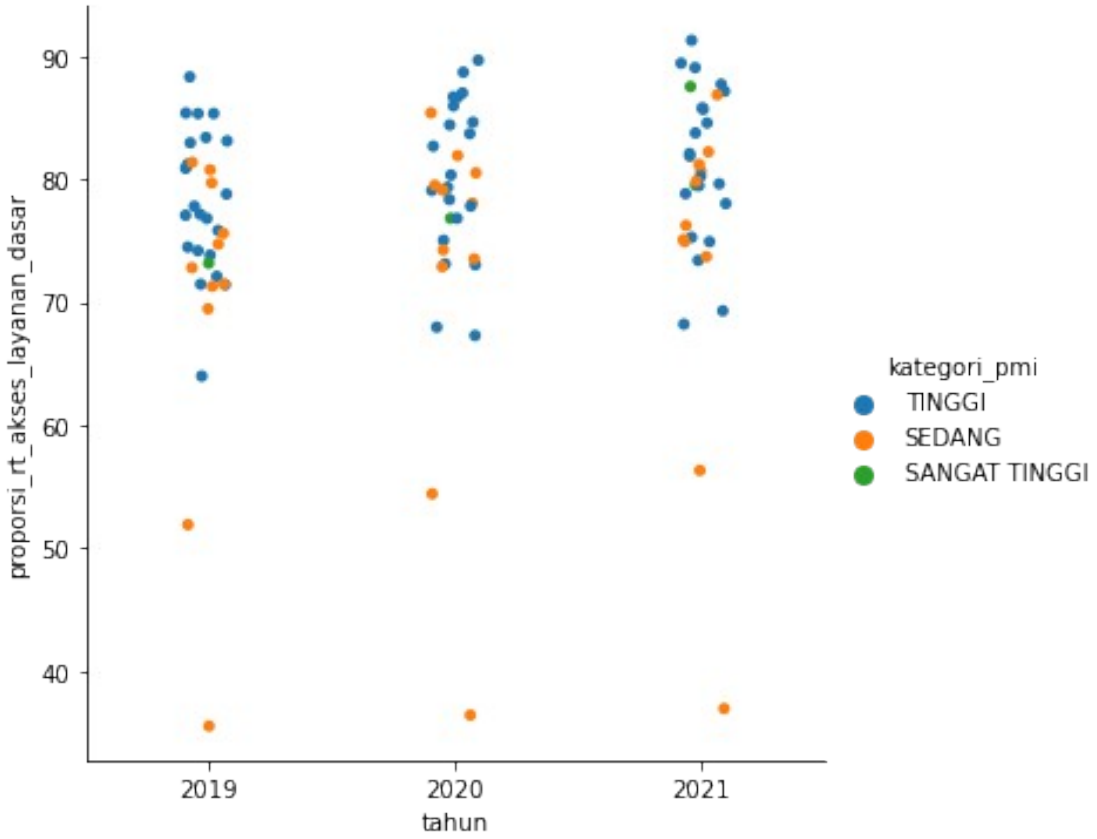
```
<seaborn.axisgrid.FacetGrid at 0x14cd42bba30>
```



dilihat dari catplot diatas, ipm/pmi dgn kategori tinggi mempunyai persentase ketidak cukupan pangan berada di kisaran 1% - 15%, sedangkan ipm dengan kategori sangat tinggi berada di rentang atau kisaran 1%-10% yang artinya kecukupan pangan masyarakat dari prov dgn ipm tinggi dan sangat tinggi sebagian besar terpenuhi. Namun tidak menutup kemungkinan bahwa ipm dengan kategori sedang juga memiliki tingkat persentase ketidak cukupan pangan di kisaran 1% - 15%, walau tingkat ketidak cukupannya pangannya lebih menyebar di >15%

```
##melihat rentang kategori pmi pada setiap tahun untuk persentase
propori akses pelayanan dasar tiap rt
sns.catplot(x="tahun", y="proporsi_rt_akses_layanan_dasar",
hue="kategori_pmi", data=df_merge)
```

```
<seaborn.axisgrid.FacetGrid at 0x14cd37ca610>
```



dilihat dari catplot diatas, untuk semua kategori ipm tiap provinsi memiliki proporsi akses pelayanan dasar yangcukup besar yaitu pada kisaran 70-90%, walau untuk kategori sedang masih mempunyai beberapa proporsi yang rendah bahkan menyentuh <40%

Klasifikasi

####masih error

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit(df_merge.kategori_pmi_x)
Y = le.transform(df_merge.kategori_pmi_x)

list(le.classes_)
```

```
['SANGAT TINGGI', 'SEDANG', 'TINGGI']
```

```
list(le.inverse_transform([1, 2, 0]))
```

```
['SEDANG', 'TINGGI', 'SANGAT TINGGI']
```

dilihat dari list diatas, pada rentang tahun 2019 sampai 2021 kategori pmi per provinsi hanya memiliki 3 kategori dari yang terendah yaitu "sedang", "tinggi", dan "sangat tinggi"

```
X = df_merge.drop("kategori_pmi_x",axis=1)
```

```
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,rando
m_state=12)
import pickle

from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=50, random_state=12)
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))

---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
<ipython-input-118-51bce608e800> in <module>
      4 from sklearn.ensemble import RandomForestClassifier
      5 clf = RandomForestClassifier(n_estimators=50, random_state=12)
----> 6 clf.fit(X_train, Y_train)
      7 Y_pred = clf.predict(X_test)
      8 acc = accuracy_score(Y_test, Y_pred)

~\anaconda3\lib\site-packages\sklearn\ensemble\_forest.py in fit(self,
X, y, sample_weight)
    302                 "sparse multilabel-indicator for y is not
supported."
    303             )
--> 304         X, y = self._validate_data(X, y, multi_output=True,
    305                                    accept_sparse="csc",
dtype=DTYPE)
    306         if sample_weight is not None:

~\anaconda3\lib\site-packages\sklearn\base.py in _validate_data(self,
X, y, reset, validate_separately, **check_params)
    431                 y = check_array(y, **check_y_params)
    432             else:
--> 433                 X, y = check_X_y(X, y, **check_params)
    434             out = X, y
    435

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
```

```
     64
     65             # extra_args > 0

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, order,
copy, force_all_finite, ensure_2d, allow_nd, multi_output,
ensure_min_samples, ensure_min_features, y_numeric, estimator)
    812         raise ValueError("y cannot be None")
    813
--> 814     X = check_array(X, accept_sparse=accept_sparse,
    815                     accept_large_sparse=accept_large_sparse,
    816                     dtype=dtype, order=order, copy=copy,

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_array(array, accept_sparse, accept_large_sparse, dtype, order,
copy, force_all_finite, ensure_2d, allow_nd, ensure_min_samples,
ensure_min_features, estimator)
    614                     array = array.astype(dtype,
casting="unsafe", copy=False)
    615                 else:
--> 616                     array = np.asarray(array, order=order,
dtype=dtype)
    617             except ComplexWarning as complex_warning:
    618                 raise ValueError("Complex data not supported\
n"

~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a,
dtype, order, like)
    100         return _asarray_with_like(a, dtype=dtype, order=order,
like=like)
    101
--> 102     return array(a, dtype, copy=False, order=order)
    103
    104

~\anaconda3\lib\site-packages\pandas\core\generic.py in
__array__(self, dtype)
   1897
   1898     def __array__(self, dtype=None) -> np.ndarray:
-> 1899         return np.asarray(self._values, dtype=dtype)
   1900
   1901     def __array_wrap__(
```

```
~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a,
dtype, order, like)
    100         return _asarray_with_like(a, dtype=dtype, order=order,
like=like)
    101
--> 102     return array(a, dtype, copy=False, order=order)
    103
    104

ValueError: could not convert string to float: 'KALIMANTAN TENGAH'
```