PRAKTIKUM 2. GAMBAR 2D, TRASLASI DAN ROTASI

RASTRI PRATHIVI, M.KOM.

SIMPAN DENGAN NAMA GAMBAR2D_NIM.IPYNB

Listing 2-5. Program CIRCLES

```
"""
CIRCLES
"""

import numpy as np
import matplotlib.pyplot as plt

plt.axis([-75,75,50,-50])

plt.axis('on')
plt.grid(True)

plt.arrow(0,0,20,0,head_length=4,head_width=3,color='k')
plt.arrow(0,0,0,20,head_length=4,head_width=3,color='k')

plt.text(16,-3,'x')
plt.text(-5,17,'y')

#————————————————————————————————-green circle
xc=0
yc=0
r=40

p1=0*np.pi/180
p2=360*np.pi/180
dp=(p2-p1)/100
xlast=xc+r*np.cos(p1)
ylast=yc+r*np.sin(p1)
for p in np.arange(p1,p2+dp,dp):
      x=xc+r*np.cos(p)
      y=yc+r*np.sin(p)
      if p > 90*np.pi/180 and p < 270*np.pi/180:
            plt.plot([xlast,x],[ylast,y],color='g',linestyle=':')
      else:
            plt.plot([xlast,x],[ylast,y],color='g')
      xlast=x
      ylast=y

plt.scatter(xc,yc,s=15,color='g')

```

```
41  #————————————————————————————————red circle
42  xc=-20
43  yc=-20
44  r=10
45
46  p1=0*np.pi/180
47  p2=360*np.pi/180
48  dp=(p2-p1)/100
49  xlast=xc+r*np.cos(p1)
50  ylast=yc+r*np.sin(p1)
51  for p in np.arange(p1,p2+dp,dp):
52        x=xc+r*np.cos(p)
53        y=yc+r*np.sin(p)
54        plt.plot([xlast,x],[ylast,y],linewidth=4,color='r')
55        xlast=x
56        ylast=y
57
58  plt.scatter(xc,yc,s=15,color='r')
59
```

```
60  #————————————————————————————————purple circle
61  xc=20
62  yc=20
63  r=50
64
65  p1=0*np.pi/180
66  p2=360*np.pi/180
67  dp=(p2-p1)/100
68  xlast=xc+r*np.cos(p1)
69  ylast=yc+r*np.sin(p1)
70  for p in np.arange(p1,p2+dp,dp):
71        x=xc+r*np.cos(p)
72        y=yc+r*np.sin(p)
73        plt.plot([xlast,x],[ylast,y],linewidth=2,color=(.8,0,.8))
74        xlast=x
75        ylast=y
76
77  plt.scatter(xc,yc,color=(.5,0,.5))
78
79  #————————————————————————————————blue disc
80  xc=-53
81  yc=-30
82  r1=0
83  r2=10
84  dr=1
85
```

```
86  p1=0*np.pi/180
87  p2=360*np.pi/180
88  dp=(p2-p1)/100
89  xlast=xc+r1*np.cos(p1)
90  ylast=yc+r1*np.sin(p1)
91  for r in np.arange(r1,r2,dr):
92        for p in np.arange(p1,p2+dp,dp):
93              x=xc+r*np.cos(p)
94              y=yc+r*np.sin(p)
95              plt.plot([xlast,x],[ylast,y],linewidth=2,color=(0,0,.8))
```

```
96              xlast=x
97              ylast=y
98
99  plt.show()
```

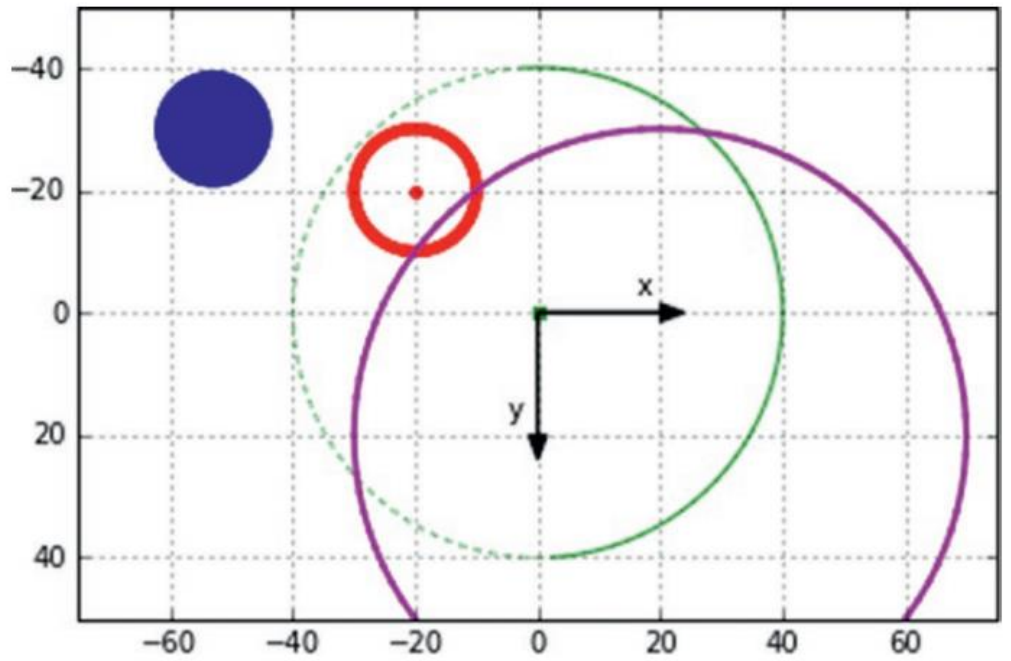

Figure 2-8. *Circles created by Listing 2-5*

Listing 2-7. Program ELLIPSES

```
"""
ELLIPSES
"""

import numpy as np
import matplotlib.pyplot as plt

plt.axis([-75,75,50,-50])

plt.axis('on')
plt.grid(True)

plt.arrow(0,0,60,0,head_length=4,head_width=3,color='k')
plt.arrow(0,0,0,45,head_length=4,head_width=3,color='k')

plt.text(58,-3,'x')
plt.text(-5,44,'y')

#————————————————————————————————red ellipse
a=40
b=20.
p1=0
p2=180*np.pi/180
dp=.2*np.pi/180

xplast=a
yplast=0
for p in np.arange(p1,p2,dp):
        xp=np.abs(a*b*(b*b+a*a*(np.tan(p))**2.)**-.5)
        yp=np.abs(a*b*(a*a+b*b/(np.tan(p)**2.))**-.5)
        if p > np.pi/2:
             xp=-xp
        plt.plot([xplast,xp],[yplast,yp],color='r')
        plt.plot([xplast,xp],[-yplast,-yp],color='r')
        xplast=xp
        yplast=yp

```

```
38 #————————————————————————————green ellipse
39 a=20.
40 b=40.
41 xp1=-a
42 xp2=a
43 dx=.1
44
45 xplast=-a
46 yplast=0
47 for xp in np.arange(xp1,xp2,dx):
48       yp=b*(1-xp**2./a**2.)**.5
49       plt.plot([xplast,xp],[yplast,yp],linewidth=1,color='g')
50       plt.plot([xplast,xp],[-yplast,-yp],linewidth=1,color='g')
51       xplast=xp
52       yplast=yp
53
```

```
54 plt.plot([xplast,a],[yplast,0],linewidth=1,color='g'
55 plt.plot([xplast,a],[-yplast,0],linewidth=1,color='g'
56
57 #————————————————————————————blue ellipse
58 a=5.
59 b=15.
60 p1=0
61 p2=180*np.pi/180
62 dp=.2*np.pi/180
63
64 for p in np.arange(p1,p2,dp):
65       xp=np.abs(a*b*(b*b+a*a*(np.tan(p))**2.)**-.5)
66       yp=np.abs(a*b*(a*a+b*b/(np.tan(p)**2.))**-.5)
67       if p > np.pi/2:
68            xp=-xp
69       plt.plot([xp,xp],[yp,-yp],linewidth=1,color='b')
70
71 plt.show()
```

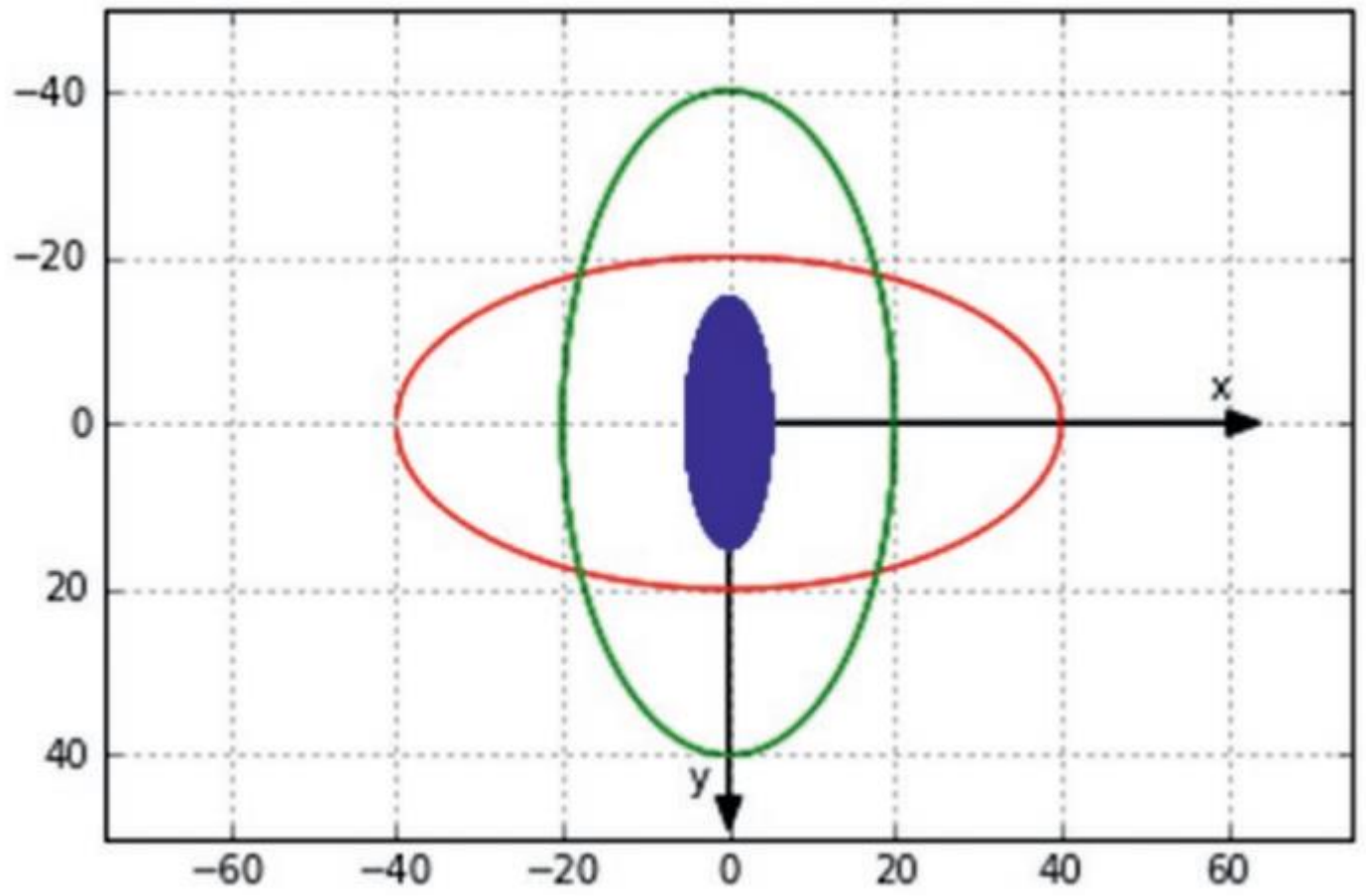

Figure 2-12. *Ellipses created by Listing 2-7*

TRANSLASI

Listing 2-9. Program 2DTRANSLATION

```
"""
2DTRANSLATION
"""

import numpy as np
import matplotlib.pyplot as plt

x1=-10
x2=140
y1=90
y2=-10
plt.axis([x1,x2,y1,y2])

plt.axis('on')
plt.grid(True)

plt.title('Translation')

#------------------------------------------------------triangle
x=[20,30,40,20]
y=[40,20,40,40]
plt.plot(x,y,color='k')
plt.plot(x,y,color='k')
plt.plot(x,y,color='k')

#-------------------------------------------translate triangle dx=60
x=[60,70,80,60]
plt.plot(x,y,color='g')
plt.plot(x,y,color='g')
plt.plot(x,y,color='g')

#-------------------------------------------translate triangle dy=40
y=[80,60,80,80]
plt.plot(x,y,color='r')
plt.plot(x,y,color='r')
plt.plot(x,y,color='r')
```

```
38  #-------------------------------------------------------box
39  x=[0,0,5,5,0]
40  y=[55,50,50,55,55]
41  plt.plot(x,y,'b')
42
43  #-------------------------------------------------translate box
44  y=[55,50,50,55,55]
45  for x in np.arange(0,130,10):
46      x=[x,x,x+5,x+5,x]
47      plt.plot(x,y,'b')
48
49  plt.show()
```

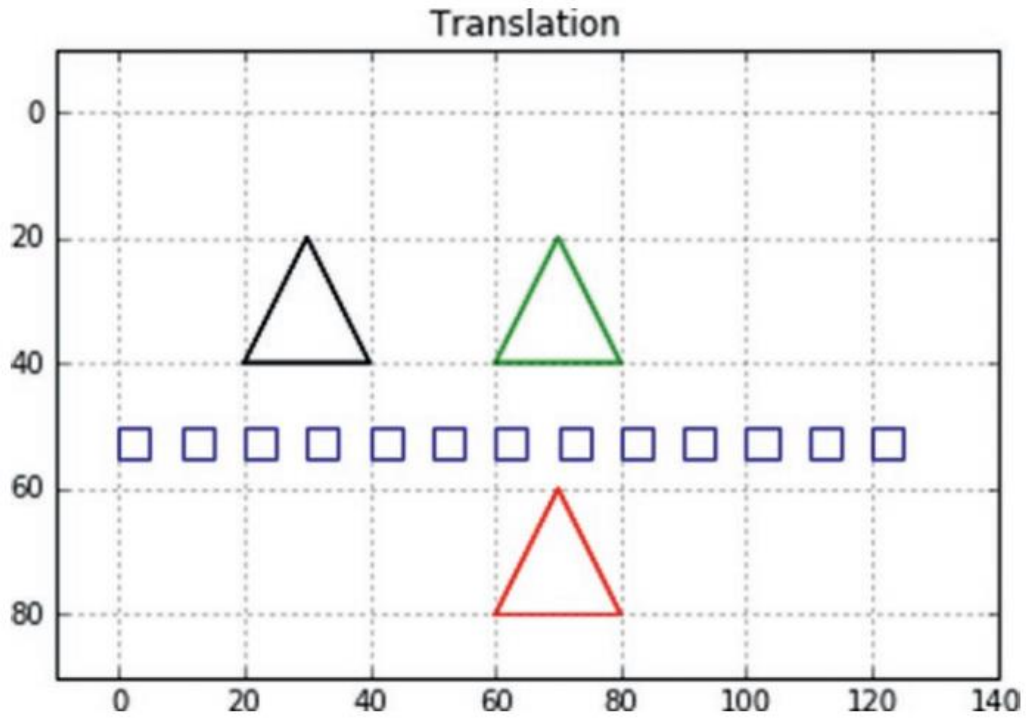

Figure 2-13. *Examples of translation created by Listing 2-9*

ROTASI

Listing 2-11. Program 2DROTRECTANGLE

```
"""
2DROTRECTANGLE
"""

import matplotlib.pyplot as plt
import numpy as np

plt.axis([-10,150,100,-10])
plt.axis('on')
plt.grid(True)

#--------------------------------------------------axes
plt.arrow(0,0,40,0,head_length=4,head_width=2,color='b')
plt.arrow(0,0,0,40,head_length=4,head_width=2,color='b')
plt.text(30,-3,'Xg',color='b')
plt.text(-8,34,'Yg',color='b')

xc=75 #-----------------center of rotation
yc=50
plt.plot([xc-40,xc+60],[yc,yc],linewidth=1,color='grey') #--X
plt.plot([xc,xc],[yc-40,yc+45],linewidth=1,color='grey') #--Y
plt.text(127,48,'X')
plt.text(70,90,'Y')

plt.scatter(xc,yc,s=20,color='k') #-plot center of rotation
plt.text(70,49,'c')

```

```
#————————————————————————————————————————define function rotz
def rotz(xp,yp,rz):
    c11=np.cos(rz)
    c12=-np.sin(rz)
    c21=np.sin(rz)
    c22=np.cos(rz)
    xpp=xp*c11+yp*c12 #————relative to xc,yc
    ypp=xp*c21+yp*c22
    xg=xc+xpp #--relative to xg,yg
    yg=yc+ypp
    return [xg,yg]

#————————————————————————————————————————plot unrotated rectangle
#————————————————rectangle corner coordinates in X,Y system
xp1=-20
xp2=+20
xp3=+20
xp4=-20
yp1=-5
yp2=-5
yp3=+5
yp4=+5

plt.text(50,45,'1') #————-label
plt.text(97,45,'2')
plt.text(97,57,'3')
plt.text(50,57,'4')
plt.scatter(xp1+xc,yp1+yc,s=10,color='k')
plt.scatter(xp2+xc,yp2+yc,s=10,color='k')
plt.scatter(xp3+xc,yp3+yc,s=10,color='k')
plt.scatter(xp4+xc,yp4+yc,s=10,color='k')

#————————————————————————————————————————plot unrotated rectangle
xg1=xc+xp1 #————-corner coordinates in Xg,Yg system
yg1=yc+yp1
xg2=xc+xp2
yg2=yc+yp2
xg3=xc+xp3
yg3=yc+yp3
xg4=xc+xp4
yg4=yc+yp4

```

```
70  xg=[xg1,xg2,xg3,xg4,xg1]
71  yg=[yg1,yg2,yg3,yg4,yg1]
72
73  plt.plot((xg),(yg),color='k')
74
75  #-----------------------rotate rectangle corner coordinates
76  rz=45
77  rz=rz*np.pi/180
78
79  #-----------------------------------point 1
80  xp=xp1
81  yp=yp1
```

```
82  [xg,yg]=rotz(xp,yp,rz)
83  [xg1,yg1]=[xg,yg]
84
85  #-----------------------------------point 2
86  xp=xp2
87  yp=yp2
88  [xg,yg]=rotz(xp,yp,rz)
89  [xg2,yg2]=[xg,yg]
90
91  #-----------------------------------point 3
92  xp=xp3
93  yp=yp3
94  [xg,yg]=rotz(xp,yp,rz)
95  [xg3,yg3]=[xg,yg]
96
```

```
97  #-----------------------------------point 4
98  xp=xp4
99  yp=yp4
100 [xg,yg]=rotz(xp,yp,rz)
101 [xg4,yg4]=[xg,yg]
102
103 #-----------------------------------plot rotated rectangle
104 xg=[xg1,xg2,xg3,xg4,xg1]
105 yg=[yg1,yg2,yg3,yg4,yg1]
106
107 plt.plot(xg,yg,color='r')
108
109 plt.show()
```

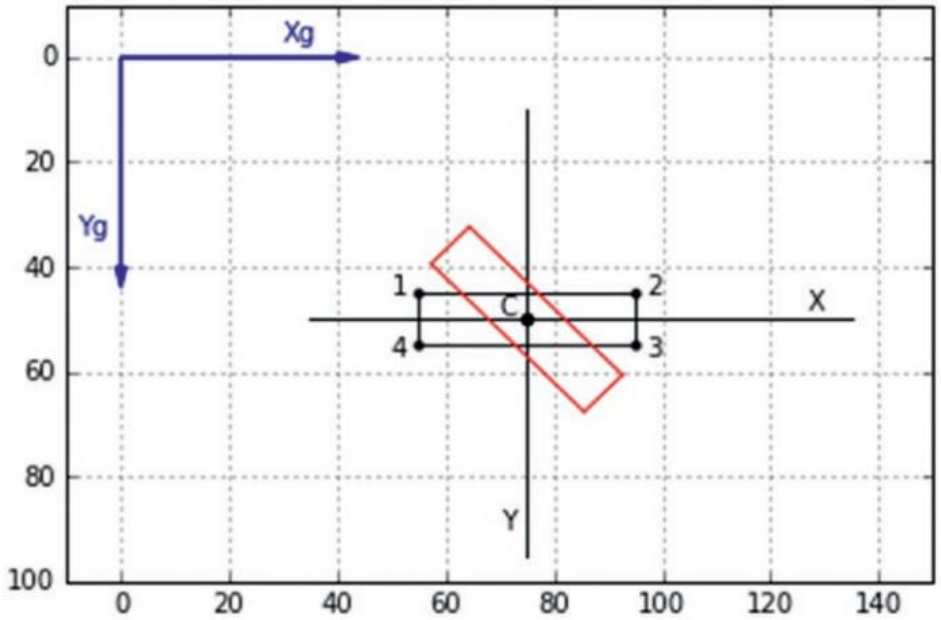

Figure 2-16. *Rotation of a rectangle around its center from Listing 2-11*