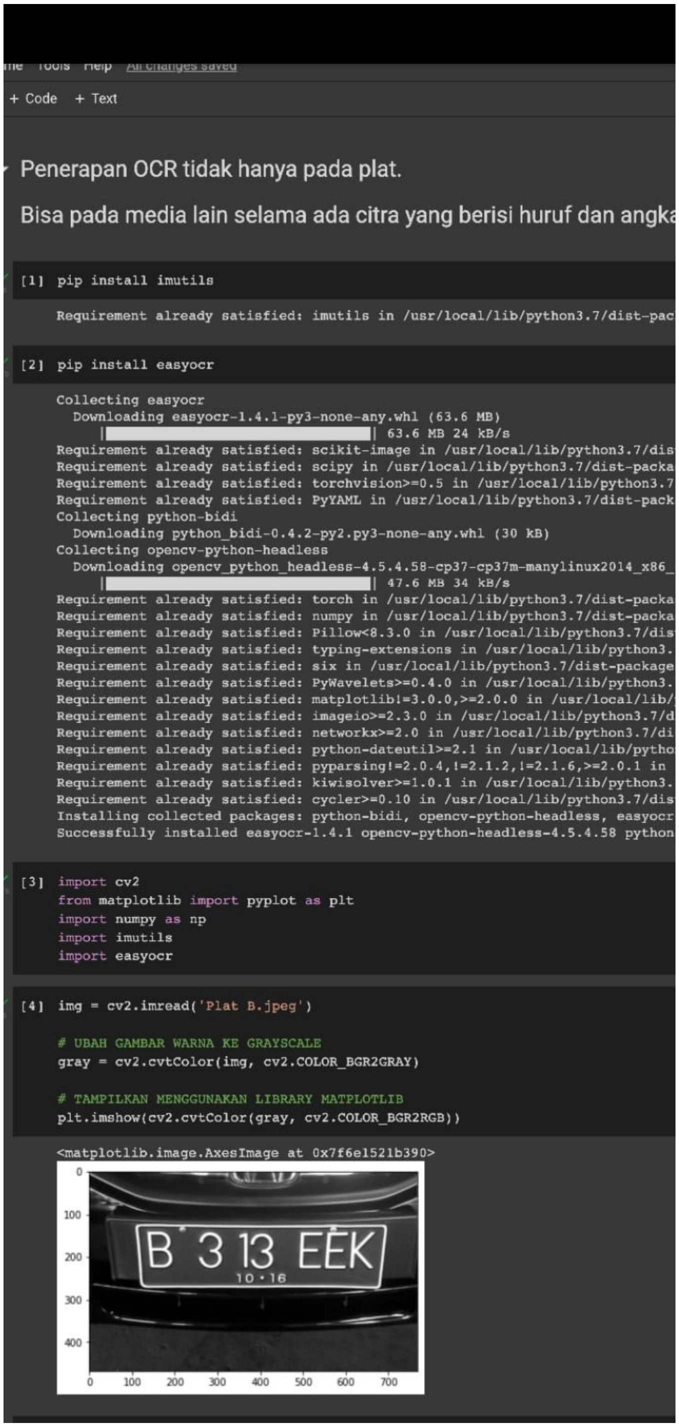
Nama : Armanita Aning

NIM : 20.01.013.002

Tugas : task 9

Ke;as : C Python



Penerapan OCR tidak hanya pada plat.

Bisa pada media lain selama ada citra yang berisi huruf dan angka

```
[1] pip install imutils

    Requirement already satisfied: imutils in /usr/local/lib/python3.7/dist-pac
```

```
[2] pip install easyocr

    Collecting easyocr
      Downloading easyocr-1.4.1-py3-none-any.whl (63.6 MB)
         |████████████████████████████████| 63.6 MB 24 kB/s
    Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dis
    Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.7
    Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-pack
    Collecting python-bidi
      Downloading python_bidi-0.4.2-py2.py3-none-any.whl (30 kB)
    Collecting opencv-python-headless
      Downloading opencv_python_headless-4.5.4.58-cp37-cp37m-manylinux2014_x86_
         |████████████████████████████████| 47.6 MB 34 kB/s
    Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packa
    Requirement already satisfied: Pillow<8.3.0 in /usr/local/lib/python3.7/dis
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-package
    Requirement already satisfied: PyWavelets>=0.4.0 in /usr/local/lib/python3.
    Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/
    Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/d
    Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/di
    Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/pytho
    Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
    Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.
    Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dis
    Installing collected packages: python-bidi, opencv-python-headless, easyocr
    Successfully installed easyocr-1.4.1 opencv-python-headless-4.5.4.58 python
```

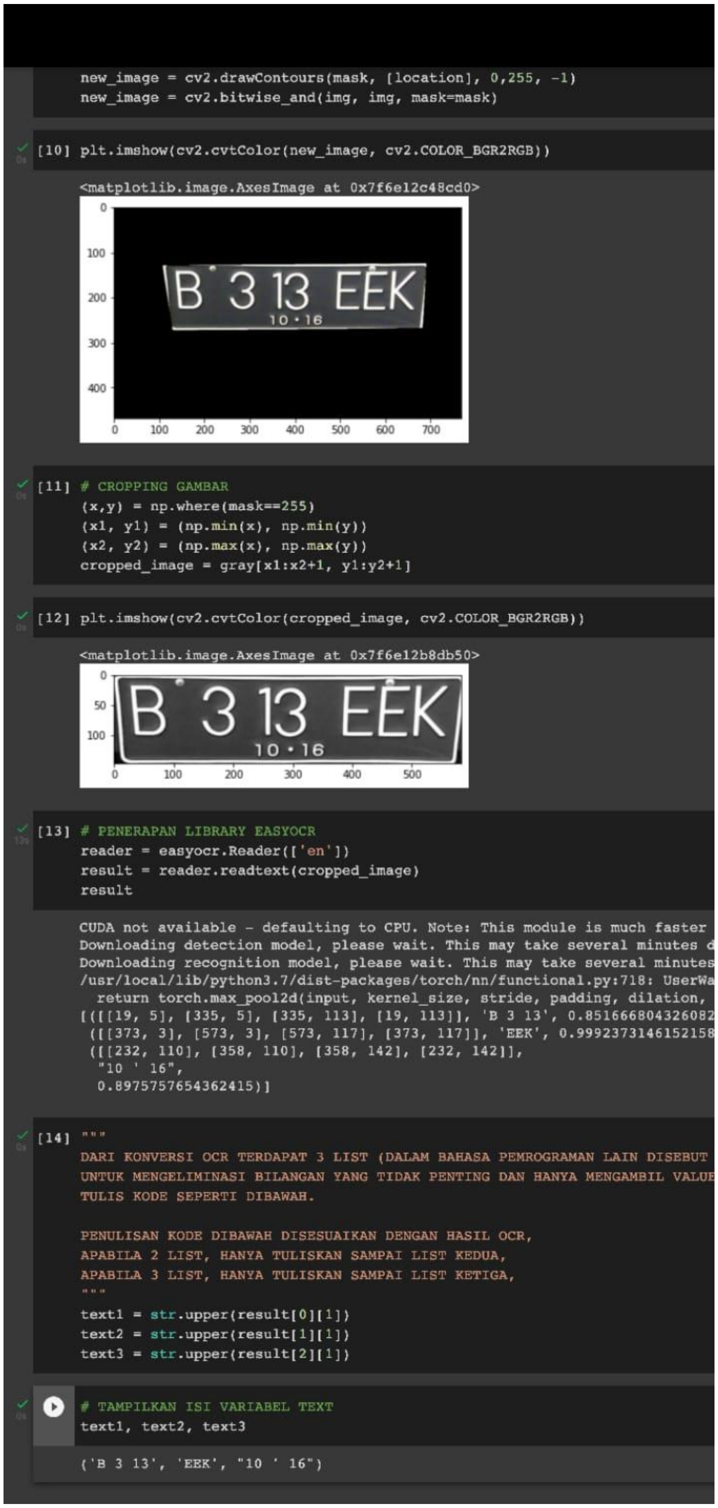
```
[3] import cv2
    from matplotlib import pyplot as plt
    import numpy as np
    import imutils
    import easyocr
```

```
[4] img = cv2.imread('Plat B.jpeg')

    # UBAH GAMBAR WARNA KE GRAYSCALE
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # TAMPILKAN MENGGUNAKAN LIBRARY MATPLOTLIB
    plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))

    <matplotlib.image.AxesImage at 0x7f6e1521b390>
```



[redacted]

```
    new_image = cv2.drawContours(mask, [location], 0,255, -1)
    new_image = cv2.bitwise_and(img, img, mask=mask)
```

```
[10] plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f6e12c48cd0>
```

```
[11] # CROPPING GAMBAR
     (x,y) = np.where(mask==255)
     (x1, y1) = (np.min(x), np.min(y))
     (x2, y2) = (np.max(x), np.max(y))
     cropped_image = gray[x1:x2+1, y1:y2+1]
```

```
[12] plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f6e12b8db50>
```

```
[13] # PENERAPAN LIBRARY EASYOCR
     reader = easyocr.Reader(['en'])
     result = reader.readtext(cropped_image)
     result
```

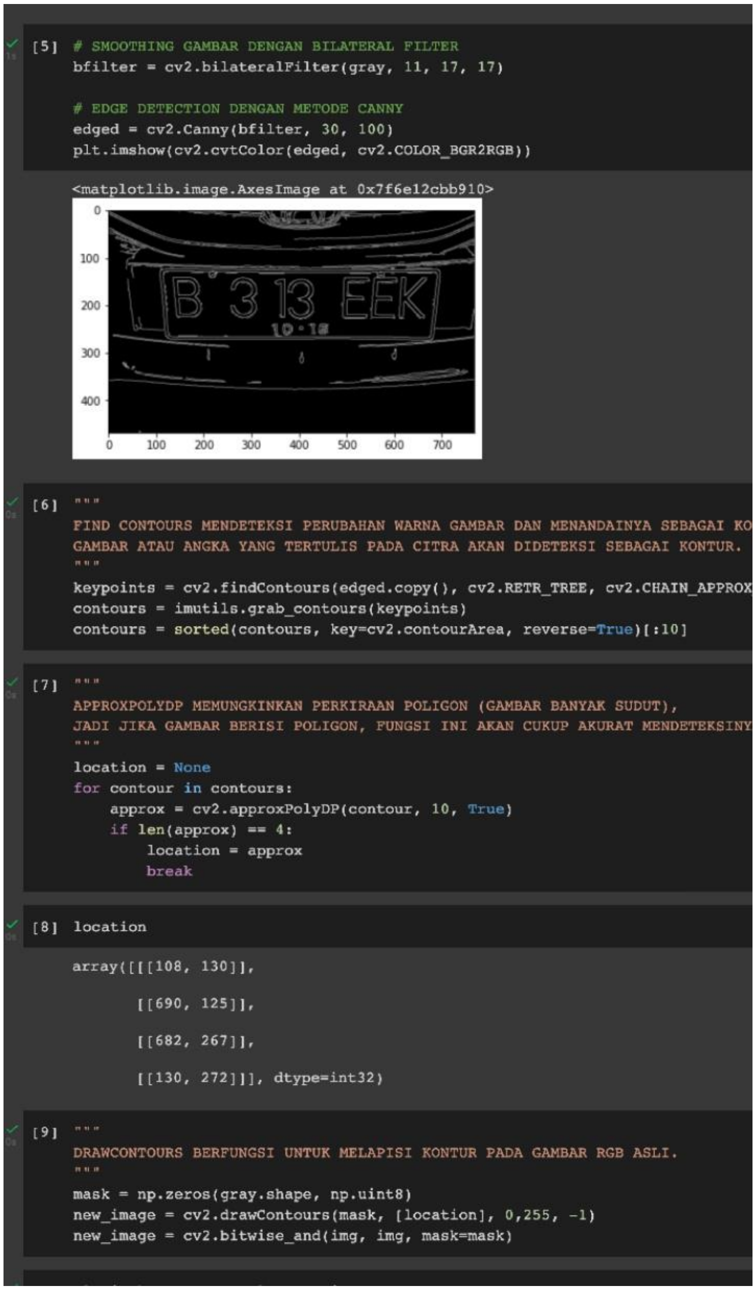
```
CUDA not available - defaulting to CPU. Note: This module is much faster
Downloading detection model, please wait. This may take several minutes d
Downloading recognition model, please wait. This may take several minutes
/usr/local/lib/python3.7/dist-packages/torch/nn/functional.py:718: UserWa
  return torch.max_pool2d(input, kernel_size, stride, padding, dilation,
[([[19, 5], [335, 5], [335, 113], [19, 113]], 'B 3 13', 0.851666804326082
 ([[373, 3], [573, 3], [573, 117], [373, 117]], 'EEK', 0.999237314615215
 ([[232, 110], [358, 110], [358, 142], [232, 142]],
  "10 ' 16",
  0.8975757654362415)]
```

```
[14] """
     DARI KONVERSI OCR TERDAPAT 3 LIST (DALAM BAHASA PEMROGRAMAN LAIN DISEBUT
     UNTUK MENGELIMINASI BILANGAN YANG TIDAK PENTING DAN HANYA MENGAMBIL VALUE
     TULIS KODE SEPERTI DIBAWAH.

     PENULISAN KODE DIBAWAH DISESUAIKAN DENGAN HASIL OCR,
     APABILA 2 LIST, HANYA TULISKAN SAMPAI LIST KEDUA,
     APABILA 3 LIST, HANYA TULISKAN SAMPAI LIST KETIGA,
     """
     text1 = str.upper(result[0][1])
     text2 = str.upper(result[1][1])
     text3 = str.upper(result[2][1])
```

```
     # TAMPILKAN ISI VARIABEL TEXT
     text1, text2, text3
```

```
('B 3 13', 'EEK', "10 ' 16")
```



```
[5] # SMOOTHING GAMBAR DENGAN BILATERAL FILTER
    bfilter = cv2.bilateralFilter(gray, 11, 17, 17)

    # EDGE DETECTION DENGAN METODE CANNY
    edged = cv2.Canny(bfilter, 30, 100)
    plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))

    <matplotlib.image.AxesImage at 0x7f6e12cbb910>
```

```
[6] """
    FIND CONTOURS MENDETEKSI PERUBAHAN WARNA GAMBAR DAN MENANDAINYA SEBAGAI KO
    GAMBAR ATAU ANGKA YANG TERTULIS PADA CITRA AKAN DIDETEKSI SEBAGAI KONTUR.
    """
    keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX
    contours = imutils.grab_contours(keypoints)
    contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

```
[7] """
    APPROXPOLYDP MEMUNGKINKAN PERKIRAAN POLIGON (GAMBAR BANYAK SUDUT),
    JADI JIKA GAMBAR BERISI POLIGON, FUNGSI INI AKAN CUKUP AKURAT MENDETEKSINY
    """
    location = None
    for contour in contours:
        approx = cv2.approxPolyDP(contour, 10, True)
        if len(approx) == 4:
            location = approx
            break
```

```
[8] location

    array([[[108, 130]],

           [[690, 125]],

           [[682, 267]],

           [[130, 272]]], dtype=int32)
```

```
[9] """
    DRAWCONTOURS BERFUNGSI UNTUK MELAPISI KONTUR PADA GAMBAR RGB ASLI.
    """
    mask = np.zeros(gray.shape, np.uint8)
    new_image = cv2.drawContours(mask, [location], 0,255, -1)
    new_image = cv2.bitwise_and(img, img, mask=mask)
```