

1. Diagram Batang (Bar Plot)

```
import numpy as np
import matplotlib.pyplot as plt

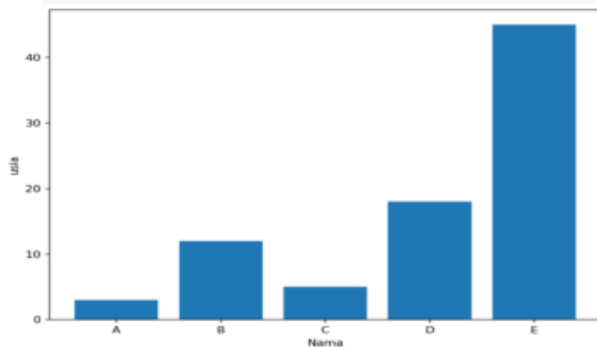
Usia = [3,12,5,18,45]          # numerik
Nama = ('A', 'B','C','D','E')  # kategori
pos_y= np.arange(len(Usia))    # reposisi data usia ke array

plt.figure(figsize=(8,6))      # atur ukuran gambar
plt.bar(pos_y, Usia)           # data numerik

plt.xticks(pos_y,Nama)         # data kategori

plt.xlabel('Nama')             # label sumbu x
plt.ylabel('usia')             # label sumbu y

plt.show()
```



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

Nama = ['A','B','C','D','E']
Usia = [39,63,45,24,13]
df = pd.DataFrame({"Nama" : Nama,
                   "Usia" : Usia})

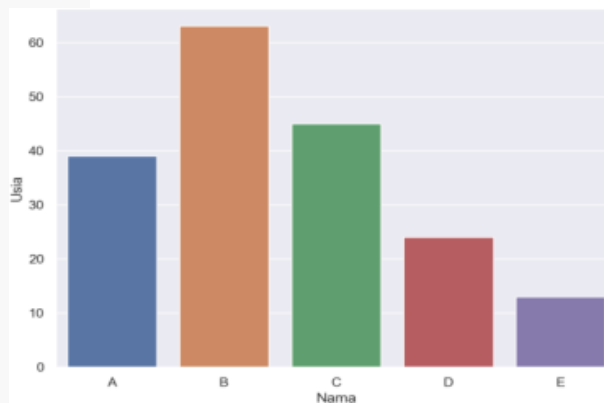
# atur tema plot dan latar belakang
sns.set(style="darkgrid")

#atur ukuran gambar
plt.figure(figsize=(8,6))

#buat diagram batang

sns.barplot(
    x = "Nama",
    y = "Usia",
    data = df,
    estimator = sum,
    ci = None)

# menampilkan grafiknya
plt.show()
```



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

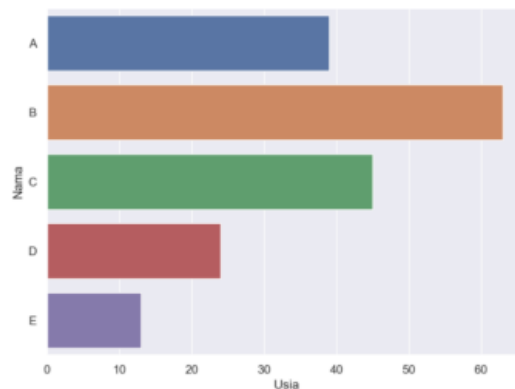
Nama = ["A", "B", "C", "D", "E"]
Usia = [39, 63, 45, 24, 13]
df = pd.DataFrame({"Nama": Nama,
                  "Usia": Usia})

# atur tema plot dan latar belakang
sns.set(style="darkgrid")

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# buat diagram batang
sns.barplot(
    x = "Usia",
    y = "Nama",
    data = df,
    estimator = sum,
    ci = None)

# menampilkan grafiknya
plt.show()
```



Jika ingin mengurutkan barplot berdasarkan peringkat grup. Misalnya, ingin memiliki grup dengan nilai tertinggi di atas, dan grup dengan nilai terendah di bawah, harus menyusun ulang data frame menggunakan fungsi `sort_values()` sebagai berikut:

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

# membentuk data frame
Nama = ["A", "B", "C", "D", "E"]
Usia = [39, 63, 45, 24, 13]
df = pd.DataFrame({"Nama": Nama,
                  "Usia": Usia})

# Susun ulang data frame
df = df.sort_values(["Usia"], ascending=False).reset_index(drop=True)

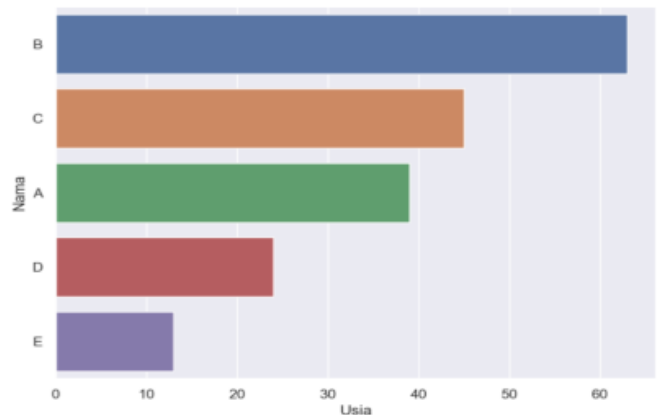
# atur tema plot dan latar belakang
sns.set(style="darkgrid")

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# buat diagram batang

sns.barplot(
    x = "Usia",
    y = "Nama",
    data = df,
    estimator = sum,
    ci = None)

# menampilkan grafiknya
plt.show()
```



Grafik Lingkaran (Piechart)

Pie chart merupakan diagram yang berbentuk lingkaran yang dibagi menjadi beberapa sektor. Pada Python, sebagian besar dilakukan dengan fungsi `pie()` dari library Matplotlib

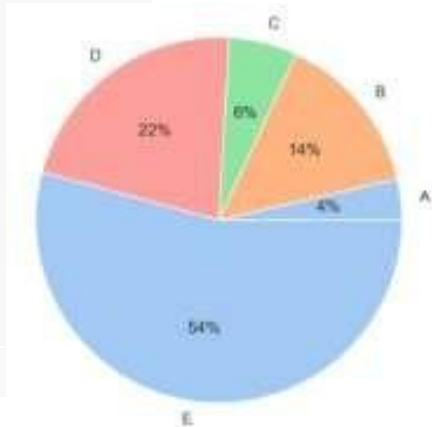
```
import numpy as np
import matplotlib.pyplot as plt

Usia = [3, 12, 5, 18, 45]           # data numerik
Nama = ('A', 'B', 'C', 'D', 'E')    # data kategori
warna = sns.color_palette('pastel')[0:4] # warna Seaborn yang akan digunakan

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# membuat diagram lingkaran
plt.pie(Usia,
        labels = Nama,
        colors = warna,
        autopct='%0.0f%%');

# menampilkan grafiknya
plt.show()
```



Grafik Peta (TreeMap)

TreeMap menampilkan data membagi bagian dengan persegi/persegi panjang. Dibagi sesuai luas. Pada python bisa menggunakan library squarify untuk menghitung posisi persegi panjang dan memplotnya.

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

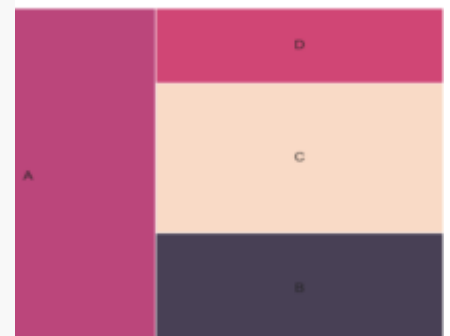
# buat data dengan data acak/
df = pd.DataFrame({'nb_people': [8, 3, 4, 2],
                  'group': ["A", "B", "C", "D"] })

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# membuat diagram
squarify.plot(sizes = df['nb_people'],
              label = df['group'],
              #color = ["red", "green", "blue", "grey"],
              alpha=.8 )

plt.axis('off')

# menampilkan grafiknya
plt.show()
```



Grafik Histogram

Histogram bisa dibuat dengan python menggunakan library seaborn dan matplotlib. Menggambar histogram sederhana dengan parameter default.

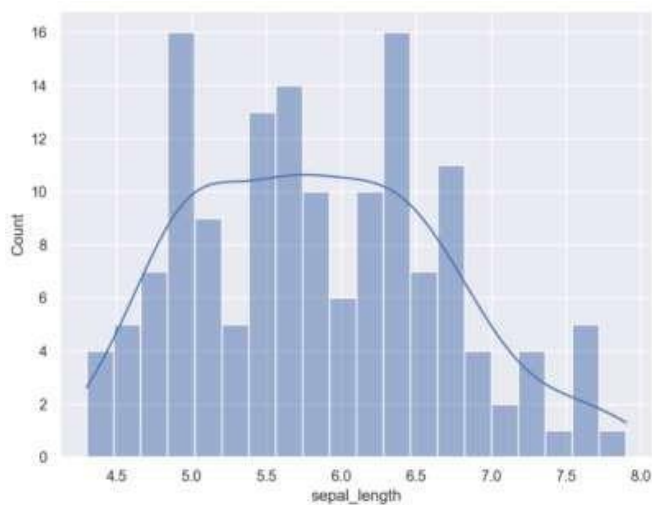
```
import seaborn as sns
import matplotlib.pyplot as plt

# ukuran gambar
plt.figure(figsize=(8, 6))

# latar belakang abu-abu
sns.set_theme(style="darkgrid")
df = sns.load_dataset("iris")

sns.histplot(data=df, x="sepal_length", bins=20, kde=True)

plt.show()
```



Jika ingin menampilkan dua variabel dalam satu grafik yang sama:

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

# atur ukuran gambar
plt.figure(figsize=(8, 6))

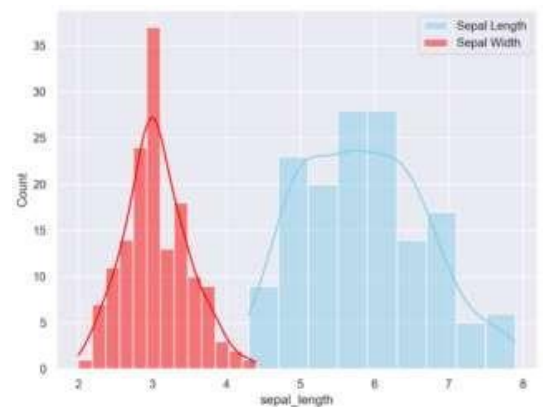
# atur latar belakang abu-abu
sns.set_theme(style="darkgrid")
df = sns.load_dataset("iris")

sns.histplot(data=df, x="sepal_length", color="skyblue", label="Sepal Length", kde=True)

sns.histplot(data=df, x="sepal_width", color="red", label="Sepal Width", kde=True)

# menampilkan grafiknya
plt.legend()

plt.show()
```



Jika ingin membagi histogram menjadi beberapa bagian, bisa menggunakan cara berikut:

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

# atur latar belakang abu-abu
sns.set_theme(style="darkgrid")
df = sns.load_dataset("iris")

#membagi grafik 2 kali 2
fig,axs = plt.subplots(2,2, figsize=(7,7))

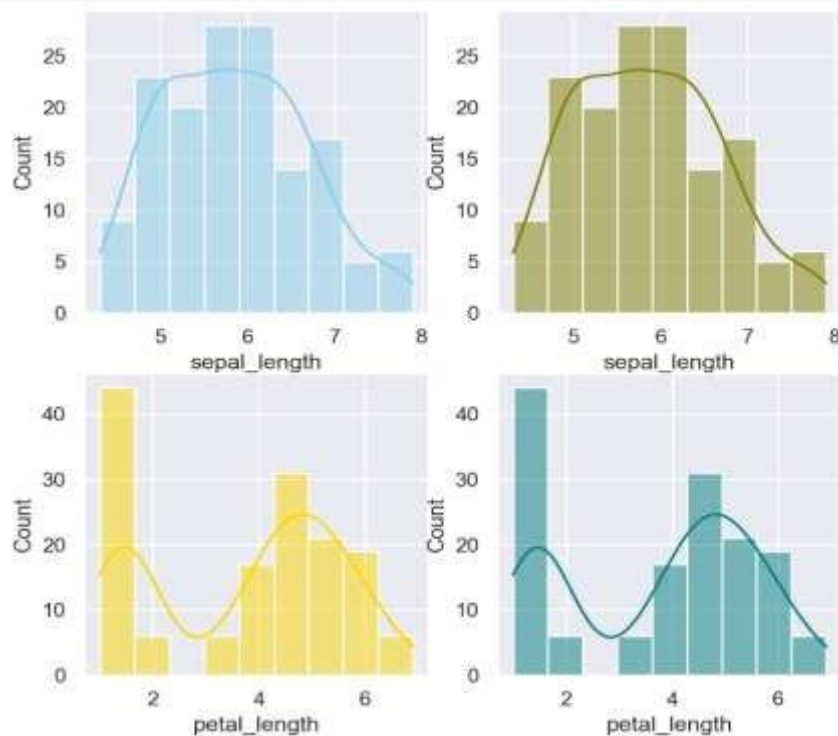
sns.histplot(data=df, x="sepal_length", color="skyblue", kde=True, ax= axs[0,0])

sns.histplot(data=df, x="sepal_length", color="olive", kde=True, ax= axs[0,1])

sns.histplot(data=df, x="petal_length", color="gold", kde=True, ax= axs[1,0])

sns.histplot(data=df, x="petal_length", color="teal", kde=True, ax= axs[1,1])

plt.show()
```



Grafik Kepadatan Kernel

Alternatif histogram adalah grafik kepadatan kernel. Grafik kepadatan kernel mencoba menggambar histogram yang dihaluskan, area di bawah kurva = 1

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

# atur latar belakang abu-abu
sns.set_theme(style="darkgrid")
df = sns.load_dataset("iris")

# membagi grafik 2 kali 2
fig, axs = plt.subplots(2, 2, figsize=(7, 7))

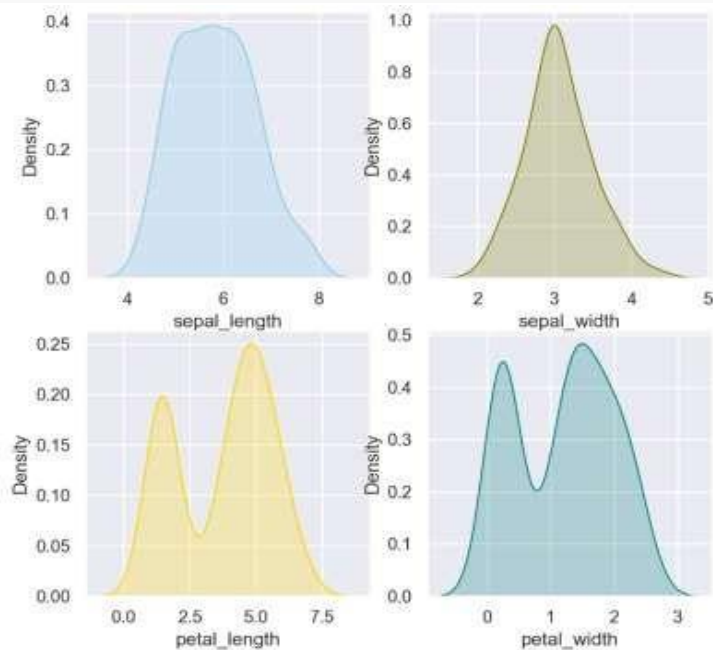
sns.kdeplot(data=df, x="sepal_length", color="skyblue", shade=True, ax=axs[0, 0])

sns.kdeplot(data=df, x="sepal_width", color="olive", shade=True, ax=axs[0, 1])

sns.kdeplot(data=df, x="petal_length", color="gold", shade=True, ax=axs[1, 0])

sns.kdeplot(data=df, x="petal_width", color="teal", shade=True, ax=axs[1, 1])

plt.show()
```



Opsi lain yang bisa digunakan adalah dengan menggabungkan semua plot tersebut. Sebenarnya bisa, tetapi akan berantakan (tumpang tindih). Solusi yang lebih mudah menggunakan transparansi.

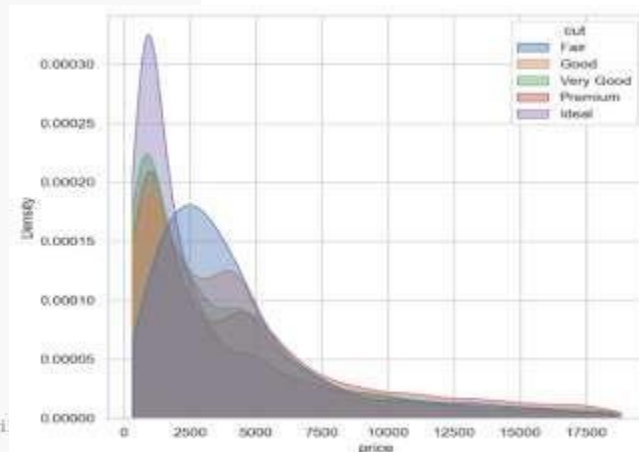
```
import seaborn as sns
import matplotlib.pyplot as plt
from plotnine.data import diamonds

# Tetapkan ukuran gambar untuk notebook
plt.rcParams["figure.figsize"] = 8, 6

# atur tema whitegrid seaborn
sns.set(style="whitegrid")

# Tanpa transparansi
sns.kdeplot(data = diamonds,
            x = "price",
            hue = "cut",
            cut = 0,
            fill = True,
            common_norm = False,
            alpha = 0.4) # penyesuaian transparansi dengan nilai

plt.show()
```



Solusi lain yang dapat dilakukan adalah menumpuk grup dengan melewati "isi" ke beberapa argumen fungsi. Hal ini memungkinkan untuk melihat grup mana yang paling sering untuk nilai tertentu, tetapi sulit untuk memahami distribusi grup yang tidak berada di bagian bawah bagan.

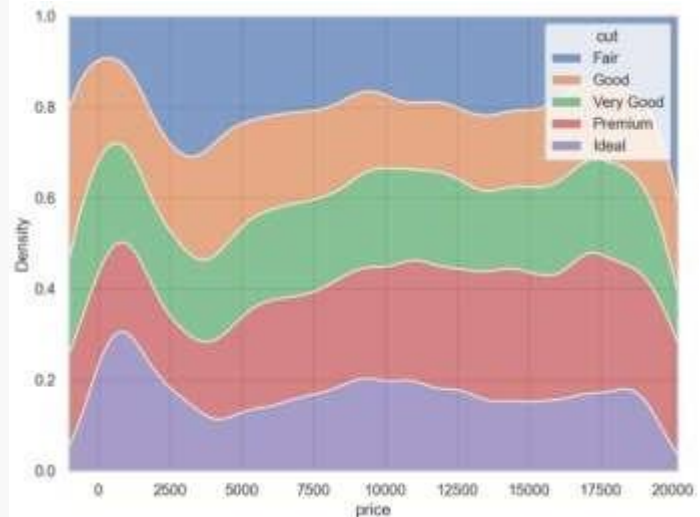
```
import seaborn as sns
import matplotlib.pyplot as plt
from plotnine.data import diamonds

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")

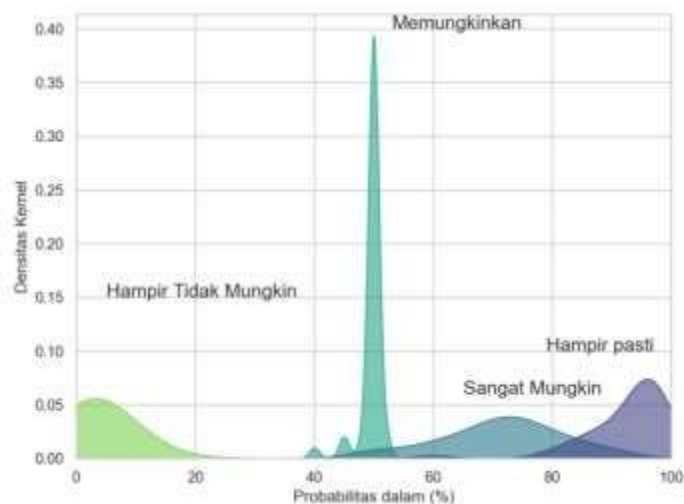
# plot kepadatan bertumpuk
sns.kdeplot(data = diamonds,
            x = "price",
            hue = "cut",
            common_norm = False,
            multiple = "fill",
            alpha = 0.7)

plt.show()
```



Contoh di bawah ini merupakan penggunaan grafik kepadatan kernel yang sesuai dengan menggunakan contoh dataset ini.

Karena grafik yang akan dibuat dari coding di bawah ini ditumpuk, maka akan lebih mudah untuk menambahkan nama/keterangan di sebelah grafik.



(grafik code dibawahnya)


```

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

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")

# memuat dataset dari GitHub dan mengubah formatnya
data = pd.read_csv("https://raw.githubusercontent.com/dscielabs/dataset/master/probly")
data = pd.melt(data, var_name='text', value_name='value')

# memuat beberapa variabel dari dataset
data = data.loc[data.text.isin(["Almost No Chance",
                                "About Even",
                                "Probable",
                                "Almost Certainly"])]

# plot kepadatan kernel
p = sns.kdeplot(data = data,
                x = "value",
                hue = "text",
                fill = True,
                common_norm = False,
                alpha = 0.6,
                palette = "viridis",
                legend = False)

# kontrol batas sumbu x
plt.xlim(0, 100)

annot = pd.DataFrame({
    'x': [5, 53, 65, 79],
    'y': [0.15, 0.4, 0.06, 0.1],
    'text': ["Hampir Tidak Mungkin",
             "Memungkinkan",
             "Sangat Mungkin",
             "Hampir pasti"]
})

# tambahkan anotasi satu per satu dengan loop
for point in range(0, len(annot)):
    p.text(annot.x[point],
           annot.y[point],
           annot.text[point],
           horizontalalignment='left',
           size='large')

# tambahkan nama sumbu
plt.xlabel("Probabilitas dalam (%)")

plt.ylabel("Densitas Kernel")

plt.show()

```


Grafik Lolipop

Grafik lolipop seperti grafik titik yang diberi tangkai di bawahnya. Grafik lolipop menjadi alternatif dari barplot. Untuk membuat lolipop menggunakan library matplotlib.

Contoh coding di bawah ini, strategi disini adalah menggunakan fungsi stem() atau meretas fungsi vline() tergantung pada format input.

```
# memuat dataframe
import pandas as pd
df = pd.DataFrame({'group':list(map(chr, range(65, 85))),
                  'values':np.random.uniform(size=20) })

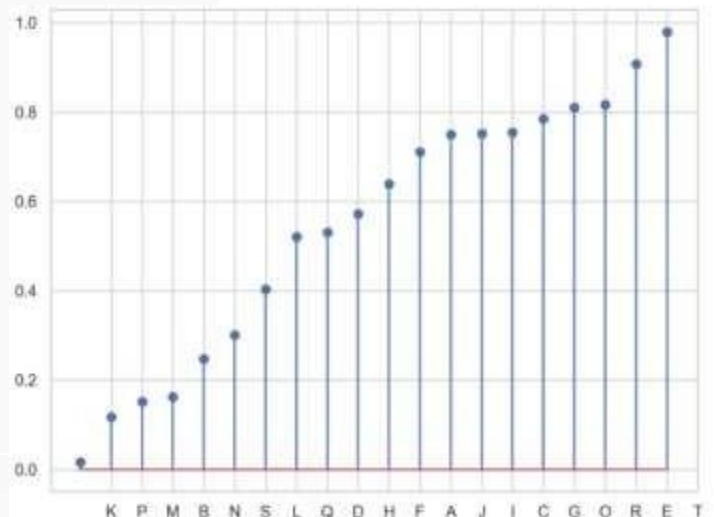
# mengurutkan berdasarkan nilai
ordered_df = df.sort_values(by='values')
my_range=range(1,len(df.index)+1)

# ukuran gambar
plt.figure(figsize=(8, 6))

# tema whitegrid seaborn
sns.set(style="whitegrid")

# buat plotnya
plt.stem(ordered_df['values'])
plt.xticks(my_range, ordered_df['group'])

plt.show()
```



Grafik di atas menjelaskan cara membuat plot lolipop vertikal pada contoh di bawah adalah grafik horizontal menggunakan fungsi hlines() dari matplotlib. “grup B” ditampilkan dengan warna dan ukuran penanda yang berbeda. Untuk melakukannya, variable my_color didefinisikan “orange” untuk grup B dan “biru langit” untuk grup yang tersisa. Variabel warna ini diteruskan ke fungsi hlines() sebagai argumen. Untuk mengubah ukuran marker, fungsi scatter() dari matplotlib digunakan.

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

df = pd.DataFrame({'group':list(map(chr, range(65, 85))),
                  'values':np.random.uniform(size=20) })

# mengurutkan berdasarkan nilai
ordered_df = df.sort_values(by='values')
my_range=range(1,len(df.index)+1)

# Buat warna jika grupnya adalah "B"
my_color=np.where(ordered_df ['group']=='B', 'orange', 'skyblue')
my_size=np.where(ordered_df ['group']=='B', 70, 30)

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")
```

```
# Plot horizontal dibuat menggunakan fungsi hline()
plt.hlines(y = my_range,
          xmin = 0,
          xmax = ordered_df['values'],
          color = my_color,
          alpha = 0.48)

plt.scatter(ordered_df['values'],
           my_range,
           color=my_color,
           s=my_size,
           alpha=1)

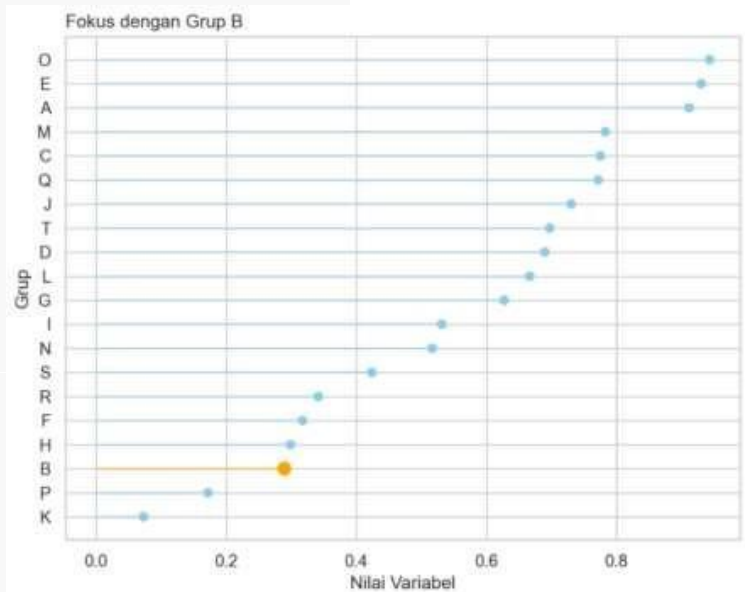
# Tambahkan judul dan nama sumbu
plt.yticks(my_range, ordered_df['group'])

plt.title("Fokus dengan Grup B", loc='left')

plt.xlabel('Nilai Variabel')

plt.ylabel('Grup')

plt.show()
```



Dua Variabel Kategori

Boxplot

```
import seaborn as sns
from warnings import filterwarnings
df = sns.load_dataset('tips')

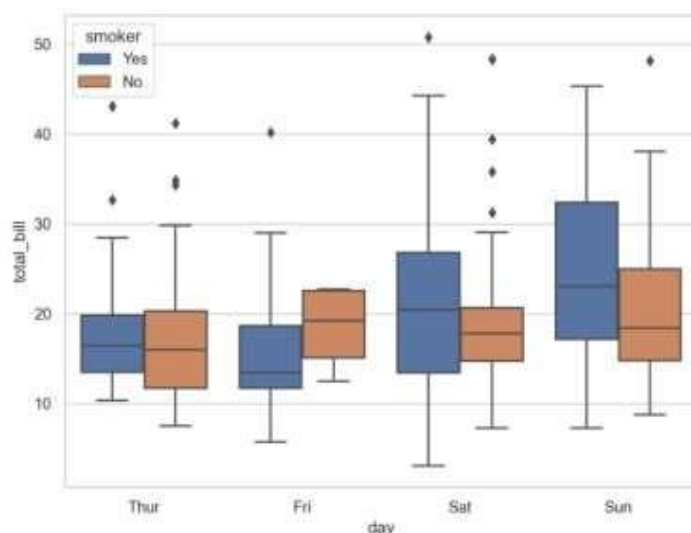
# impor perpustakaan seaborn
# untuk menghindari peringatan
# memuat dataset dari seaborn

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")

# membuat grafik boxplotnya
sns.boxplot(x = 'day',
            y = 'total_bill',
            data = df,
            hue = 'smoker')

# menampilkan grafiknya
plt.show()
```



Grafik Biola (Violinplot)

Library Seaborn secara khusus disesuaikan untuk membuat grafik biola(). Grafikbiola ini visualisasinya lebih “bagus” dari boxplot karena pada violin plot

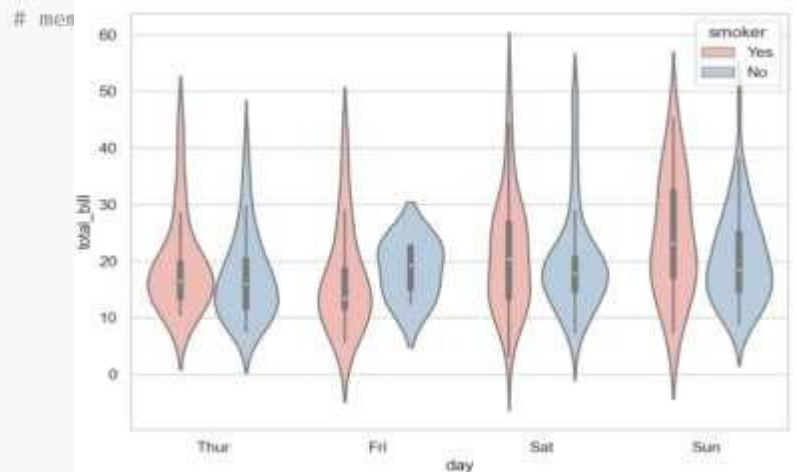
```
import seaborn as sns
from warnings import filterwarnings
import matplotlib.pyplot as plt
df = sns.load_dataset('tips')
```

```
# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")
```

```
# Grouped violinplot
sns.violinplot(x = "day",
               y = "total_bill",
               hue = "smoker",
               data = df,
               palette = "Pastell")
```

```
plt.show()
```



Numerik vs Numerik

Biasa ditampilkan menggunakan scatterplots dan grafik garis.

Plot Pencar (Scatterplot).

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

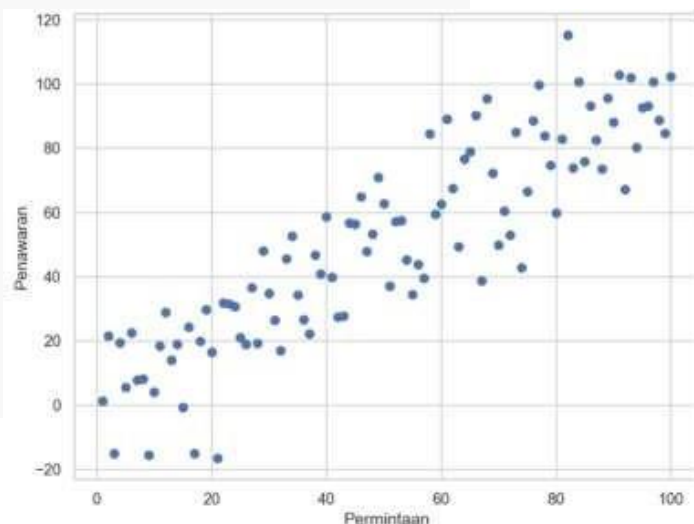
```
# atur ukuran gambar
plt.figure(figsize=(8, 6))
```

```
# atur tema whitegrid seaborn
sns.set(style="whitegrid")
```

```
df=pd.DataFrame({'x_values': range(1,101),
                 'y_values': np.random.randn(100)*15+range(1,101) })
```

```
plt.plot('x_values',
         'y_values',
         data = df,
         linestyle = 'none',
         marker = 'o')
```

```
# menambahkan label
plt.xlabel('Permintaan')
plt.ylabel('Penawaran')
plt.show()
```



Grafik Garis

Grafik garis menunjukkan grafik dalam bentuk garis. Grafik garis menampilkan evolusi satu atau beberapa variabel numerik.

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

# atur ukuran gambar
plt.figure(figsize=(8, 6))

# atur tema whitegrid seaborn
sns.set(style="whitegrid")

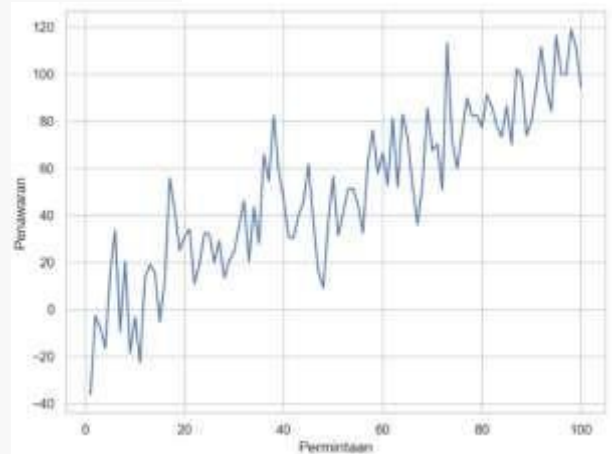
df=pd.DataFrame({'x_values': range(1,101),
                 'y_values': np.random.randn(100)*15+range(1,101) })

plt.plot('x_values',
         'y_values',
         data = df
        )

# menambahkan label
plt.xlabel('Permintaan')

plt.ylabel('Penawaran')

plt.show()
```



Grafik Scatter dan Grup

```
import seaborn as sns
import matplotlib.pyplot as plt
df = sns.load_dataset('iris')

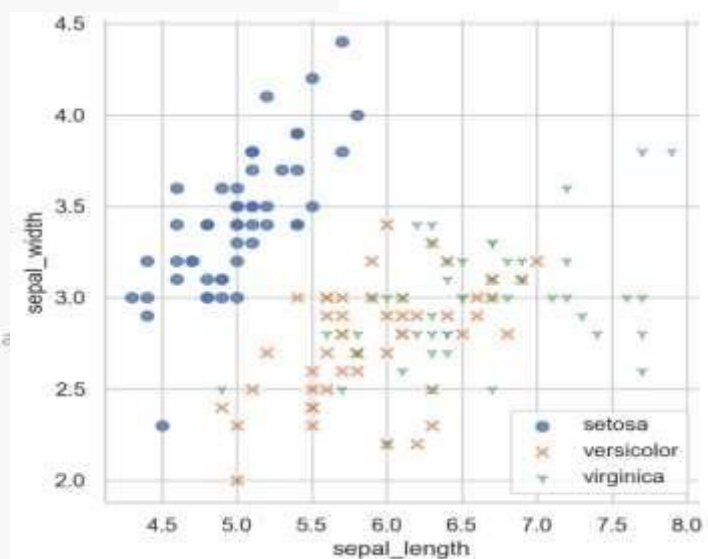
# ukuran gambar
plt.figure(figsize=(8, 6))

# tema
sns.set(style="whitegrid")

# Gunakan argumen 'hue' untuk menambahkan variabel
sns.lmplot(x = "sepal_length",
          y = "sepal_width",
          data = df,
          fit_reg = False,
          hue = 'species',
          legend = False,
          markers=["o", "x", "1"]);

plt.legend(loc='lower right')

plt.show()
```

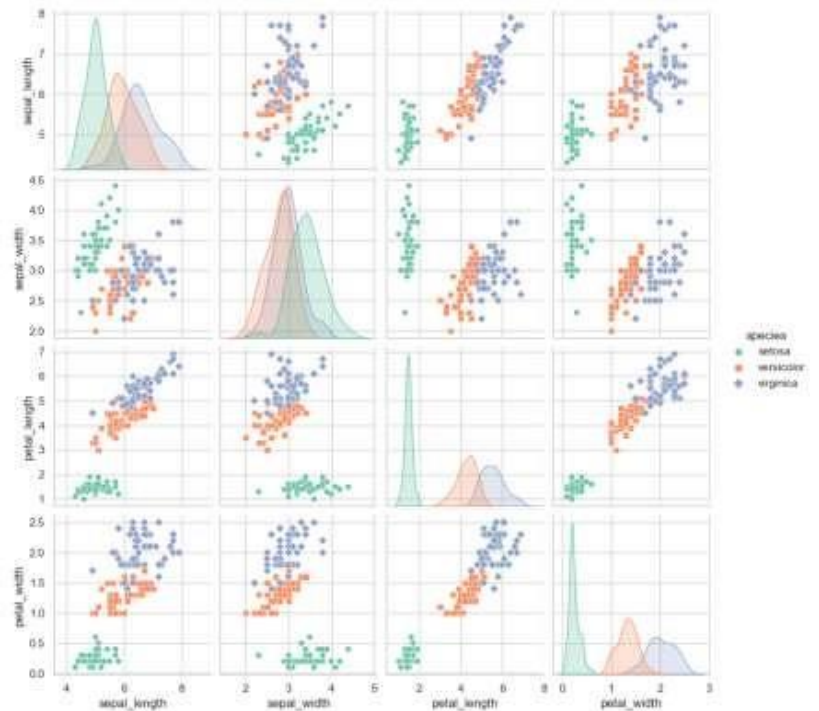


Grafik Korelogram

```
import matplotlib.pyplot as plt
import seaborn as sns
df = sns.load_dataset('iris')

# memuat grafik korelogram
sns.pairplot(df,
             kind = "scatter",
             hue = "species",
             markers = ["o", "s", "D"],
             palette = "Set2")

plt.show()
```



Grafik Berlapis (Faceting)

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

my_count=["France","Australia","Japan","USA","Germany",
          "Congo","China","England","Spain","Greece",
          "Marocco","South Africa","Indonesia","Peru",
          "Chili","Brazil"]

df = pd.DataFrame({
    "country":np.repeat(my_count, 10),
    "years":list(range(2000, 2010)) * 16,
    "value":np.random.rand(160)
})

# Inisialisasi grafik berlapis
g = sns.FacetGrid(df, col='country', hue='country', col_wrap=4)

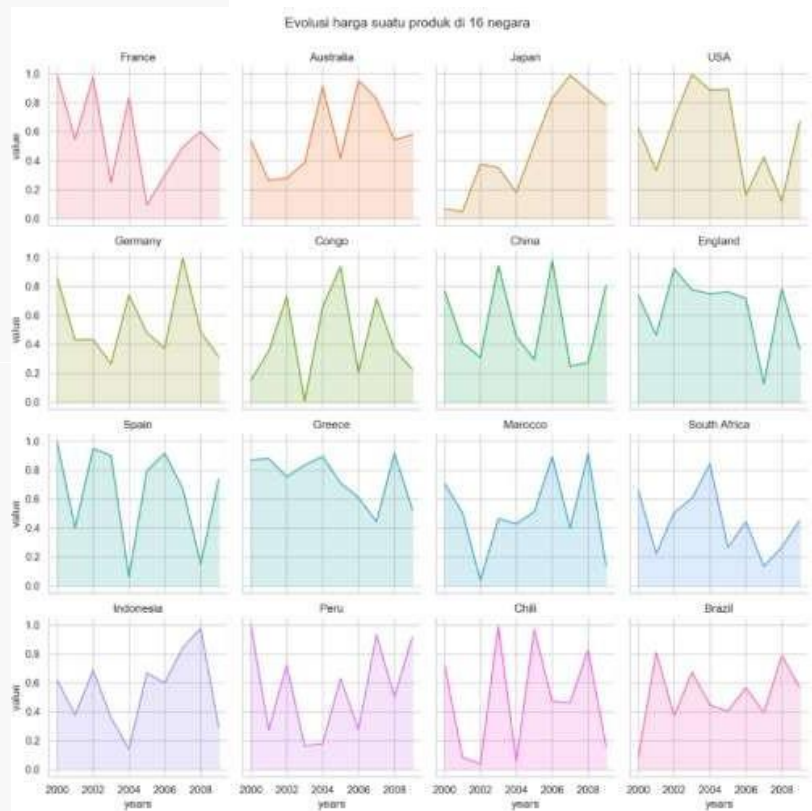
# Tambahkan garis di atas area dengan fungsi plot
g = g.map(plt.plot, 'years', 'value')

# Isi area dengan fill_between
g = g.map(plt.fill_between,
          'years',
          'value',
          alpha=0.2).set_titles("{col_name} country")

# Pengaturan nama variabel setiap aspek
g = g.set_titles("{col_name}")

# Tambahkan nama variabel untuk keseluruhan plot
plt.subplots_adjust(top=0.92)
g = g.fig.suptitle('Evolusi harga suatu produk di 16 negara ')

plt.show()
```



TUGAS

1. Buatlah grafik Batang untuk data berikut :
 - a. Buat data csv dengan nama : **data_soal_1.csv** dengan isi data sebagai berikut:

ID Pelanggan	Nama	Jenis Kelamin	Pendapatan	Produk	Harga	Jumlah	Total
1	Arif	1	600000	A	100000	4	400000
2	Dian	2	1200000	D	250000	4	1000000
3	Dinda	2	950000	D	250000	3	750000
4	Fajar	1	400000	A	100000	2	200000
5	Ika	2	1200000	D	250000	4	1000000
6	Ilham	1	800000	B	150000	4	600000
7	Indra	1	950000	B	150000	5	750000
8	Kartika	2	1100000	E	300000	3	900000
9	Lestari	2	800000	E	300000	2	600000
10	Lia	2	1700000	E	300000	5	1500000
11	Maria	2	600000	A	100000	4	400000
12	Maya	2	950000	B	150000	5	750000
13	Mila	2	400000	C	200000	1	200000
14	Nurul	2	6450000	D	250000	5	1250000

15	Retno	2	1000000	C	200000	4	800000
16	Rini	2	800000	B	150000	4	600000
17	Rizki	1	1200000	C	200000	5	1000000
18	Sari	2	700000	D	250000	2	500000
19	Tyas	2	600000	A	100000	4	400000
20	Wahyu	1	800000	C	200000	3	600000

- b. Dengan menggunakan library pandas import/baca file csv yang sudah dibuat.
 - c. Tampilkan deskripsi statistik dari data frame tersebut.
 - d. Buatlah grafik pencar dari data frame tersebut, data yang ditampilkan adalah atribut: **Pendapatan (x)** dan **Total (y)**.
 - e. Buatlah grafik batang dengan sumbu x = Nama dan sumbu y = pendapatan.
2. Dengan menggunakan librari yang ada kerjakan soal berikut :
- a. Import library yang dibutuhkan.
 - b. Buatlah dataframe dengan isi data sebagai berikut :

Id	Year	Bears	Dolphins	Whales
1	2017	8	150	80
2	2018	54	77	54
3	2019	93	32	100
4	2020	116	11	76
5	2021	137	6	93
6	2022	184	1	72

- c. Tampilkan data teratas
- d. Tampilkan data terbawah
- e. Tampilkan data statistik dari data di atas.
- f. Tampilkan grafik garis (line Chart) untuk data Tahun dan Bear
- g. Tampilkan grafik garis (line chart) untuk Bears, Dolphins dan Whales berdasarkan tahun.
- h. Tampilkan grafik area untuk data whales berdasarkan tahun.