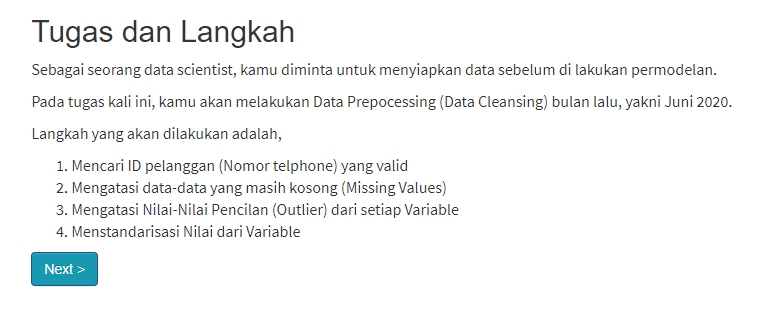
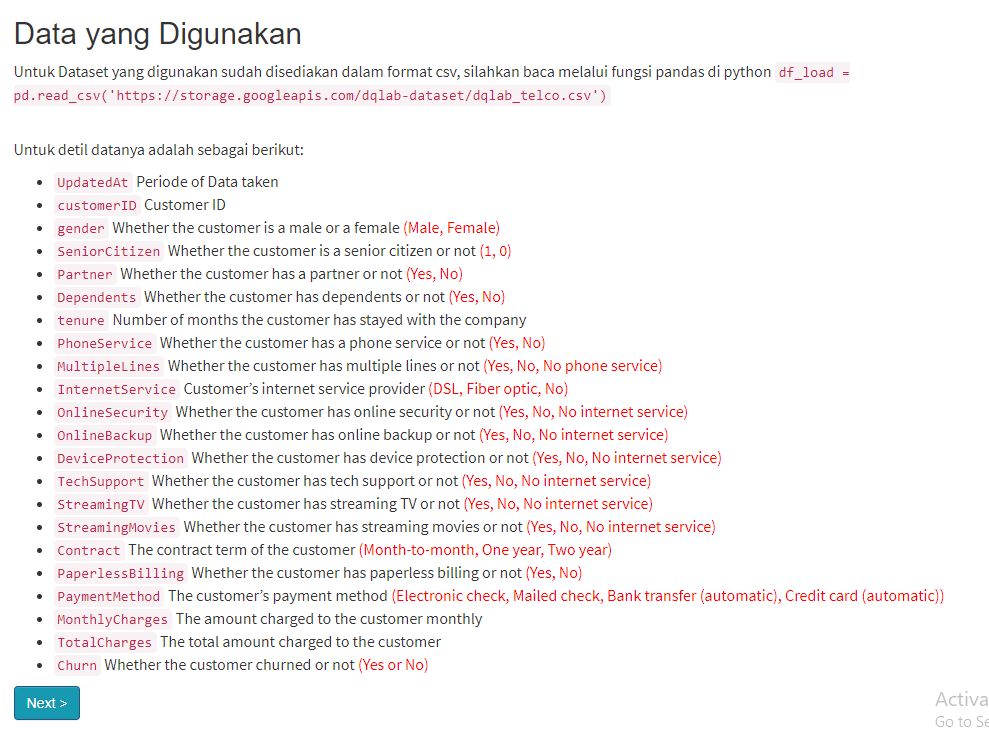
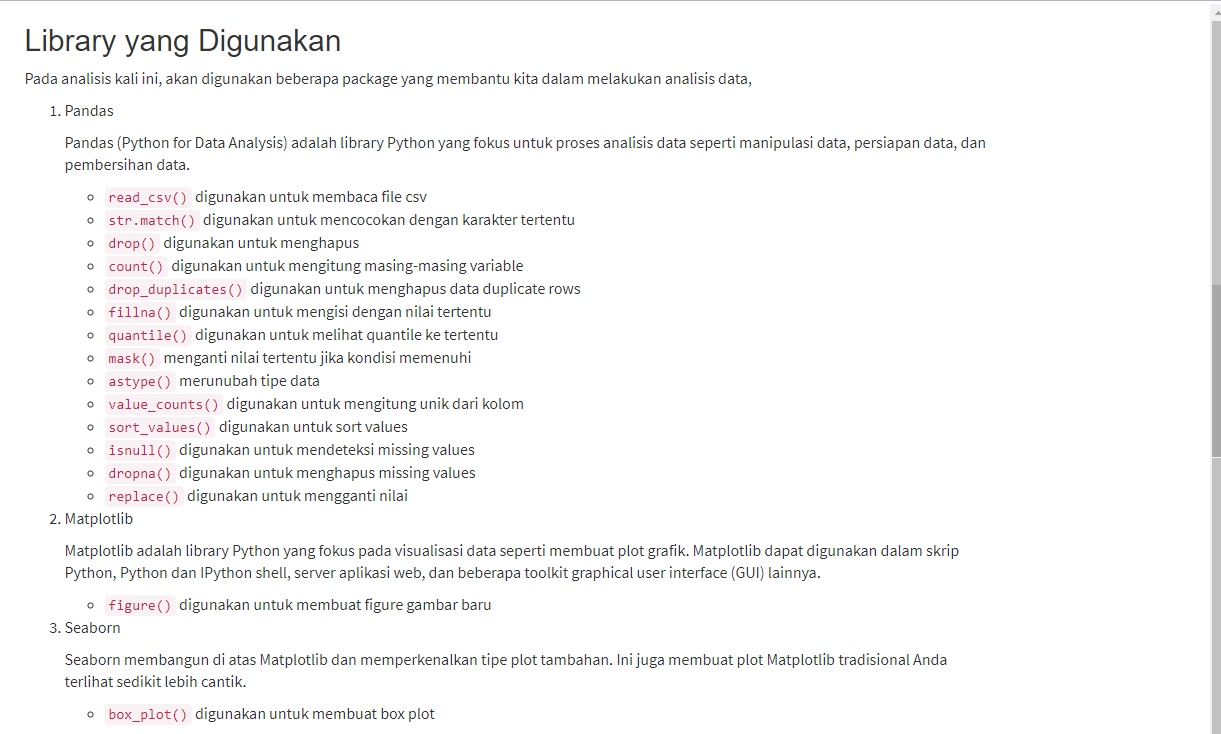
Data Science in Telco: Data Cleansing

Refrence : https://www.hendra.my.id/2020/09/data-science-in-telco-data-cleansing.html







#import library

import pandas as pd

pd.options.display.max\_columns = 50

#import dataset

df\_load = pd.read\_csv('https://storage.googleapis.com/dqlab-dataset/dqlab\_telco.csv')

#Tampilkan jumlah baris dan kolom

print(df\_load.shape)

#Tampilkan 5 data teratas

print(df\_load.head(5))

#Jumlah ID yang unik

print(df\_load.customerID.nunique())

(7113, 22)

UpdatedAt customerID gender SeniorCitizen Partner Dependents tenure \

0 202006 45759018157 Female 0 Yes No 1.0

1 202006 45557574145 Male 0 No No 34.0

2 202006 45366876421 Male 0 No No 2.0

3 202006 45779536532 Male 0 No No 45.0

4 202006 45923787906 Female 0 No No 2.0

PhoneService MultipleLines InternetService OnlineSecurity OnlineBackup \

0 No No phone service DSL No Yes

1 Yes No DSL Yes No

2 Yes No DSL Yes Yes

3 No No phone service DSL Yes No

4 Yes No Fiber optic No No

DeviceProtection TechSupport StreamingTV StreamingMovies Contract \

0 No No No No Month-to-month

1 Yes No No No One year

2 No No No No Month-to-month

3 Yes Yes No No One year

4 No No No No Month-to-month

PaperlessBilling PaymentMethod MonthlyCharges TotalCharges \

0 Yes Electronic check 29.85 29.85

1 No Mailed check 56.95 1889.50

2 Yes Mailed check 53.85 108.15

3 No Bank transfer (automatic) 42.30 1840.75

4 Yes Electronic check 70.70 151.65

Churn

0 No

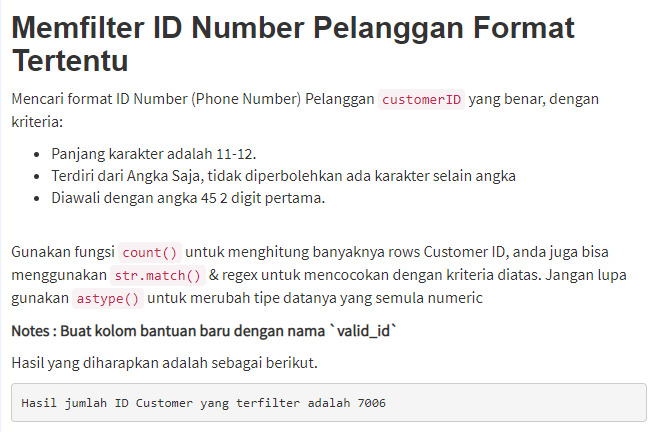
1 No

2 Yes

3 No

4 Yes

7017

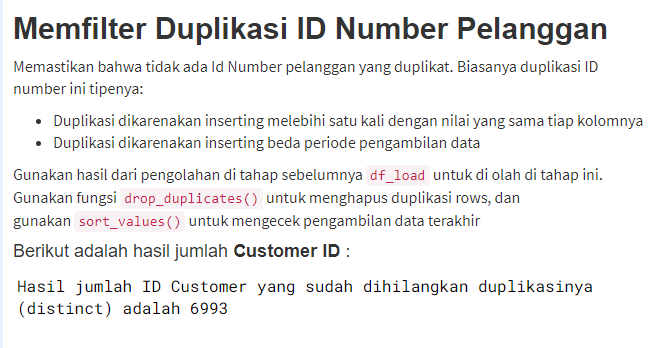


df\_load['valid\_id'] = df\_load['customerID'].astype(str).str.match (r'(45\d{9,10})')

df\_load = (df\_load[df\_load['valid\_id'] == True]).drop('valid\_id', axis = 1)

print('Hasil jumlah ID Customer yang terfilter adalah',df\_load['customerID'].count())

Hasil jumlah ID Customer yang terfilter adalah 7006



# Drop Duplicate Rows-Baris

df\_load.drop\_duplicates()

# Drop duplicate ID sorted by Periode -Urut periode

df\_load = df\_load.sort\_values('UpdatedAt', ascending=False).drop\_duplicates(['customerID'])

print('Hasil jumlah ID Customer yang sudah dihilangkan duplikasinya (distinct) adalah',df\_load['customerID'].count())

Hasil jumlah ID Customer yang sudah dihilangkan duplikasinya (distinct) adalah 6993



# Missing Values

print('Total missing values data dari kolom Churn',df\_load['Churn'].isnull().sum())

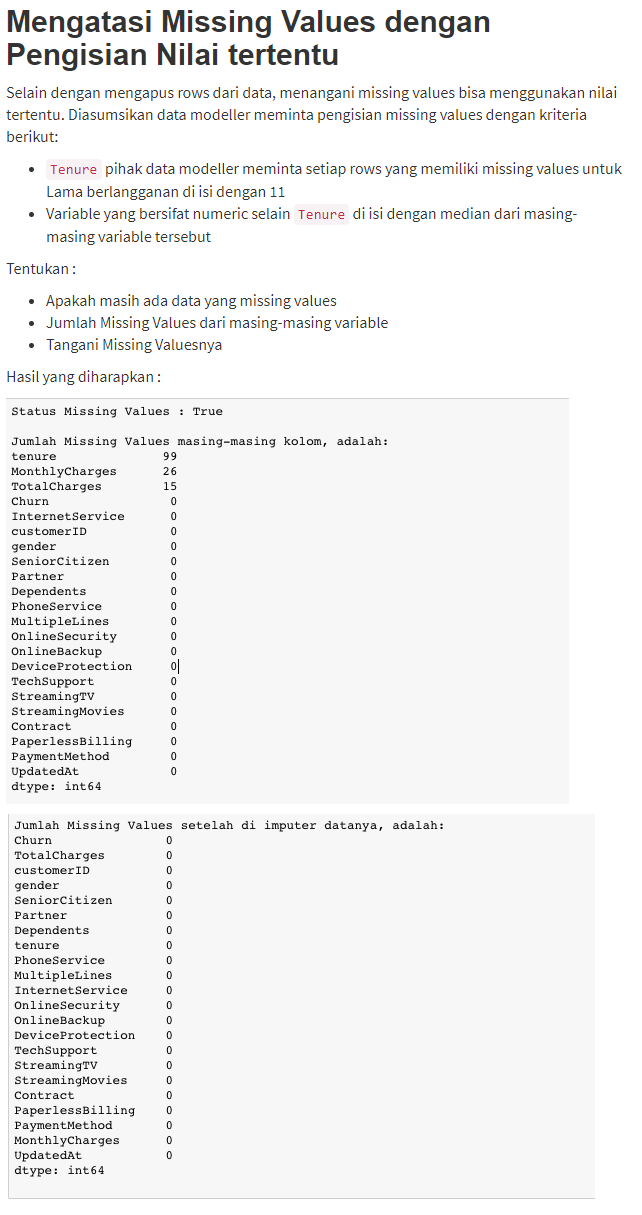
#Drop Rows and Column

df\_load.dropna(subset=['Churn'], inplace=True)

print('Total Rows dan kolom Data setelah dihapus data Missing Values adalah', df\_load.shape)

Total missing values data dari kolom Churn 0

Total Rows dan kolom Data setelah dihapus data Missing Values adalah (6950, 22)



print('Status Missing Values :',df\_load.isnull().values.any())

print('\nJumlah Missing Values masing-masing kolom, adalah:')

print(df\_load.isnull().sum().sort\_values(ascending=False))

#handling missing values Tenure fill with 11

df\_load['tenure'].fillna(11, inplace=True)

#Loop

#Handling missing values num vars (except Tenure)

for col\_name in list(['MonthlyCharges', 'TotalCharges']):

median = df\_load[col\_name].median()

df\_load[col\_name].fillna(median, inplace=True)

print('\nJumlah Missing Values setelah di imputer datanya, adalah:')

print(df\_load.isnull().sum().sort\_values(ascending=False))

#write your command here

Status Missing Values : False

Jumlah Missing Values masing-masing kolom, adalah:

Churn 0

TotalCharges 0

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

UpdatedAt 0

dtype: int64

Jumlah Missing Values setelah di imputer datanya, adalah:

Churn 0

TotalCharges 0

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

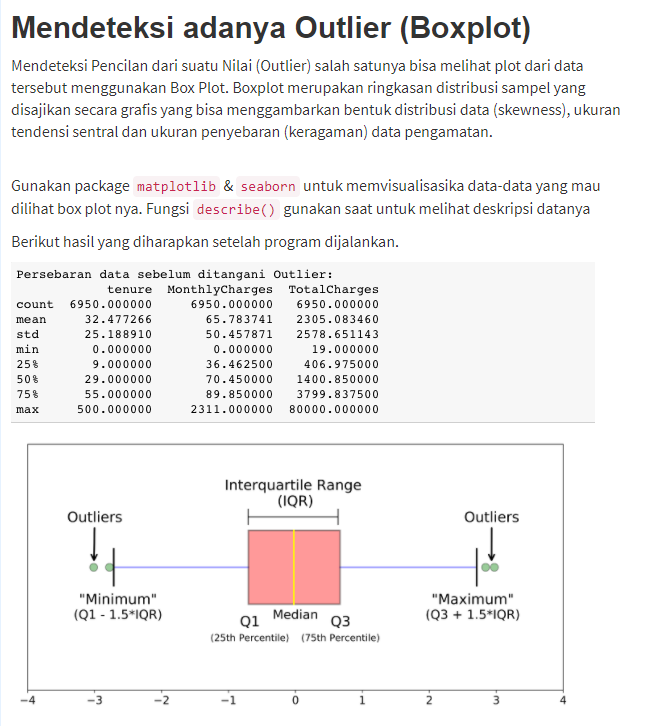
PaperlessBilling 0

PaymentMethod 0

MonthlyCharges 0

UpdatedAt 0

dtype: int64



print('\nPersebaran data sebelum ditangani Outlier: ')

print(df\_load[['tenure','MonthlyCharges','TotalCharges']].describe())

# Creating Box Plot

import matplotlib.pyplot as plt

import seaborn as sns

# Masukkan variable

plt.figure() # untuk membuat figure baru

sns.boxplot(x=df\_load['tenure'])

plt.show()

plt.figure() # untuk membuat figure baru

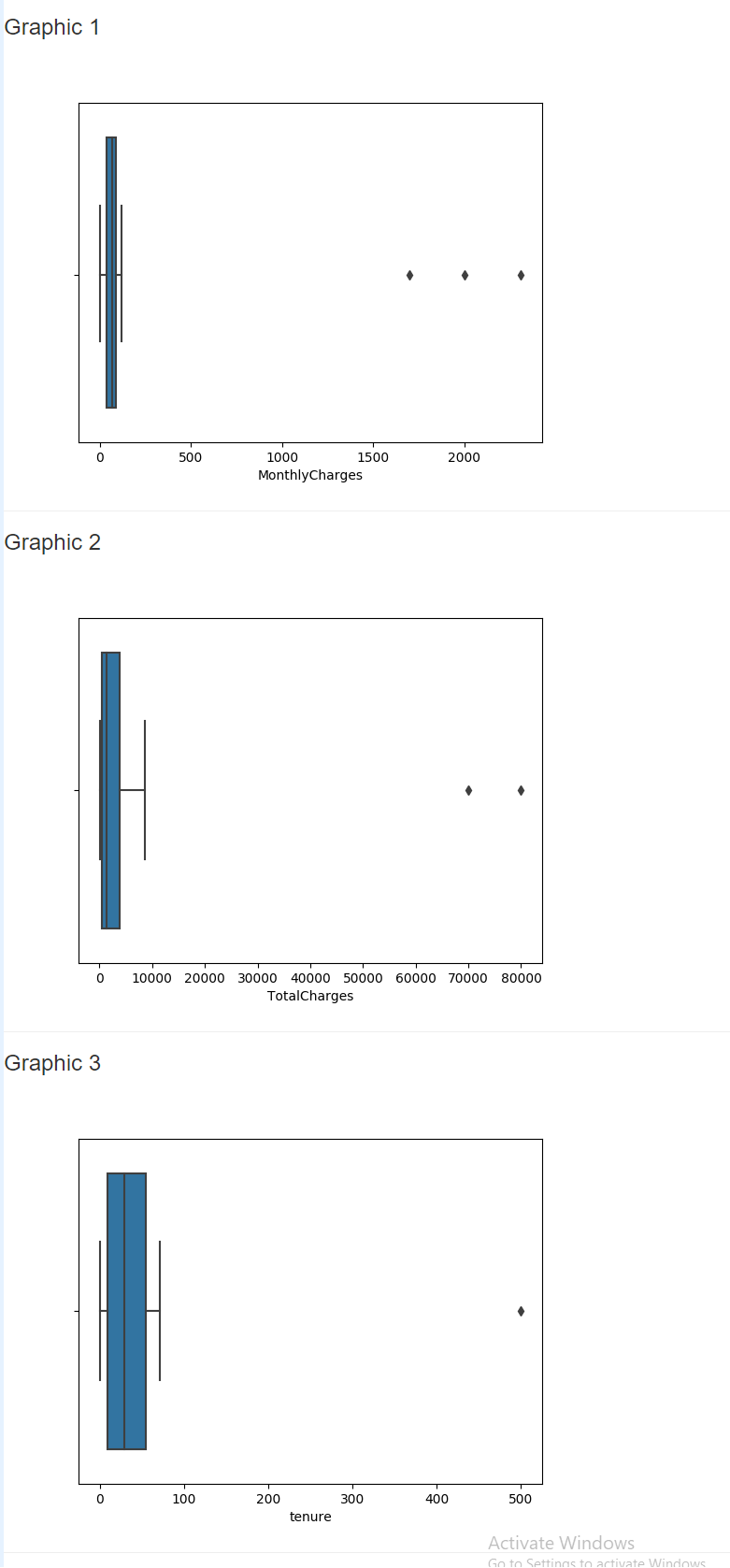
sns.boxplot(x=df\_load['MonthlyCharges'])

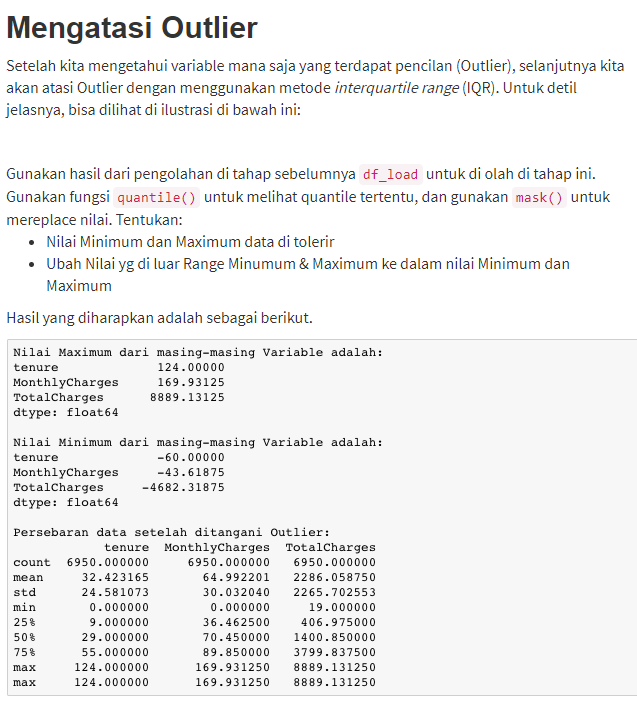
plt.show()

plt.figure() # untuk membuat figure baru

sns.boxplot(x=df\_load['TotalCharges'])

plt.show()





# Handling with IQR

Q1 = (df\_load[['tenure','MonthlyCharges','TotalCharges']]).quantile(0.25)

Q3 = (df\_load[['tenure','MonthlyCharges','TotalCharges']]).quantile(0.75)

IQR = Q3 - Q1

maximum = Q3 + (1.5\*IQR)

print('Nilai Maximum dari masing-masing Variable adalah: ')

print(maximum)

minimum = Q1 - (1.5\*IQR)

print('\nNilai Minimum dari masing-masing Variable adalah: ')

print(minimum)

more\_than = (df\_load > maximum)

lower\_than = (df\_load < minimum)

df\_load = df\_load.mask(more\_than, maximum, axis=1)

df\_load = df\_load.mask(lower\_than, minimum, axis=1)

print('\nPersebaran data setelah ditangani Outlier: ')

print(df\_load[['tenure','MonthlyCharges','TotalCharges']].describe())

Nilai Maximum dari masing-masing Variable adalah:

tenure 124.00000

MonthlyCharges 169.93125

TotalCharges 8889.13125

dtype: float64

Nilai Minimum dari masing-masing Variable adalah:

tenure -60.00000

MonthlyCharges -43.61875

TotalCharges -4682.31875

dtype: float64

Persebaran data setelah ditangani Outlier:

tenure MonthlyCharges TotalCharges

count 6950.000000 6950.000000 6950.000000

mean 32.423165 64.992201 2286.058750

std 24.581073 30.032040 2265.702553

min 0.000000 0.000000 19.000000

25% 9.000000 36.462500 406.975000

50% 29.000000 70.450000 1400.850000

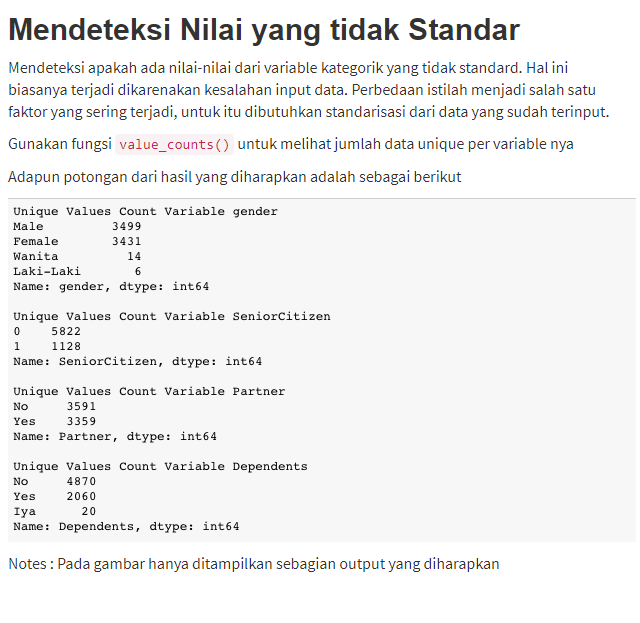
75% 55.000000 89.850000 3799.837500

max 124.000000 169.931250 8889.131250

Kesimpulan

Dari ketiga boxplot dengan variable *'tenure','MonthlyCharges' & 'TotalCharges'* terlihat jelas bahwasannya ada outlier. Hal ini bisa di identifikasi dari adanya titik titik yang berada jauh dari gambar boxplotnya. Kemudian kalo kita liat persebaran datanya dari kolom **max** nya juga ada nilai yang sangat tinggi sekali.

Kemudian nilai outlier tersebut ditangani dengan cara merubah nilainya ke nilai Maximum & Minimum dari *interquartile range* (IQR). Setelah di tangani outliernya, dan dilihat perseberan data nya, terlihat sudah tidak ada lagi nilai yang outlier.



#Loop

for col\_name in list(['gender','SeniorCitizen','Partner','Dependents','PhoneService','MultipleLines','InternetService','OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport','StreamingTV','StreamingMovies','Contract','PaperlessBilling','PaymentMethod','Churn']):

print('\nUnique Values Count \033[1m' + 'Before Standardized \033[0m Variable',col\_name)

print(df\_load[col\_name].value\_counts())

Unique Values Count Before Standardized Variable gender

Male 3499

Female 3431

Wanita 14

Laki-Laki 6

Name: gender, dtype: int64

Unique Values Count Before Standardized Variable SeniorCitizen

0 5822

1 1128

Name: SeniorCitizen, dtype: int64

Unique Values Count Before Standardized Variable Partner

No 3591

Yes 3359

Name: Partner, dtype: int64

Unique Values Count Before Standardized Variable Dependents

No 4870

Yes 2060

Iya 20

Name: Dependents, dtype: int64

Unique Values Count Before Standardized Variable PhoneService

Yes 6281

No 669

Name: PhoneService, dtype: int64

Unique Values Count Before Standardized Variable MultipleLines

No 3346

Yes 2935

No phone service 669

Name: MultipleLines, dtype: int64

Unique Values Count Before Standardized Variable InternetService

Fiber optic 3057

DSL 2388

No 1505

Name: InternetService, dtype: int64

Unique Values Count Before Standardized Variable OnlineSecurity

No 3454

Yes 1991

No internet service 1505

Name: OnlineSecurity, dtype: int64

Unique Values Count Before Standardized Variable OnlineBackup

No 3045

Yes 2400

No internet service 1505

Name: OnlineBackup, dtype: int64

Unique Values Count Before Standardized Variable DeviceProtection

No 3054

Yes 2391

No internet service 1505

Name: DeviceProtection, dtype: int64

Unique Values Count Before Standardized Variable TechSupport

No 3431

Yes 2014

No internet service 1505

Name: TechSupport, dtype: int64

Unique Values Count Before Standardized Variable StreamingTV

No 2774

Yes 2671

No internet service 1505

Name: StreamingTV, dtype: int64

Unique Values Count Before Standardized Variable StreamingMovies

No 2747

Yes 2698

No internet service 1505

Name: StreamingMovies, dtype: int64

Unique Values Count Before Standardized Variable Contract

Month-to-month 3823

Two year 1670

One year 1457

Name: Contract, dtype: int64

Unique Values Count Before Standardized Variable PaperlessBilling

Yes 4114

No 2836

Name: PaperlessBilling, dtype: int64

Unique Values Count Before Standardized Variable PaymentMethod

Electronic check 2337

Mailed check 1594

Bank transfer (automatic) 1519

Credit card (automatic) 1500

Name: PaymentMethod, dtype: int64

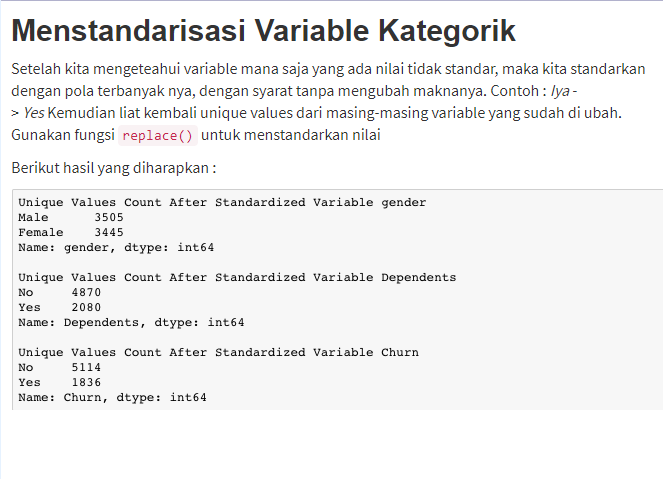
Unique Values Count Before Standardized Variable Churn

No 5114

Yes 1827

Churn 9

Name: Churn, dtype: int64



df\_load = df\_load.replace(['Wanita','Laki-Laki','Churn','Iya'],['Female','Male','Yes','Yes'])

# Masukkan variable

for col\_name in list(['gender','Dependents','Churn']):

print('\nUnique Values Count \033[1m' + 'After Standardized \033[0mVariable',col\_name)

print(df\_load[col\_name].value\_counts())

Unique Values Count After Standardized Variable gender

Male 3505

Female 3445

Name: gender, dtype: int64

Unique Values Count After Standardized Variable Dependents

No 4870

Yes 2080

Name: Dependents, dtype: int64

Unique Values Count After Standardized Variable Churn

No 5114

Yes 1836

Name: Churn, dtype: int64

