Telco Churn Data Insight

Faishal Syams Afif _ 056

LANGKAH – LANGKAH
EXPLORE DATA TELCO CHURN

1. Mengimport module/library yang diperlukan

```
import numpy as np
import pandas as pd
import seaborn as sns
import pandas as pd
print("Setup Complete")
```

```
Setup Complete
```

2. Melihat isi dataset

```
[ ] df = pd.read_csv('/content/drive/MyDrive/Dataset/WA_Fn-UseC_-Telco-Customer-Churn (1).csv')
    df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

5 rows × 21 columns

3. Melihat Detail DataFrame Telco Churn

Detail DataFrame

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Diketahui kolom 'Total Charges' itu ber type data string padahal dia bentuk angka float. maka untuk membuat visualisasi data dari total charges kita harus merubah type datanya terlebih dahulu

```
[ ] df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

4. Melihat apakah dataset telco churn mempunyai missing value

Missing Value

```
df.isna().sum()
# atau
df.isnull().sum()
```

0d

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

Tidak ada missing value di datase telco churn

5. Merubah Data Type TotalCharges dari string menjadi float

```
df = df.drop(df[df['TotalCharges'] == ' '].index)
df['TotalCharges'] = df['TotalCharges'].astype(float)
df.head(3)
```

PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	pct_PaymentMethod
Yes	Electronic check	29.85	29.85	No	33.58
Yes	Electronic check	70.70	151.65	Yes	33.58
Yes	Electronic check	99.65	820.50	Yes	33.58

DATA ENCODING

Label Encoding

```
df['gender'] = df['gender'].astype('category').cat.codes
df.head(3)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenu
0	7590-VHVEG	0	0	Yes	No	
1	5575-GNVDE	1	0	No	No	
2	3668-QPYBK	1	0	No	No	

3 rows × 21 columns

Merubah value gender dengan mengcategorikan kan female = 0 dan male = 1

Ordinal Encoding

```
[33] df['PaymentMethod'].value_counts()
```

```
Electronic check             2365
Mailed check                 1604
Bank transfer (automatic)    1542
Credit card (automatic)      1521
Name: PaymentMethod, dtype: int64
```

```
[38] map_PaymentMethod = {'Electronic check':1,
                          'Mailed check':2,
                          'Bank transfer (automatic)':3,
                          'Credit card (automatic)':4
                          }
     df['PaymentMethod_cat'] = df['PaymentMethod'].map(map_PaymentMethod)
     df[['PaymentMethod','PaymentMethod_cat']].head()
```

	PaymentMethod	PaymentMethod_cat
0	Electronic check	1
1	Mailed check	2
2	Mailed check	2
3	Bank transfer (automatic)	3
4	Electronic check	1

Membuat colom baru dengan value berdasarkan value kolom payment method yang dikategorikan 1, 2, 3, dan 4

One Hot Encoding

Membuat kolom dumies dari value kolom embark_town. berisi 1 dan 0 berdasarkan value

```
dummies_embark_town = pd.get_dummies(df['embark_town'],prefix='embark_town')
dummies_embark_town.head()
```

	embark_town_Cherbourg	embark_town_Queenstown	embark_town_Southampton
0	0	0	1
1	1	0	0
2	0	0	1
3	0	0	1
4	0	0	1

menambahkan kolom dumies diatas ke dataset telcochurn

```
[77] df = pd.concat([df, dummies_embark_town], axis=1)
df.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone	class_cat	embark_town_Cherbourg	embark_town_Queenstown	embark_town_Southampton
0	0	3	1	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False	3	0	0	1
1	1	1	0	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False	1	1	0	0
2	1	3	0	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True	3	0	0	1
3	1	1	0	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False	1	0	0	1
4	0	3	1	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True	3	0	0	1

menghapus kolom embark town, karena sudah di gantikan oleh kolom dummies dari embark town

```
[78] df = df.drop('embark_town',axis=1)
df.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	alive	alone	class_cat	embark_town_Cherbourg	embark_town_Queenstown	embark_town_Southampton
0	0	3	1	22.0	1	0	7.2500	S	Third	man	True	NaN	no	False	3	0	0	1
1	1	1	0	38.0	1	0	71.2833	C	First	woman	False	C	yes	False	1	1	0	0
2	1	3	0	26.0	0	0	7.9250	S	Third	woman	False	NaN	yes	True	3	0	0	1
3	1	1	0	35.0	1	0	53.1000	S	First	woman	False	C	yes	False	1	0	0	1
4	0	3	1	35.0	0	0	8.0500	S	Third	man	True	NaN	no	True	3	0	0	1

Frequesi Encoding

```
[86] freq_pm = df['PaymentMethod'].value_counts().reset_index()
     freq_pm.rename(columns={"index": "PaymentMethod", "PaymentMethod": "freq_PaymentMethod"}, inplace = True)
     freq_pm['pct_PaymentMethod'] = round((freq_pm['freq_PaymentMethod']/freq_pm['freq_PaymentMethod'].sum())*100,2)
     freq_pm
```

	PaymentMethod	freq_PaymentMethod	pct_PaymentMethod
0	Electronic check	2365	33.58
1	Mailed check	1612	22.89
2	Bank transfer (automatic)	1544	21.92
3	Credit card (automatic)	1522	21.61

1. Menghitung value di kolom "PaymentMethod"
2. Kemudian merubah nama kolom "PaymentMethod" menjadi "freq_PaymentMethod".

3. Mmembagi jumlah kemunculan masing-masing kota keberangkatan dengan total frekuensi untuk menghitung kemunculan masing-masing kota keberangkatan.

*100: mengalikan proporsi dengan 100 untuk mendapatkan nilai dalam bentuk persentase.

round(....,2): membulatkan nilai persentase menjadi 2 angka desimal.

```
df = df.merge(freq_pm[['PaymentMethod','pct_PaymentMethod']], on='PaymentMethod', how='inner')
df.head()
```

iorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	pct_PaymentMethod
0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No	33.58
0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes	33.58
0	No	No	8	Yes	Yes	Fiber optic	No	...	No	Yes	Yes	Month-to-month	Yes	Electronic check	99.65	820.5	Yes	33.58
0	Yes	No	28	Yes	Yes	Fiber optic	No	...	Yes	Yes	Yes	Month-to-month	Yes	Electronic check	104.80	3046.05	Yes	33.58
0	No	No	25	Yes	No	Fiber optic	Yes	...	Yes	Yes	Yes	Month-to-month	Yes	Electronic check	105.50	2686.05	No	33.58

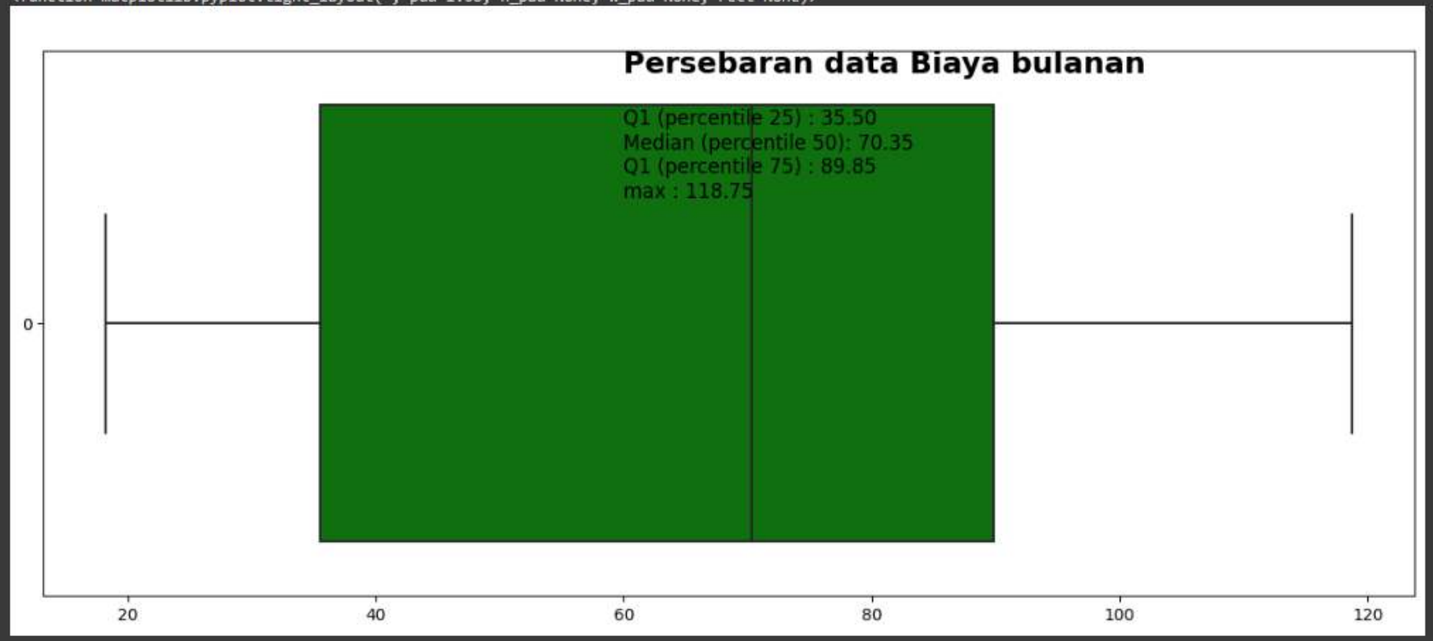
Melakukan penggabungan (merge) antara dua DataFrames, yaitu "df" dan "freq_et[['embark_town','pct_embark_town']]" berdasarkan kolom "embark_town".

ANOMALIES AND OUTLIER HANDLING

```
[89] fig, ax = plt.subplots(figsize=(15,6))
     sns.boxplot(df['MonthlyCharges'],color='green',orient='h')

     title = '''
     Persebaran data Biaya bulanan
     '''
     ax.text(60,-0.4,title,horizontalalignment='left',color='black',fontsize=18,fontweight='bold')

     text = '''
     Q1 (percentile 25) : 35.50
     Median (percentile 50): 70.35
     Q1 (percentile 75) : 89.85
     max : 118.75

     '''
     ax.text(60,-0.14,text,horizontalalignment='left',color='black',fontsize=12,fontweight='normal')

     plt.tight_layout
```

text = : didapatkan dari df.describe()

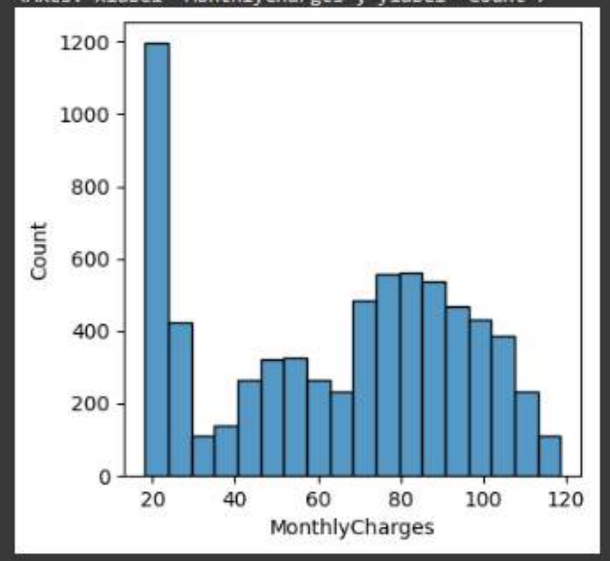
```
<function matplotlib.pyplot.tight_layout(*, pad=1.08, h_pad=None, w_pad=None, rect=None)>
```



Berdasarkan boxplot yang ditampilkan, dapat dilihat bahwa persebaran data biaya bulanan (MonthlyCharges) cukup luas, dengan nilai minimum sekitar 18 dan nilai maksimum sekitar 119. Median (percentile 50) berada di sekitar 70.

```
sns.histplot(df['MonthlyCharges'])
```

```
<Axes: xlabel='MonthlyCharges', ylabel='Count'>
```

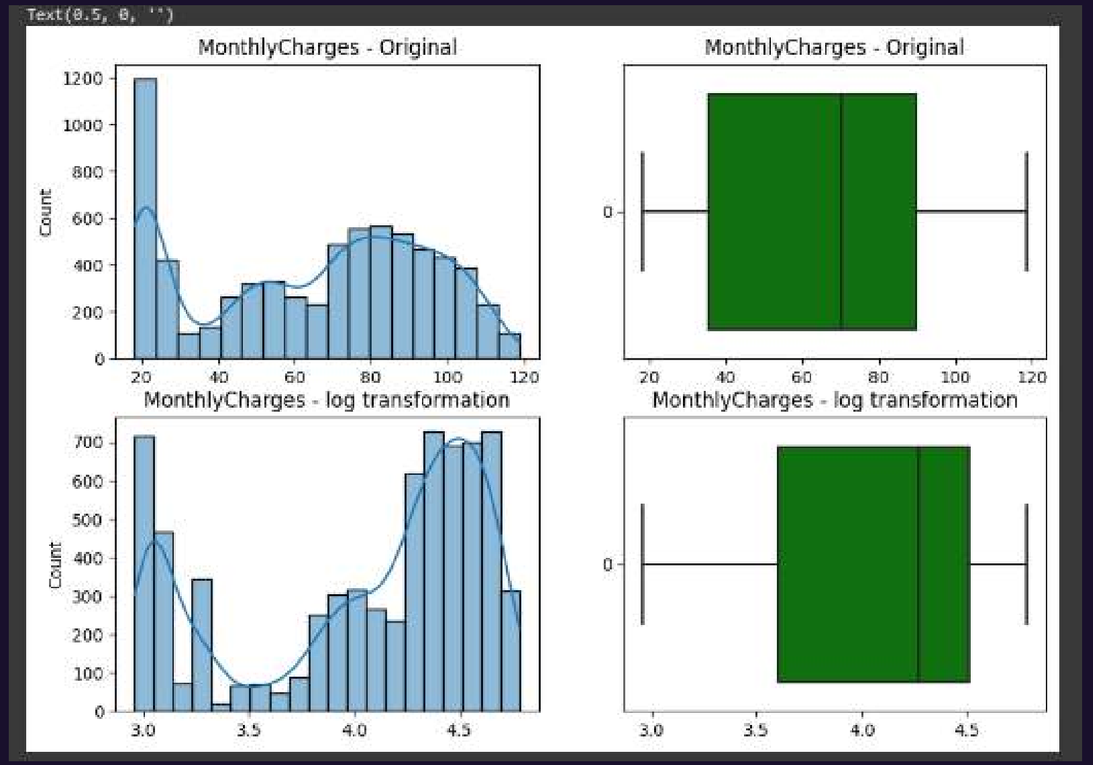


LOG TRANFORMATION

```
[91] df['log_MonthlyCharges'] = np.log(df['MonthlyCharges']+1)
     # Membuat kolom baru Log_Monthly Charge
```

```
[110] f,ax = plt.subplots(2,2,figsize=(10,7))

      g = sns.histplot(df['MonthlyCharges'],kde=True, ax=ax[0,0])
      ax[0,0].set_title('MonthlyCharges - Original')
      ax[0,0].set_xlabel('')

      g = sns.boxplot(df['MonthlyCharges'],color='green',orient='h', ax=ax[0,1])
      ax[0,1].set_title('MonthlyCharges - Original')
      ax[0,1].set_xlabel('')

      g = sns.histplot(np.log(df['MonthlyCharges']+1),kde=True, ax=ax[1,0])
      ax[1,0].set_title('MonthlyCharges - log transformation')
      ax[1,0].set_xlabel('')

      g = sns.boxplot(np.log(df['MonthlyCharges']+1),color='green',orient='h', ax=ax[1,1])
      ax[1,1].set_title('MonthlyCharges - log transformation')
      ax[1,1].set_xlabel('')
```

mengubah distribusi data sehingga menjadi lebih simetris dan terdistribusi secara normal.

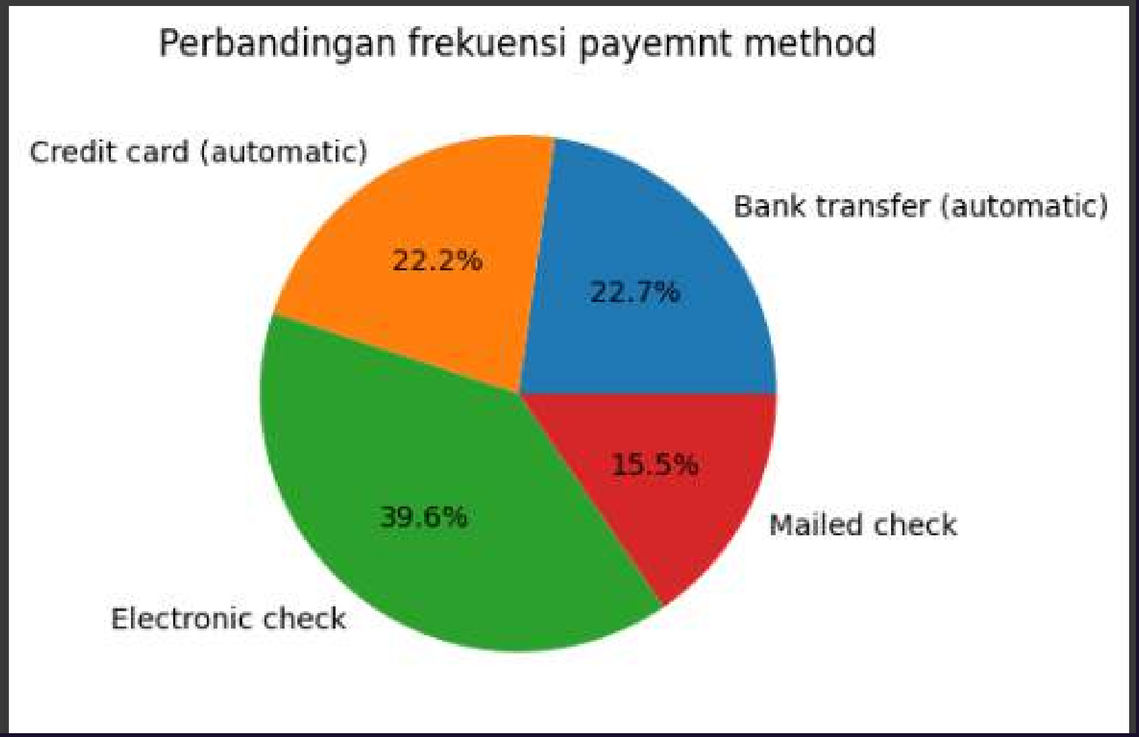


How We're Doing

A review of business year 2025

PERBANDINGAN FREKUENSI PENGGUNAAN PAYMENT METHOD

```
sums = df.groupby(df["PaymentMethod"])["MonthlyCharges"].sum()
plt.axis('equal');
plt.title('Perbandingan frekuensi payemnt method')
plt.pie(sums, labels=sums.index, autopct='%1.1f%%');
plt.rcParams['figure.figsize'] = (4,4)
plt.show()
```



Electronic check payment method paling sering digunakan oleh customer

RATA – RATA PENDAPATAN PERBULAN BERDASARKAN PAYMENT METHOD

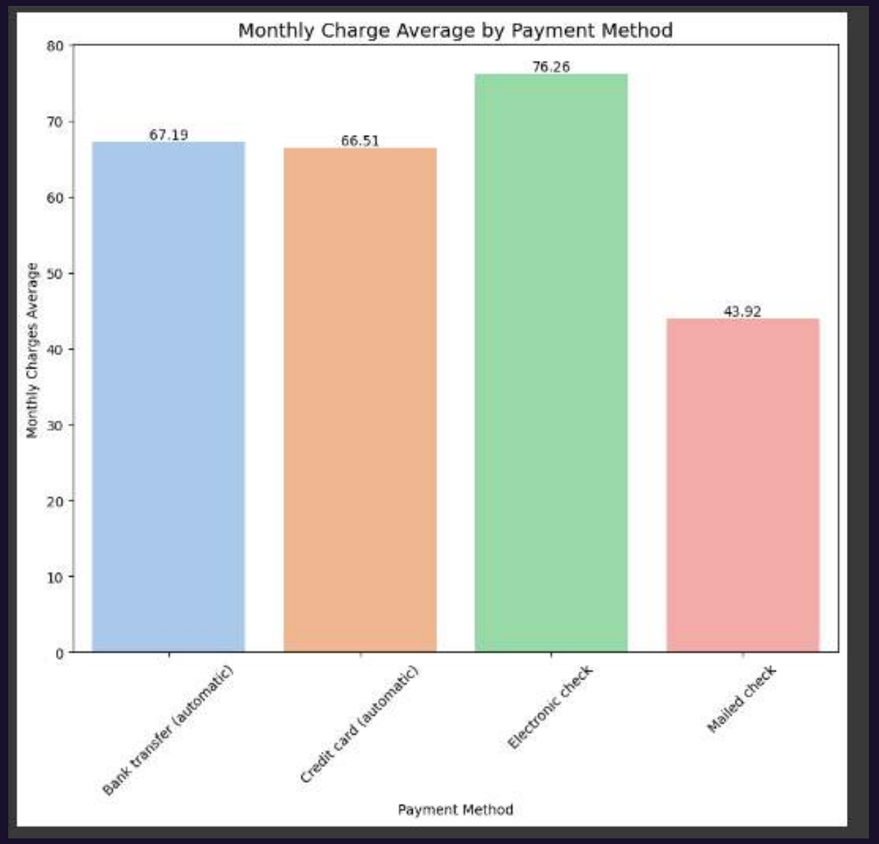
```
monthly_charge_avg = df.groupby('PaymentMethod').agg({'MonthlyCharges' : 'mean'}).reset_index()
monthly_charge_avg.columns = ['Payment Method', 'Monthly Charges Average']
monthly_charge_avg
```

	Payment Method	Monthly Charges Average
0	Bank transfer (automatic)	67.192649
1	Credit card (automatic)	66.512385
2	Electronic check	76.255814
3	Mailed check	43.917060

Menghitung rata-rata penghasilan perbulan berdasarkan payment method

```
# create barplot

fig, ax = plt.subplots(1, 1, figsize = (10,8))

sns.barplot(x = 'Payment Method',
            y = 'Monthly Charges Average',
            data=monthly_charge_avg,
            palette = 'pastel')
ax.set_title('Monthly Charge Average by Payment Method', fontsize = 14)
plt.xticks(rotation = 45)

# add annotation

x = monthly_charge_avg['Payment Method']
y = monthly_charge_avg['Monthly Charges Average']
for i in range(len(x)):
    ax.text(i, y[i], round(y[i],2), ha='center', va='bottom')
```

Membuat chart dengan pyhton



berdasarkan visualisasi data tersebut electronic check menghasilkan pemasukan yang tinggi di banding metode pembayaran lainya

Ini wajar terjadi karena metode electronic check adalah yang paling banyak digunakan oleh customer untuk membayar tagihannya. seperti apa yang di visualisasikan oleh pie chart sebelumnya

Thank you for listening!

Link Google Collab Telco Churn : https://colab.research.google.com/drive/1sqQnqrtJUm_Tz2TSbC87qP5E_v-v-fUo?usp=sharing

Link Google Collab Soal 6.ipynb : https://colab.research.google.com/drive/1uO_wJFF1Cg9wnVzmd0P72PVGGdVuYolR?usp=sharing