

Tugas Mandiri Pertemuan 6_Andrea Yonathan_Institut Teknologi Sepuluh Nopember

Latihan (1)

Sebelum menelaah data dengan metode visualisasi, kita perlu memanggil modul visualisasi (seaborn & matplotlib) terlebih dahulu.

```
In [1]: # memanggil modul Pandas and Seaborn
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

import warnings
warnings.filterwarnings('ignore')

In [2]: bunga = 'Iris.csv'
df = pd.read_csv(bunga)

In [3]: # menampilkan 5 baris data
df.head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Latihan (2)

Karena kita tidak membutuhkan kolom "Id" dalam melakukan visualisasi kita dapat menghapus kolom "Id" menggunakan fungsi .drop()

```
In [4]: # menghapus kolom "Id"
df = df.drop(columns = 'Id', axis = 1)
df.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Latihan (3)

Lakukan pengecekan nilai yang hilang (missing value) pada dataset. Dengan function info()

```
In [5]: # memeriksa missing values pada dataset
df.info()

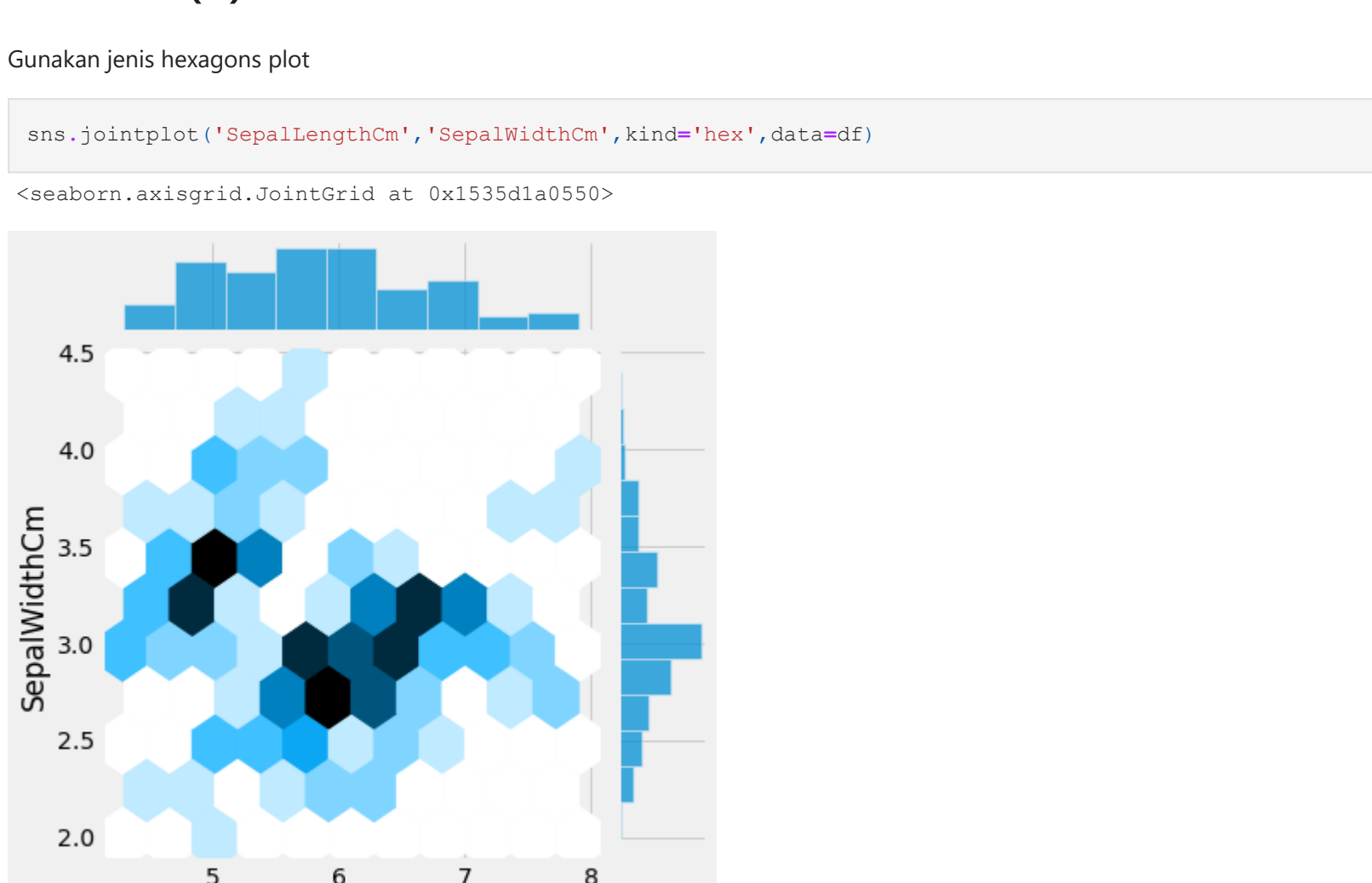
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ---
 0   SepalLengthCm         150 non-null    float64
 1   SepalWidthCm          150 non-null    float64
 2   PetalLengthCm         150 non-null    float64
 3   PetalWidthCm          150 non-null    float64
 4   Species               150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

Latihan (4)

Tampilkan visualisasi dari data yang telah menggunakan fungsi describe() untuk mendapatkan informasi umum statistik tentang dataset

```
In [6]: # melakukan visualisasi dari data describe

In [7]: df.describe().plot(kind = 'bar', figsize = (10,5), colormap='Blues')
plt.xlabel('Statistics',fontsize=14)
plt.ylabel('Value',fontsize=14)
plt.title("General Statistics of Iris Dataset\n",fontsize=18)
plt.show()
```



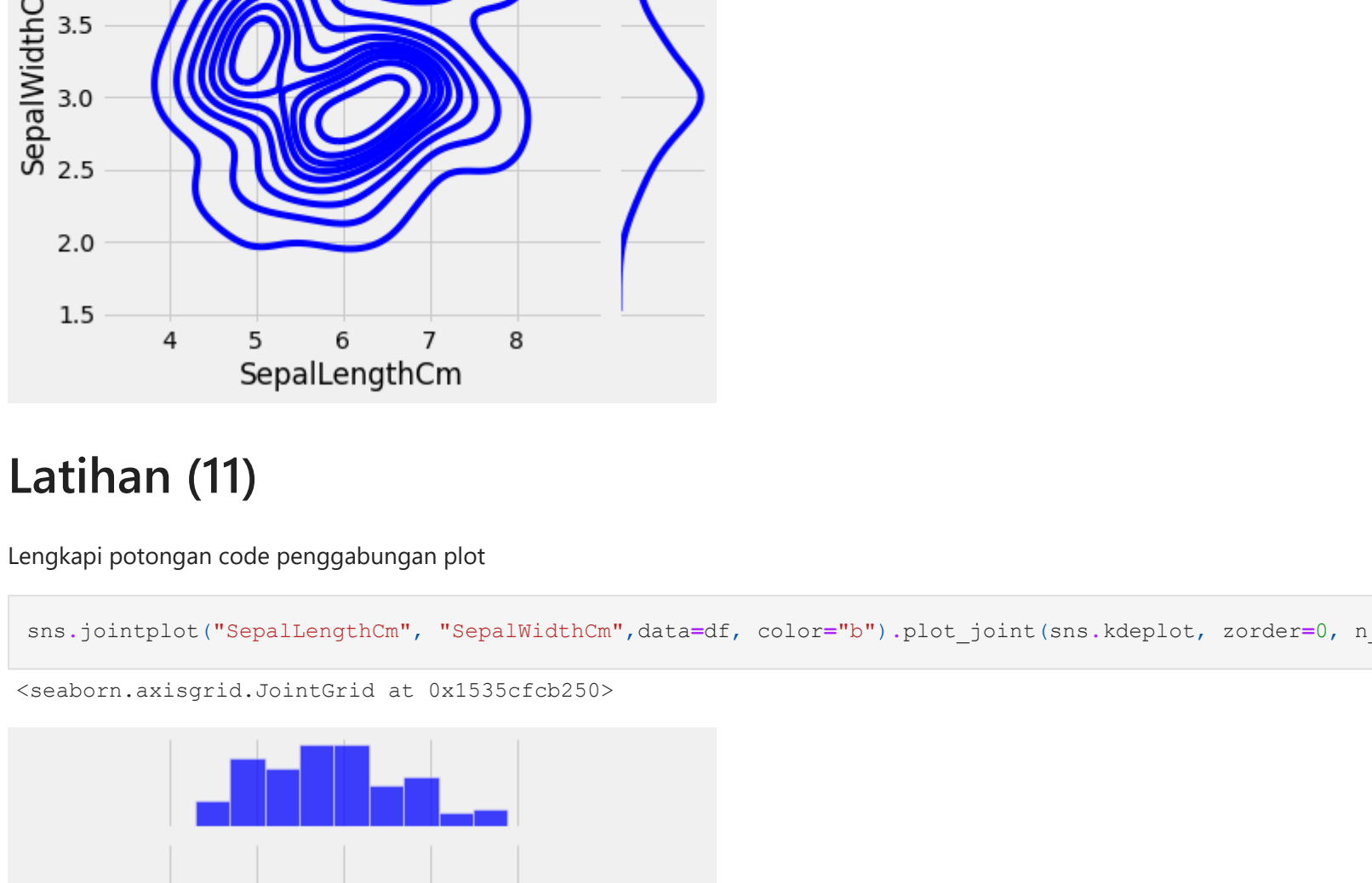
Latihan (5)

Tampilkan visualisasi bar plot dan pie chart untuk menghitung frekuensi setiap species dalam dataset iris

```
In [8]: # visualisasi bar plot dan pie chart

In [9]: f,ax=plt.subplots(1,2,figsize=(18,8))
df['Species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],autopct='%1.1f%%',ax=ax[0],shadow=True)
ax[0].set_title('Iris Species Count')
sns.countplot('Species',data=df,ax=ax[1])
ax[1].set_title('Iris Species Count')

Out[9]: Text(0.5, 1.0, 'Iris Species Count')
```

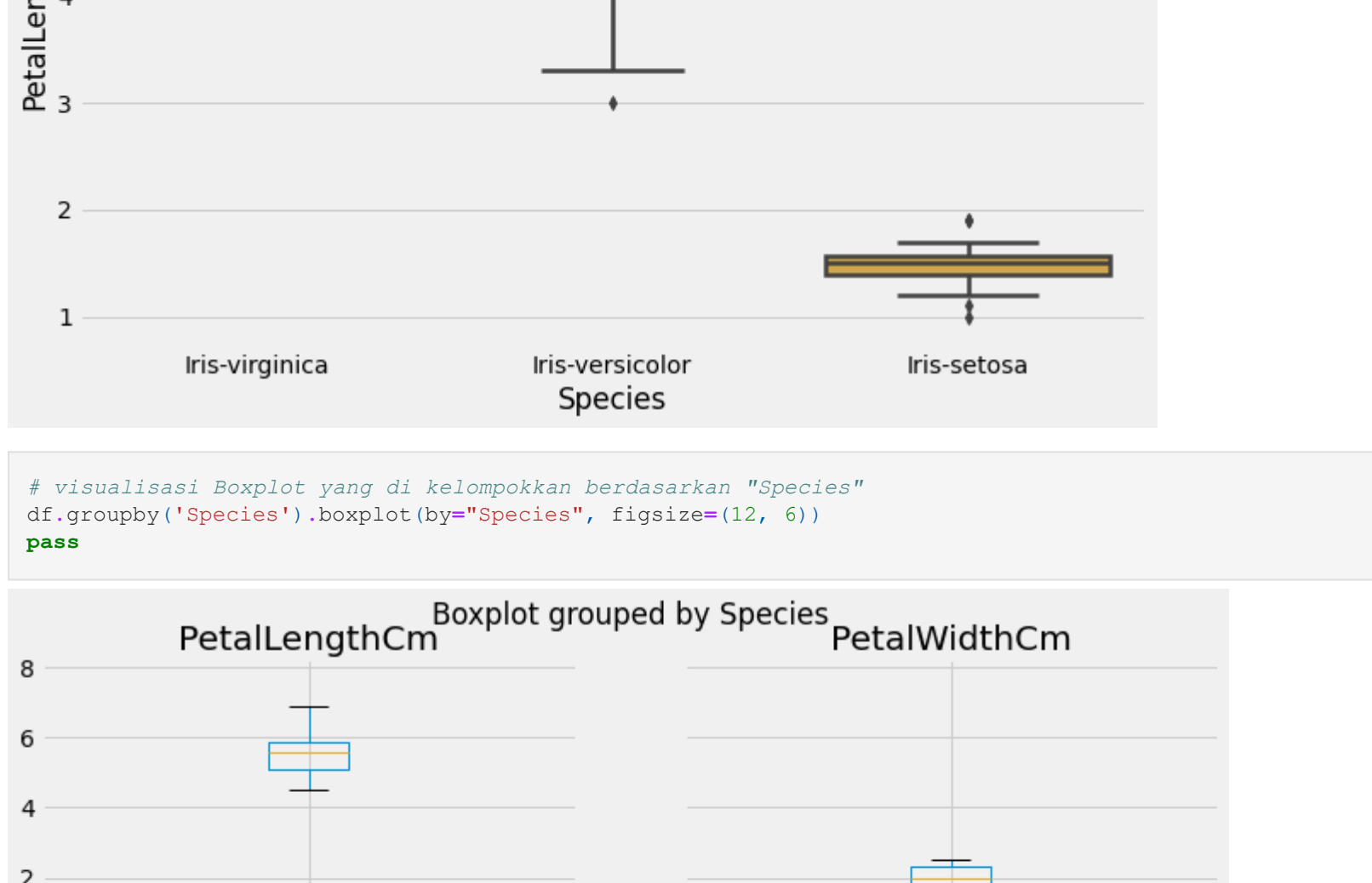


Latihan (6)

Gunakan jenis plot residual

```
In [10]: sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind='resid',data=df)

Out[10]: <seaborn.axisgrid.JointGrid at 0x1535d188b50>
```

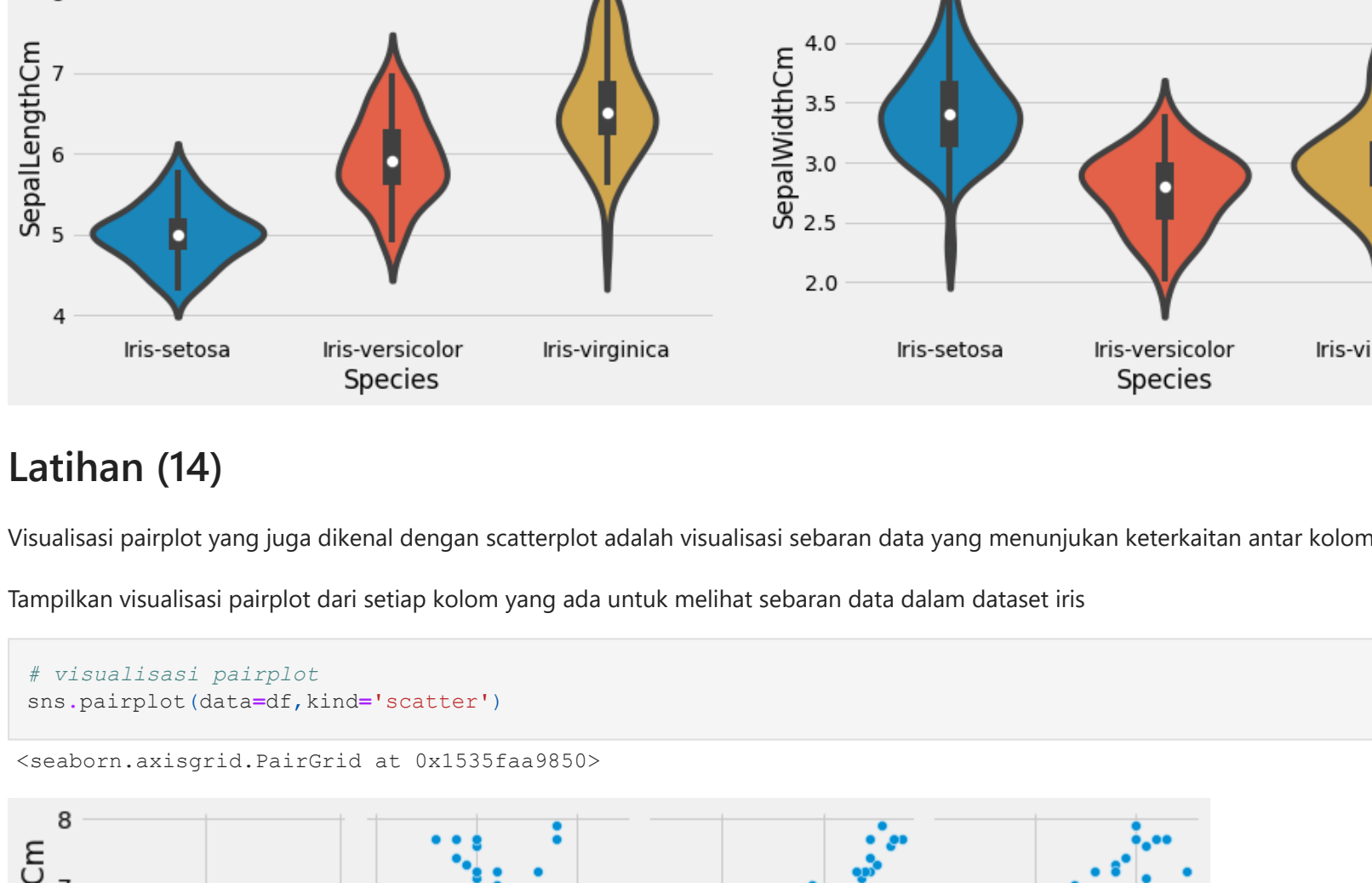


Latihan (7)

Gunakan jenis scatter plot

```
In [11]: sns.jointplot(x='SepalLengthCm',y='SepalWidthCm',kind='scatter',data=df)

Out[11]: <seaborn.axisgrid.JointGrid at 0x1535d1e1a9a0>
```

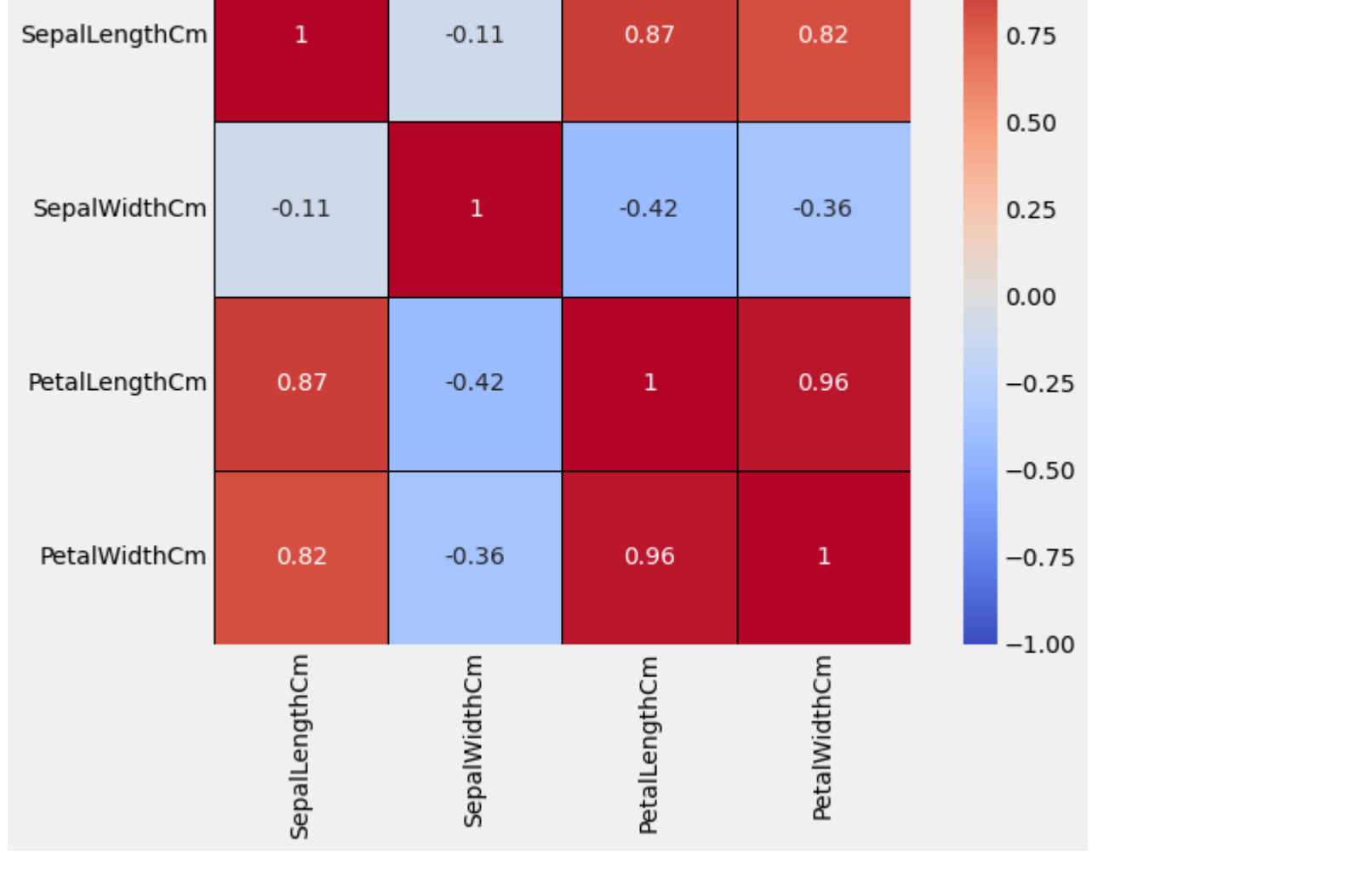


Latihan (8)

Gunakan jenis hexagons plot

```
In [12]: sns.jointplot('SepalLengthCm', 'SepalWidthCm', kind='hex', data=df)

Out[12]: <seaborn.axisgrid.JointGrid at 0x1535d1efcd50>
```

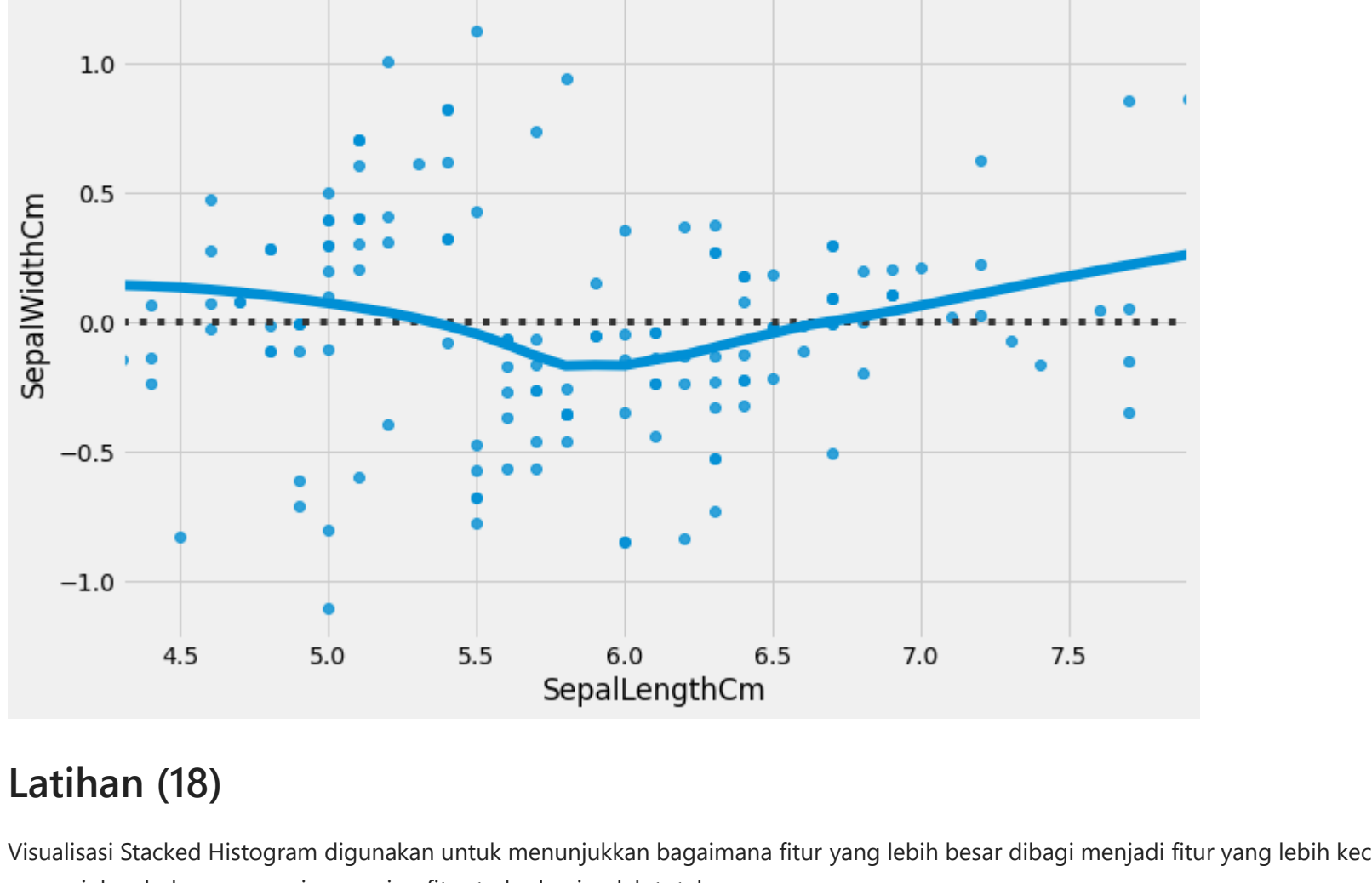


Latihan (9)

Gunakan jenis Linear regression line plot

```
In [13]: sns.jointplot('SepalLengthCm', 'SepalWidthCm', data=df, kind='reg')

Out[13]: <seaborn.axisgrid.JointGrid at 0x1535d1ef7000>
```

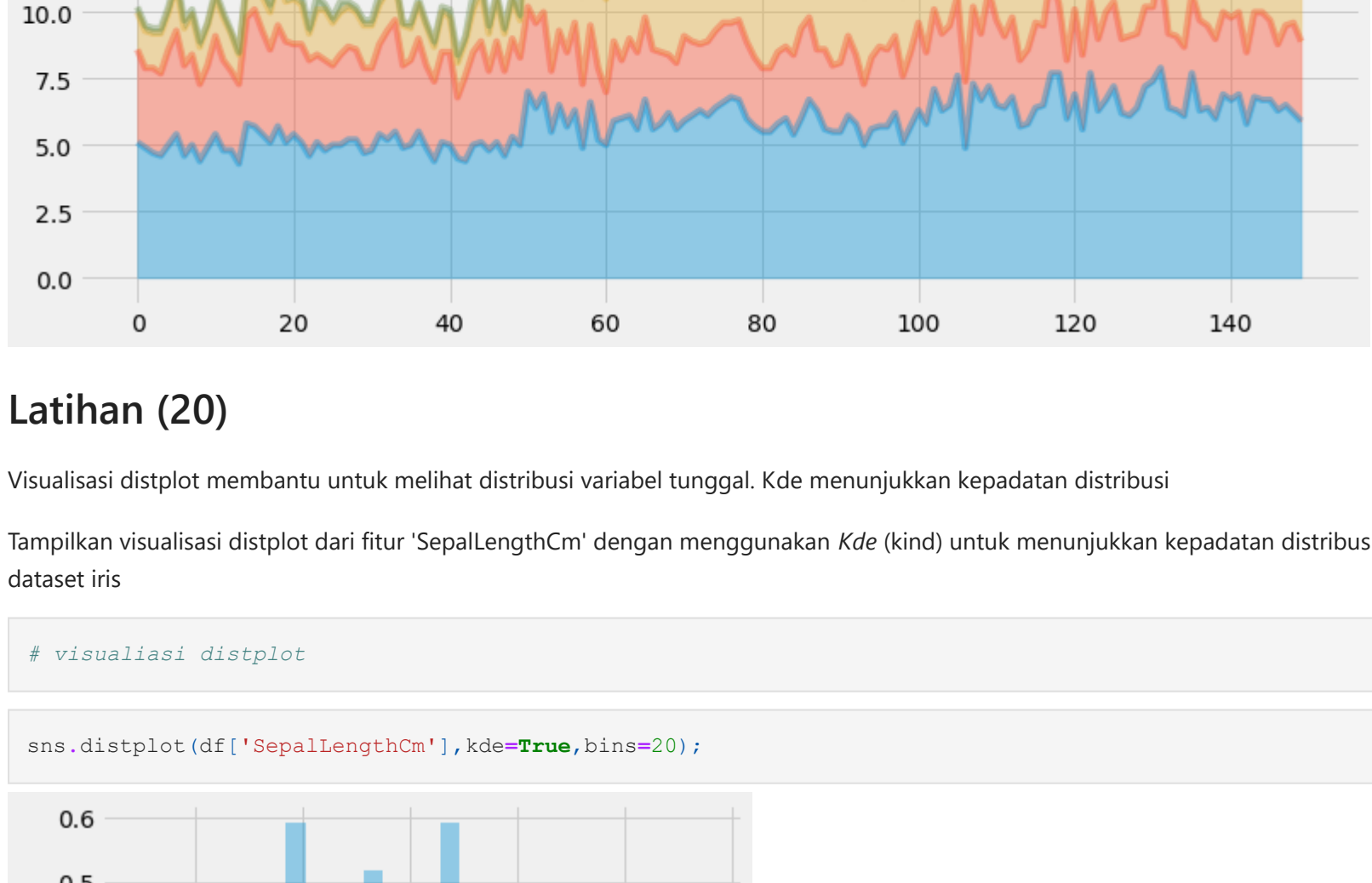


Latihan (10)

Gunakan jenis kernel density estimate plot

```
In [14]: sns.jointplot("SepalLengthCm", "SepalWidthCm", data=df, kind='kde',color='b')

Out[14]: <seaborn.axisgrid.JointGrid at 0x1535d1efcd60>
```



Latihan (11)

Lengkapi potongan code penggabungan plot

```
In [15]: sns.jointplot("SepalLengthCm", "SepalWidthCm",data=df, color='b').plot_joint(sns.kdeplot, sorder=0, n_levels=5)

Out[15]: <seaborn.axisgrid.JointGrid at 0x1535d1ef250>
```


Latihan (12)

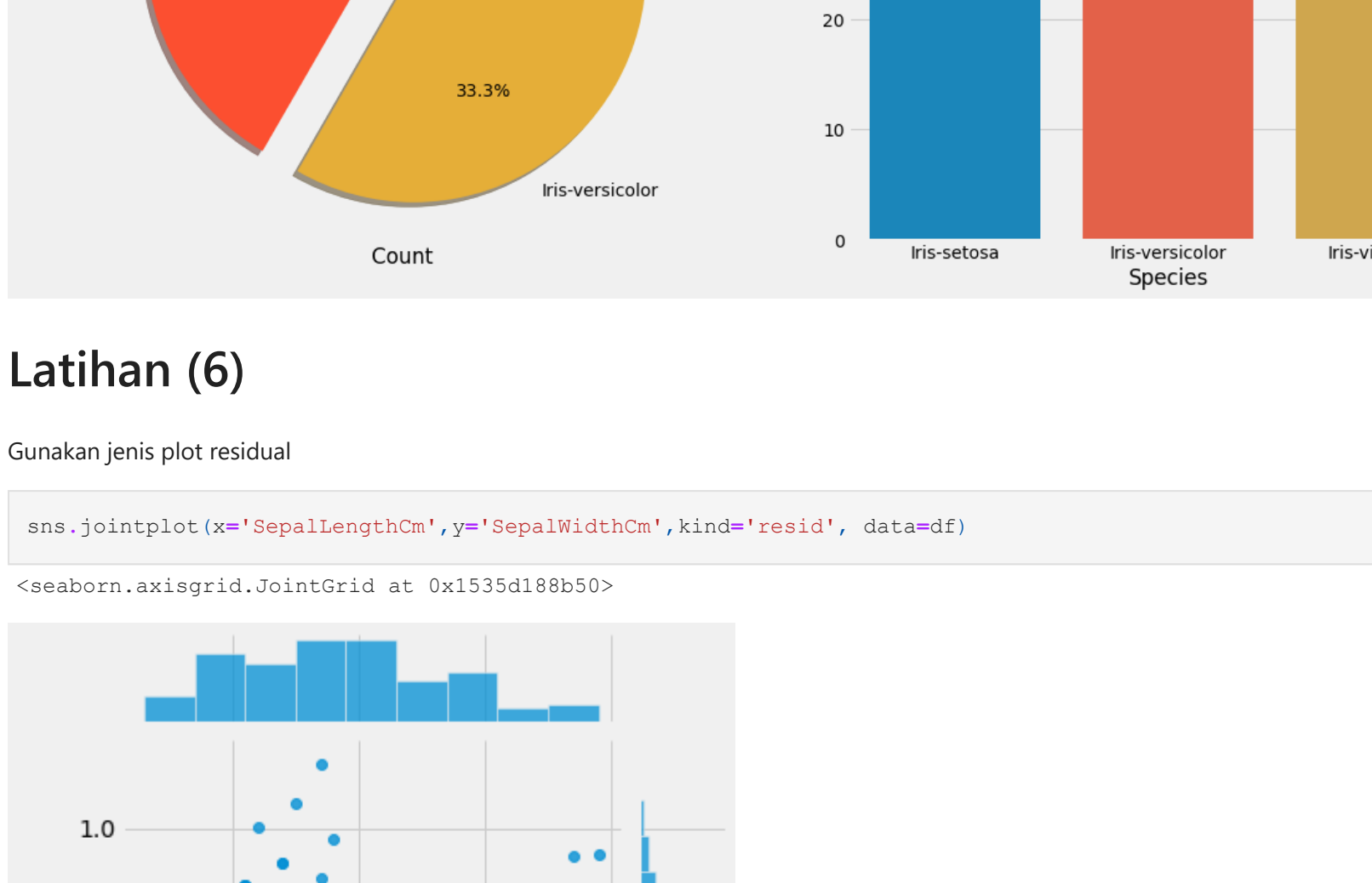
Visualisasi Boxplot untuk memberikan ringkasan statis dari fitur yang diplot.

- Garis atas mewakili nilai maksimal
- Tepi atas kotak adalah Kuartil ketiga
- Tepi tengah adalah median,
- Tepi bawah adalah nilai kuartil pertama.
- Garis paling bawah adalah nilai minimum.
- Ketinggian kotak disebut sebagai rentang Interkuartil.
- Titik titik hitam pada plot adalah nilai outlier dalam data.

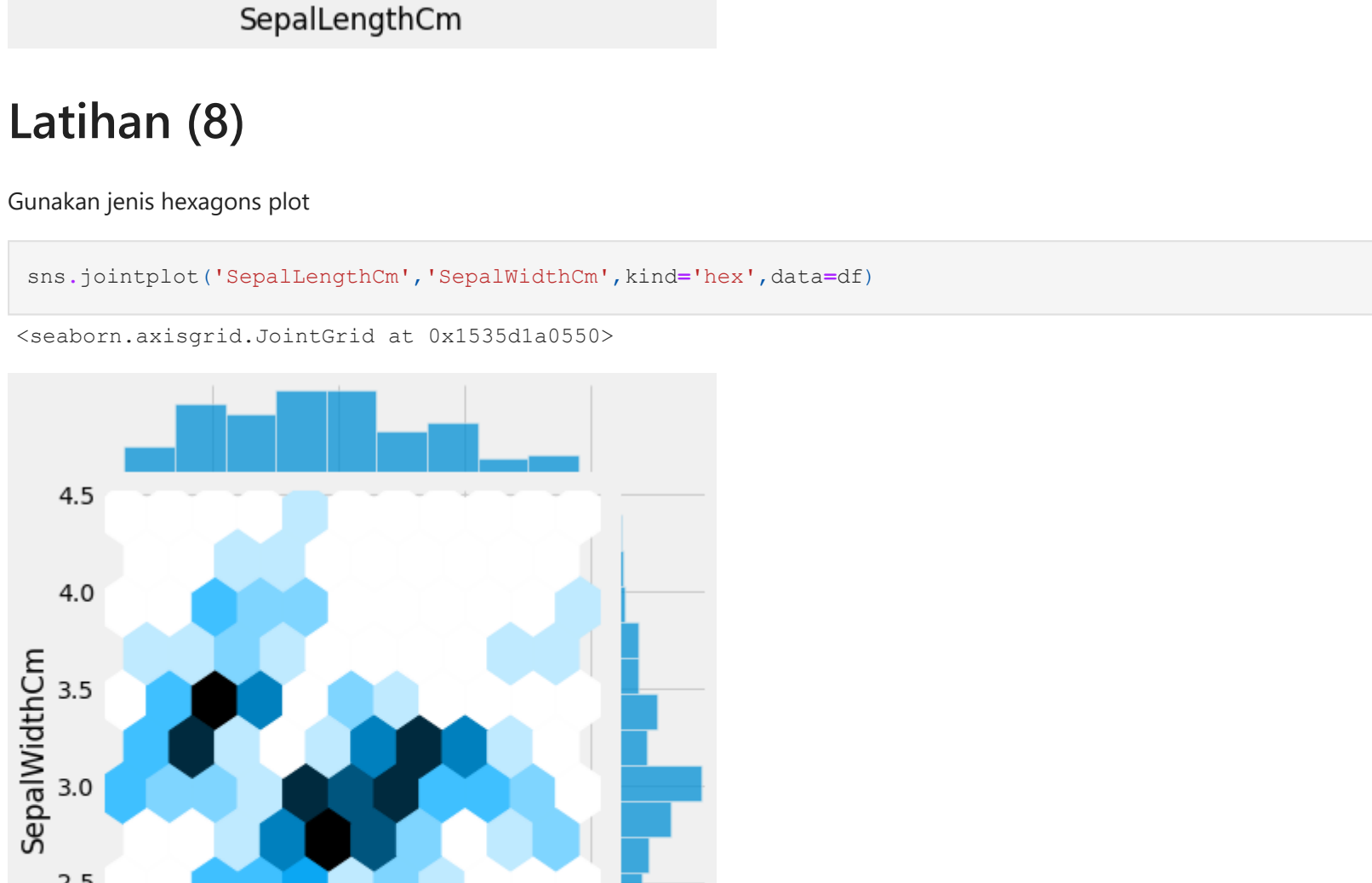
Tampilkan visualisasi boxplot menggunakan kolom "Species" dan "PetalLengthCm" dalam dataset iris

```
In [16]: # visualisasi Boxplot

In [17]: fig=plt.gcf()
fig.set_size_inches(10,7)
fig.suptitle('Species',y='PetalLengthCm',data=df,
figsns.boxplot(x='Species',y='PetalLengthCm',data=df,
order=['Iris-virginica','Iris-versicolor','Iris-setosa'],
linewidth=2.5,orient='v',dodge=False)
```



```
In [18]: # Visualisasi Boxplot yang di kelompokkan berdasarkan "Species"
figsns.boxplot('Species').boxplot(by='Species',figsize=(12, 6))
pass
```



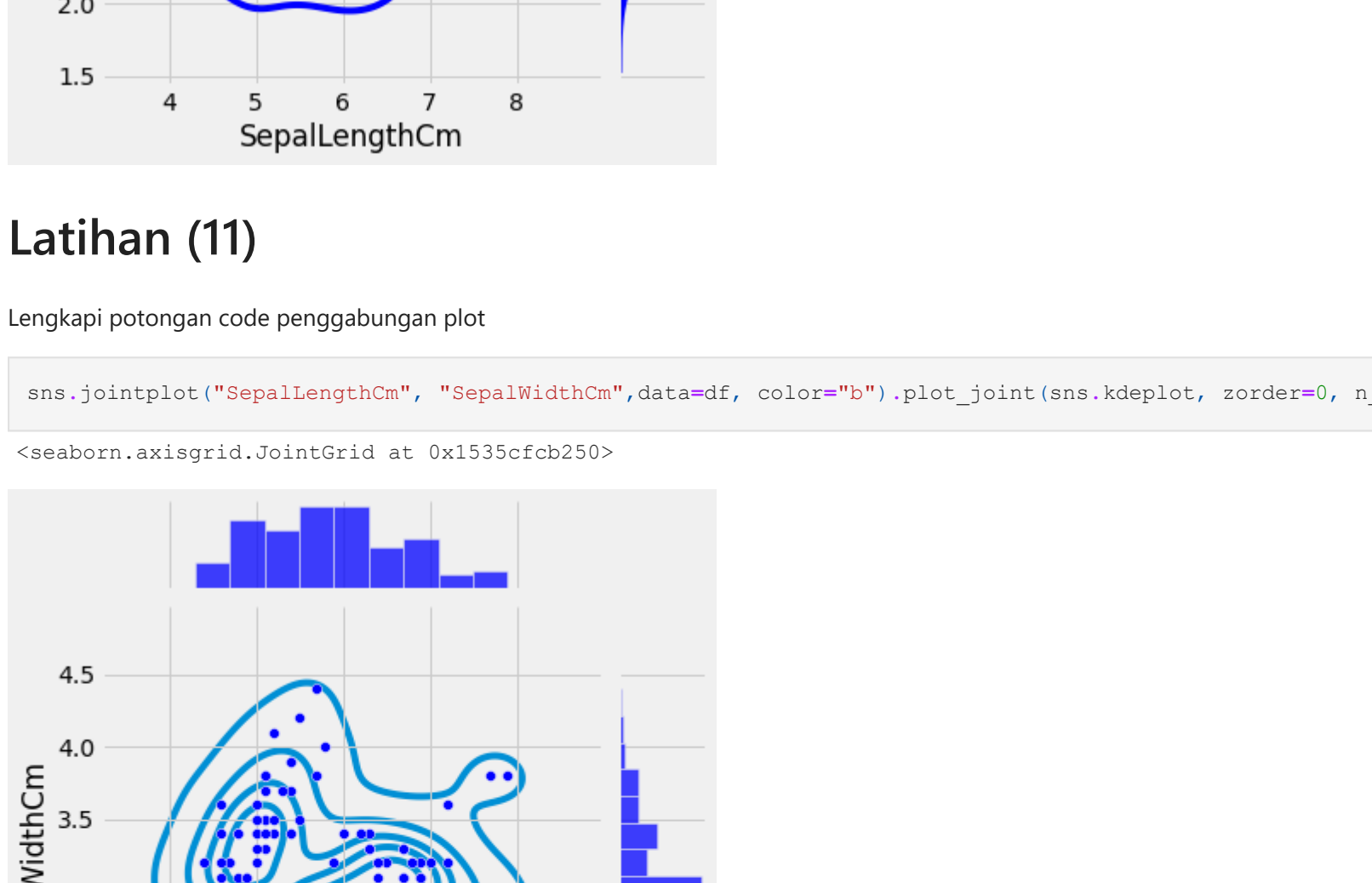
Latihan (13)

Visualisasi Violin Plot untuk memvisualisasikan sebaran data dan distribusi probabilitas.

- Bilah hitam tebal di tengah mewakili rentang interkuartil
- Garis hitam tipis yang memanjang darinya mewakili interval kepercayaan 95%
- Titik putih adalah median.

Tampilkan visualisasi Violin Plot dengan menggunakan setiap kolom yang ada untuk melihat sebaran data terhadap kolom "Species" dalam dataset iris

```
In [19]: plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.violinplot(x='Species',y='PetalLengthCm',data=df)
plt.subplot(2,2,2)
sns.violinplot(x='Species',y='PetalWidthCm',data=df)
plt.subplot(2,2,3)
sns.violinplot(x='Species',y='SepalLengthCm',data=df)
plt.subplot(2,2,4)
sns.violinplot(x='Species',y='SepalWidthCm',data=df)
<AxesSubplot: xlabel='Species', ylabel='SepalWidthCm'>
```



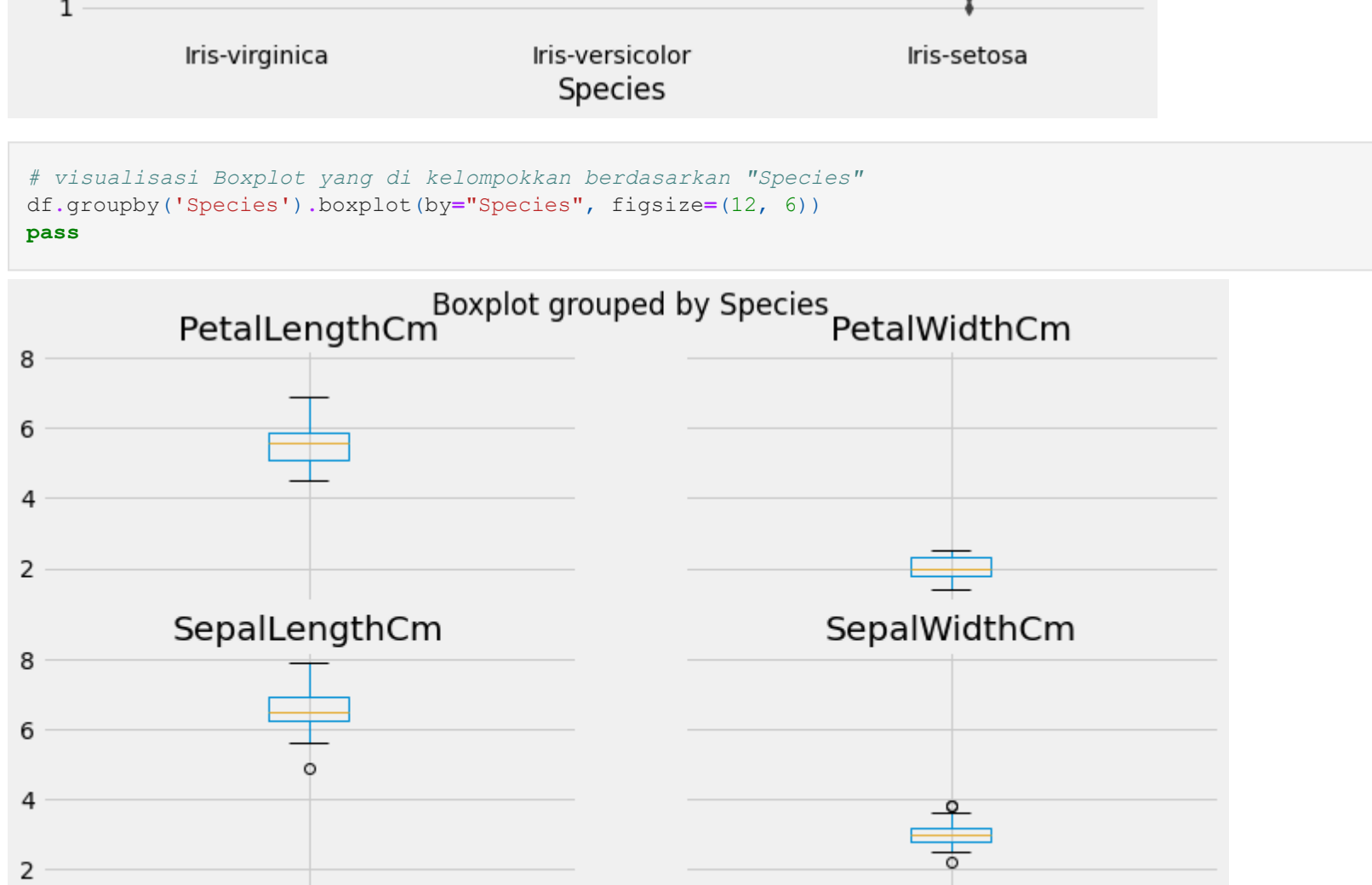
Latihan (14)

Visualisasi pairplot yang juga dikenal dengan scatterplot adalah visualisasi sebaran data yang menunjukkan keterkaitan antar kolom.

Tampilkan visualisasi pairplot dari setiap kolom yang ada untuk melihat sebaran data dalam dataset iris

```
In [20]: # visualisasi pairplot
sns.pairplot(data=df,kind='scatter')

Out[20]: <seaborn.axisgrid.PairGrid at 0x1535faa98f50>
```



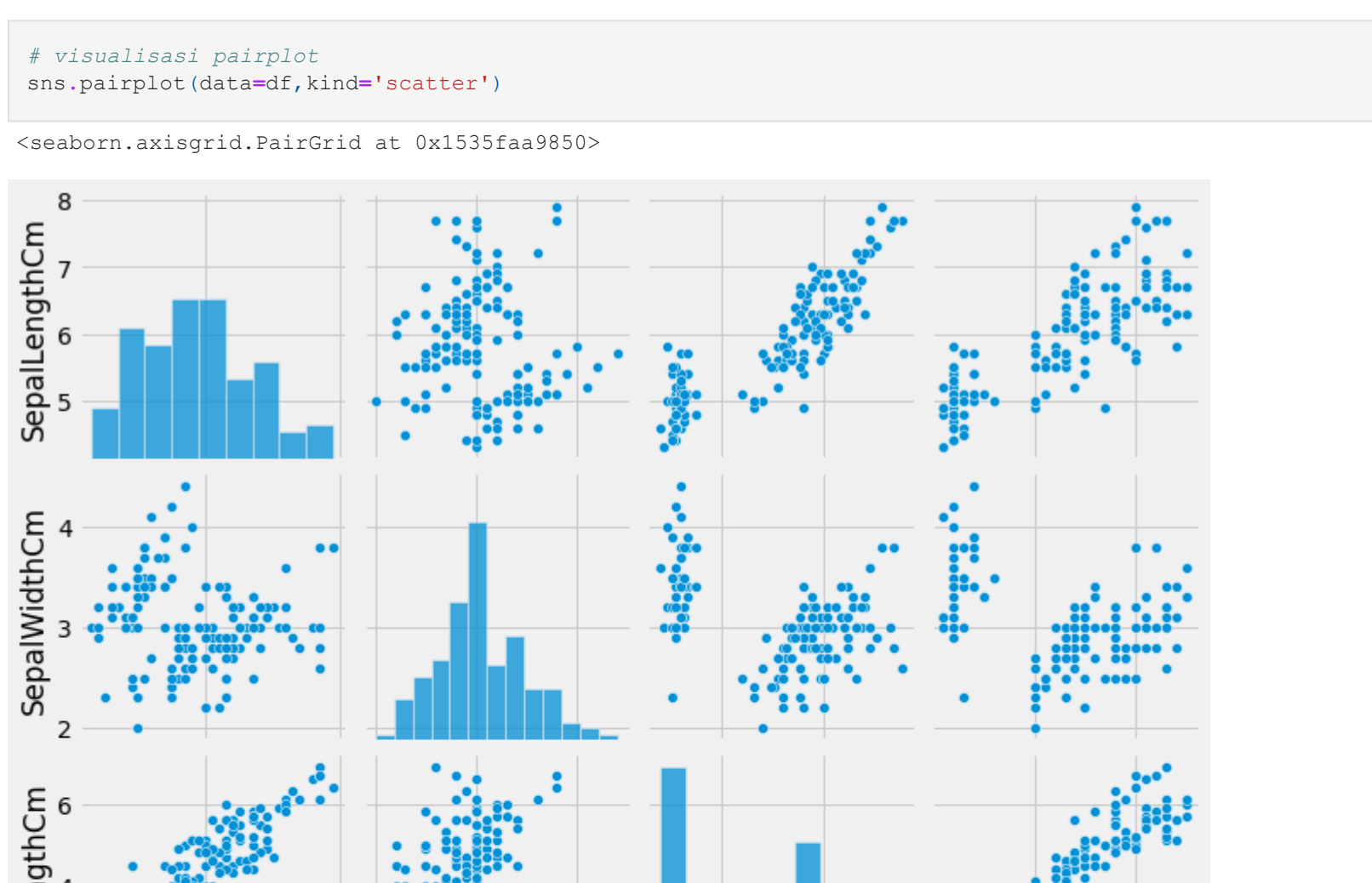
Latihan (15)

Visualisasi heatmap digunakan untuk mengetahui korelasi setiap kolom dalam dataset. Nilai positif atau negatif yang tinggi menunjukkan bahwa fitur tersebut memiliki korelasi yang tinggi. Hal ini membantu kita memilih parameter untuk machine learning.

Tampilkan visualisasi heatmap dari korelasi setiap fitur dalam dataset iris

```
In [21]: # visualisasi heatmap

In [22]: fig=plt.gcf()
fig.set_size_inches(10,7)
figsns.heatmap(df.corr(),annot=True,cmap='coolwarm',linewidth=1,linestyle='k',square=True,
mask=False,vmin=-1,vmax=1,bar_kws={"orientation": "vertical"},bar=True)
```



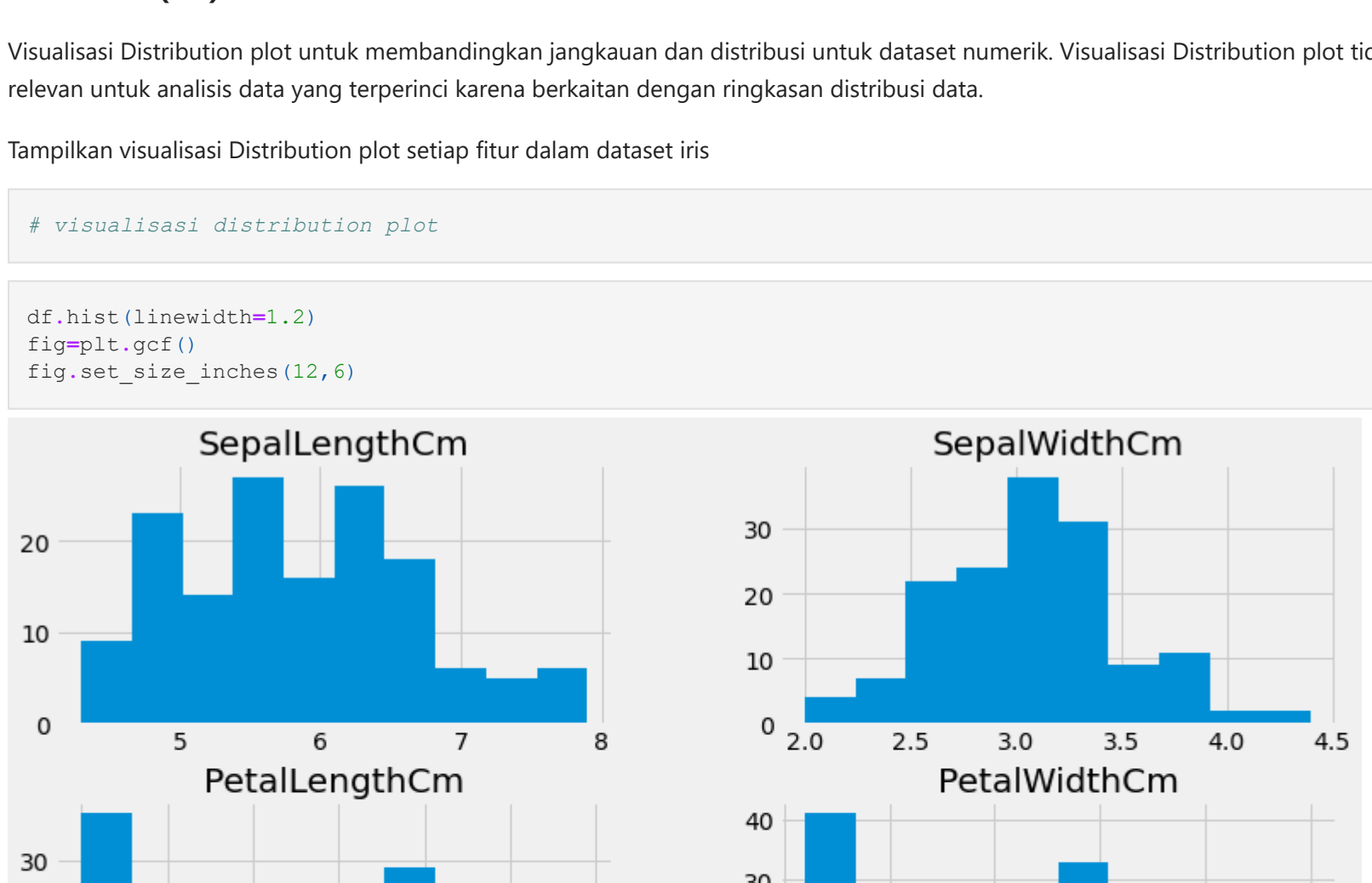
Latihan (16)

Visualisasi Distribution plot untuk membandingkan jangkauan dan distribusi untuk dataset numerik. Visualisasi Distribution plot tidak relevan untuk analisis data yang terperi ini karena berkaitan dengan ringkasan distribusi data.

Tampilkan visualisasi Distribution plot setiap fitur dalam dataset iris

```
In [23]: # visualisasi distribution plot

In [24]: df.hist(linewidth=1,2)
fig=plt.gcf()
fig.set_size_inches(12,6)
```



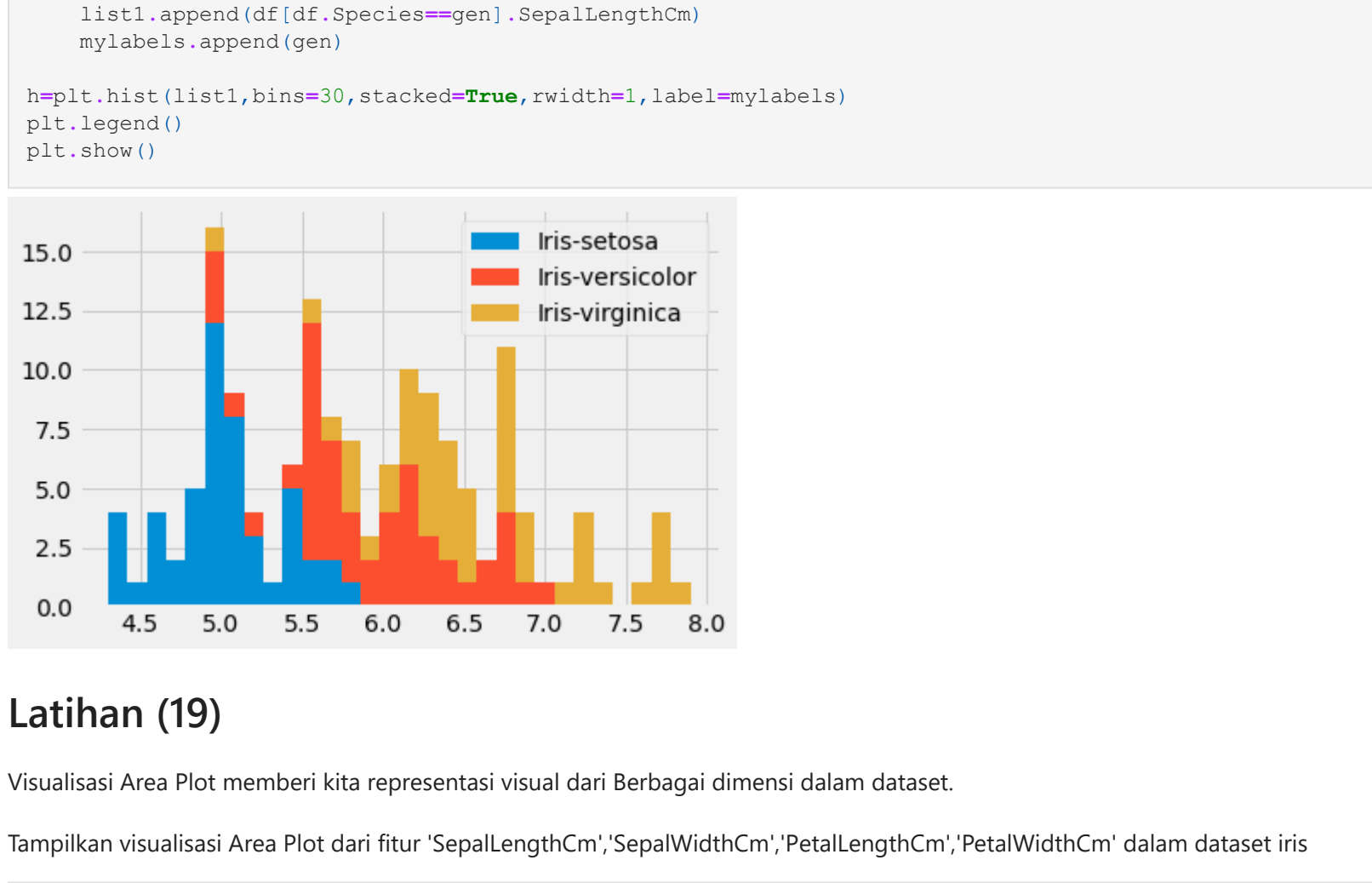
Latihan (17)

Visualisasi Residual Plot adalah visualisasi untuk memplot residu dengan nilai prediksi pada sumbu x, dan residu pada sumbu y. Jarak dari garis 0 adalah seberapa buruk prediksi untuk nilai yang di plot.

Tampilkan visualisasi Residual Plot dari fitur "SepalLengthCm" dan SepalWidthCm dalam dataset iris

```
In [25]: # Visualisasi Residual Plot

In [26]: fig=plt.gcf()
fig.set_size_inches(10,7)
figsns.residplot('SepalLengthCm', 'SepalWidthCm',data=df,lowess=True)
```



Latihan (18)

Visualisasi Stacked Histogram digunakan untuk menunjukkan bagaimana fitur yang lebih besar dibagi menjadi fitur yang lebih kecil dan menunjukkan hubungan masing-masing fitur terhadap jumlah total

Tampilkan visualisasi Stacked Histogram dari fitur "Species" dengan mengubah tipe datanya menjadi category (astype) dalam dataset iris

```
In [27]: # visualisasi stacked histogram

In [28]: df['Species'] = df['Species'].astype('category')
df.head()
list=list()
mylabels=list()
for gen in df.Species.cat.categories:
list.append(df[df.Species==gen].SepalLengthCm)
mylabels.append(gen)

plt.hist(list,bins=30,stacked=True,width=1,label=mylabels)
plt.legend()
plt.show()
```


Latihan (19)

Visualisasi Area Plot memberi kita representasi visual dari Berbagai dimensi dalam dataset.

Tampilkan visualisasi Area Plot dari fitur "SepalLengthCm",SepalWidthCm",PetalLengthCm",PetalWidthCm" dalam dataset iris

```
In [29]: # Visualisasi Area Plot

In [30]: df.plot.area(y=['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm'],alpha=0.4,figsize=(12, 6));
```


Latihan (20)

Visualisasi distplot membantu untuk melihat distribusi variabel tunggal. Kde menunjukkan kepadatan distribusi

Tampilkan visualisasi distplot dari fitur "SepalLengthCm" dengan menggunakan Kde (kind) untuk menunjukkan kepadatan distribusi dalam dataset iris

```
In [31]: # visualisasi distplot

In [32]: sns.distplot(df['SepalLengthCm'],kde=True,bins=20);
```

