In [1]:
```
import cv2
import numpy as np

# load the shape template
image = cv2.imread('gambar/star01.jpeg')
template = cv2.resize(image, (300, 600))

template_gray = cv2.cvtColor(template,cv2.COLOR_BGR2GRAY)
cv2.imshow('Template', template_gray)
cv2.waitKey()
cv2.destroyAllWindows()
```

In [2]:
```
#Load the target
image_target = cv2.imread('gambar/star02.jpeg')
target = cv2.resize(image_target, (300, 600))
target_gray = cv2.cvtColor(target,cv2.COLOR_BGR2GRAY)
```

In [6]:
```
#Load the target
image_target = cv2.imread('gambar/star02.jpeg')
target = cv2.resize(image_target, (300, 600))
target_gray = cv2.cvtColor(target,cv2.COLOR_BGR2GRAY)


# Canny edge
edged_template = cv2.Canny(template_gray,80,250)
edged_target = cv2.Canny(target_gray,80,250)

#Threshold both image
ret, thresh00 = cv2.threshold(template_gray,190,255,0)
ret, thresh01 = cv2.threshold(target_gray,200,255,0)

cv2.imshow('Filtered template', thresh00)
#cv2.imshow('Filtered target', thresh01)


#Find contour in template
#contour, ct, hierarchy =cv2.findContours(thresh1, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)
edged00,c00, h00 = cv2.findContours(thresh00, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_NONE)
edged01,c01, h01 = cv2.findContours(thresh01, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_NONE)


# Sort contour by area
sc00 = sorted(c00, key=cv2.contourArea, reverse=True)
sc01 = sorted(c01, key=cv2.contourArea, reverse=True)

imgA00=cv2.drawContours(template,sc00,1,(0,255,0),3)
cv2.imshow('Template', imgA00)

imgA01=cv2.drawContours(target,sc01,1,(0,255,0),3)
cv2.imshow('Target', imgA01)



cv2.waitKey()
cv2.destroyAllWindows()

print('Number of contour =' +str(len(c00)))


# Similarity check
template_contour = sc00[1]
target_contour = sc01[1]

match = cv2.matchShapes(template_contour,target_contour,1,0.0)

print(match)
```

```
Number of contour =18
8.63437500567431
```

In [ ]:

```
#Find contour in template
contour, hierarchy =cv2.findContours(thresh1, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)

#sort contour by area

sorted_contours = sorted(contours, key=cv2.contourArea, reverse=True)

# extract tje second largest contour which is the template contour
template_contour = contours[1]

# extract contour from the second target image
contours, hierarchy = cv2.findContours(thresh2, cv2.RETR_CCOMP, cv2.CHAIN_APPROX_SIMPLE)

for c in contours:
    macth = cv2.matchShapes(template_contour,c,1,0.0)
    print match

    if match < 0.15:
        closest_contour = c
    else:
        closest_contour = []

cv2.drawContours(target, [closest_contour],-1,(0,255,0),3)
cv2.imshow('output', target)
cv2.waitKey()
cv2.destroyAllWindows()
```