Tugas 4 Pengolahan Citra Digital

Nama : Rosa Amalia Nursinta

NIM : 11190940000041

Import Modul

In [49]:

```
from PIL import Image
import numpy as np
import random
import matplotlib.pyplot as plt
%matplotlib inline
import imageio
import cv2
```

Nomor 1

In [50]:

```
# Load gambar 1
img = Image.open('forest.jpeg')
pic = np.array(img)
pic
```

Out[50]:

```
array([[[165, 163, 166],
        [165, 163, 166],
        [165, 163, 166],
        ...,
        [156, 160, 163],
        [156, 160, 163],
        [156, 160, 163]],

       [[247, 245, 248],
        [247, 245, 248],
        [247, 245, 248],
        ...,
        [234, 238, 241],
        [234, 238, 241],
        [234, 238, 241]],

       [[245, 245, 247],
        [245, 245, 247],
```

In [51]:

```
# Load gambar 2
img2 = Image.open('gajah.jpeg')
pic2 = np.array(img2)
pic2
```

Out[51]:

```
array([[[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        ...,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        ...,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
        [0, 0, 0],
```

In [52]:

```
# Melihat ukuran pixel gambar 1
print(f'shape of the first image{pic.shape}')
print(f'heigt {pic.shape[0]} pixels')
print(f'width {pic.shape[1]} pixels')

# Melihat ukuran pixel gambar 2
print(f'shape of the second image{pic2.shape}')
print(f'heigt {pic2.shape[0]} pixels')
print(f'width {pic2.shape[1]} pixels')
```

```
shape of the first image(1334, 750, 3)
heigt 1334 pixels
width 750 pixels
shape of the second image(1334, 750, 3)
heigt 1334 pixels
width 750 pixels
```

In [53]:

```
# periksa ukuran dimensi kedua gambar

def deteksi_dimensi(pic,pic2):
    if pic.shape[0] == pic2.shape[0]:
        result = print('Berdimensi sama')
    else:
        result = print('Berbeda dimensi')

    return result
```

In [54]:

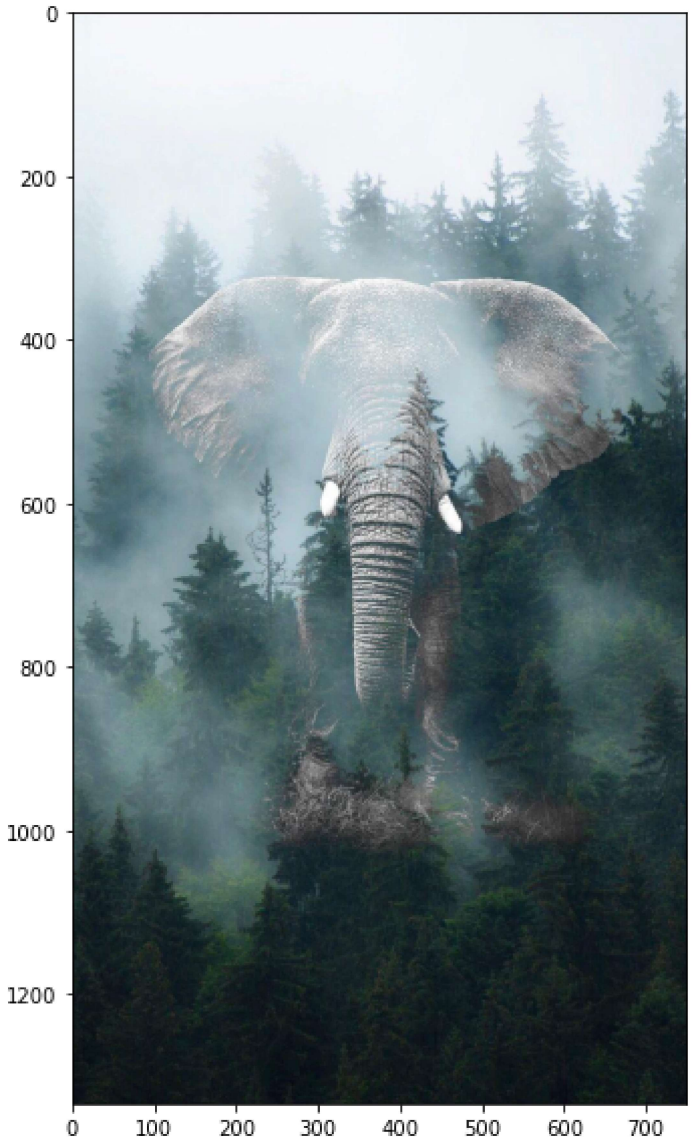
```
def operasi_min_max(pic,pic2):
    if pic.shape != pic2.shape:
        size_min = deteksi_dimensi(pic,pic2)
        dim = (size_min.shape[0], size_min.shape[1])
        resize_image = cv2.resize(size_min, dim, interpolation=cv2.INTER_AREA)
        print('dimensi berbeda sehingga size diubah menjadi nilai minimal dimensi dari kedu
    else:
        maxIMG = cv2.max(pic,pic2)
        minIMG = cv2.min(pic,pic2)
        imageio.imwrite('Nomor1_Max1.jpeg', maxIMG)
        imageio.imwrite('Nomor1_Min1.jpeg', minIMG)
```

In [55]:

```
operasi_min_max(pic,pic2)
```

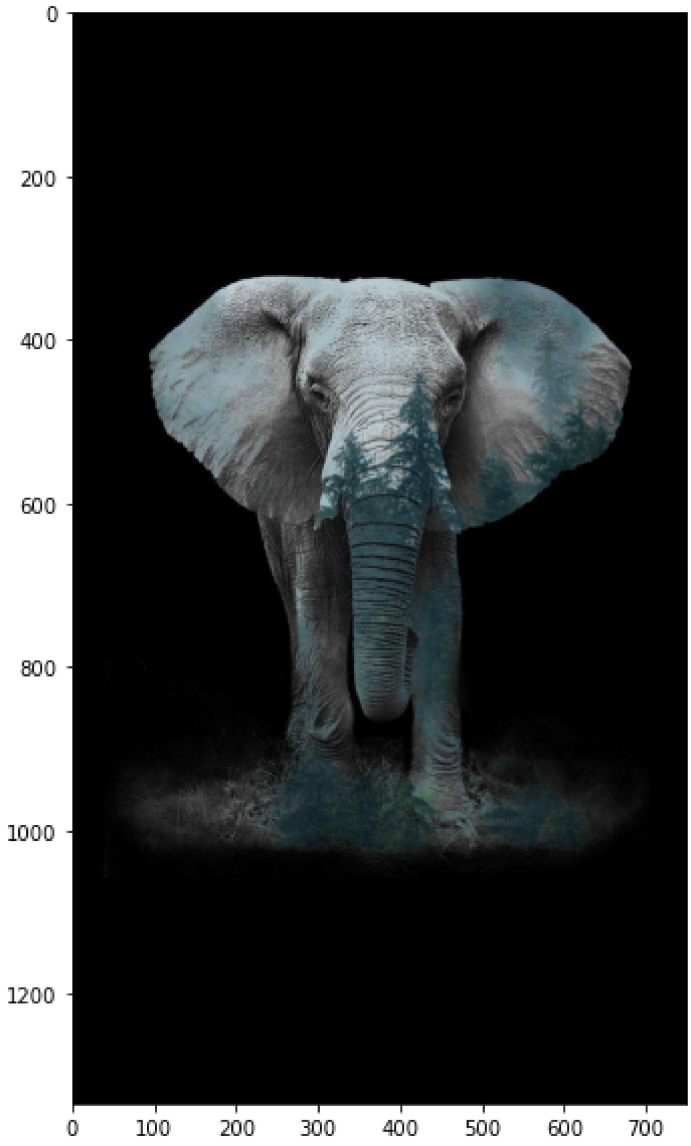
In [56]:

```
hasil_max1 = Image.open('Nomor1_Max1.jpeg')
plt.figure(figsize = (10,10))
plt.imshow(hasil_max1)
plt.show()
```



In [57]:

```
hasil_min1 = Image.open('Nomor1_Min1.jpeg')
plt.figure(figsize = (10,10))
plt.imshow(hasil_min1)
plt.show()
```



Nomor 2

In [58]:

```
img3 = Image.open('moon_.jpeg')
pic3 = np.array(img3)

img4 = Image.open('quran_ayat.jpeg')
pic4 = np.array(img4)
```

In [59]:

```
# Melihat ukuran pixel gambar 1
print(f'shape of the first image{pic3.shape}')
print(f'heigt {pic3.shape[0]} pixels')
print(f'width {pic3.shape[1]} pixels')

# Melihat ukuran pixel gambar 2
print(f'shape of the second image{pic4.shape}')
print(f'heigt {pic4.shape[0]} pixels')
print(f'width {pic4.shape[1]} pixels')
```

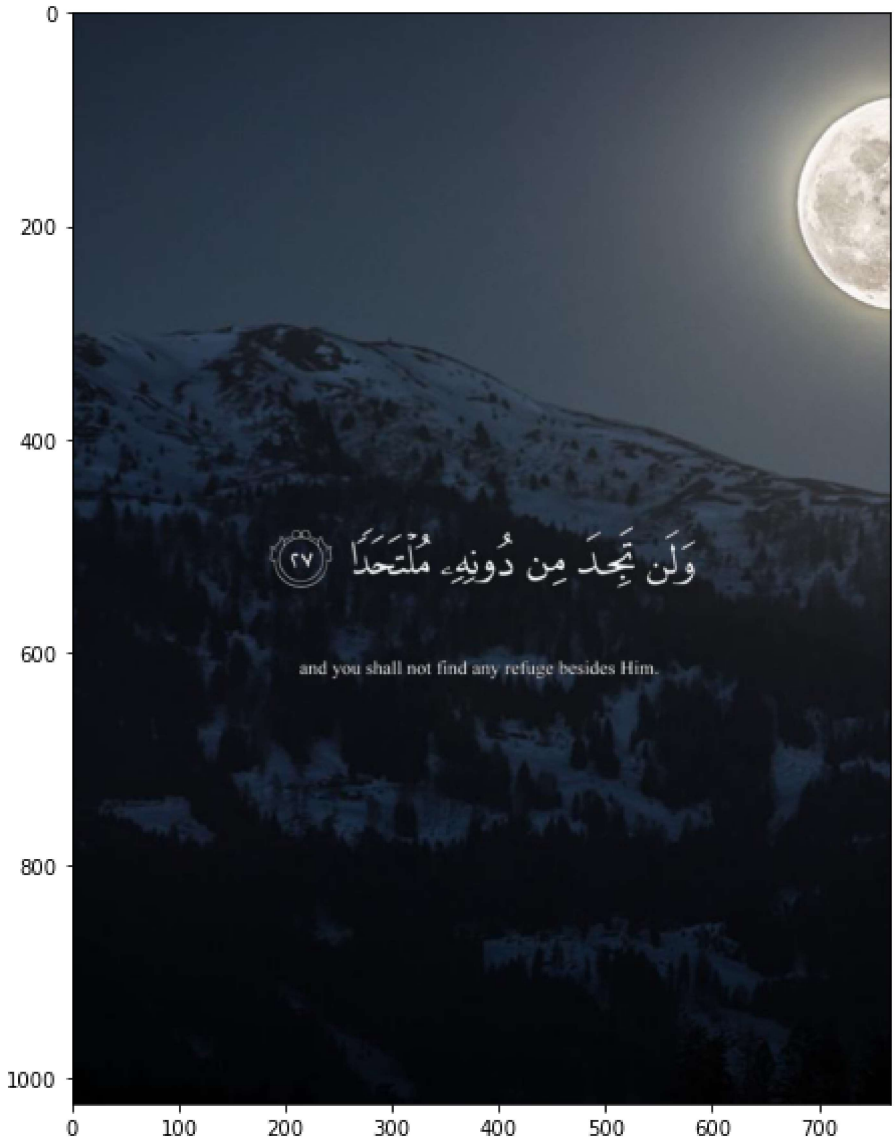
```
shape of the first image(1024, 768, 3)
heigt 1024 pixels
width 768 pixels
shape of the second image(1024, 768, 3)
heigt 1024 pixels
width 768 pixels
```

In [60]:

```
operasi_min_max(pic3,pic4)
```

In [61]:

```
hasil_max2 = Image.open('Nomor2_Max2.jpeg')
plt.figure(figsize = (10,10))
plt.imshow(hasil_max2)
plt.show()
```



Nomor 3

In [105]:

```
### Diabetics Retinopathy ###

img5 = Image.open('diabeticsa.jpg')
pic5 = np.array(img5)

img6 = Image.open('diabeticsb.jpg')
pic6 = np.array(img6)

img7 = Image.open('diabeticsc.jpg')
pic7 = np.array(img7)
```

In [106]:

```
def show_image(img, cmap='gray'):
    fig = plt.figure(figsize=(20,20))
    axes = fig.add_subplot(111)
    axes.imshow(img, cmap=cmap)
```

In [121]:

```
# A
mean6 = pic6.mean()
print('Mean: %.3f' % mean6)
```

```
Mean: 86.744
```

In [122]:

```
# B
B6, G6, R6 = cv2.split(pic6)
B6[B6 > 86.744] = 255
G6[G6 > 86.744] = 255
R6[R6 > 86.744] = 255
```

In [123]:

```
image6 = (cv2.merge([B6,G6,R6])).astype(np.uint8)
```

In [124]:

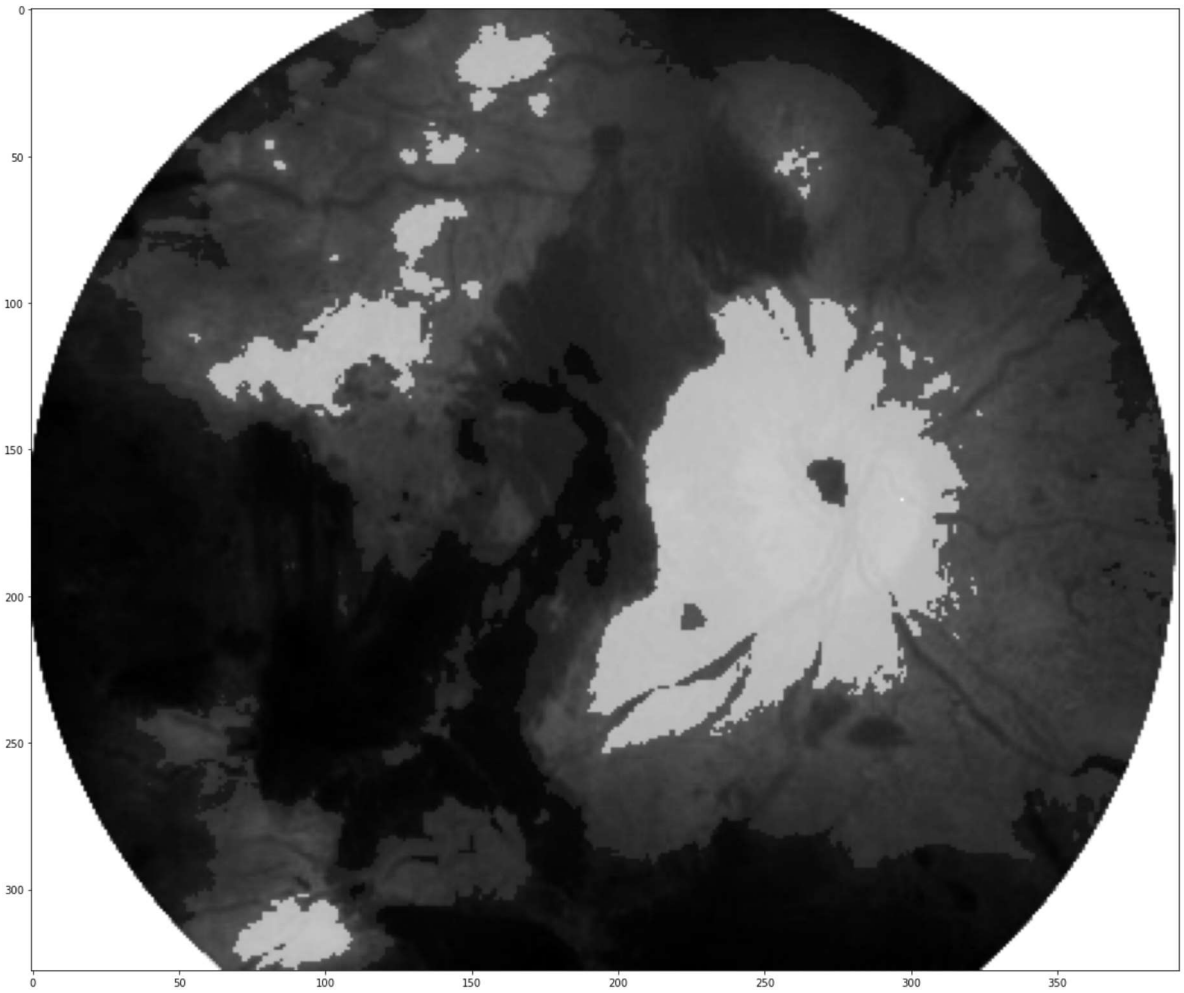
```
image6.shape
```

Out[124]:

```
(328, 392, 3)
```

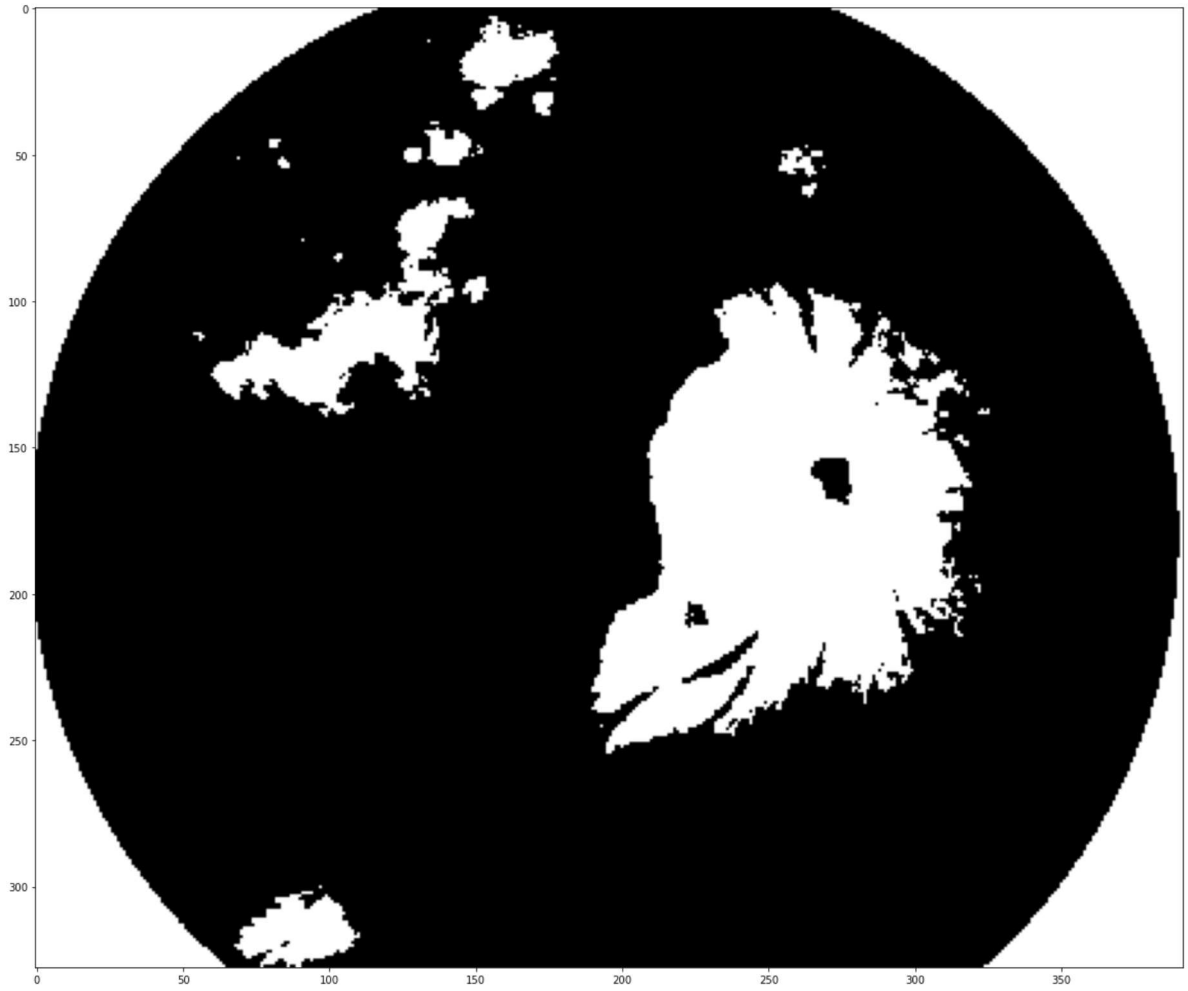
In [125]:

```
# C
gray6 = cv2.cvtColor(image6,cv2.COLOR_BGR2GRAY)
show_image(gray6)
```



In [126]:

```
# D
tresh6 = cv2.threshold(gray6, 86.744, 255, cv2.THRESH_BINARY)[1]
show_image(tresh6)
```



In [127]:

```
# A
mean5 = pic5.mean()
print('Mean: %.3f' % mean5)
```

```
Mean: 99.394
```

In [128]:

```
# B
B5, G5, R5 = cv2.split(pic5)
B5[B5 > 99.394] = 255
G5[G5 > 99.394] = 255
R5[R5 > 99.394] = 255
```

In [129]:

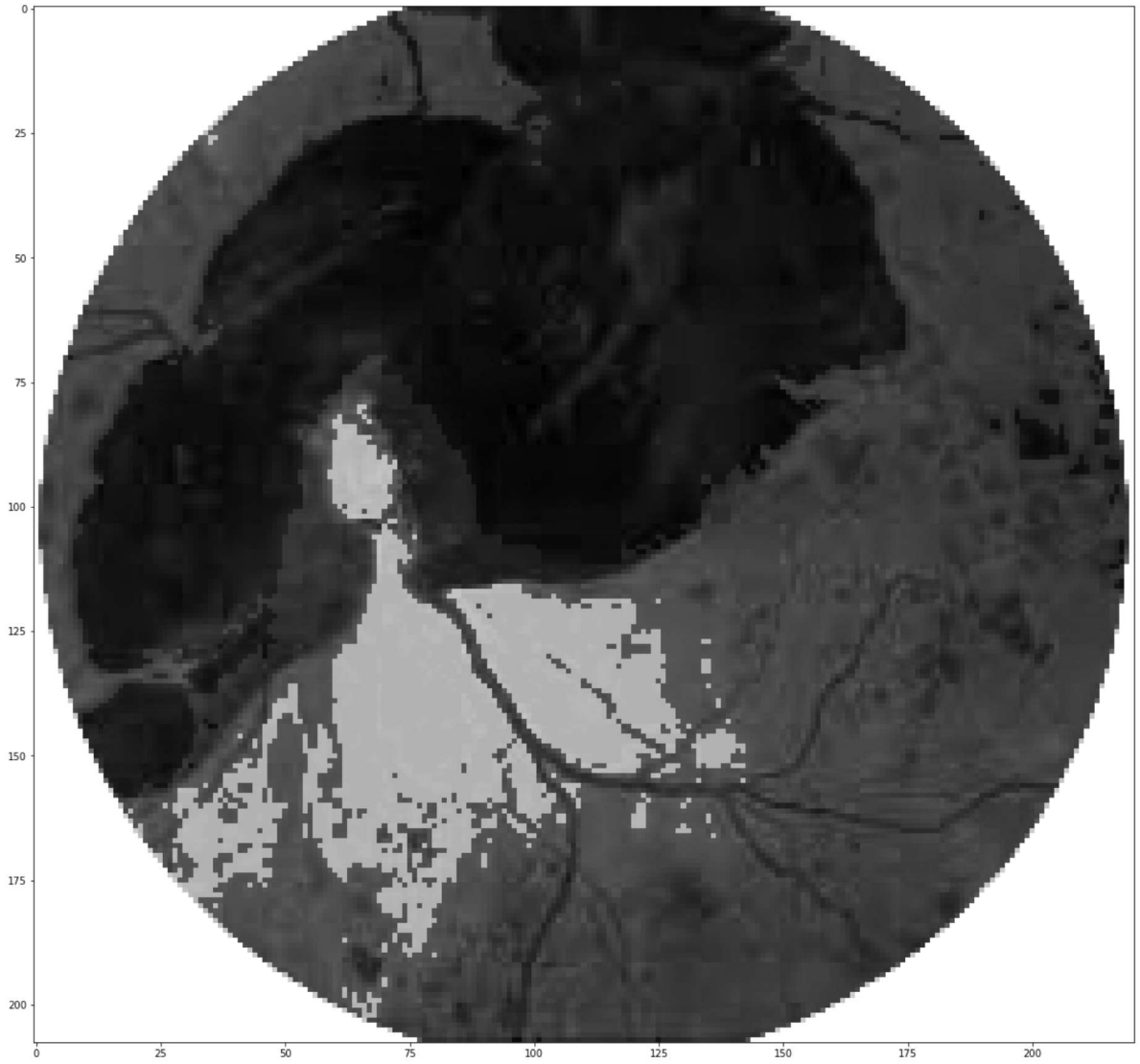
```
image5 = (cv2.merge([B5,G5,R5])).astype(np.uint8)
image5.shape
```

Out[129]:

```
(208, 221, 3)
```

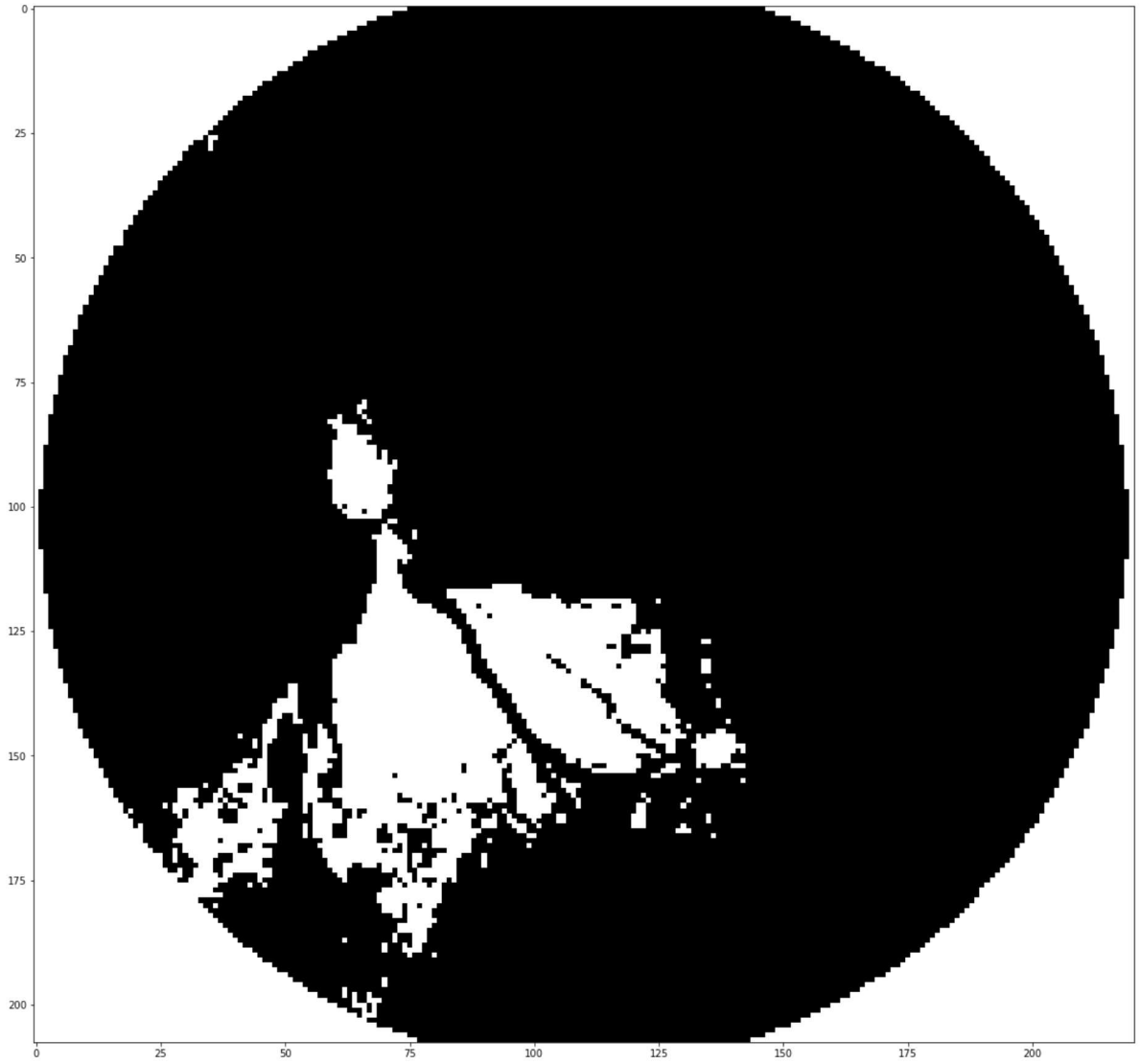
In [130]:

```
# C
gray5 = cv2.cvtColor(image5,cv2.COLOR_BGR2GRAY)
show_image(gray5)
```



In [131]:

```
# D
tresh5 = cv2.threshold(gray5, 99.394, 255, cv2.THRESH_BINARY)[1]
show_image(tresh5)
```



In [ ]:

```
```

In [132]:

```
# A
mean7 = pic7.mean()
print('Mean: %.3f' % mean7)
```

```
Mean: 132.354
```

In [133]:

```
# B
B7, G7, R7 = cv2.split(pic7)
B7[B7 > 132.354] = 255
G7[G7 > 132.354] = 255
R7[R7 > 132.354] = 255
```

In [135]:

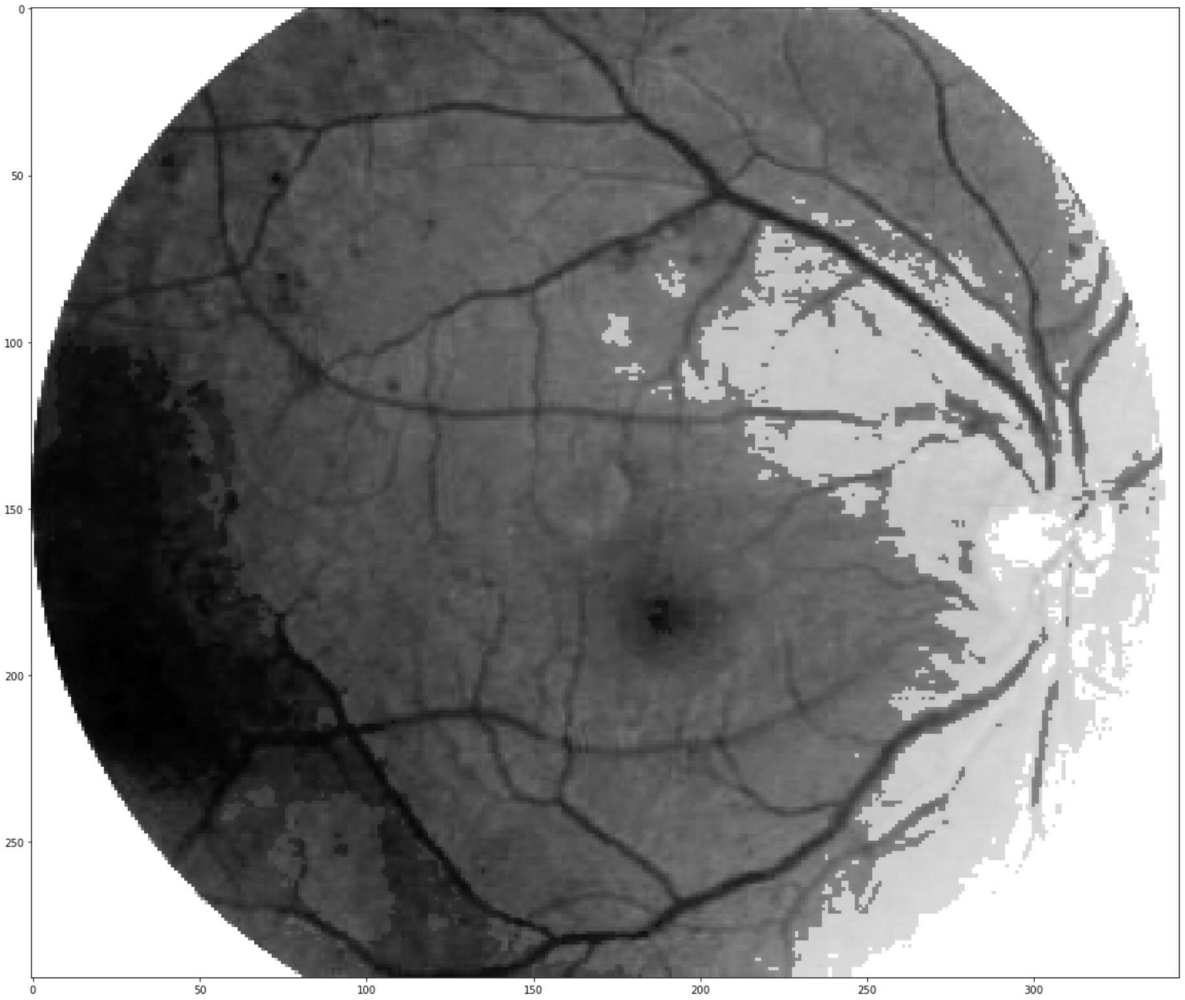
```
image7 = (cv2.merge([B7,G7,R7])).astype(np.uint8)
image7.shape
```

Out[135]:

```
(291, 344, 3)
```

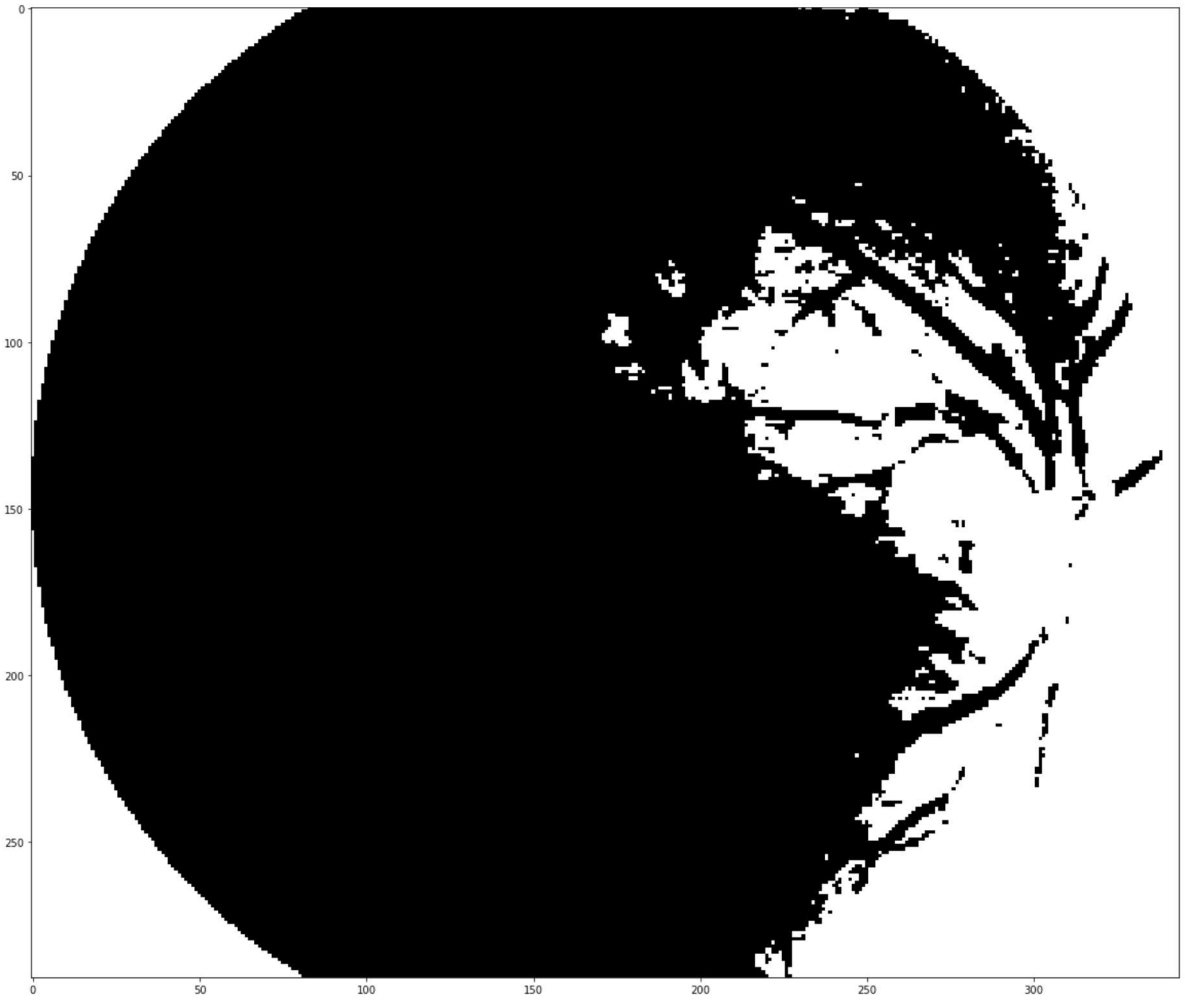
In [136]:

```
# C
gray7 = cv2.cvtColor(image7,cv2.COLOR_BGR2GRAY)
show_image(gray7)
```



In [137]:

```
# D
tresh7 = cv2.threshold(gray7, 132.354, 255, cv2.THRESH_BINARY)[1]
show_image(tresh7)
```



In [ ]:

```
```

In [113]:

```
### Normal Retina ###

img8 = Image.open('retina_normal1.jpg')
pic8 = np.array(img8)

img9 = Image.open('retina_normal2.jpg')
pic9 = np.array(img9)

img10 = Image.open('retina_normal3.jpg')
pic10 = np.array(img10)
```

In [114]:

```
def show_image(img, cmap='gray'):
    fig = plt.figure(figsize=(20,20))
    axes = fig.add_subplot(111)
    axes.imshow(img, cmap=cmap)
```

In [115]:

```
# A
mean8 = pic8.mean()
print('Mean: %.3f' % mean8)
```

```
Mean: 151.238
```

In [116]:

```
# B
B8, G8, R8 = cv2.split(pic8)
B8[B8 > 151.238] = 255
G8[G8 > 151.238] = 255
R8[R8 > 151.238] = 255
```

In [117]:

```
image8 = (cv2.merge([B8,G8,R8])).astype(np.uint8)
```

In [118]:

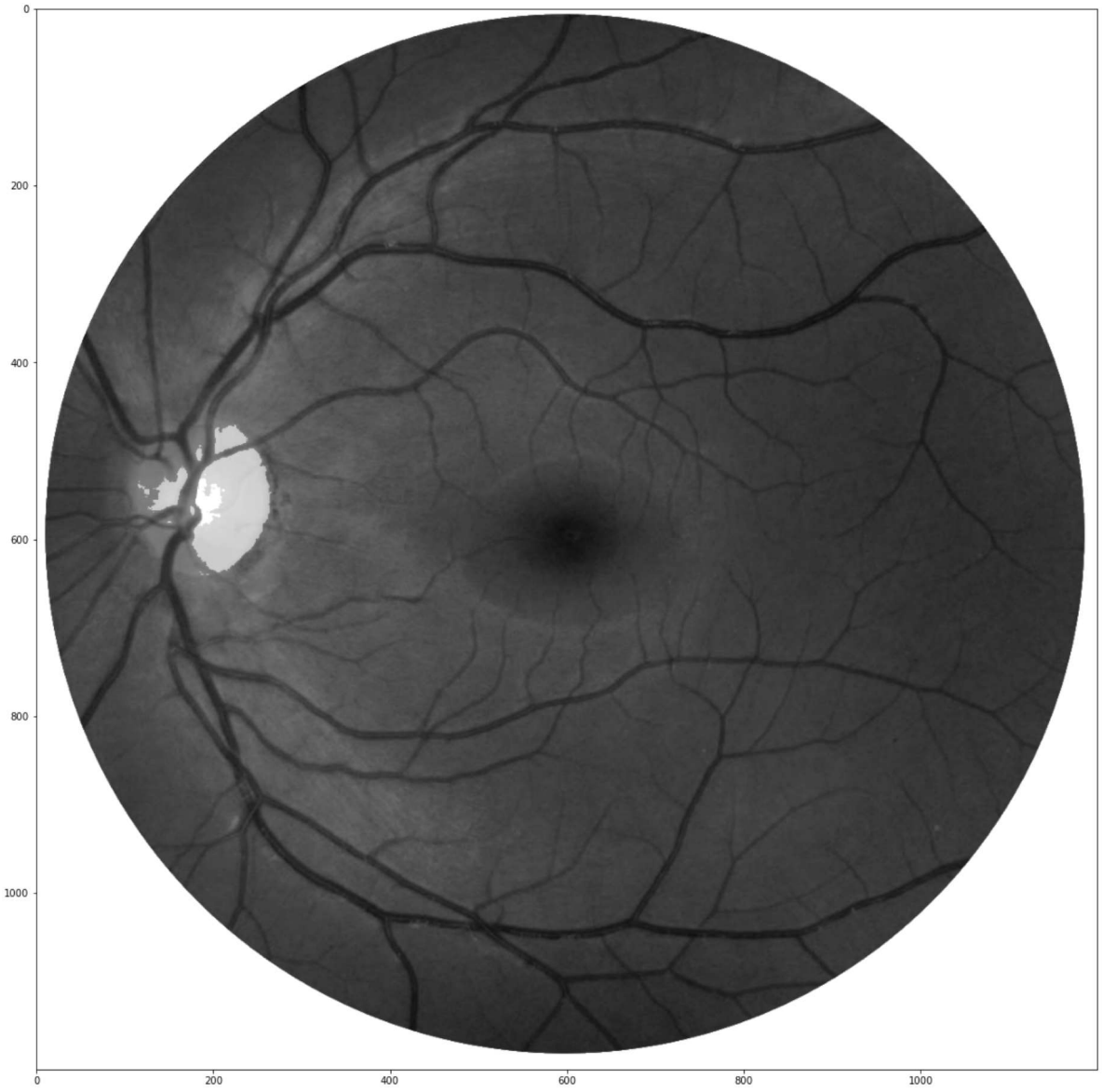
```
image8.shape
```

Out[118]:

```
(1200, 1200, 3)
```

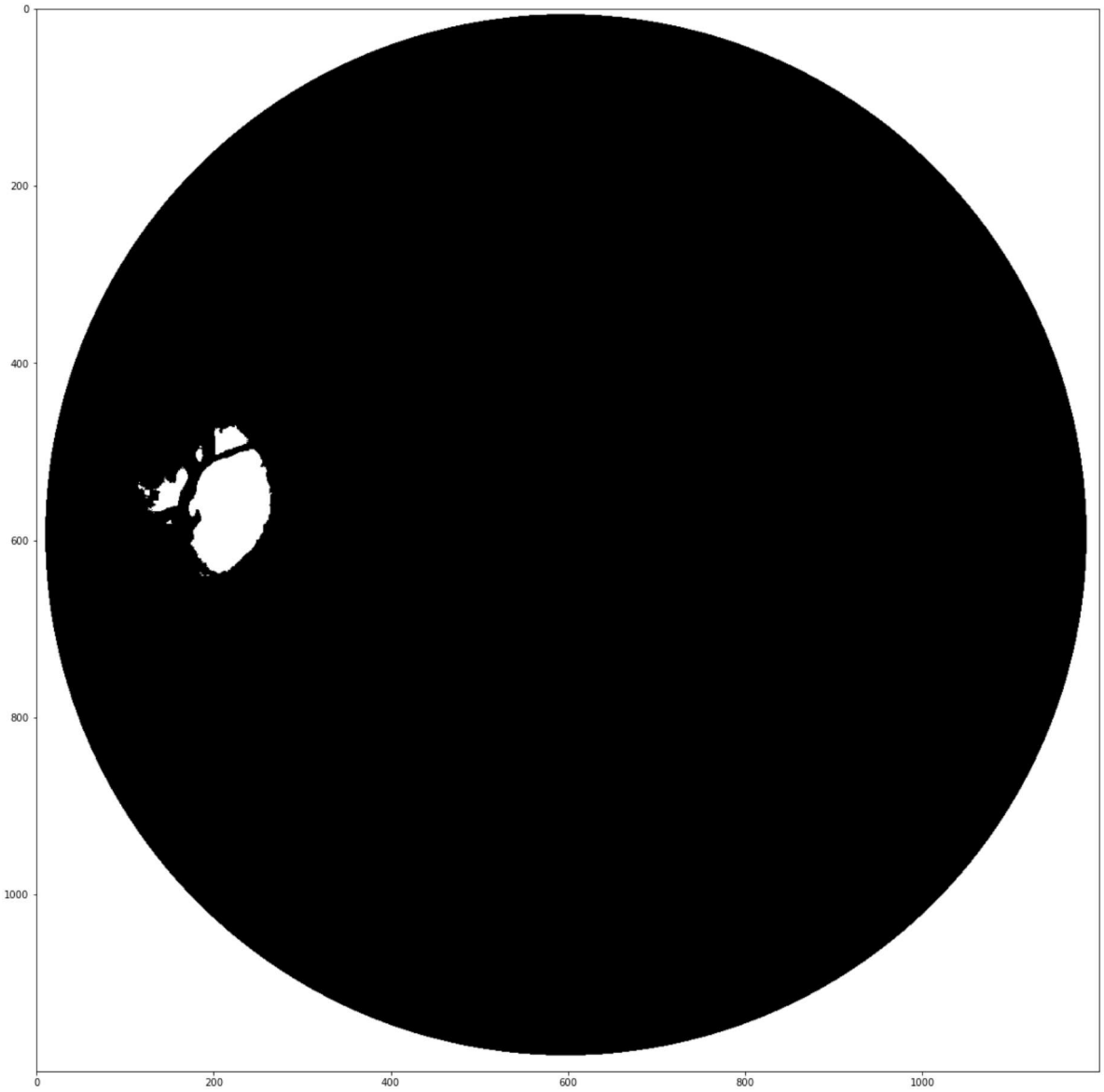
In [119]:

```
# C
gray8 = cv2.cvtColor(image8,cv2.COLOR_BGR2GRAY)
show_image(gray8)
```



In [138]:

```
# D
tresh8 = cv2.threshold(gray8, 151.238, 255, cv2.THRESH_BINARY)[1]
show_image(tresh8)
```



In [139]:

```
# A
mean9 = pic9.mean()
print('Mean: %.3f' % mean9)
```

```
Mean: 164.631
```

In [140]:

```
# B
B9, G9, R9 = cv2.split(pic9)
B9[B9 > 164.631] = 255
G9[G9 > 164.631] = 255
R9[R9 > 164.631] = 255
```

In [141]:

```
image9 = (cv2.merge([B9,G9,R9])).astype(np.uint8)
image9.shape
```

Out[141]:

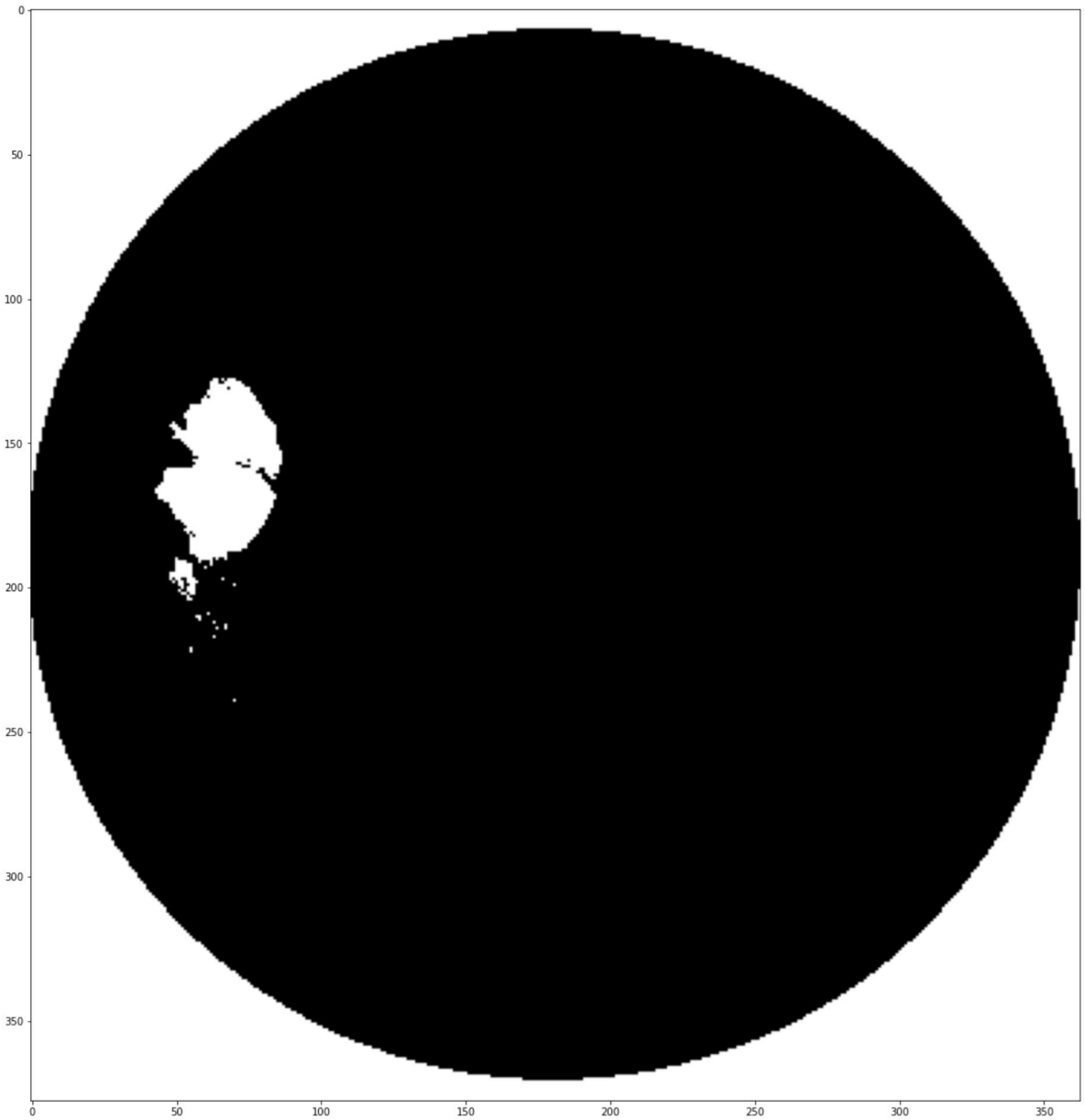
```
(378, 363, 3)
```

In [142]:

```
# C
gray9 = cv2.cvtColor(image9,cv2.COLOR_BGR2GRAY)
show_image(gray9)
```

In [143]:

```
# D
tresh9 = cv2.threshold(gray9, 164.631, 255, cv2.THRESH_BINARY)[1]
show_image(tresh9)
```



In [144]:

```
# A
mean10 = pic10.mean()
print('Mean: %.3f' % mean10)
```

```
Mean: 162.378
```

In [145]:

```
# B
B10, G10, R10 = cv2.split(pic10)
B10[B10 > 162.378] = 255
G10[G10 > 162.378] = 255
R10[R10 > 162.378] = 255
```

In [146]:

```
image10 = (cv2.merge([B10,G10,R10])).astype(np.uint8)
image10.shape
```

Out[146]:

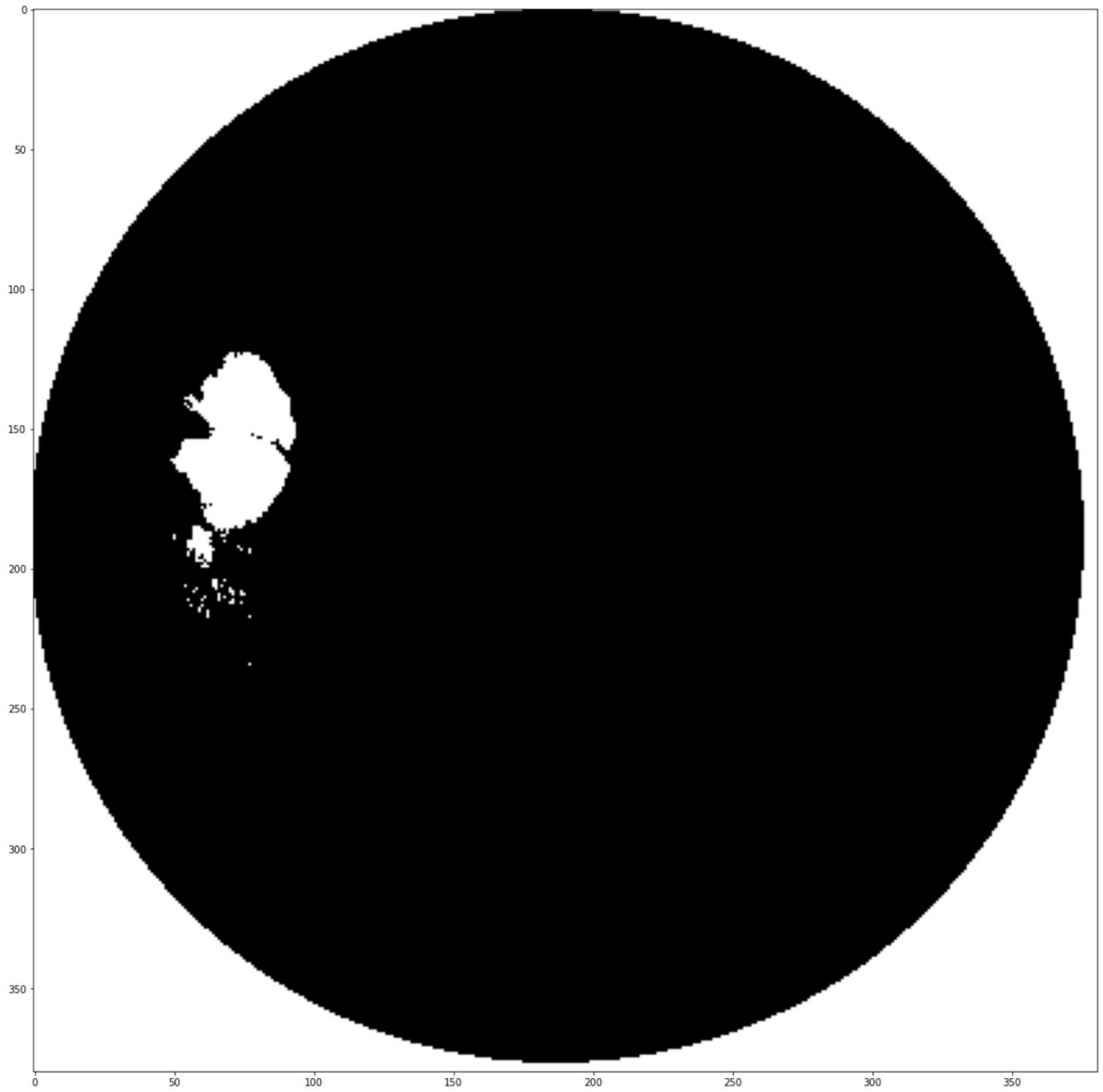
```
(380, 381, 3)
```

In [147]:

```
# C
gray10 = cv2.cvtColor(image10,cv2.COLOR_BGR2GRAY)
show_image(gray10)
```

In [148]:

```
# D
tresh10 = cv2.threshold(gray10, 162.378, 255, cv2.THRESH_BINARY)[1]
show_image(tresh10)
```



Secara visual terlihat perbedaan yang signifikan antara citra retina normal dengan retina pada penderita diabetes dan dengan parameter threshold kita mampu mendeteksi adanya gangguan Retinopati Diabetik pada penderita diabetes.

In [ ]:

```
```