

PYTHON

 DATA SCIENCE

by Kevin Perdana



PYTHON

Matplotlib

Matplotlib adalah library Python 2D yang dapat menghasilkan plot dengan kualitas tinggi dalam berbagai format dan dapat digunakan di banyak platform.

Matplotlib dapat digunakan sebagai pembuat grafik dalam berbagai platform, seperti Python dan Jupyter. Grafik yang dapat dibuat beragam, seperti grafik garis, batang, lingkaran, histogram, dsb.

3.1 Jenis - Jenis Plot

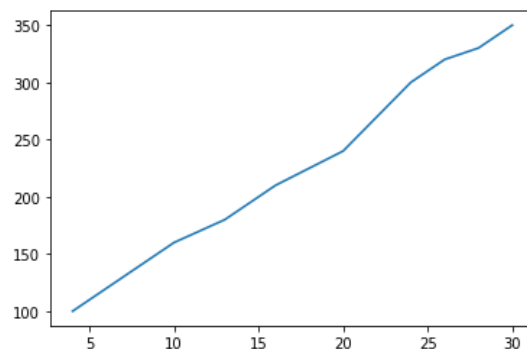
Berikut adalah jenis plot/grafik yang dapat dibuat oleh Matplotlib.

3.1.1 Plot Garis

Merupakan representasi berupa garis yang menghubungkan antar posisi koordinat data. Cara membuatnya sebagai berikut.

```
In [6]: # matplotlib adalah library python yang digunakan untuk menampilkan grafik
from matplotlib import pyplot as plt

x = [4,6,10,13,16,20,22,24,26,28,30]
y = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(x,y)
plt.show()
```



Gambar 36 Plot Garis

PYTHON

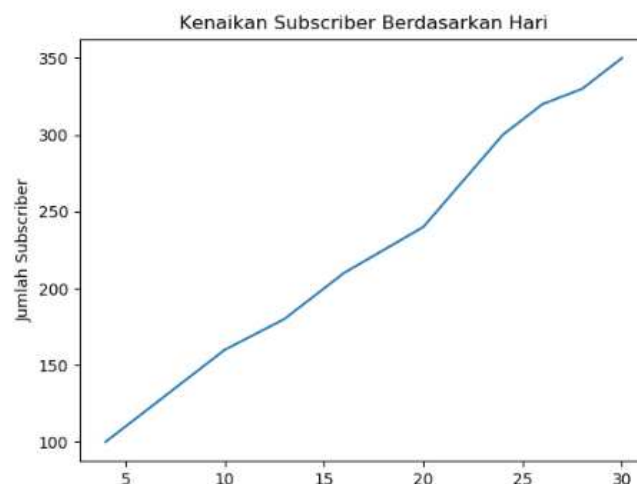
1. Menambah Label dan Title

```
In [17]: # %matplotlib notebook untuk melakukan zoom in,out,move (berinteraksi dengan plot)
%matplotlib notebook

from matplotlib import pyplot as plt

sumbox = [4,6,10,13,16,20,22,24,26,28,30]
sumbuy = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(sumbox,sumbuy)

#menambah label dan title
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.show()
```



Gambar 37 Label dan Title

2. Plot Garis Multiline

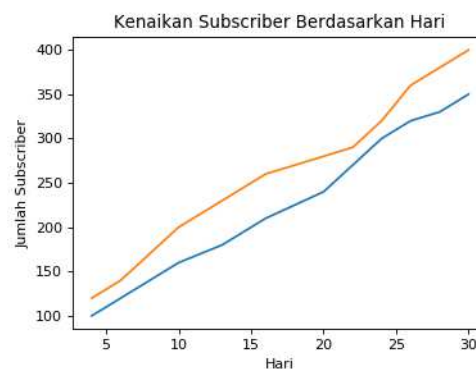
```
In [1]: %matplotlib notebook
from matplotlib import pyplot as plt

hari = [4,6,10,13,16,20,22,24,26,28,30]

# sumbu y pertama
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex)

# sumbu y kedua
atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta)

#menambah label dan title
plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')
plt.show()
```



Gambar 38 Plot Garis Multiline

PYTHON

3. Legend

```
In [23]: %matplotlib notebook

from matplotlib import pyplot as plt

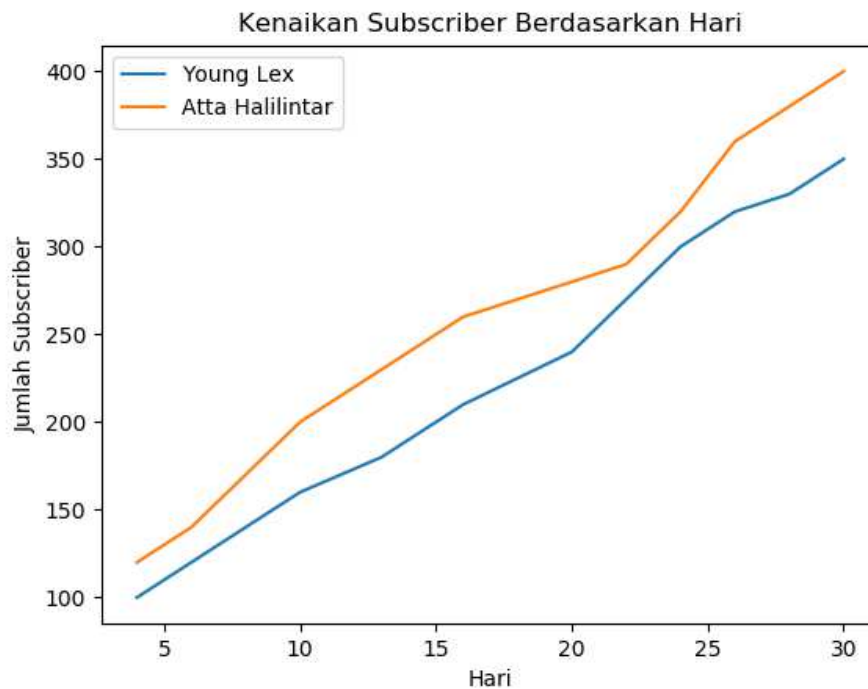
hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex)

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta)

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

#Legend, digunakan untuk memberikan info masing-masing plot
plt.legend(['Young Lex','Atta Halilintar'])

plt.show()
```



Gambar 39 Legend

PYTHON

4. Color, Linestyle, Marker

```
In [2]: %matplotlib notebook

from matplotlib import pyplot as plt

hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]

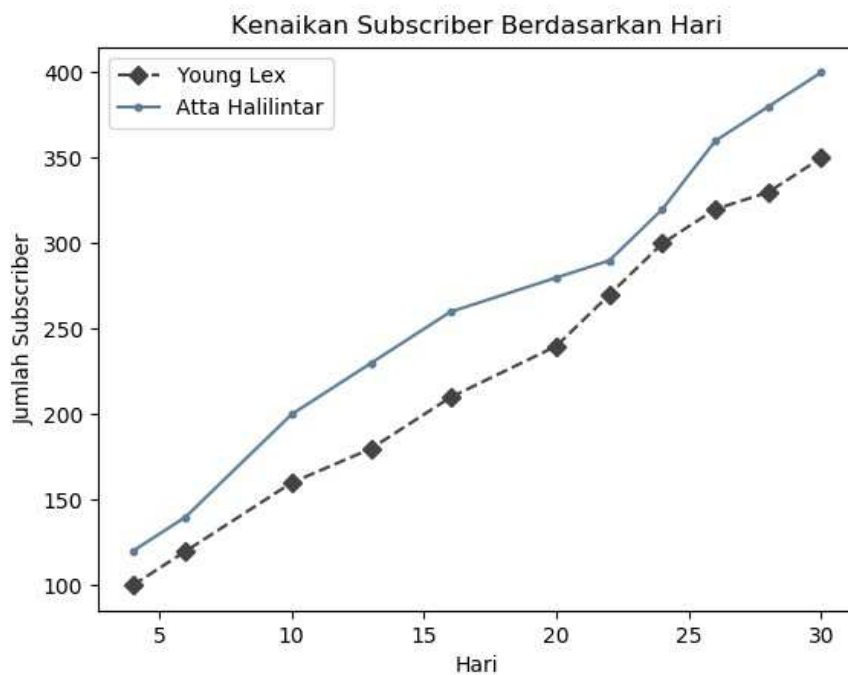
# color, untuk warna garis
# linestyle, untuk jenis garis
# -.
# --
# marker, untuk penanda pertemuan sumbu x dan y|
# p, pentagon
# D, diamond
plt.plot(hari,young_lex, color='#444444', linestyle='--', marker='D')

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', marker='.')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend(['Young Lex','Atta Halilintar'])

plt.show()
```



Gambar 40 Color, Linestyle, Marker

Untuk menggunakan lebih banyak varias **linestyle**, **marker**, dan **color** silahkan baca dokumentasi dibawah ini.

PYTHON

Markers

character	description
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
' '	vline marker
'_'	hline marker

Line Styles

character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

Example format strings:

```
'b'      # blue markers with default shape
'or'     # red circles
'-g'     # green solid line
'--'     # dashed line with default color
'^k:'    # black triangle_up markers connected by a dotted line
```

Colors

The supported color abbreviations are the single letter codes

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black

Gambar 41 Variasi Color, Linestyle, Marker yang Bisa Digunakan

PYTHON

5. Linewidth

```
In [31]: %matplotlib notebook

from matplotlib import pyplot as plt

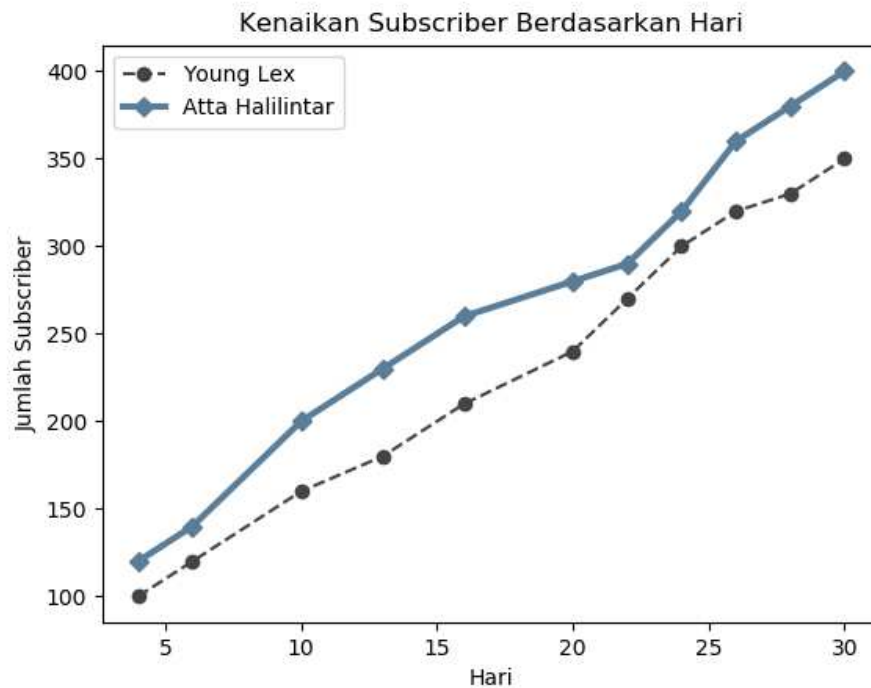
hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex, color='#444444', linestyle='--', marker='o')

# linewidth, untuk menebalkan garis
atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', marker='D', linewidth='3')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend(['Young Lex', 'Atta Halilintar'])

plt.show()
```



Gambar 42 Linewidth

PYTHON

6. Grid

```
In [32]: %matplotlib notebook

from matplotlib import pyplot as plt

hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex, color='#444444', linestyle='--', marker='o')

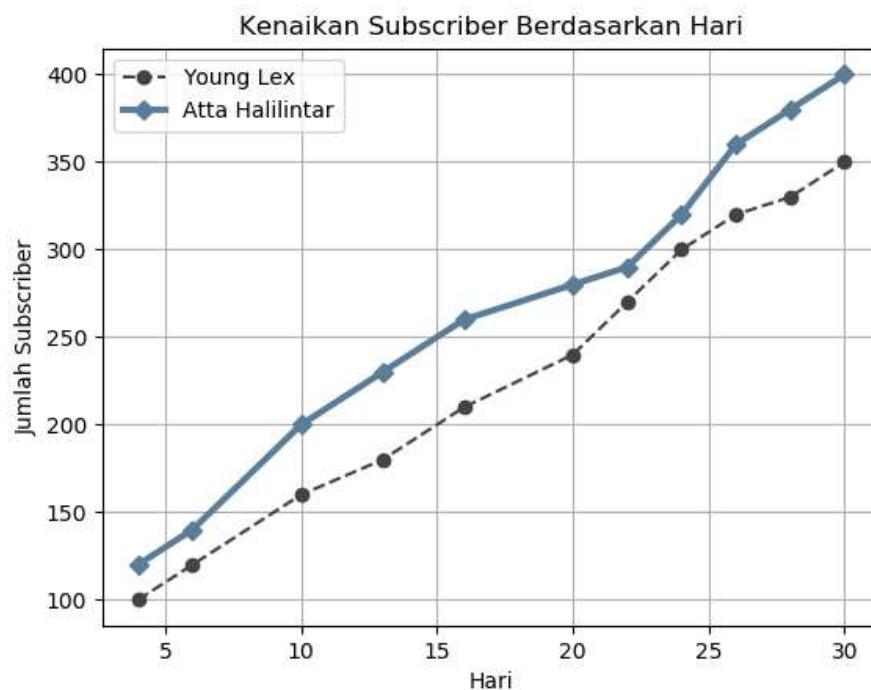
atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', marker='D', linewidth='3')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend(['Young Lex', 'Atta Halilintar'])

# grid, untuk menampilkan layout grid (susunan vertikal dan horizontal)
plt.grid(True)

plt.show()
```



Gambar 43 Grid

PYTHON

7. Menggunakan Style (fivethirtyeight) dan Tight Layout

```
In [35]: %matplotlib notebook

from matplotlib import pyplot as plt

# menggunakan style fivethirtyeight
plt.style.use('fivethirtyeight')
#plt.style.use('ggplot')

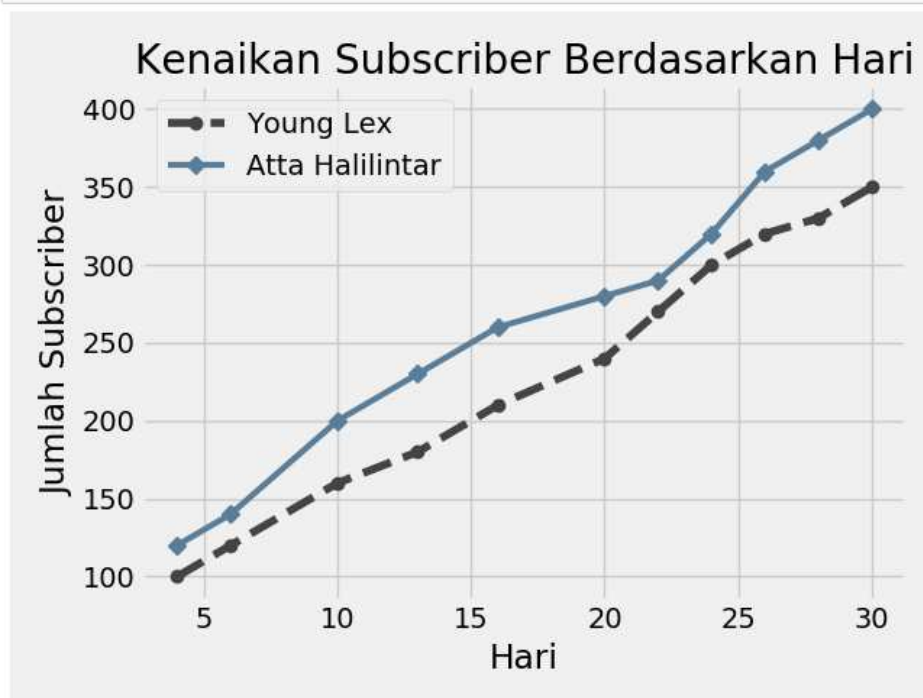
hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex, color='#444444', linestyle='--', marker='o')

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', marker='D', linewidth='3')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend(['Young Lex', 'Atta Halilintar'])
plt.grid(True)

# tight_layout(), digunakan untuk menyesuaikan plot, label, title di dalam area gambar
plt.tight_layout()
plt.show()
```



Gambar 44 Style & Tight Layout

PYTHON

8. Menggunakan Style ggplot

```
In [37]: %matplotlib notebook

from matplotlib import pyplot as plt

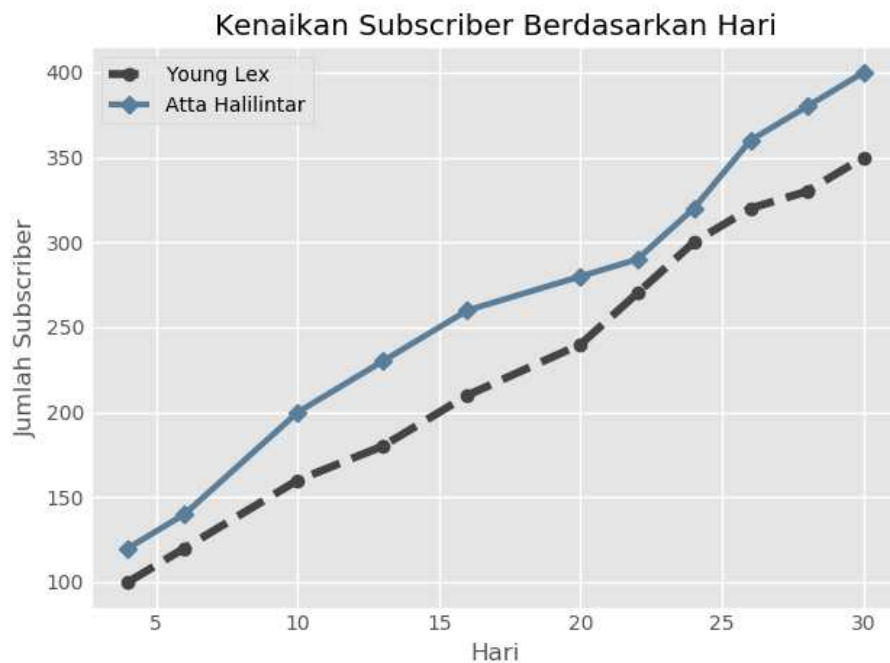
# menggunakan style ggplot
plt.style.use('ggplot')

hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.plot(hari,young_lex, color='#444444', linestyle='--', marker='o')

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', marker='D', linewidth='3')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend(['Young Lex', 'Atta Halilintar'])
plt.grid(True)
plt.tight_layout()
plt.show()
```



Gambar 45 Style ggplot

PYTHON

3.1.2 Plot Garis dan Bar

Ketika menggunakan plot bar, **tidak boleh** menggunakan **linestyle** dan **marker**. Ini hanya berlaku untuk plot garis.

```
In [44]: %matplotlib notebook

from matplotlib import pyplot as plt

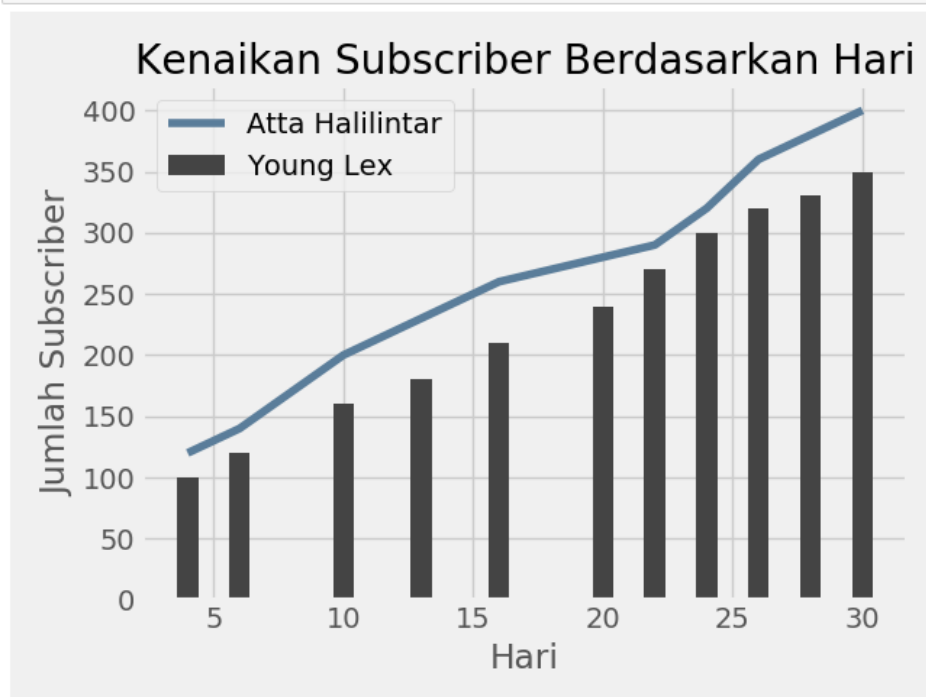
plt.style.use('fivethirtyeight')

hari = [4,6,10,13,16,20,22,24,26,28,30]
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.bar(hari,young_lex, color='#444444', label='Young Lex')

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.plot(hari,atta, color='#5A7D9A', label='Atta Halilintar')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



Gambar 46 Plot Garis dan Bar

PYTHON

3.1.3 Plot Bar

Ketika menggunakan plot bar. Jika ada 2 data (2 sumbu y), maka **dataset** yang bilangannya **paling tinggi** harus diletakkan **diatas**. Sehingga pada grafik akan tampil **dibelakang** bar lain.

```
In [90]: %matplotlib notebook

from matplotlib import pyplot as plt

plt.style.use('fivethirtyeight')

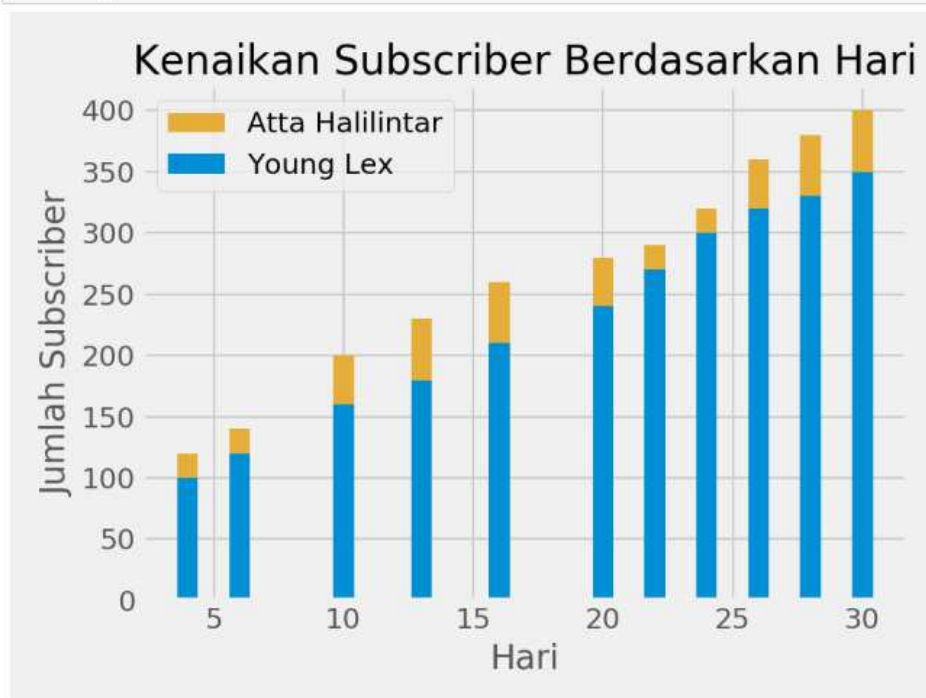
hari = [4,6,10,13,16,20,22,24,26,28,30]

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.bar(hari,atta, color='#E5AE38', label='Atta Halilintar')

young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.bar(hari,young_lex, color='#008FD5', label='Young Lex')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



Gambar 47 Plot Bar

PYTHON

3.1.4 Plot Bar (Side by Side)

```
In [95]: %matplotlib notebook

from matplotlib import pyplot as plt
import numpy as np

plt.style.use('fivethirtyeight')

hari = [4,6,10,13,16,20,22,24,26,28,30]

# menghitung posisi
# np dari modul numpy untuk mengelola data array
# len untuk menghitung jumlah karakter atau array/list
x_indeks = np.arange(len(hari))

# lebar bar adalah 0.35 per 1 unit bar
lebar_bar = 0.35
print(len(hari))
print(x_indeks)
print(x_indeks-lebar_bar)

atta = [120,140,200,230,260,280,290,320,360,380,400]
plt.bar(x_indeks - lebar_bar, atta, width=lebar_bar, color='#E5AE38', label='Atta Halililar')

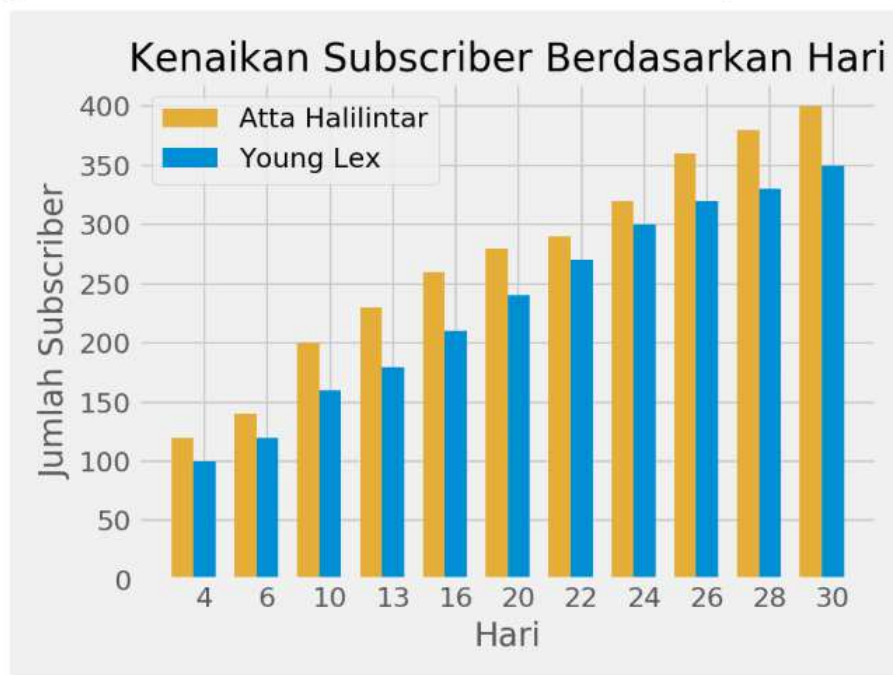
young_lex = [100,120,160,180,210,240,270,300,320,330,350]
plt.bar(x_indeks, young_lex, width=lebar_bar, color='#008FD5', label='Young Lex')

plt.title('Kenaikan Subscriber Berdasarkan Hari')
plt.xlabel('Hari')
plt.ylabel('Jumlah Subscriber')

# xticks digunakan untuk menentukan posisi dan meng-set label sesuai data list
plt.xticks(ticks=x_indeks, labels=hari)

plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

11
[ 0  1  2  3  4  5  6  7  8  9 10]
[-0.35  0.65  1.65  2.65  3.65  4.65  5.65  6.65  7.65  8.65  9.65]
```



Gambar 48 Plot Bar (Side by Side)

PYTHON

3.2 Matplotlib - Import Data CSV

Dibawah ini adalah praktikum cara menampilkan plot dengan menggunakan file CSV.

3.2.1 Dengan Plot Bar

Ikuti langkah-langkah dibawah untuk menampilkan data dari CSV menjadi plot bar.

1. Tampilkan Key dan Value

```
In [9]: from matplotlib import pyplot as plt
import numpy as np
import csv

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    # menampilkan key dan value
    row = next(csv_reader)
    print(row)

OrderedDict([('Id', '1'), ('JumlahPengguna', 'HTML/CSS;Java;JavaScript;Python')])
```

Gambar 49 Tampilkan Key dan Value

2. Tampilkan Value Berdasarkan Nama Key

```
In [11]: from matplotlib import pyplot as plt
import numpy as np
import csv

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    # hanya menampilkan value berdasarkan nama "key"
    row = next(csv_reader)
    print(row['JumlahPengguna'].split(';'))

['HTML/CSS', 'Java', 'JavaScript', 'Python']
```

Gambar 50 Tampilkan Value Berdasarkan Nama Key

PYTHON

3. Hitung Semua Jumlah Pengguna

```
In [13]: from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    # hitung semua jumlah pengguna
    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    print(hitung)

Counter({'JavaScript': 59219, 'HTML/CSS': 55466, 'SQL': 47544, 'Python': 36443, 'Java': 35917, 'Bash/Shell/PowerShell': 31991, 'C#': 27097, 'PHP': 23030, 'C++': 20524, 'TypeScript': 18523, 'C': 18017, 'Other(s)': 7920, 'Ruby': 7331, 'Go': 7201, 'Assembly': 5833, 'Swift': 5744, 'Kotlin': 5620, 'R': 5048, 'VBA': 4781, 'Objective-C': 4191, 'Scala': 3309, 'Rust': 2794, 'Dart': 1683, 'Elixir': 1260, 'Clojure': 1254, 'WebAssembly': 1015, 'F#': 973, 'Erlang': 777})
```

Gambar 51 Hitung Semua Jumlah Pengguna

4. Tampilkan Jumlah Pengguna Teratas dan Batas Tertentu

```
In [14]: from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    # most_common(15) adalah menampilkan jumlah paling banyak 15 teratas
    print(hitung.most_common(15))

[('JavaScript', 59219), ('HTML/CSS', 55466), ('SQL', 47544), ('Python', 36443), ('Java', 35917), ('Bash/Shell/PowerShell', 31991), ('C#', 27097), ('PHP', 23030), ('C++', 20524), ('TypeScript', 18523), ('C', 18017), ('Other(s)', 7920), ('Ruby', 7331), ('Go', 7201), ('Assembly', 5833)]
```

Gambar 52 Tampilkan Jumlah Pengguna Teratas dan Batas Tertentu

5. Pisahkan Value Bahasa dan Jumlah Pengguna ke Dalam Masing - Masing List

```
In [16]: from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    bahasa = []
    pengguna = []

    for item in hitung.most_common(15):
        bahasa.append(item[0])
        pengguna.append(item[1])

    print(bahasa)
    print(pengguna)

['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java', 'Bash/Shell/PowerShell', 'C#', 'PHP', 'C++', 'TypeScript', 'C', 'Other(s)', 'Ruby', 'Go', 'Assembly']
[59219, 55466, 47544, 36443, 35917, 31991, 27097, 23030, 20524, 18523, 18017, 7920, 7331, 7201, 5833]
```

Gambar 53 Pisahkan Value Bahasa dan Jumlah Pengguna ke Dalam Masing - Masing List

PYTHON

6. Tampilkan Menjadi Plot Bar

```
from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    bahasa = []
    pengguna = []

    for item in hitung.most_common(15):
        bahasa.append(item[0])
        pengguna.append(item[1])

plt.bar(bahasa, pengguna)

plt.title("Bahasa Pemrograman yang Paling Banyak Digunakan")
plt.xlabel("Bahasa Pemrograman")
plt.ylabel("Jumlah Pengguna")
plt.show
```



Gambar 54 Tampilkan Menjadi Plot Bar

PYTHON

7. Ubah Menjadi Horizontal (Jika Diperlukan)

```
from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    bahasa = []
    pengguna = []

    for item in hitung.most_common(15):
        bahasa.append(item[0])
        pengguna.append(item[1])

plt.barh(bahasa, pengguna)

plt.title("Bahasa Pemrograman yang Paling Banyak Digunakan")
plt.ylabel("Bahasa Pemrograman")
plt.xlabel("Jumlah Pengguna")
plt.show
```



Gambar 55 Ubah Menjadi Horizontal (Jika Diperlukan)

PYTHON

8. Reverse (Tampilkan Data dari Banyak ke Sedikit)

```
from matplotlib import pyplot as plt
import numpy as np
import csv
from collections import Counter

plt.style.use('fivethirtyeight')

with open('penggunabahasapemrograman.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)

    hitung = Counter()

    for row in csv_reader:
        hitung.update(row['JumlahPengguna'].split(';'))

    bahasa = []
    pengguna = []

    for item in hitung.most_common(15):
        bahasa.append(item[0])
        pengguna.append(item[1])

    # menampilkan data dari yang paling banyak
    # reverse, digunakan untuk menampilkan kebalikan dari jumlah sebelumnya
    bahasa.reverse()
    pengguna.reverse()

    plt.barh(bahasa, pengguna)

    plt.title("Bahasa Pemrograman yang Paling Banyak Digunakan")
    plt.ylabel("Bahasa Pemrograman")
    plt.xlabel("Jumlah Pengguna")
    plt.show
```



Gambar 56 Reverse

PYTHON

3.3 Plot Pie Chart

Dibawah ini merupakan cara menampilkan data menjadi Pie Chart.

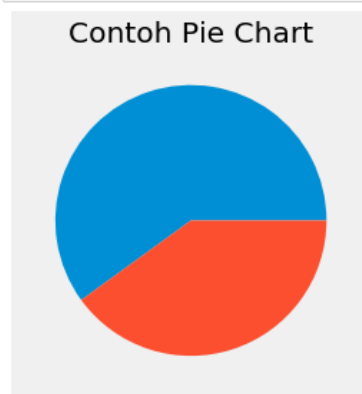
1. Slices

```
In [52]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

# slices digunakan untuk membagi pie menjadi beberapa bagian
slices = [60, 40]
plt.pie(slices)

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 57 Slices

2. Labels

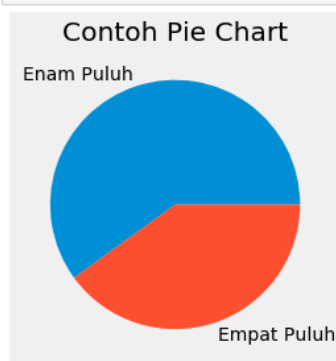
```
In [5]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

# slices digunakan untuk membagi pie menjadi beberapa bagian
slices = [60, 40]

# labels untuk menampilkan label per bagian
label = ['Enam Puluh', 'Empat Puluh']
plt.pie(slices, labels=label)

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 58 Labels

PYTHON

3. Wedgeprops

```
In [6]: from matplotlib import pyplot as plt

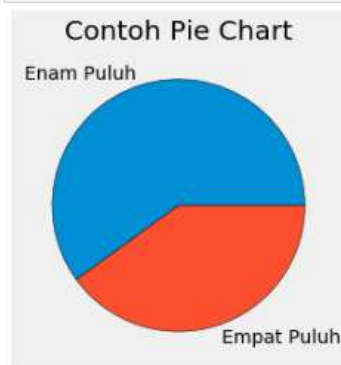
plt.style.use("fivethirtyeight")

# slices digunakan untuk membagi pie menjadi beberapa bagian
slices = [60, 40]

# labels untuk menampilkan label per bagian
label = ['Enam Puluh', 'Empat Puluh']

# wedgeprops untuk membuat garis pinggir/border pie
plt.pie(slices, labels=label, wedgeprops={'edgecolor': 'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 59 Wedgeprops

4. Colors

```
In [8]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

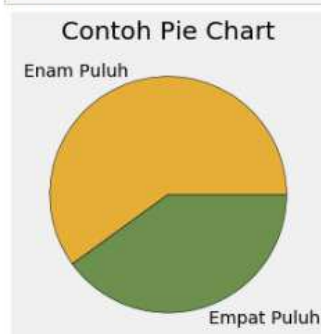
# slices digunakan untuk membagi pie menjadi beberapa bagian
slices = [60, 40]

# labels untuk menampilkan label per bagian
label = ['Enam Puluh', 'Empat Puluh']

# colors merubah warna pie
warna = ['#E5AE37', '#6D904F']

# wedgeprops
plt.pie(slices, labels=label, colors=warna, wedgeprops={'edgecolor': 'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 60 Colors

PYTHON

5. Explode

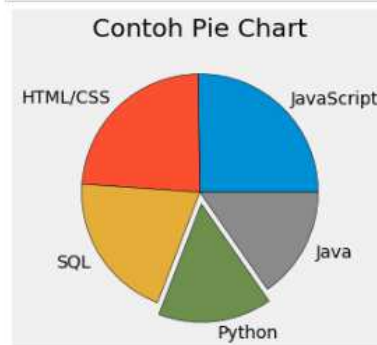
```
In [29]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

slices = [59219, 55466, 47544, 36443, 35917]
label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
warna = ['#E5AE37', '#6D904F']

# explode artinya fraksi atau pecahan pie
explode = [0, 0, 0, 0.1, 0]
plt.pie(slices, labels=label, explode=explode, wedgeprops={'edgecolor':'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 61 Explode

6. Shadow

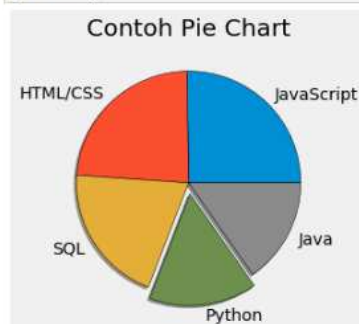
```
In [17]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

slices = [59219, 55466, 47544, 36443, 35917]
label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
warna = ['#E5AE37', '#6D904F']
explode = [0, 0, 0, 0.1, 0]

# shadow=True untuk menampilkan bayangan
plt.pie(slices, labels=label, explode=explode, shadow=True, wedgeprops={'edgecolor':'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 62 Shadow

PYTHON

7. Startangle

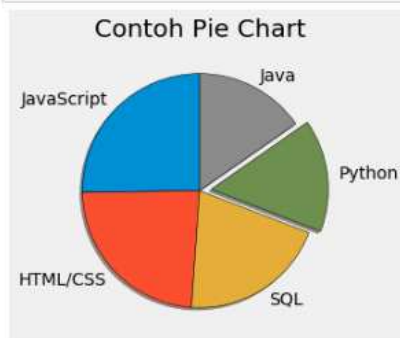
```
In [54]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

slices = [59219, 55466, 47544, 36443, 35917]
label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
warna = ['#E5AE37', '#6D904F']
explode = [0, 0, 0, 0.1, 0]

# angle digunakan untuk memutar derajat pie
# 90 artinya memutar 90 derajat berlawanan arah jarum jam
# -90 artinya memutar 90 derajat searah jarum jam
plt.pie(slices, labels=label, explode=explode, shadow=True, startangle=90, wedgeprops={'edgecolor':'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 63 Startangle

8. Autopct

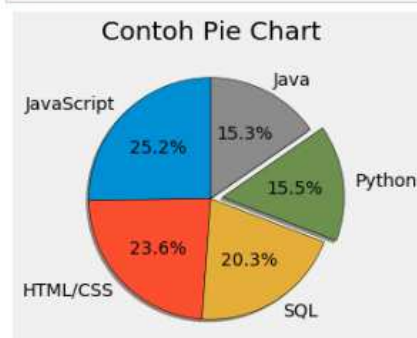
```
In [50]: from matplotlib import pyplot as plt

plt.style.use("fivethirtyeight")

slices = [59219, 55466, 47544, 36443, 35917]
label = ['JavaScript', 'HTML/CSS', 'SQL', 'Python', 'Java']
warna = ['#E5AE37', '#6D904F']
explode = [0, 0, 0, 0.1, 0]

# autopct untuk menampilkan persen bagian pie
# %f artinya menampilkan bilangan float. Harus diawali %
# %0.1f menampilkan bilangan float dengan 1 angka dibelakang koma
# %0.1f%% menampilkan simbol % itu sendiri dan ditutup dengan %
plt.pie(slices, labels=label, explode=explode, shadow=True,
        startangle=90, autopct='%0.1f%%', wedgeprops={'edgecolor':'black'})

plt.title("Contoh Pie Chart")
plt.tight_layout()
plt.show()
```



Gambar 64 Autopct