PCD Pertemuan 7 & 8

Rahma Fairuz Rania
J0303201065

Pertemuan 7

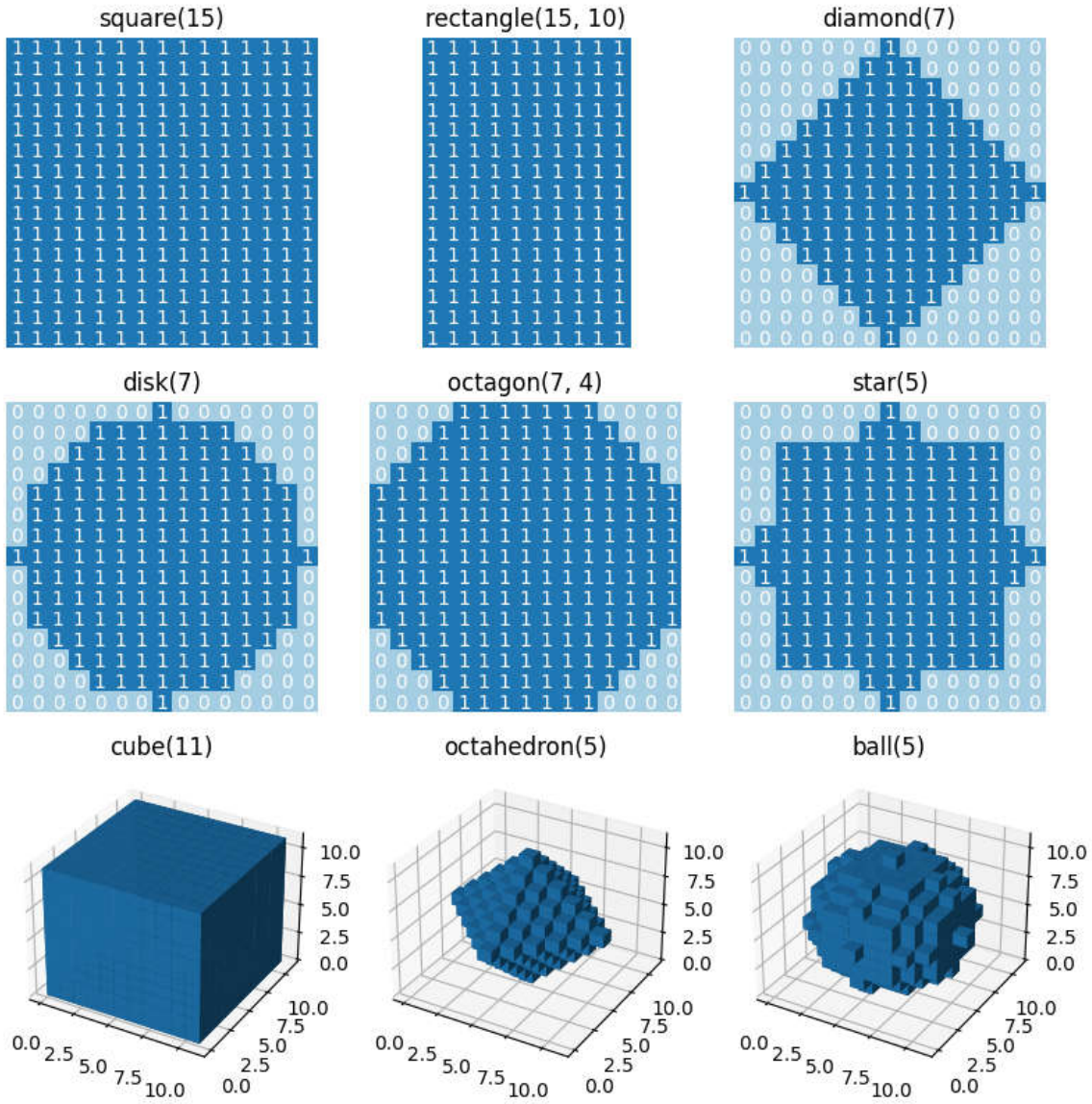
skimage.morphology untuk menghasilkan elemen penataan.

In [1]:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from skimage.morphology import (square, rectangle, diamond, disk, cube, octahedron, ball,
```

In [5]:

```
# Generate 2D and 3D structuring elements.
struc_2d = {
    "square(15)": square(15),
    "rectangle(15, 10)": rectangle(15, 10),
    "diamond(7)": diamond(7),
    "disk(7)": disk(7),
    "octagon(7, 4)": octagon(7, 4),
    "star(5)": star(5)
}

struc_3d = {
    "cube(11)": cube(11),
    "octahedron(5)": octahedron(5),
    "ball(5)": ball(5)
}

# Visualize the elements.
fig = plt.figure(figsize=(8, 8))

idx = 1
for title, struc in struc_2d.items():
    ax = fig.add_subplot(3, 3, idx)
    ax.imshow(struc, cmap="Paired", vmin=0, vmax=12)
    for i in range(struc.shape[0]):
        for j in range(struc.shape[1]):
            ax.text(j, i, struc[i, j], ha="center", va="center", color="w")
    ax.set_axis_off()
    ax.set_title(title)
    idx += 1

for title, struc in struc_3d.items():
    ax = fig.add_subplot(3, 3, idx, projection=Axes3D.name)
    ax.voxels(struc)
    ax.set_title(title)
    idx += 1

fig.tight_layout()
plt.show()
```



Segmentasi dan Morphology Gambar Koin

In [72]:

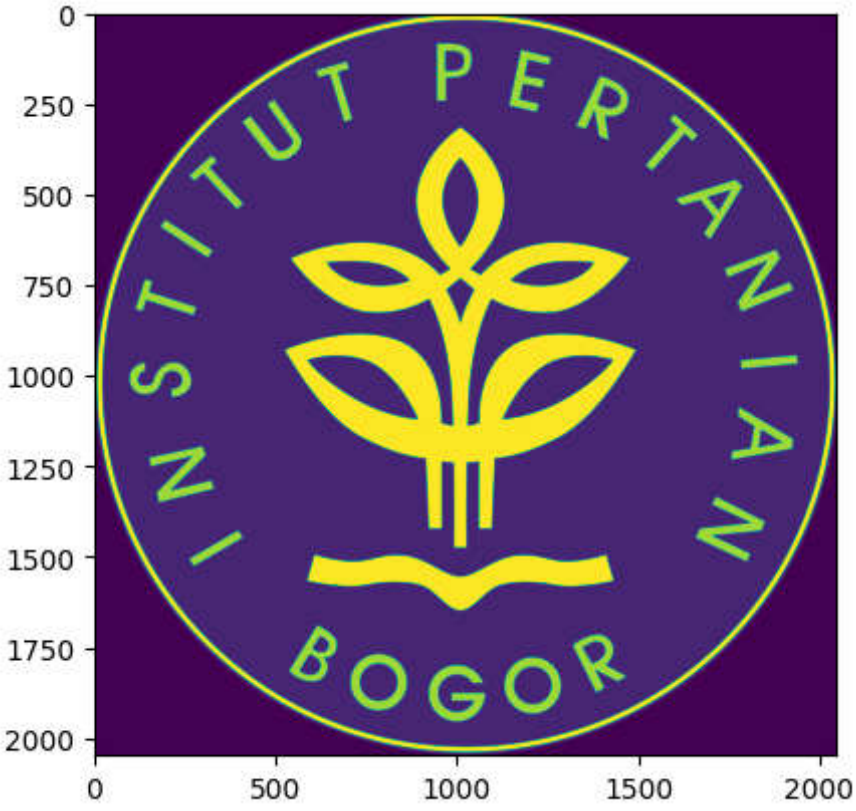
```
import numpy as np
import cv2
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import matplotlib.pyplot as plt
from skimage.data import coins
from skimage.filters import threshold_otsu
from skimage.segmentation import clear_border
from skimage.morphology import label, closing, square
from skimage.measure import regionprops
from skimage.color import lab2rgb
%matplotlib inline
```

In [117]:

```
def show(img, cmap=None):
    cmap = cmap or plt.cm.gray
    fig, ax = plt.subplots(1, 1, figsize=(8, 6))
    ax.imshow(img, cmap=cmap)
    ax.set_axis_off()
    plt.show()

img = cv2.imread('ip.png')
img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(img)
```

Out[117]:

```
<matplotlib.image.AxesImage at 0x15b7b8b6460>
```



In [118]:

```
threshold_otsu(img)
107
show(img > 107)
```

In [119]:

```
img_bin = clear_border(closing(img > 120, square(5)))
show(img_bin)
```

In [120]:

```
labels = label(img_bin)
show(labels, cmap=plt.cm.rainbow)
```

In [123]:

```
regions = regionprops(labels)
boxes = np.array([label['BoundingBox']
 for label in regions
if label['Area'] > 100])
print(f"There are {len(boxes)} words including space.")
```

```
There are 25 words including space.
```

In [125]:

```
fig, ax = plt.subplots(1, 1, figsize=(8, 6))
ax.imshow(img, cmap=plt.cm.gray)
ax.set_axis_off()

# Get the coordinates of the boxes.
xs = boxes[:, [1, 3]].mean(axis=1)
ys = boxes[:, [0, 2]].mean(axis=1)

# We reorder the boxes by increasing
# column first, and row second.
#for row in range(4):
#    # We select the coins in each of the four rows.
#    if row < 3:
#        ind = ((ys[6 * row] <= ys) & (ys < ys[6 * row + 6]))
#    else:
#         ind = (ys[6 * row] <= ys)

#     We reorder by increasing x coordinate.
#     ind = np.nonzero(ind)[0]
#     reordered = ind[np.argsort(xs[ind])]
#     xs_row = xs[reordered]
#     ys_row = ys[reordered]

    # We display the coin number.
#     for col in range(6):
#         n = 6 * row + col
#         ax.text(xs_row[col] - 5, ys_row[col] + 5, str(n), fontsize=20)
```

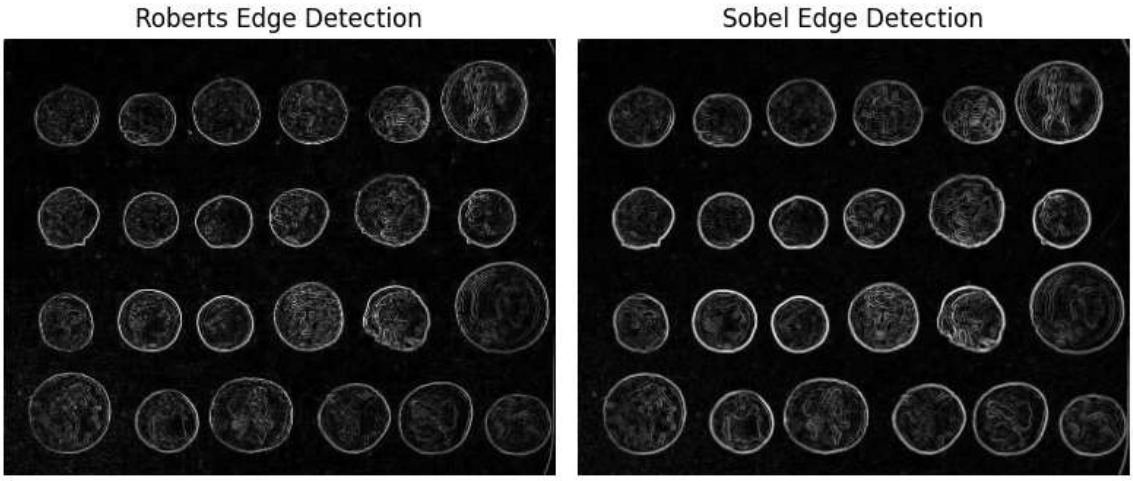
Pertemuan 8

In [77]:

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import filters
from skimage.data import camera
from skimage.util import compare_images
```

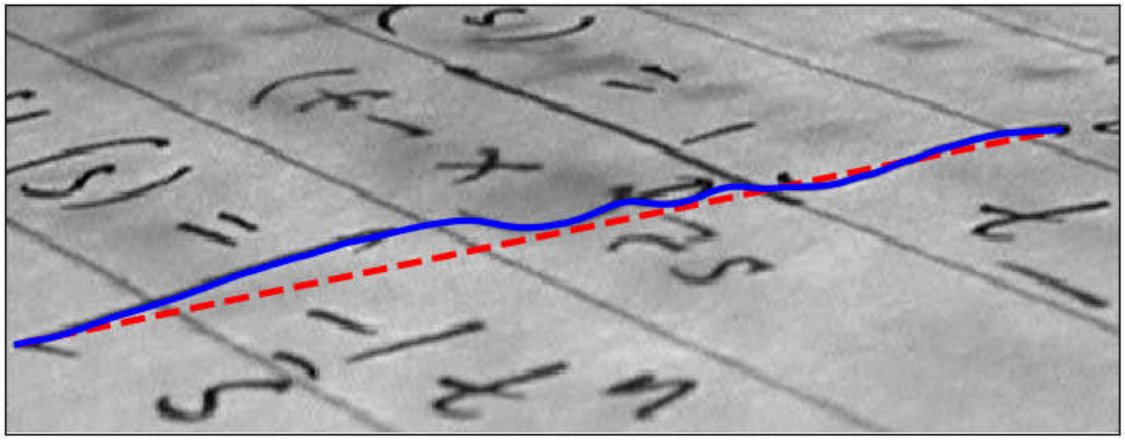
In [126]:

```
# Deteksi tepi
image = coins()
edge_roberts = filters.roberts(image)
edge_sobel = filters.sobel(image)
fig, axes = plt.subplots(ncols=2, sharex=True,
sharey=True,
 figsize=(8, 4))
axes[0].imshow(edge_roberts, cmap=plt.cm.gray)
axes[0].set_title('Roberts Edge Detection')
axes[1].imshow(edge_sobel, cmap=plt.cm.gray)
axes[1].set_title('Sobel Edge Detection')
for ax in axes:
 ax.axis('off')
plt.tight_layout()
plt.show()
```



In [127]:

```
# active contour
import numpy as np
import matplotlib.pyplot as plt
from skimage.color import rgb2gray
from skimage import data
from skimage.filters import gaussian
from skimage.segmentation import active_contour
img = data.text()
r = np.linspace(136, 50, 100)
c = np.linspace(5, 424, 100)
init = np.array([r, c]).T
snake = active_contour(gaussian(img, 1), init,
boundary_condition='fixed',
 alpha=0.1, beta=1.0,
w_line=-5, w_edge=0, gamma=0.1)
fig, ax = plt.subplots(figsize=(9, 5))
ax.imshow(img, cmap=plt.cm.gray)
ax.plot(init[:, 1], init[:, 0], '--r', lw=3)
ax.plot(snake[:, 1], snake[:, 0], '-b', lw=3)
ax.set_xticks([]), ax.set_yticks([])
ax.axis([0, img.shape[1], img.shape[0], 0])
plt.show()
```

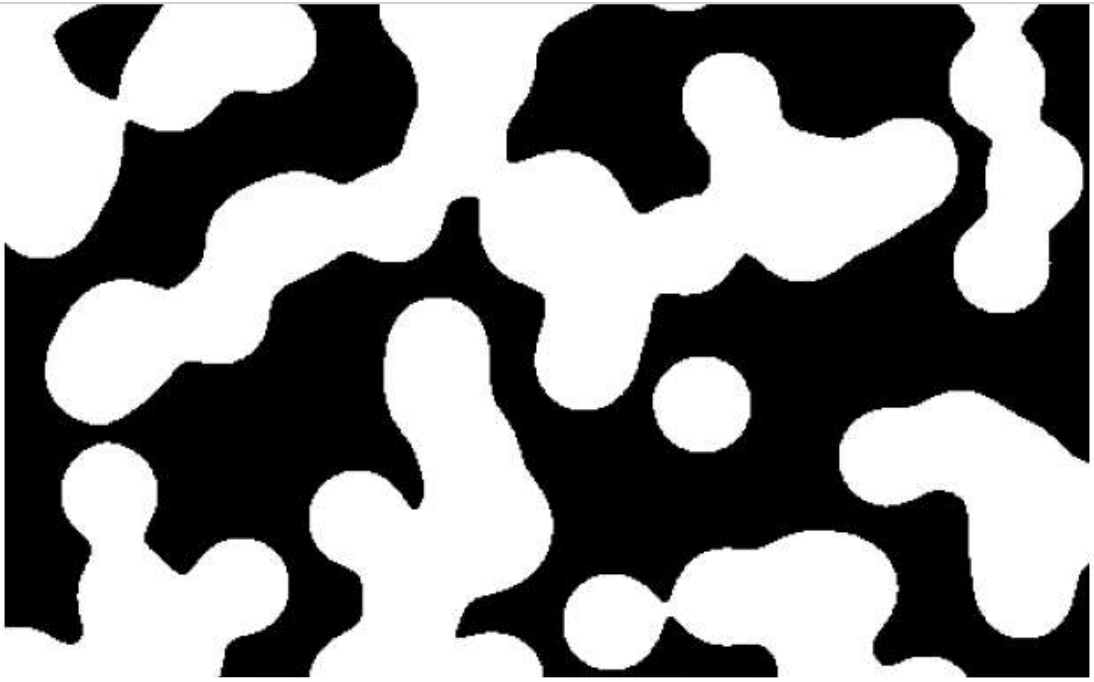


In [82]:

```
# gaussian noise
import numpy as np
import matplotlib.pyplot as plt
import skimage
import skimage.color as skic
import skimage.filters as skif
import skimage.data as skid
import skimage.util as sku
%matplotlib inline
```

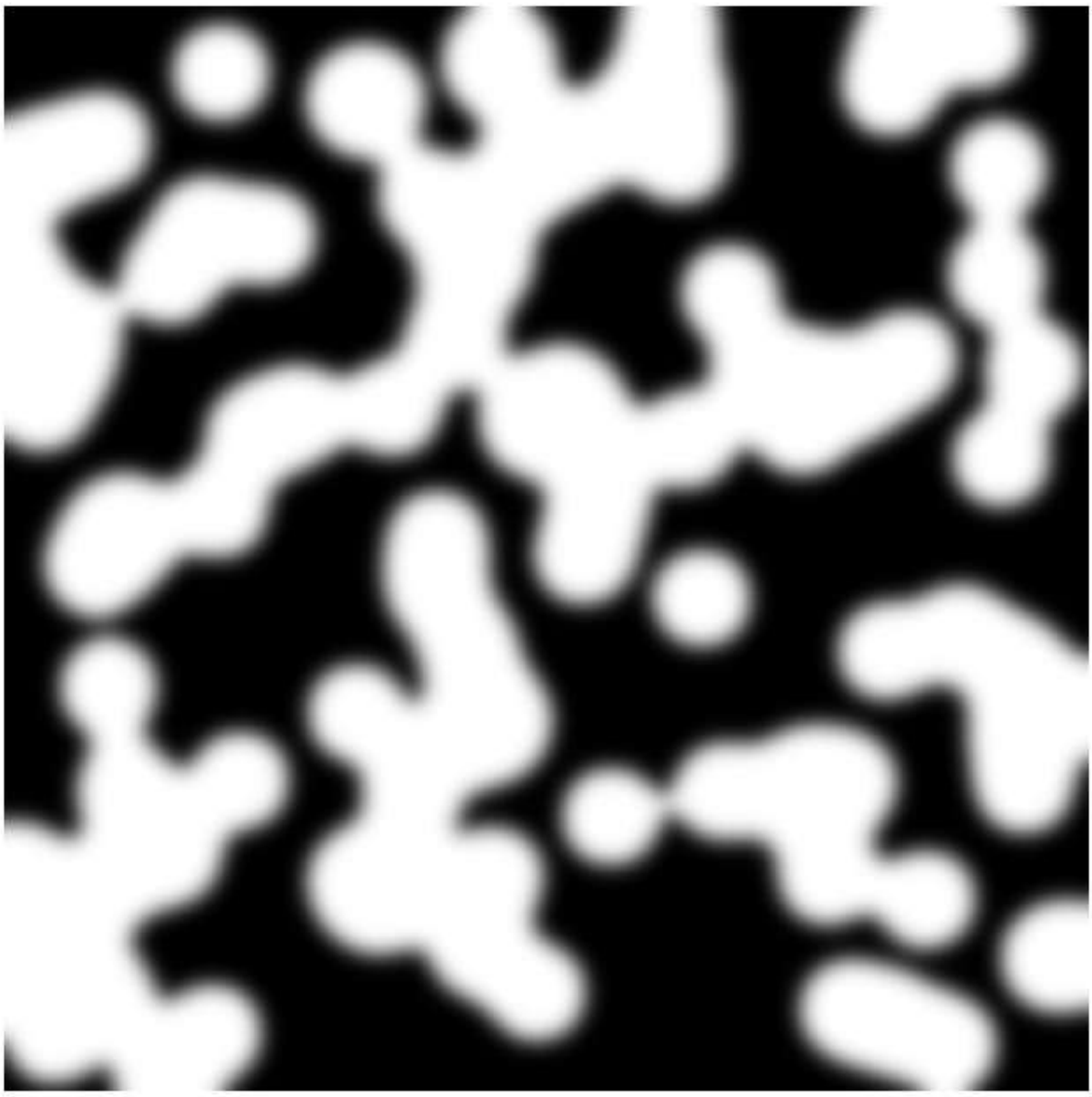
In [134]:

```
def show(img):
    fig, ax = plt.subplots(1, 1, figsize=(8, 8))
    ax.imshow(img, cmap=plt.cm.gray)
    ax.set_axis_off()
    plt.show()

img = skic.rgb2gray(skimage.data.binary_blobs())
show(img)
```



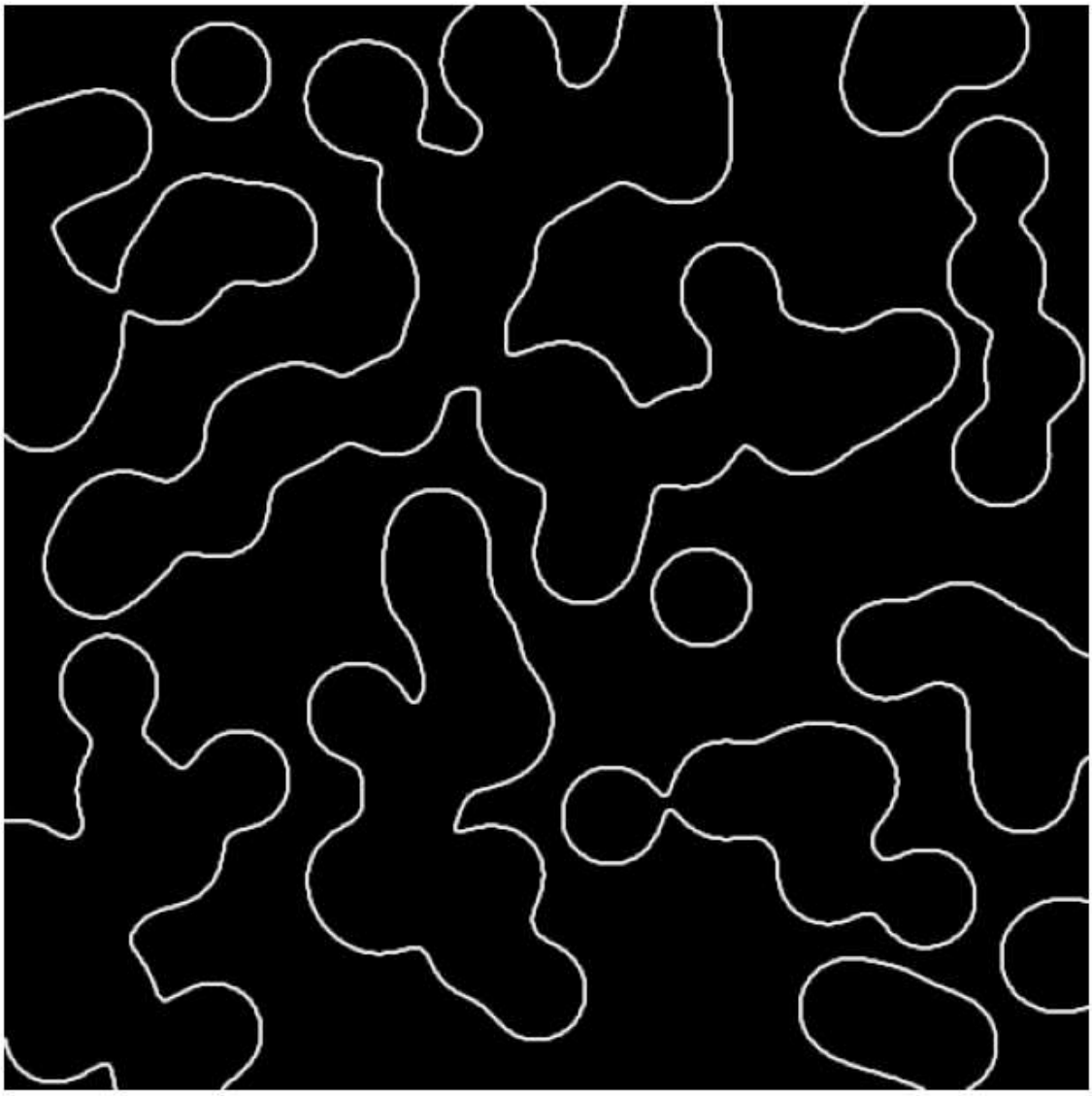
In [135]:

```
show(skif.gaussian(img, 5.))
```



In [136]:

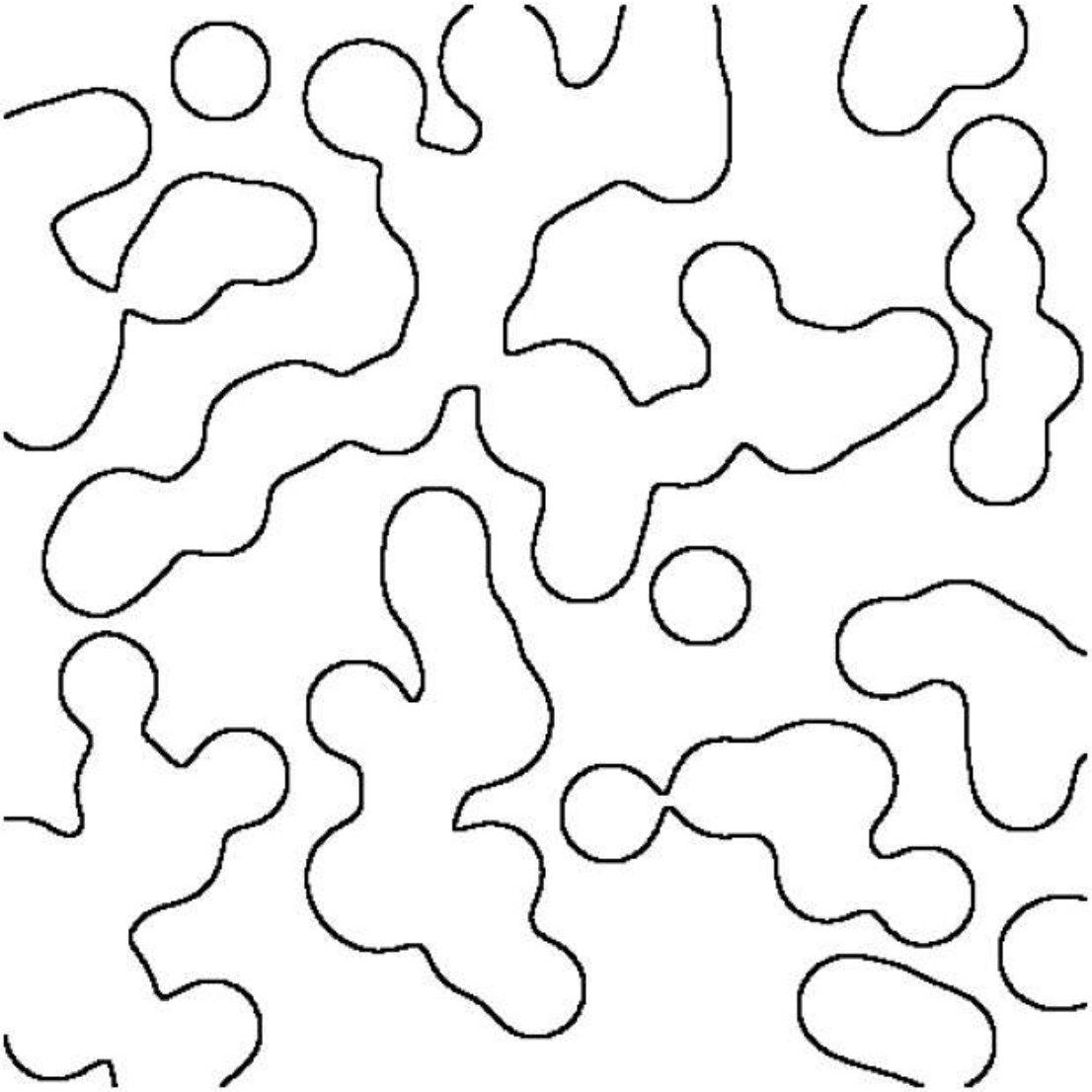
```
sobimg = skif.sobel(img)
show(sobimg)
```



In [137]:

```
from ipywidgets import widgets
@widgets.interact(x=(0.01, .2, .005))
def edge(x):
    show(sobimg < x)
```

x 0.10



In [147]:

```
img = Image.open('rahma.png')
img = skimage.img_as_float(img)
# We take a portion of the image to show the details.
img = img[150:500, 150:300]
# We add Gaussian noise.
img_n = sku.random_noise(img)
show(img_n)
```

In [148]:

```
img_r = skimage.restoration.denoise_tv_bregman(img_n, 5.)
fig, (ax1, ax2, ax3) = plt.subplots(
 1, 3, figsize=(12, 8))
ax1.imshow(img_n)
ax1.set_title('With noise')
ax1.set_axis_off()
ax2.imshow(img_r)
ax2.set_title('Denoised')
ax2.set_axis_off()
ax3.imshow(img)
ax3.set_title('Original')
ax3.set_axis_off()
```