

# **CERDAS MENGUASAI PYTHON**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Matplotlib</b>
----------	-------------------

<b>1</b>
----------



# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1    Matplotlib</b>	<b>1</b>
1.1    Bakti Qilan Mufid   1174083	1
1.1.1    Buatlah library fungsi (file terpisah/library dengan nama NPM_bar.py) untuk plot dengan jumlah subplot adalah $NPM \bmod 3 + 2$	1
	<b>ix</b>



1.1.2	Buatlah library fungsi (file terpisah/library dengan nama NPM_scatter.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	3
1.1.3	Buatlah library fungsi (file terpisah/library dengan nama NPM_pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	4
1.1.4	Buatlah library fungsi (file terpisah/library dengan nama NPM_plot.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2	5
1.1.5	Keterampilan Penanganan Error	7
1.2	Mochamad Arifqi Ramadhan   1174074	7
1.2.1	Keterampilan Pemograman	7
1.3	Advent Nopele Olansi Damiahan Sihite / 1174089)	12
1.3.1	Teori	12
1.3.2	Praktek	24
1.3.3	Penanganan Error	29
1.3.4	Screenshoot Plagiat	30
1.3.5	Screenshoot Kode Program	31
	Daftar Pustaka	35
	Index	37

# DAFTAR GAMBAR

---

1.1	Hasil compile membuat diagram menggunakan Matplotlib.	13
1.2	Hasil compile membuat bar graph menggunakan Matplotlib.	14
1.3	Hasil compile membuat histogram menggunakan Matplotlib.	15
1.4	Hasil compile membuat scatter plot menggunakan Matplotlib.	16
1.5	Hasil compile membuat diagram menggunakan Matplotlib.	17
1.6	Hasil compile membuat Pie Plot menggunakan Matplotlib.	18
1.7	Hasil compile membuat diagram menggunakan Matplotlib.	19
1.8	Hasil compile membuat diagram menggunakan Matplotlib.	20
1.9	Hasil compile membuat subplot menggunakan Matplotlib.	22
1.10	Hasil compile membuat diagram menggunakan Matplotlib.	23
1.11	Hasil compile membuat fungsi Bar Plot menggunakan Matplotlib.	25
1.12	Hasil compile membuat fungsi Scatter Plot menggunakan Matplotlib.	26

1.13	Hasil compile membuat fungsi Pie Plot menggunakan Matplotlib.	27
1.14	Hasil compile membuat fungsi Plot menggunakan Matplotlib.	28
1.15	Hasil compile membuat fungsi penanganan error.	30

# DAFTAR TABEL

---



# Listings

---

src/6/1174083/Praktek/1174083_bar.py	1
src/6/1174083/Praktek/1174083_scatter.py	3
src/6/1174083/Praktek/1174083_pie.py	4
src/6/1174083/Praktek/1174083_plot.py	5
src/6/1174083/Praktek/error.py	7
src/6/1174074/Praktek/a1174074_bar.py	7
src/6/1174074/Praktek/main.py	8
src/6/1174074/Praktek/a1174074_scatter.py	8
src/6/1174074/Praktek/main.py	9
src/6/1174074/Praktek/a1174074_pie.py	9
src/6/1174074/Praktek/main.py	10
src/6/1174074/Praktek/a1174074_plot.py	10
src/6/1174074/Praktek/main.py	11
src/6/1174074/Praktek/1174074_error.py	11
src/6/1174089/Praktek/1174089.py	12
src/6/1174089/Praktek/1174089.py	12
src/6/1174089/Praktek/1174089.py	12
src/6/1174089/Praktek/1174089.py	12

1.1	Kode program membuat diagram menggunakan Matplotlib.	13
1.2	Kode program membuat bar graph menggunakan Matplotlib.	14
1.3	Kode program membuat histogram menggunakan Matplotlib.	15
1.4	Kode program membuat scatter plot menggunakan Matplotlib.	16
1.5	Kode program membuat diagram menggunakan Matplotlib.	17
1.6	Kode program membuat Pie Plot menggunakan Matplotlib.	18
1.7	Kode program membuat diagram menggunakan Matplotlib.	18
1.8	Kode program menggunakan parameter label dengan Matplotlib.	19
1.9	Kode program memanggil fungsi legend dengan Matplotlib.	20
1.10	Kode program membuat diagram menggunakan Matplotlib.	20
1.11	Kode program membuat subplot menggunakan Matplotlib.	21
1.12	Kode program membuat diagram menggunakan Matplotlib.	23
1.13	Kode program membuat fungsi Bar Plot menggunakan Matplotlib.	24
1.14	Kode program membuat fungsi Scatter Plot menggunakan Matplotlib.	25
1.15	Kode program membuat fungsi Pie Plot menggunakan Matplotlib.	26
1.16	Kode program membuat fungsi Plot menggunakan Matplotlib.	27
1.17	Kode program membuat fungsi penanganan error.	29

# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa





# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan flask sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.



# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald





# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$



# BAB 1

---

## MATPLOTLIB

---

### 1.1 Bakti Qilan Mufid | 1174083

#### 1.1.1 Buatlah library fungsi (file terpisah/library dengan nama NPM\_bar.py) untuk plot dengan jumlah subplot adalah $NPM \bmod 3 + 2$

Pertama-tama kita harus mengetahui hasil dari  $NPM \bmod 3 + 2$  terlebih dahulu, lalu sesudah itu kita membuat subplotnya, seperti pada kode berikut:

```
1 from matplotlib import pyplot as plot
2 import numpy as np
3 def bar(b):
4     if b == 4:
5         x = [2,5,6,2,8]
6         y = [1,3,7,8,9]
7         x1 = [10,23,22,11,25,22,17,12,15,18]
8         y1 = [23,22,25,24,21,22,26,29,11,16]
9         x2 = [1,2,3,4,5,6,7,8,9,10]
10        y2 = [11,12,13,14,15,16,17,18,19,20]
11        x3 = [2,4,6,8,10,12]
12        y3 = [3,5,7,9,11,13]
```

```

13     plot.subplot(221)
14     plot.bar(x,y)
15     plot.subplot(222)
16     plot.bar(x1,y1)
17     plot.subplot(223)
18     plot.bar(x2,y2)
19     plot.subplot(224)
20     plot.bar(x3,y3)
21     plot.show()
22 elif b == 3:
23     x = [1,2,3,4,5]
24     y = [2,3,6,5,7]
25     x1 = [10,23,22,11,25,22,17,12,15,18]
26     y1 = [23,22,25,24,21,22,26,29,11,16]
27     x2 = [1,2,3,4,5,6,7,8,9,10]
28     y2 = [11,12,13,14,15,16,17,18,19,20]
29     plot.subplot(221)
30     plot.bar(x,y)
31     plot.subplot(222)
32     plot.bar(x1,y1)
33     plot.subplot(223)
34     plot.bar(x2,y2)
35     plot.show()
36 elif b == 2:
37     x = np.random.randint(0, 10, 10)
38     y = np.random.randint(0, 10, 10)
39     x1 = np.random.randint(10, 20, 10)
40     y1 = np.random.randint(10, 20, 10)
41     plot.subplot(211)
42     plot.bar(x,y)
43     plot.subplot(212)
44     plot.bar(x1,y1)
45     plot.show()
46 elif b == 1:
47     x = np.random.randint(0, 10, 10)
48     y = np.random.randint(0, 10, 10)
49     x1 = np.random.randint(10, 20, 10)
50     y1 = np.random.randint(10, 20, 10)
51     plot.subplot(111)
52     plot.bar(x,y)
53     plot.show()
54 else:
55     print("Data tidak Valid")
56
57
58 npm = input("Masukkan NPM Anda: ")
59 modnpm = int(npm)%3+2
60 print(modnpm)
61
62 if modnpm == 4:
63     bar(4)
64 elif modnpm == 3:
65     bar(3)
66 elif modnpm == 2:
67     bar(2)
68 elif modnpm == 1:

```

```

69     bar(1)
70 else :
71     print("Data tidak Valid")

```

### 1.1.2 Buatlah library fungsi (file terpisah/library dengan nama NPM\_scatter.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Pertama-tama kita harus mengetahui hasil dari NPM mod 3 + 2 terlebih dahulu, lalu sesudah itu kita membuat subplotnya, seperti pada kode berikut:

```

1 from matplotlib import pyplot as plot
2 import numpy as np
3 def scatter(s):
4     if s == 4:
5         x = [2,5,6,2,8]
6         y = [1,3,7,8,9]
7         x1 = [10,23,22,11,25,22,17,12,15,18]
8         y1 = [23,22,25,24,21,22,26,29,11,16]
9         x2 = [1,2,3,4,5,6,7,8,9,10]
10        y2 = [11,12,13,14,15,16,17,18,19,20]
11        x3 = [2,4,6,8,10,12]
12        y3 = [3,5,7,9,11,13]
13        plot.subplot(221)
14        plot.scatter(x,y)
15        plot.subplot(222)
16        plot.scatter(x1,y1)
17        plot.subplot(223)
18        plot.scatter(x2,y2)
19        plot.subplot(224)
20        plot.scatter(x3,y3)
21        plot.show()
22    elif s == 3:
23        x = [1,2,3,4,5]
24        y = [2,3,6,5,7]
25        x1 = [10,23,22,11,25,22,17,12,15,18]
26        y1 = [23,22,25,24,21,22,26,29,11,16]
27        x2 = [1,2,3,4,5,6,7,8,9,10]
28        y2 = [11,12,13,14,15,16,17,18,19,20]
29        plot.subplot(221)
30        plot.scatter(x,y)
31        plot.subplot(222)
32        plot.scatter(x1,y1)
33        plot.subplot(223)
34        plot.scatter(x2,y2)
35        plot.show()
36    elif s == 2:
37        x = np.random.randint(0, 10, 10)
38        y = np.random.randint(0, 10, 10)
39        x1 = np.random.randint(10, 20, 10)
40        y1 = np.random.randint(10, 20, 10)
41        plot.subplot(211)
42        plot.scatter(x,y)
43        plot.subplot(212)

```



```

44     plot.scatter(x1,y1)
45     plot.show()
46     elif s == 1:
47         x = np.random.randint(0, 10, 10)
48         y = np.random.randint(0, 10, 10)
49         x1 = np.random.randint(10, 20, 10)
50         y1 = np.random.randint(10, 20, 10)
51         plot.subplot(111)
52         plot.scatter(x,y)
53         plot.show()
54     else:
55         print("Data tidak Valid")
56
57
58 npm = input("Masukkan NPM Anda: ")
59 modnpm = int(npm)%3+2
60 print(modnpm)
61
62 if modnpm == 4:
63     scatter(4)
64 elif modnpm == 3:
65     scatter(3)
66 elif modnpm == 2:
67     scatter(2)
68 elif modnpm == 1:
69     scatter(1)
70 else:
71     print("Data tidak Valid")

```

### 1.1.3 Buatlah library fungsi (file terpisah/library dengan nama NPM\_pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Pertama-tama kita harus mengetahui hasil dari NPM mod 3 + 2 terlebih dahulu, lalu sesudah itu kita membuat subplotnya, seperti pada kode berikut:

```

1 from matplotlib import pyplot as plt
2
3 def pie(mod):
4     if mod == 2:
5         merk= [5,10,2,3]
6         prog = [7,5,6,1]
7         jns_merk = ['Acer','Asus','Lenovo','HP']
8         programming = ['Perl','Java','C++','Python']
9         cols = ['silver','lime','gold','aqua']
10        plt.subplot(121)
11        plt.pie(merk,
12               labels=jns_merk,
13               colors=cols,
14               startangle=0,
15               shadow= True,
16               explode=(0.2,0,0,0),
17               autopct='%1.1f%%')
18        plt.title('Pie Laptop')

```

```

19         plt.subplot(122)
20         plt.pie(prog,
21                 labels=programming,
22                 colors=cols,
23                 startangle=90,
24                 shadow=True,
25                 explode=(.1,0,0,0),
26                 autopct='%1.1f%%')
27         plt.title('Pie Programming')
28         plt.show()
29     else:
30         print("Data tidak ada")
31
32
33 npm = input("Masukkan NPM Anda: ")
34 modnpm = int(npm)%3+2
35 print(modnpm)
36
37 if modnpm == 4:
38     print("Data tidak ada")
39 elif modnpm == 3:
40     print("Data tidak ada")
41 elif modnpm == 2:
42     pie(2)
43 elif modnpm == 1:
44     print("Data tidak ada")
45 else:
46     print("Data tidak Valid")

```

### 1.1.4 Buatlah library fungsi (file terpisah/library dengan nama NPM\_plot.py) untuk plot dengan jumlah subplot adalah $NPM \bmod 3 + 2$

Pertama-tama kita harus mengetahui hasil dari  $NPM \bmod 3 + 2$  terlebih dahulu, lalu sesudah itu kita membuat subplotnya, seperti pada kode berikut:

```

1 from matplotlib import pyplot as plt
2 import numpy as np
3 def plot(p):
4     if p == 4:
5         x = [2,5,6,2,8]
6         y = [1,3,7,8,9]
7         x1 = [10,23,22,11,25,22,17,12,15,18]
8         y1 = [23,22,25,24,21,22,26,29,11,16]
9         x2 = [1,2,3,4,5,6,7,8,9,10]
10        y2 = [11,12,13,14,15,16,17,18,19,20]
11        x3 = [2,4,6,8,10,12]
12        y3 = [3,5,7,9,11,13]
13        plt.subplot(221)
14        plt.plot(x,y)
15        plt.subplot(222)
16        plt.plot(x1,y1)
17        plt.subplot(223)
18        plt.plot(x2,y2)

```

```

19     plt.subplot(224)
20     plt.plot(x3,y3)
21     plt.show()
22 elif p == 3:
23     x = [1,2,3,4,5]
24     y = [2,3,6,5,7]
25     x1 = [10,23,22,11,25,22,17,12,15,18]
26     y1 = [23,22,25,24,21,22,26,29,11,16]
27     x2 = [1,2,3,4,5,6,7,8,9,10]
28     y2 = [11,12,13,14,15,16,17,18,19,20]
29     plt.subplot(221)
30     plt.plot(x,y)
31     plt.subplot(222)
32     plt.plot(x1,y1)
33     plt.subplot(223)
34     plt.plot(x2,y2)
35     plt.show()
36 elif p == 2:
37     x = [1,2,3,4,5]
38     y = [2,3,6,5,7]
39     x1 = [2,4,6,8,10,12,14,16,18,20]
40     y1 = [23,22,25,24,21,22,26,29,11,16]
41     plt.subplot(211)
42     plt.plot(x,y)
43     plt.subplot(212)
44     plt.plot(x1,y1)
45     plt.show()
46 elif p == 1:
47     x = np.random.randint(0, 10, 10)
48     y = np.random.randint(0, 10, 10)
49     x1 = np.random.randint(10, 20, 10)
50     y1 = np.random.randint(10, 20, 10)
51     plt.subplot(111)
52     plt.plot(x,y)
53     plt.show()
54 else:
55     print("Data tidak Valid")
56
57
58 npm = input("Masukkan NPM Anda: ")
59 modnpm = int(npm)%3+2
60 print(modnpm)
61
62 if modnpm == 4:
63     plot(4)
64 elif modnpm == 3:
65     plot(3)
66 elif modnpm == 2:
67     plot(2)
68 elif modnpm == 1:
69     plot(1)
70 else:
71     print("Data tidak Valid")

```

## 1.1.5 Keterampilan Penanganan Error

Fungsi Penanganan error sebagai berikut:

```

1 from matplotlib import pyplot as plot
2
3 def PenangananError():
4     try:
5         a=[1,2,3]
6         b=[5,2,4]
7         plot.plot(a,b)
8         plot.show()
9     except SyntaxError:
10        print("Kesalahan pada penulisan syntax")
11    except NameError:
12        print("Variable yang anda maksud tidak ada")
13    except TypeError:
14        print("Tipe data salah")
15    except:
16        print("Terjadi error")
17
18 PenangananError()
```

## 1.2 Mochamad Arifqi Ramadhan | 1174074

### 1.2.1 Keterampilan Pemograman

**1.2.1.1 Buatlah library fungsi (file terpisah/library dengan nama *NPM/textunderscore bar.py*) untuk plot dengan jumlah subplot adalah  $NPM \bmod 3 + 2$**  Ini adalah fungsi untuk membuat plot bar sesuai dengan hasil modulus:

```

1 # No 1
2 from matplotlib import pyplot as plt
3 import numpy as np
4 def bar(mod):
5     if mod == 3:
6         x = [1,2,3,4,5]
7         y = [2,3,6,5,7]
8         x1 = [10,23,22,11,25,22,17,12,15,18]
9         y1 = [23,22,25,24,21,22,26,29,11,16]
10        x2 = [1,2,3,4,5,6,7,8,9,10]
11        y2 = [11,12,13,14,15,16,17,18,19,20]
12        plt.subplot(221)
13        plt.bar(x,y)
14        plt.subplot(222)
15        plt.bar(x1,y1)
16        plt.subplot(223)
17        plt.bar(x2,y2)
18        plt.show()
19    elif mod == 2:
20        x = np.random.randint(0, 10, 10)
21        y = np.random.randint(0, 10, 10)
22        x1 = np.random.randint(10, 20, 10)
23        y1 = np.random.randint(10, 20, 10)
```

```

24     plt.subplot(211)
25     plt.bar(x,y)
26     plt.subplot(212)
27     plt.bar(x1,y1)
28     plt.show()
29     elif mod == 1:
30         x = np.random.randint(0, 10, 10)
31         y = np.random.randint(0, 10, 10)
32         x1 = np.random.randint(10, 20, 10)
33         y1 = np.random.randint(10, 20, 10)
34         plt.subplot(111)
35         plt.bar(x,y)
36         plt.show()
37     else:
38         print("Data tidak Valid")

```

Berikut ini cara pemanggilannya:

```

1  ### Pemanggilan Plot Bar
2  import all74074_bar as bar
3
4  npm = input("Masukkan NPM Anda: ")
5  modnpm = int(npm)%3+2
6  print(modnpm)
7
8  if modnpm == 3:
9      bar.bar(3)
10 elif modnpm == 2:
11     bar.bar(2)
12 elif modnpm == 1:
13     bar.bar(1)
14 else:
15     print("NPM Salah")

```

### 1.2.1.2 Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore scatter.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2

Ini adalah fungsi untuk membuat plot scatter sesuai dengan hasil modulus:

```

1  #No 2
2  from matplotlib import pyplot as plt
3  import numpy as np
4  def scatter(mod):
5      if mod == 3:
6          x = [1,2,3,4,5]
7          y = [2,4,6,8,10]
8          x1 = [10,23,22,11,25,22,17,12,15,18]
9          y1 = [23,22,25,24,21,22,26,29,11,16]
10         x2 = [1,2,3,4,5,6,7,8,9,10]
11         y2 = [11,12,13,14,15,16,17,18,19,20]
12         plt.subplot(221)
13         plt.scatter(x,y)
14         plt.subplot(222)
15         plt.scatter(x1,y1)
16         plt.subplot(223)
17         plt.scatter(x2,y2)
18         plt.show()

```

```

19 elif mod == 2:
20     x = np.random.randint(0, 10, 10)
21     y = np.random.randint(0, 10, 10)
22     x1 = np.random.randint(10, 20, 10)
23     y1 = np.random.randint(10, 20, 10)
24     plt.subplot(211)
25     plt.scatter(x,y)
26     plt.subplot(212)
27     plt.scatter(x1,y1)
28     plt.show()
29 elif mod == 1:
30     x = np.random.randint(0, 10, 10)
31     y = np.random.randint(0, 10, 10)
32     x1 = np.random.randint(10, 20, 10)
33     y1 = np.random.randint(10, 20, 10)
34     plt.subplot(111)
35     plt.scatter(x,y)
36     plt.show()
37 else:
38     print("Data tidak Valid")

```

Berikut ini cara pemanggilannya:

```

1 ##### Pemanggilan Plot Scatter
2 import a1174074.scatter as scatter
3
4 npm = input("Masukkan NPM Anda: ")
5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     scatter.scatter(3)
10 elif modnpm == 2:
11     scatter.scatter(2)
12 elif modnpm == 1:
13     scatter.scatter(1)
14 else:
15     print("NPM Salah")

```

**1.2.1.3 Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore pie.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2** Ini adalah fungsi untuk membuat plot pie sesuai dengan hasil modulus:

```

1 #No 3
2 from matplotlib import pyplot as plt
3
4 def pie(mod):
5     if mod == 3:
6         Jobs = [8,1,14,2]
7         movie = [10,18,4]
8         sport = [7,5,6,1]
9         Profesi = ['Programmer', 'Atlit', 'Gamer', 'Youtuber']
10        Film = ['Avengers', 'John Wick', 'Shazam']
11        Olahraga = ['Basket', 'Futsal', 'Voli', 'Baseball']
12        cols = ['m', 'r', 'c', 'b']
13        plt.subplot(221)

```

```

14     plt.pie(Jobs,
15             labels=Profesi,
16             colors=cols,
17             startangle=0,
18             shadow=True,
19             explode=(0.2,0,0,0),
20             autopct='%1.1f%%')
21     plt.title('Pie Profesi')
22
23     plt.subplot(222)
24     plt.pie(movie,
25             labels=Film,
26             colors=cols,
27             startangle=90,
28             shadow=True,
29             explode=(.3,0.1,0),
30             autopct='%1.1f%%')
31     plt.title('Pie Movie')
32
33     plt.subplot(223)
34     plt.pie(sport,
35             labels=Olahraga,
36             colors=cols,
37             startangle=90,
38             shadow=True,
39             explode=(.1,0,0,0),
40             autopct='%1.1f%%')
41     plt.title('Pie Olahraga')
42     plt.show()
43 else:
44     print("Data belum dimasukkan")

```

Berikut ini cara pemanggilannya:

```

1  """ Pemanggilan Plot Pie
2  import all74074_pie as pie
3
4  npm = input("Masukkan NPM Anda: ")
5  modnpm = int(npm)%3+2
6  print(modnpm)
7
8  if modnpm == 3:
9      pie.pie(3)
10 elif modnpm == 2:
11     pie.pie(2)
12 elif modnpm == 1:
13     pie.pie(1)
14 else:
15     print("NPM Salah")

```

**1.2.1.4 Buatlah library fungsi (file terpisah/library dengan nama NPM/textunderscore pie.py) untuk plot dengan jumlah subplot adalah  $NPM \bmod 3 + 2$**  Ini adalah fungsi untuk membuat subplot sesuai dengan hasil modulus:

```

1  # No 4
2  from matplotlib import pyplot as plt

```

```

3 import numpy as np
4 def plot(mod):
5     if mod == 3:
6         x = [1,2,3,4,5]
7         y = [2,4,6,8,10]
8         x1 = [10,23,22,11,25,22,17,12,15,18]
9         y1 = [23,22,25,24,21,22,26,29,11,16]
10        x2 = [1,2,3,4,5,6,7,8,9,10]
11        y2 = [11,12,13,14,15,16,17,18,19,20]
12        plt.subplot(221)
13        plt.plot(x,y)
14        plt.subplot(222)
15        plt.plot(x1,y1)
16        plt.subplot(223)
17        plt.plot(x2,y2)
18        plt.show()
19    elif mod == 2:
20        x = np.random.randint(0, 10, 10)
21        y = np.random.randint(0, 10, 10)
22        x1 = np.random.randint(10, 20, 10)
23        y1 = np.random.randint(10, 20, 10)
24        plt.subplot(211)
25        plt.plot(x,y)
26        plt.subplot(212)
27        plt.plot(x1,y1)
28        plt.show()
29    elif mod == 1:
30        x = np.random.randint(0, 10, 10)
31        y = np.random.randint(0, 10, 10)
32        x1 = np.random.randint(10, 20, 10)
33        y1 = np.random.randint(10, 20, 10)
34        plt.subplot(111)
35        plt.plot(x,y)
36        plt.show()
37    else:
38        print("Data tidak Valid")

```

Berikut ini cara pemanggilannya:

```

1 #%% Pemanggilan Plot Error
2 import all174074_plot as plot
3
4 npm = input("Masukkan NPM Anda: ")
5 modnpm = int(npm)%3+2
6 print(modnpm)
7
8 if modnpm == 3:
9     plot.plot(3)
10 elif modnpm == 2:
11     plot.plot(2)
12 elif modnpm == 1:
13     plot.plot(1)
14 else:
15     print("NPM Salah")

```

```

1 def PenangananError():

```



```

2     try :
3         from matplotlib import pyplot as plt
4     except SyntaxError:
5         print("Ada kesalahan dalam penulisan Code")
6     except NameError:
7         print("Variable yang dimasukkan tidak ada")
8     except TypeError:
9         print("Ada yang salah pada type data")
10    except:
11        print("Terjadi kesalahan")
12
13    PenangananError()

```

## 1.3 Advent Nopele Olansi Damiahan Sihite / 1174089)

### 1.3.1 Teori

#### 1.3.1.1 Soal No. 1

Apa itu fungsi library matplotlib?

Matplotlib merupakan salah satu library Python 2D yang dapat menghasilkan plot dengan kualitas yang tinggi dalam berbagai format dan dapat digunakan di berbagai platform. Matplotlib berfungsi sebagai pembuat grafik di berbagai platform, seperti Python dan Jupyter. Grafik yang dibuat menggunakan Matplotlib bisa dibuat dalam berbagai bentuk, seperti grafik garis, batang, lingkaran, histogram, dan sebagainya.

#### 1.3.1.2 Soal No. 2

Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib!

1. Pertama import library Matplotlib.

```
1 from matplotlib import pyplot as plt
```

2. Buat variabel x yang menampung list untuk sumbu x dan variabel y yang menampung list untuk sumbu y.

```
1 x=[1,2,3]
2 y=[5,2,4]
```

3. Panggil fungsi plot dan isi parameter pertama dengan variabel x dan parameter kedua dengan variabel y.

```
1 plt.plot(x,y)
```

4. Lalu panggil plot tadi dengan memanggil fungsi show.

```
1 plt.show()
```

## Kode Program

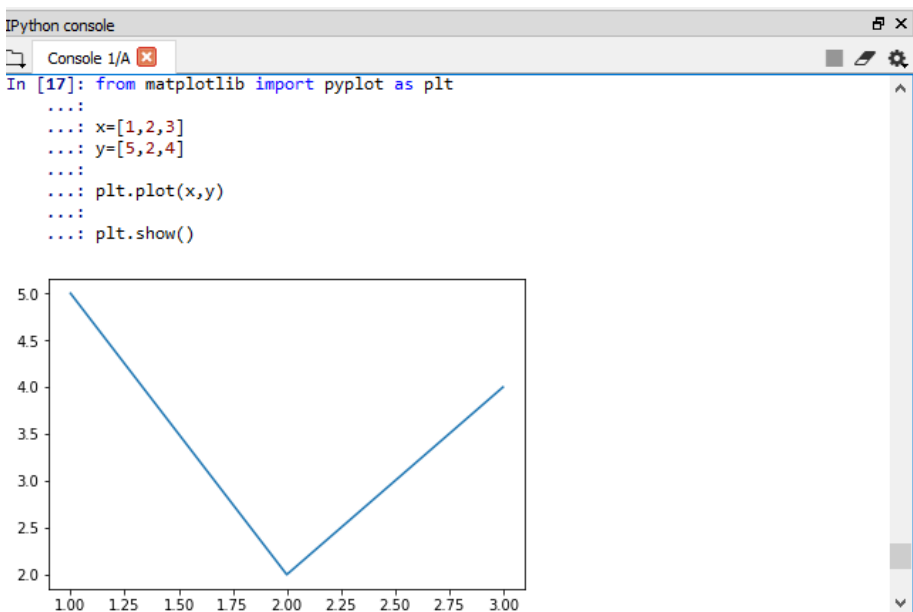
```

1 from matplotlib import pyplot as plt
2
3 x=[1,2,3]
4 y=[5,2,4]
5
6 plt.plot(x,y)
7
8 plt.show()

```

**Listing 1.1** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.1** Hasil compile membuat diagram menggunakan Matplotlib.

### 1.3.1.3 Soal No. 3

Jelaskan bagaimana perbedaan fungsi dan cara pakai untuk berbagai jenis (bar, histogram, scatter, line, dll) jenis plot di matplotlib!

#### 1. Bar Graph

Perbedaan bar graph dengan jenis plot yang lain adalah bar graph menggunakan bar atau batang untuk membandingkan data di antara berbagai kategori.

#### Kode Program

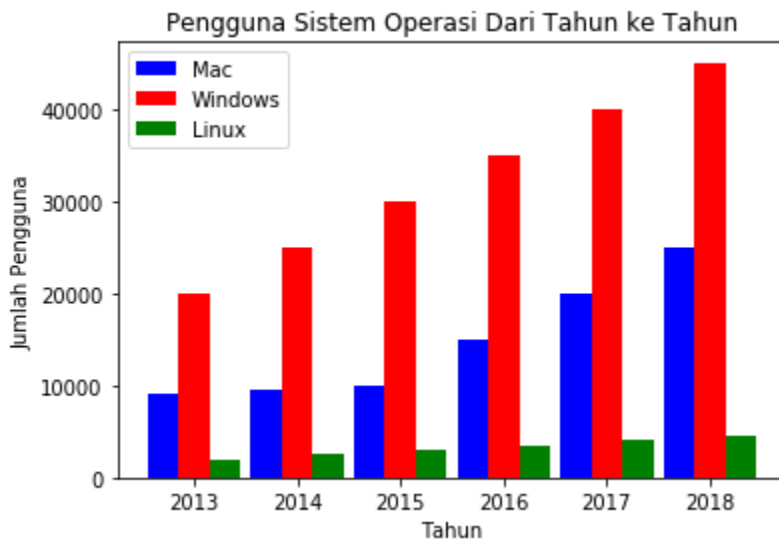
```

1 from matplotlib import pyplot as plt
2
3 plt.bar
4     ([2012.7,2013.7,2014.7,2015.7,2016.7,2017.7],[9000,9500,10000,15000,20000,25000])
5
6 label="Mac",color='b',width=.3)
7 plt.bar
8     ([2013,2014,2015,2016,2017,2018],[20000,25000,30000,35000,40000,45000])
9
10 label="Windows",color='r',width=.3)
11 plt.bar
12     ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[2000,2500,3000,3500,4000,4500])
13
14 label="Linux",color='g',width=.3)
15 plt.legend()
16 plt.xlabel('Tahun')
17 plt.ylabel('Jumlah Pengguna')
18 plt.title('Pengguna Sistem Operasi Dari Tahun ke Tahun')
19 plt.show()

```

**Listing 1.2** Kode program membuat bar graph menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.2** Hasil compile membuat bar graph menggunakan Matplotlib.

## 2. Histogram

Perbedaan histogram dengan jenis plot yang lain adalah histogram akan membuat plot dimana plot yang dimunculkan merupakan gabungan dari beberapa data yang telah dikelompokkan.

## Kode Program

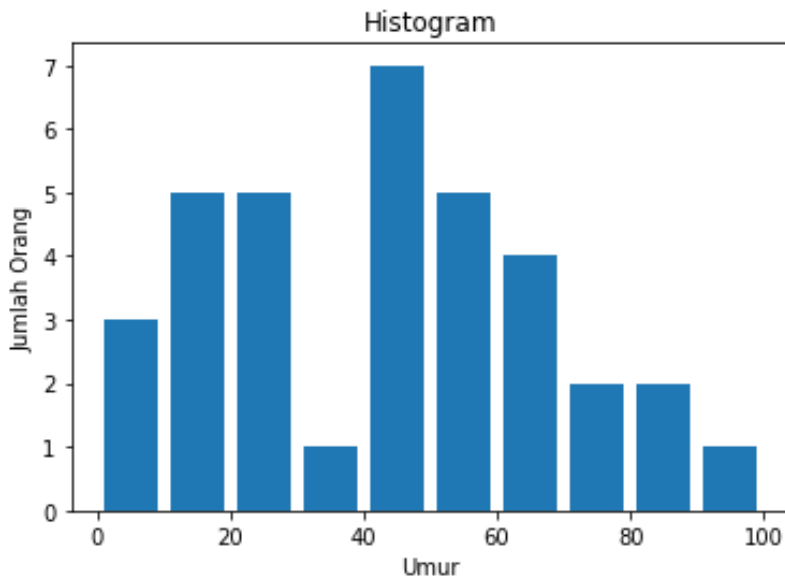
```

1 import matplotlib.pyplot as plt
2 orang =
   [11,22,16,9,10,15,22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,7]
3 umur = [0,10,20,30,40,50,60,70,80,90,100]
4 plt.hist(orang, umur, histtype='bar', rwidth=0.8)
5 plt.xlabel('Umur')
6 plt.ylabel('Jumlah Orang')
7 plt.title('Histogram')
8 plt.show()

```

**Listing 1.3** Kode program membuat histogram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.3** Hasil compile membuat histogram menggunakan Matplotlib.

## 3. Scatter Plot

Perbedaan scatter plot dengan jenis plot lain adalah scatter plot menampilkan data sebagai kumpulan titik, masing-masing memiliki nilai satu variabel yang menentukan posisi pada sumbu horizontal dan nilai variabel lain menentukan posisi pada sumbu vertikal.

### Kode Program

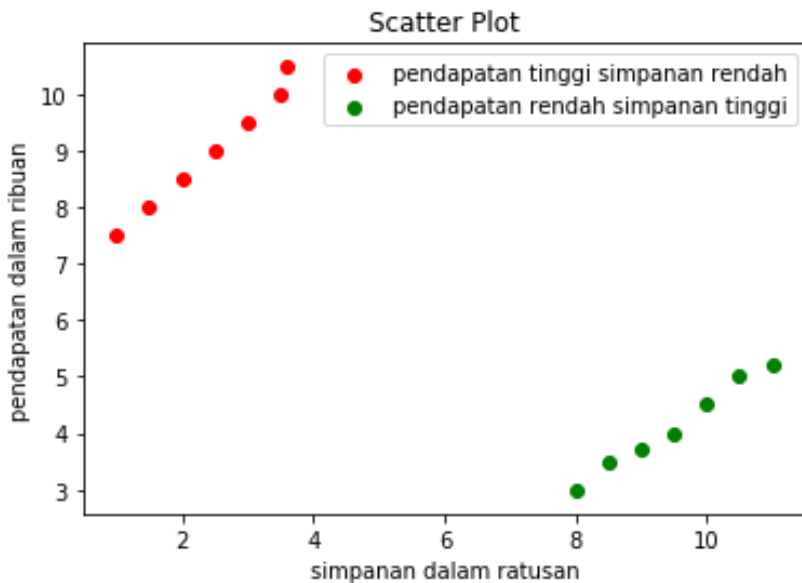
```

1 import matplotlib.pyplot as plt
2 x = [1,1.5,2,2.5,3,3.5,3.6]
3 y = [7.5,8,8.5,9,9.5,10,10.5]
4
5 x1=[8,8.5,9,9.5,10,10.5,11]
6 y1=[3,3.5,3.7,4,4.5,5,5.2]
7
8 plt.scatter(x,y, label='pendapatan tinggi simpanan rendah',color=
    'r')
9 plt.scatter(x1,y1, label='pendapatan rendah simpanan tinggi',color
    ='g')
10 plt.xlabel('simpanan dalam ratusan')
11 plt.ylabel('pendapatan dalam ribuan')
12 plt.title('Scatter Plot')
13 plt.legend()
14 plt.show()

```

**Listing 1.4** Kode program membuat scatter plot menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.4** Hasil compile membuat scatter plot menggunakan Matplotlib.

## 4. Area Plot

Perbedaan area plot dengan jenis plot lain adalah area plot digunakan untuk melacak perubahan dari waktu ke waktu untuk dua atau lebih kelompok terkait yang membentuk satu kategori secara keseluruhan.

## Kode Program

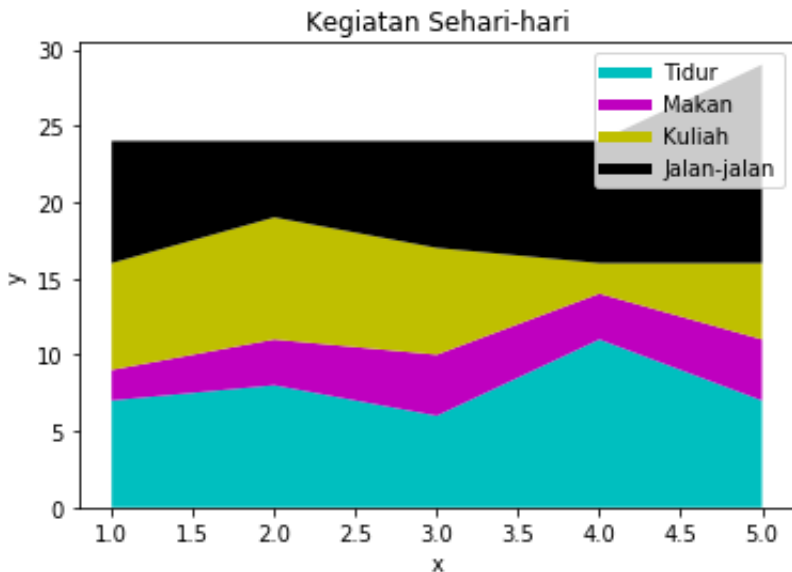
```

1 import matplotlib.pyplot as plt
2 hari = [1,2,3,4,5]
3
4 tidur =[7,8,6,11,7]
5 makan = [2,3,4,3,4]
6 kuliah =[7,8,7,2,5]
7 jalan_jalan = [8,5,7,8,13]
8
9 plt.plot([],[],color='c', label='Tidur', linewidth=5)
10 plt.plot([],[],color='m', label='Makan', linewidth=5)
11 plt.plot([],[],color='y', label='Kuliah', linewidth=5)
12 plt.plot([],[],color='k', label='Jalan-jalan', linewidth=5)
13
14 plt.stackplot(hari,tidur ,makan,kuliah,jalan_jalan , colors=[ 'c','m
    ', 'y','k' ])
15
16 plt.xlabel('x')
17 plt.ylabel('y')
18 plt.title('Kegiatan Sehari-hari')
19 plt.legend()
20 plt.show()

```

**Listing 1.5** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.5** Hasil compile membuat diagram menggunakan Matplotlib.

## 5. Pie Plot

Perbedaan pie plot dengan jenis plot lain adalah pie plot digunakan untuk menunjukkan persentase atau data proporsional di mana setiap potongan pie mewakili kategori.

### Kode Program

```
1 import matplotlib.pyplot as plt
2
3 potong = [7,2,2,12]
4 kegiatan = ['Tidur', 'Makan', 'Kuliah', 'Jalan-jalan']
5 kolom = ['c', 'm', 'y', 'g']
6
7 plt.pie(potong,
8         labels=kegiatan,
9         colors=kolom,
10        startangle=90,
11        shadow=True,
12        explode=(0.1,0,0,0),
13        autopct='%1.1f%%')
14
15 plt.title('Kegiatan Sehari-hari')
16 plt.show()
```

**Listing 1.6** Kode program membuat Pie Plot menggunakan Matplotlib.

### Hasil Compile



In [19]:

**Gambar 1.6** Hasil compile membuat Pie Plot menggunakan Matplotlib.

## 6. Line Graph

Perbedaan line graph dengan jenis plot lain adalah line graph menampilkan diagram dalam bentuk garis.

### Kode Program

```
1 from matplotlib import pyplot as plt
2
```

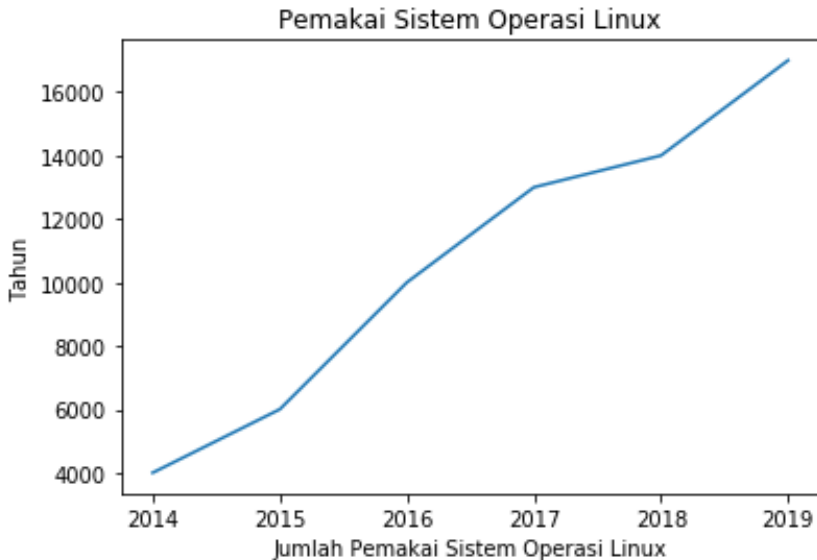
```

3 y = [4000,6000,10000,13000,14000,17000]
4 x = [2014,2015,2016,2017,2018,2019]
5 plt.plot(x,y)
6 plt.title('Pemakai Sistem Operasi Linux')
7 plt.ylabel('Tahun')
8 plt.xlabel('Jumlah Pemakai Sistem Operasi Linux')
9 plt.show()

```

**Listing 1.7** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.7** Hasil compile membuat diagram menggunakan Matplotlib.

### 1.3.1.4 Soal No. 4

Jelaskan bagaimana cara menggunakan legend dan label serta kaitannya dengan fungsi tersebut!

1. Untuk menggunakan legend definisikan parameter label di tiap fungsi plot. Parameter label digunakan untuk memberikan label pada line sebagai pembeda antar line.

```

1 plt.plot(x,y,'b',label='Team Captain America', linewidth=1)
2 plt.plot(x2,y2,'r',label='Team Iron Man',linewidth=1)

```

**Listing 1.8** Kode program menggunakan parameter label dengan Matplotlib.

2. Kemudian panggil fungsi legend.



```
1 plt.legend()
```

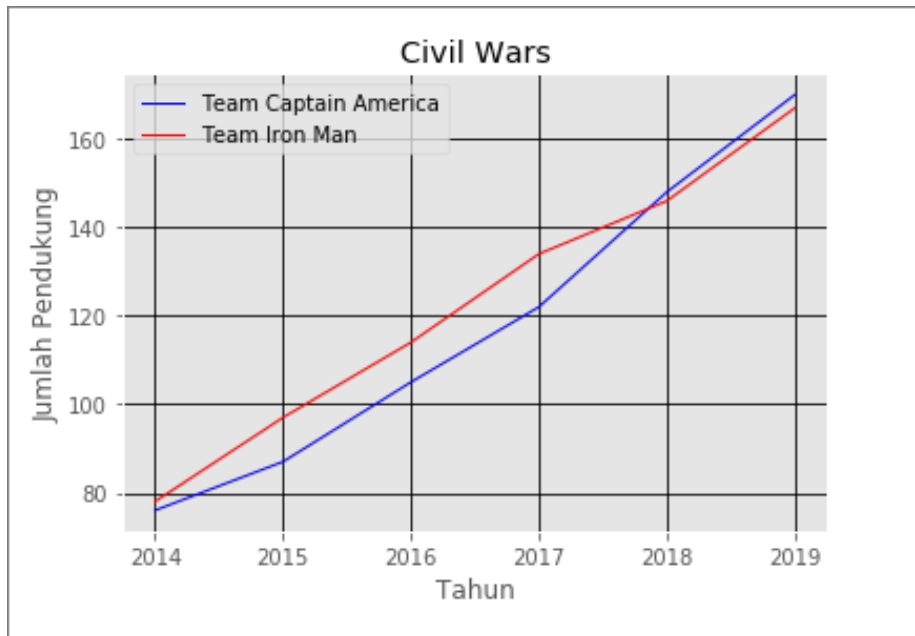
**Listing 1.9** Kode program memanggil fungsi legend dengan Matplotlib.

## Kode Program

```
1 from matplotlib import pyplot as plt
2
3 x = [2014,2015,2016,2017,2018,2019]
4 y = [76,87,105,122,148,170]
5 x2 = [2014,2015,2016,2017,2018,2019]
6 y2 = [78,97,114,134,146,167]
7 plt.plot(x,y,'b',label='Team Captain America', linewidth=1)
8 plt.plot(x2,y2,'r',label='Team Iron Man',linewidth=1)
9 plt.title('Civil Wars')
10 plt.ylabel('Jumlah Pendukung')
11 plt.xlabel('Tahun')
12 plt.legend()
13 plt.grid(True,color='k')
14 plt.show()
```

**Listing 1.10** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.8** Hasil compile membuat diagram menggunakan Matplotlib.

### 1.3.1.5 Soal No. 5

Jelaskan apa fungsi dari subplot di matplotlib, dan bagaimana cara kerja dari fungsi subplot, sertakan ilustrasi dan gambar sendiri dan apa parameternya jika ingin menggambar plot dengan 9 subplot di dalamnya!

Fungsi subplot adalah untuk membuat beberapa plot di dalam satu gambar.

Cara kerja subplot, yaitu fungsi subplot memiliki parameter pertama adalah jumlah kolom, parameter kedua adalah jumlah baris, dan parameter ketiga adalah index plot keberapanya.

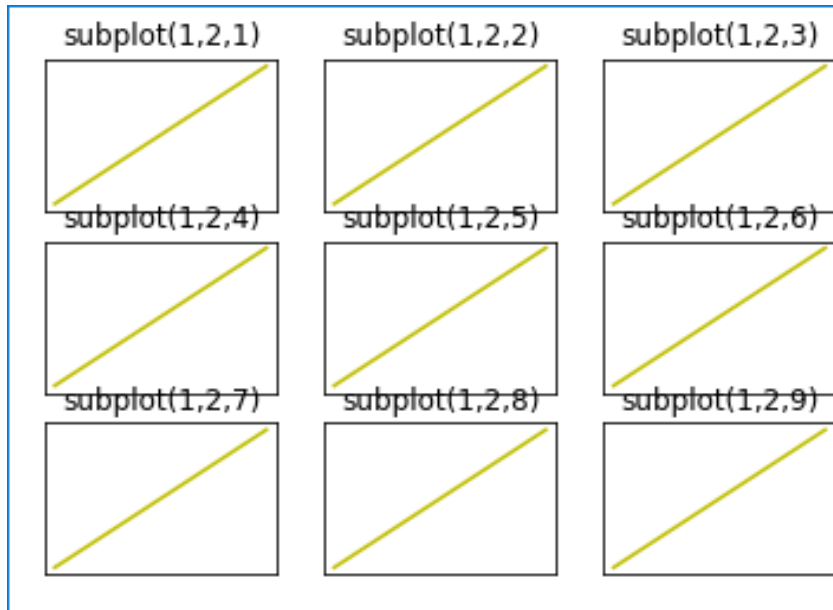
### Kode Program

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t = np.arange(0.0, 9.0, 1)
5 s = [1,2,3,4,5,6,7,8,9]
6
7 for i in range(1, 10):
8     plt.subplot(3,3,i)
9     plt.xticks([], plt.yticks([]))
10    plt.title('subplot(1,2,'+str(i)+'')')
11    plt.plot(t,s,'-y')
12
13 plt.show()
```

**Listing 1.11** Kode program membuat subplot menggunakan Matplotlib.

### Hasil Compile



**Gambar 1.9** Hasil compile membuat subplot menggunakan Matplotlib.

#### 1.3.1.6 Soal No. 6

Sebutkan semua parameter color yang bisa digunakan (contoh: m,c,r,k,... dkk)!

- 'b' (blue)
- 'g' (green)
- 'r' (red)
- 'c' (cyan)
- 'm' (magenta)
- 'y' (yellow)
- 'k' (black)
- 'w' (white)

#### 1.3.1.7 Soal No. 7

Jelaskan bagaimana cara kerja dari fungsi hist, sertakan ilustrasi dan gambar sendiri!

Cara kerja dari fungsi hist yaitu fungsi hist akan menerima parameter yang diberikan, kemudian fungsi hist akan dieksekusi sesuai dengan parameter yang diberikan.

#### Kode Program

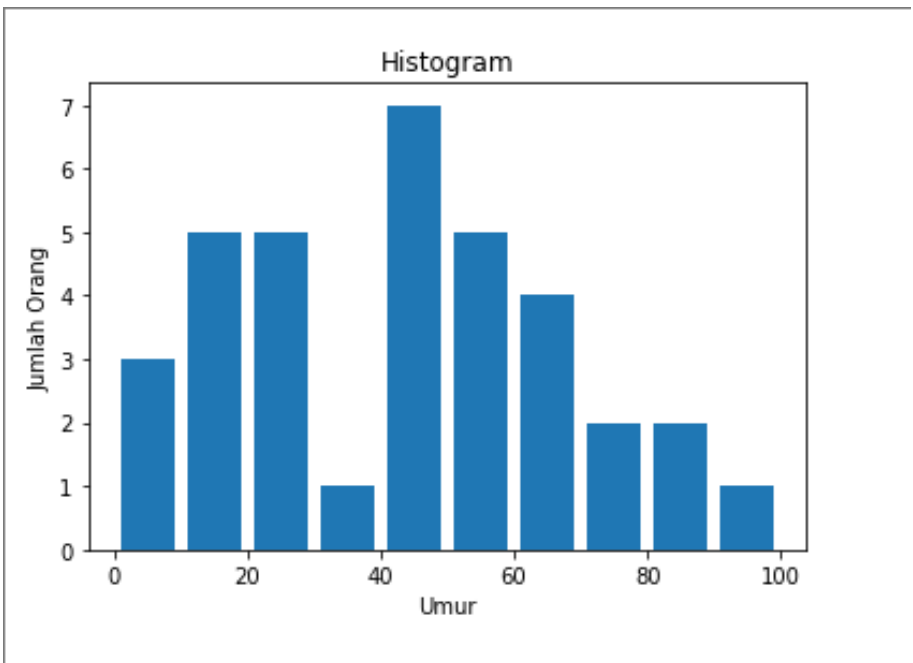
```

1 import matplotlib.pyplot as plt
2 orang =
   [11,22,16,9,10,15,22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,62,
3 umur = [0,10,20,30,40,50,60,70,80,90,100]
4 plt.hist(orang, umur, histtype='bar', rwidth=0.8)
5 plt.xlabel('Umur')
6 plt.ylabel('Jumlah Orang')
7 plt.title('Histogram')
8 plt.show()

```

**Listing 1.12** Kode program membuat diagram menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.10** Hasil compile membuat diagram menggunakan Matplotlib.

### 1.3.1.8 Soal No. 8

Jelaskan lebih mendalam tentang parameter dari fungsi pie diantaranya labels, colors, startangle, shadow, explode, autopct!

- labels : untuk memberikan label di tiap persentase.
- colors : untuk memberikan warna di tiap persentase.
- startangle : untuk memutar plot sesuai dengan derajat yang ditentukan.

- shadow : untuk memberikan bayangan pada plot.
- explode : untuk memisahkan antar tiap potongan pie pada plot.
- autopct : untuk menentukan jumlah angka dibelakang koma.

## 1.3.2 Praktek

### 1.3.2.1 Soal No. 1

Buatlah librari fungsi (file terpisah/library dengan nama NPMbar.py) untuk plot dengan jumlah subplot adalah NPM mod 3 + 2!

### Kode Program

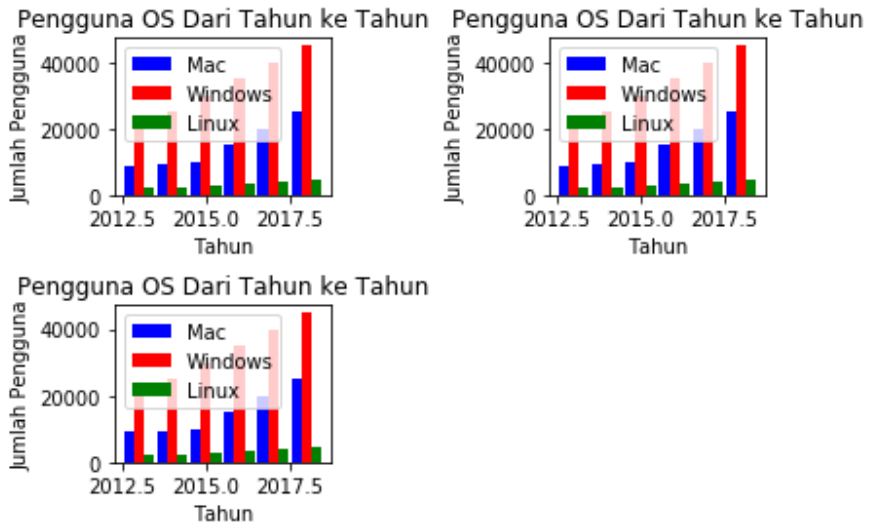
```

1 from matplotlib import pyplot as plt
2
3 def bar():
4
5     hasil = 1174089 % 3 + 2
6
7     for i in range(1, hasil+1):
8         plt.subplot(2,2,i)
9         plt.bar
10        ([2012.7,2013.7,2014.7,2015.7,2016.7,2017.7],[9000,9500,10000,15000,20000,25000],
11         label="Mac",color='b',width=.3)
12        plt.bar
13        ([2013,2014,2015,2016,2017,2018],[20000,25000,30000,35000,40000,45000],
14         label="Windows",color='r',width=.3)
15        plt.bar
16        ([2013.3,2014.3,2015.3,2016.3,2017.3,2018.3],[2000,2500,3000,3500,4000,4500],
17         label="Linux",color='g',width=.3)
18        plt.legend()
19        plt.xlabel('Tahun')
20        plt.ylabel('Jumlah Pengguna')
21        plt.title('Pengguna OS Dari Tahun ke Tahun')
22        plt.subplots_adjust(wspace=1, hspace=.7)
23
24    plt.show()

```

**Listing 1.13** Kode program membuat fungsi Bar Plot menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.11** Hasil compile membuat fungsi Bar Plot menggunakan Matplotlib.

### 1.3.2.2 Soal No. 2

Buatlah library fungsi (file terpisah/library dengan nama NPMscatter.py) untuk plot dengan jumlah subplot NPM mod 3 + 2!

#### Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def scatter():
4
5     hasil = 1174089 % 3 + 2
6
7     x = [1,1.5,2,2.5,3,3.5,3.6]
8     y = [7.5,8,8.5,9,9.5,10,10.5]
9
10    x1=[8,8.5,9,9.5,10,10.5,11]
11    y1=[3,3.5,3.7,4,4.5,5,5.2]
12
13    for i in range(1, hasil+1):
14        plt.subplot(2,2,i)
15        plt.scatter(x,y, label='pendapatan tinggi simpanan rendah',
16            color='r')
17        plt.scatter(x1,y1, label='pendapatan rendah simpanan tinggi',
18            color='g')
19        plt.xlabel('simpanan dalam ratusan')
20        plt.ylabel('pendapatan dalam ribuan')
21        plt.title('Scatter Plot')
22        plt.legend()
23        plt.subplots_adjust(wspace=1.1, hspace=.7)

```

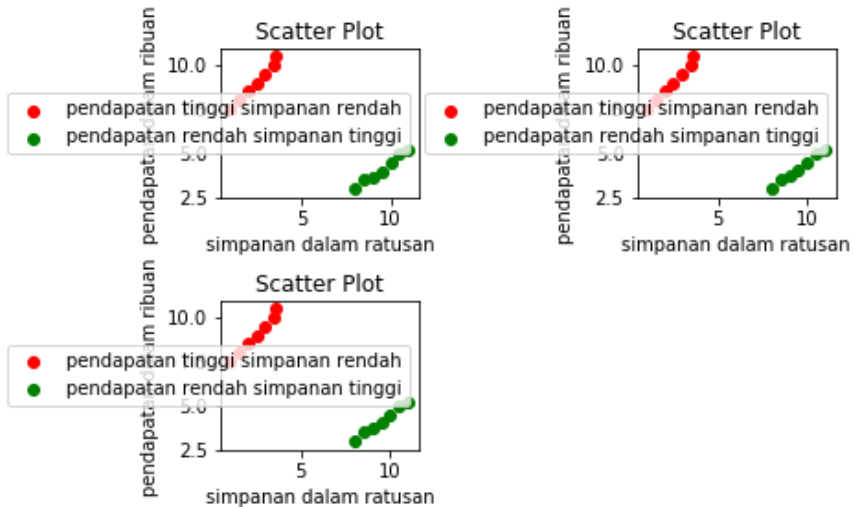
```

22
23 plt.show()

```

**Listing 1.14** Kode program membuat fungsi Scatter Plot menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.12** Hasil compile membuat fungsi Scatter Plot menggunakan Matplotlib.

### 1.3.2.3 Soal No. 3

Buatlah librari fungsi (file terpisah/library dengan nama NPMpie.py) untuk plot dengan jumlah subplot NPM mod 3 + 2!

## Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def pie():
4
5     hasil = 1174089 % 3 + 2
6
7     potong = [7,4,2,12]
8     kegiatan = ['Tidur', 'Makan', 'Kuliah', 'Jalan-jalan']
9     kolom = ['c', 'm', 'y', 'g']
10
11     for i in range(1, hasil+1):
12         plt.subplot(2,2,i)
13         plt.pie(potong,
14               labels=kegiatan,
15               colors=kolom,
16               startangle=90,

```

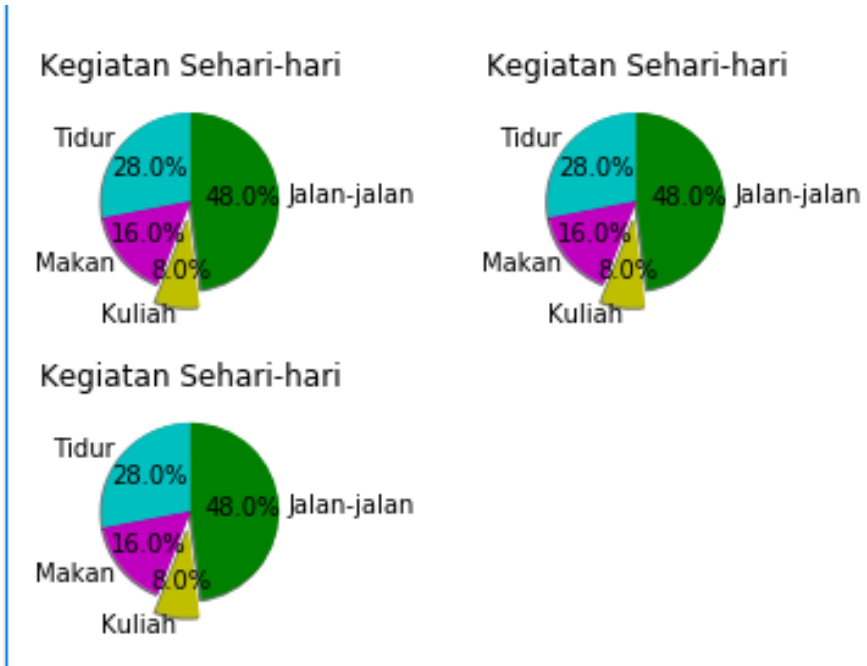
```

17     shadow= True ,
18     explode=(0.1,0,0,0) ,
19     autopct='%1.1f%%')
20     plt.title('Kegiatan Sehari-hari')
21     plt.subplots_adjust(hspace=.4)
22
23 plt.show()

```

**Listing 1.15** Kode program membuat fungsi Pie Plot menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.13** Hasil compile membuat fungsi Pie Plot menggunakan Matplotlib.

### 1.3.2.4 Soal No. 4

Buatlah librari fungsi (file terpisah/library dengan nama NPMplot.py) untuk plot dengan jumlah subplot NPM mod 3 + 2

## Kode Program

```

1 from matplotlib import pyplot as plt
2
3 def plot():
4
5     hasil = 1174089 % 3 + 2
6

```



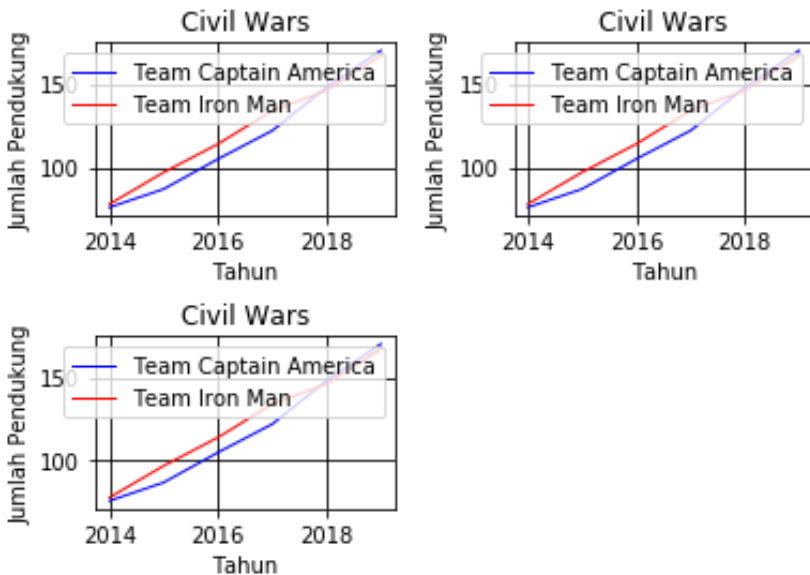
```

7  x = [2014,2015,2016,2017,2018,2019]
8  y = [76,87,105,122,148,170]
9  x2 = [2014,2015,2016,2017,2018,2019]
10 y2 = [78,97,114,134,146,167]
11
12 for i in range(1, hasil+1):
13     plt.subplot(2,2,i)
14     plt.plot(x,y,'b',label='Team Captain America', linewidth=1)
15     plt.plot(x2,y2,'r',label='Team Iron Man',linewidth=1)
16     plt.title('Civil Wars')
17     plt.ylabel('Jumlah Pendukung')
18     plt.xlabel('Tahun')
19     plt.legend()
20     plt.grid(True, color='k')
21     plt.subplots_adjust(wspace=.4, hspace=.7)
22
23 plt.show()

```

**Listing 1.16** Kode program membuat fungsi Plot menggunakan Matplotlib.

## Hasil Compile



**Gambar 1.14** Hasil compile membuat fungsi Plot menggunakan Matplotlib.

### 1.3.3 Penanganan Error

Tuliskan peringatan error yang didapat dari mengerjakan praktek keenam ini, dan jelaskan cara penanganan error tersebut. dan Buatlah satu fungsi yang menggunakan try except untuk menanggulangi error tersebut.

Peringatan error di praktek kelima ini, yaitu:

- **Syntax Errors** Syntax Errors adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- **Name Error** NameError adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- **Type Error** TypeError adalah exception yang akan terjadi apabila pada saat dilakukannya eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusi dari error ini adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

Fungsi yang menggunakan try except untuk menanggulangi error.

#### Kode Program

```

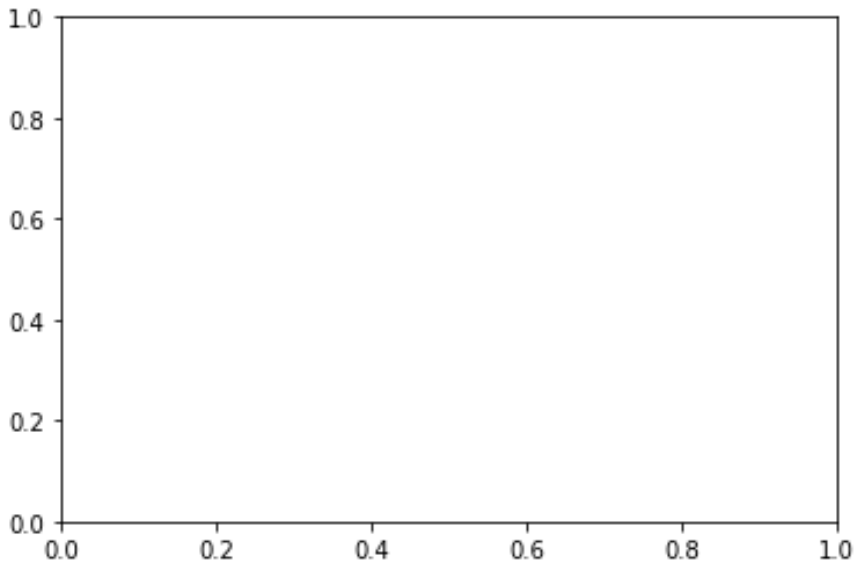
1 from matplotlib import pyplot as plt
2
3 def tryExceptError():
4     try:
5         a=[1,2,3]
6         y=[5,2,4]
7         plt.plot(x,y)
8         plt.show()
9     except SyntaxError:
10        print("Kesalahan penulisan syntax")
11    except NameError:
12        print("Variable tersebut tidak ada")
13    except TypeError:
14        print("Tipe data salah")
15    except:
16        print("Terjadi sebuah kesalahan")
17
18 tryExceptError()

```

**Listing 1.17** Kode program membuat fungsi penanganan error.

#### Hasil Compile

Terjadi sebuah kesalahan



**Gambar 1.15** Hasil compile membuat fungsi penanganan error.

### 1.3.4 Screenshoot Plagiat

RESULTS

100%

Completed: 100% Checked

0%

Plagiarism

100%

Unique

📄

Sentence Wise Result

📄

Matched Sources

📄

Document View

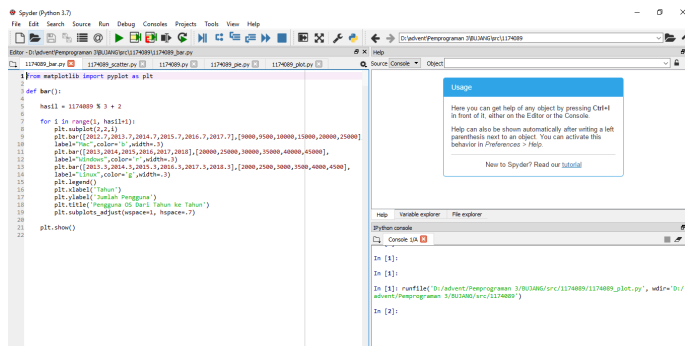
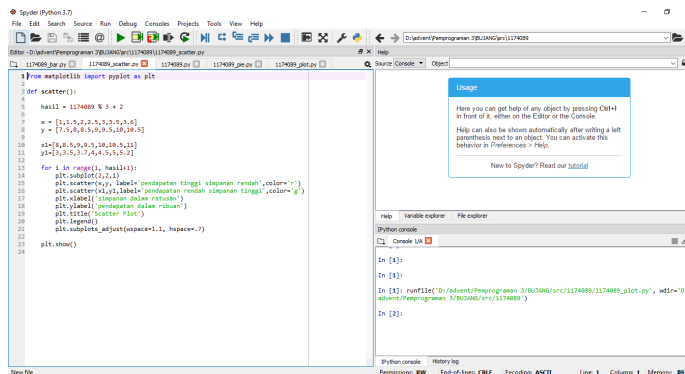
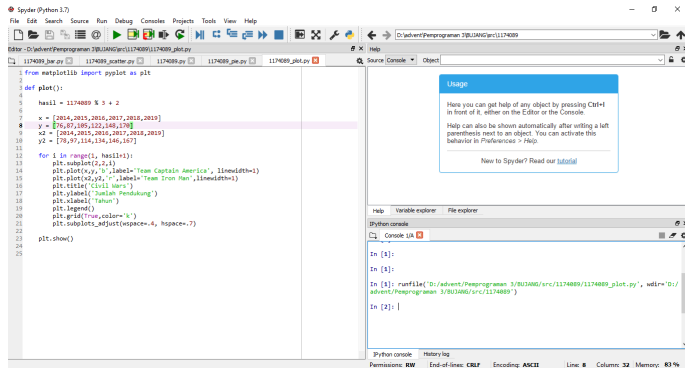
UNIQUE	Matplotlib merupakan salah satu library Python 2D yang dapat menghasilkan plot ...
UNIQUE	Matplotlib berfungsi sebagai pembuat grafik di berbagai platform, seperti Python d...
UNIQUE	Grafik yang dibuat menggunakan Matplotlib bisa dibuat dalam berbagai bentuk, se...
UNIQUE	Jelaskan langkah-langkah membuat sumbu X dan Y di matplotlib!
UNIQUE	Item Buat variabel x yang menampung list untuk sumbu x dan variabel y yang men...
UNIQUE	Item Panggil fungsi plot dan isi parameter pertama dengan variabel x dan paramet...
UNIQUE	Item Lalu panggil plot tadi dengan memanggil fungsi show.

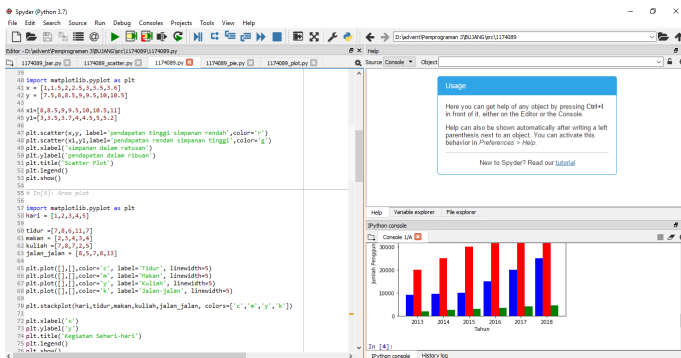
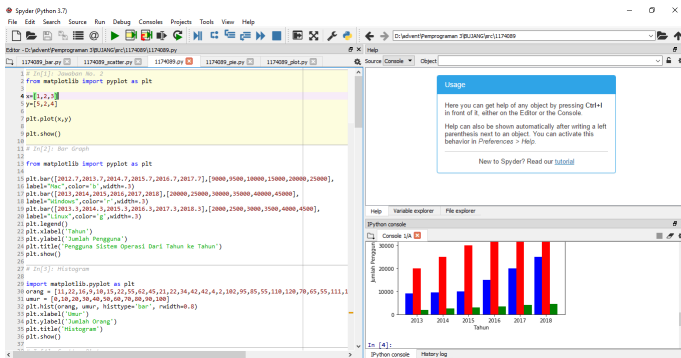
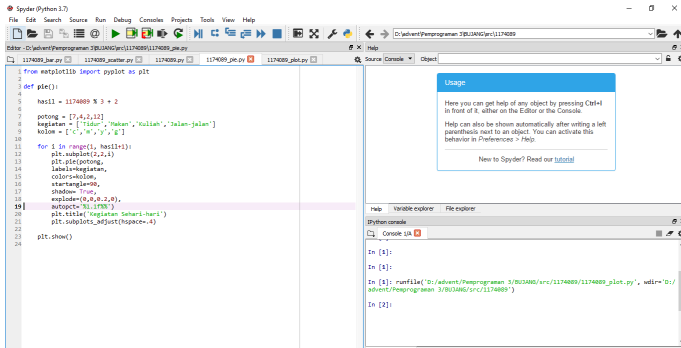
🚫

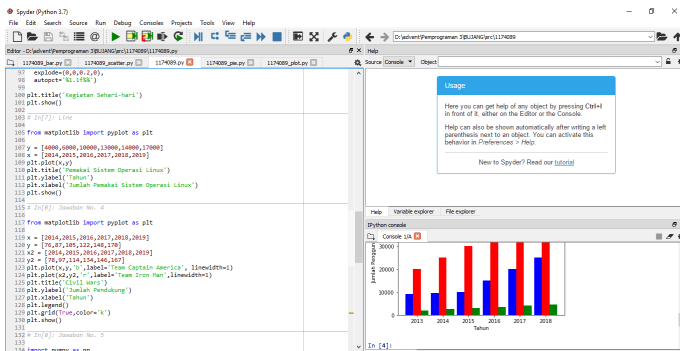
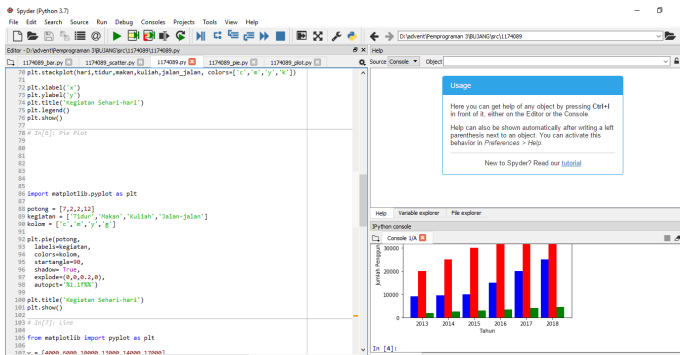
🍏

🤖

### 1.3.5 Screenshot Kode Program







Spyder (Python 3.7)

File Edit Search Source Run Debug Console Projects Tools View Help

Editor: D:\labent\Programman 3BUJAN\py1174089\py1174089.py

```

153 plt.xlabel('umur')
154 plt.ylabel('jumlah orang')
155 plt.title('Histogram')
156 plt.show()
157
158 # Isi[] : Perulangan Error
159
160 from matplotlib import pyplot as plt
161
162 def tryexceptError():
163     try:
164         a=[1,2,3]
165         b=[5,6,7]
166         plt.plot(a,b)
167         plt.show()
168     except SyntaxError:
169         print("kesalahan penulisan syntax")
170     except NameError:
171         print("Variable tersebut tidak ada")
172     except TypeError:
173         print("tipe data salah")
174     except:
175         print("terjadi sebuah kesalahan")
176
177 tryexceptError()
178
179 # Isi[] : Notus
180
181
182 bar = _import_t('1174089_bar.py')
183 scat = _import_t('1174089_scat.py')
184 pie = _import_t('1174089_pie.py')
185 plt = _import_t('1174089_plot.py')
186
187 bar.bar()
188 scat.scatter()
189 pie.pie()
190 plt.plot()

```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Python console

Console (R)

Python console

Variable explorer

File explorer

Console (R)

Python console

History log

# DAFTAR PUSTAKA

---

1. R. Awangga, "Sampeu: Servicing web map tile service over web map service to increase computation performance," in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.





# Index

---

disruptif, xxix  
modern, xxix