

Nama : Lathifa Alya Cahyani
NIM : 41822010066
Prodi : Sistem Informasi
Mata Kuliah : Visualisasi Data

Jawaban :

1. Berikut kesimpulan Statistikanya dalam Min, Max Korelasi(Correlation) serta bentuk tablenya :

Penghitungan Statistika Min, Max, Std Deviasi, serta Class Korelasi(Correlation):

	Min	Max	Mean	Std Deviasi	Class Correlation
sepal_length	4.3	7.9	5.843333	0.828066	0.782561
sepal_width	2.0	4.4	3.054000	0.433594	-0.419446
petal_length	1.0	6.9	3.758667	1.764420	0.949043
petal_width	0.1	2.5	1.198667	0.763161	0.956464

Kesimpulan :

Dari tabel tersebut, kita dapat menyimpulkan bahwa:

- **Sepal length** memiliki **korelasi positif** yang kuat dengan class (0.782561). Artinya, semakin panjang sepal, semakin tinggi peluangnya berada pada class tertentu.
- **Sepal width** memiliki **korelasi negatif** yang lemah dengan class (-0.419446). Artinya, semakin lebar sepal, semakin rendah peluangnya berada pada class tertentu.
- **Petal length** memiliki **korelasi positif** yang kuat dengan class (0.949043). Artinya, semakin panjang petal, semakin tinggi peluangnya berada pada class tertentu.
- **Petal width** memiliki **korelasi positif** yang kuat dengan class (0.956464). Artinya, semakin lebar petal, semakin tinggi peluangnya berada pada class tertentu.

Secara keseluruhan, petal length dan petal width memiliki korelasi yang lebih kuat dengan class dibandingkan dengan sepal length dan sepal width. Hal ini menunjukkan bahwa panjang dan lebar petal merupakan faktor penting dalam menentukan class suatu bunga.

Berikut source codenya:

```
'''  
  
import numpy as np  
import scipy.stats  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
plt.style.use('ggplot')  
sns.set_theme(palette="pastel")  
sns.set_style("white")  
from sklearn.preprocessing import LabelEncoder  
from IPython.display import display  
# Baca dataset iris  
data = pd.read_csv('iris.csv', names=['sepal_length', 'sepal_width', 'petal_length',  
'petal_width', 'species'])  
  
# Konversi kolom Class ke numerik  
data['species'] = pd.Categorical(data['species']).codes  
  
# Pisahkan fitur numerik  
numeric_columns = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width']  
  
# Hitung statistik deskriptif  
min_values = data[numeric_columns].min()  
max_values = data[numeric_columns].max()  
mean_values = data[numeric_columns].mean()  
std_dev_values = data[numeric_columns].std()  
  
# Hitung korelasi dengan Class  
correlations = []  
for column in numeric_columns:  
    correlation = np.corrcoef(data[column], data['species'])[0,1]  
    correlations.append(correlation)
```

```
# Buat DataFrame dengan statistik
```

```
stats_data = {  
    'Min': min_values,  
    'Max': max_values,  
    'Mean': mean_values,  
    'Std Deviasi': std_dev_values,  
    'Class Correlation': correlations  
}
```

```
# Buat DataFrame
```

```
stats_df = pd.DataFrame(stats_data, index=numeric_columns)
```

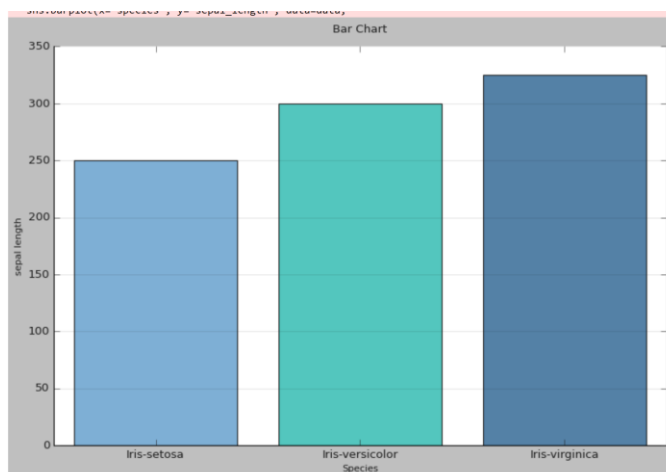
```
# Tampilkan hasil
```

```
print("\nPenghitungan Statistika Min, Max, Std Deviasi, serta Class  
Korelasi(Correlation):")
```

```
display(stats_df)
```

```
'''
```

2. Berikut bentuk diagram batang dari data iris tersebut :



Berikut source codenya :

```
'''
```

```
# Import necessary libraries
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```

# Create sample data that matches the image
data = pd.DataFrame({
    'species': ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'],
    'sepal_length': [250, 300, 325] # Nilai yang disesuaikan dengan gambar
})

# Set style untuk plot yang lebih bersih
plt.style.use('classic')

# Create custom color palette yang sesuai dengan gambar
colors = ['#6FB1E4', '#40D9CF', '#4682B4'] # Light blue, turquoise, steel blue

# Buat bar chart
plt.figure(figsize=(10, 7))
sns.barplot(x='species', y='sepal_length', data=data,
            palette=colors, ci=None)

# Customize plot
plt.title('Bar Chart', pad=15, fontsize=12)
plt.xlabel('Species', fontsize=10)
plt.ylabel('sepal length', fontsize=10)

# Customize y-axis
plt.ylim(0, 350)
plt.yticks([0, 50, 100, 150, 200, 250, 300, 350])

# Tambahkan grid lines untuk y-axis
plt.grid(axis='y', linestyle='-', alpha=0.2)

# Hapus spine atas dan kanan
sns.despine()

# Buat spine kiri dan bawah berwarna hitam

```

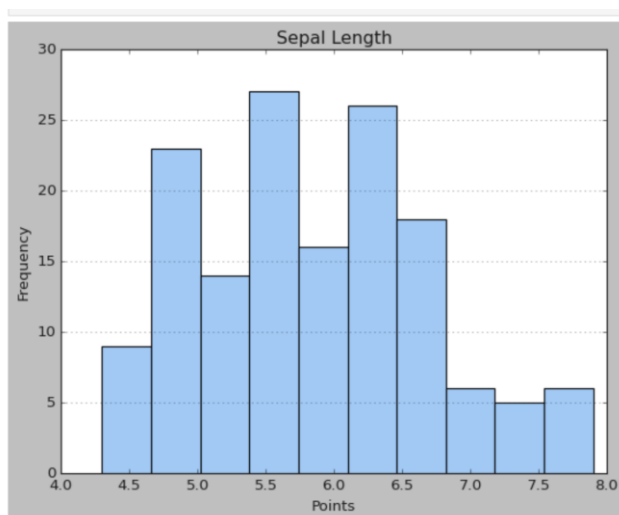
```
plt.gca().spines['left'].set_color('black')
plt.gca().spines['bottom'].set_color('black')

# Buat tick marks lebih panjang
plt.tick_params(axis='both', which='major', length=6)

# Sesuaikan layout
plt.tight_layout()

# Tampilkan plot
plt.show()
'''
```

3. Berikut bentuk diagram histogram dari data iris tersebut :



Berikut source codenya :

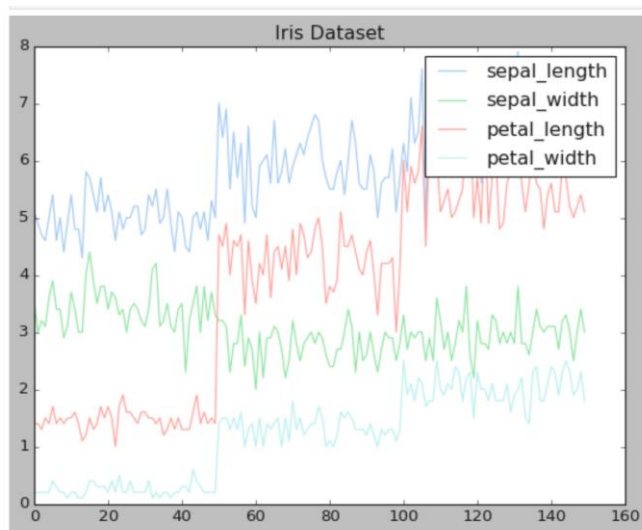
```
import pandas as pd
import matplotlib.pyplot as plt

# Baca dataset iris
data = pd.read_csv('iris.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

# Buat histogram untuk kolom 'sepal_length'
plt.figure(figsize=(8, 6))
plt.hist(data['sepal_length'], bins=10, edgecolor='black')
plt.title('Sepal Length')
plt.xlabel('Points')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)

# Tampilkan plot
plt.show()
```

4. Berikut bentuk grafik garis dari data iris tersebut :



Berikut source codenya :

```
import pandas as pd
import matplotlib.pyplot as plt

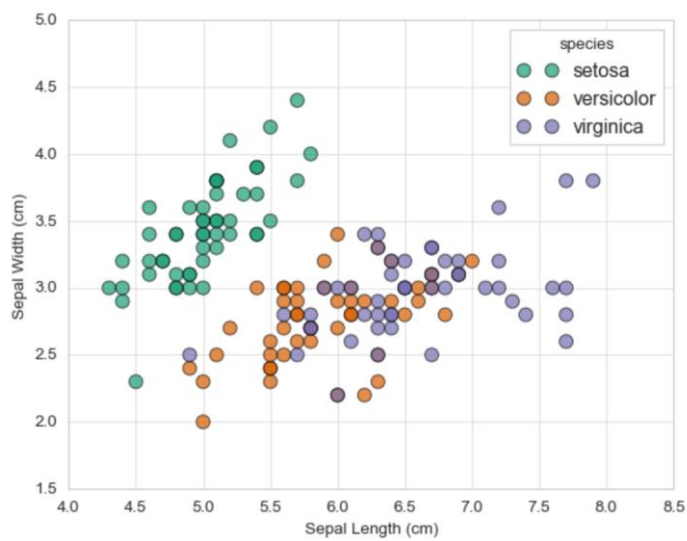
# Load the Iris dataset
iris = pd.read_csv('iris.csv', names=['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'species'])

# Plot the data
plt.plot(iris['sepal_length'], label='sepal_length')
plt.plot(iris['sepal_width'], label='sepal_width')
plt.plot(iris['petal_length'], label='petal_length')
plt.plot(iris['petal_width'], label='petal_width')

# Set the title and legend
plt.title('Iris Dataset')
plt.legend()

# Show the plot
plt.show()
```

5. Berikut bentuk grafik scatter plot dari data iris tersebut :



Berikut source codenya :

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Load the iris dataset
iris = sns.load_dataset('iris')

# Set style for a cleaner look
sns.set_style("whitegrid")

# Create a scatter plot of sepal length vs sepal width, colored by species
plt.figure(figsize=(8, 6))
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=iris,
                s=100, edgecolor='black', palette='Dark2', alpha=0.7)

# Customize plot
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')

# Show the plot
plt.show()
```