Pengenalan Pola Sidik Jari

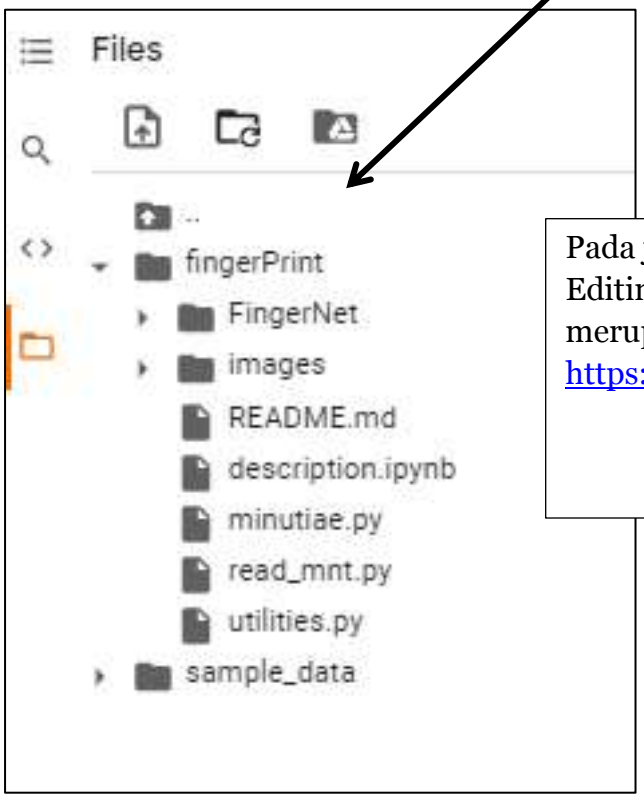
(File di cloning dari Github: https://github.com/skconan/fingerPrint.git)

Silakan Saudara ketikkan source code di bawah ini (meskipun nomor baris tidak urut, tidak apa-apa)

Import / Cloning dari Github

```
[ ]   1 !git clone https://github.com/skconan/fingerPrint.git
```

```
Cloning into 'fingerPrint'...
remote: Enumerating objects: 44, done.
remote: Total 44 (delta 0), reused 0 (delta 0), pack-reused 44
Unpacking objects: 100% (44/44), done.
```



Pada jendela sebelah kiri dari Source Code Editing akan terlihat folder fingerPrint yang merupakan hasil cloning dari github https://github.com/skconan/fingerPrint.git

Mengakses folder fingerPrint

```
[ ]  1 import sys
     2 sys.path.append('/content/fingerPrint')
```

```
[ ]  1 import cv2 as cv
     2 from google.colab.patches import cv2_imshow
```

```
[ ]  1 print(cv.__version__)
```

```
4.1.2
```

```
[ ]  1 import numpy as np
     2 import math
     3 from operator import itemgetter
     4 import urllib
     5 import urllib.request
     6 from google.colab import files
     7 import matplotlib.pyplot as plt
```

```
[ ]
degree_range = 30
radian_range = math.radians(degree_range)
radius_range = 50
```

```
[ ]
def distance(p1,p2,mode="L2"):
    p1 = np.array(p1)
    p2 = np.array(p2)
    if mode == "L1":
        return np.linalg.norm(p1-p2,1)
    elif mode == "L2":
        return np.linalg.norm(p1-p2)

def implot(rgb):
    bgr = cv.cvtColor(rgb, cv.COLOR_BGR2RGB)
  #     plt.imshow(bgr)
#     plt.show()
    cv2_imshow(rgb)

def imread(url):
    data = urllib.request.urlopen(url)
    data = data.read()

    arr = np.fromstring(data, np.uint8)
    return cv.imdecode(arr, 1)
```

```
[ ]
def read_mnt(file_path):
    data = open(file_path)
    data = data.read()
    lines = data.split("\n")
   # print(lines)
    result = []
    for line in lines[2:]:
        data = line.split(" ")
        if len(data) < 3:
          continue
        data_dict = {}
        data_dict['x'] = int(data[0])
        data_dict['y'] = int(data[1])
        data_dict['rad'] = float(data[2])
        result.append(data_dict)
    print(result[:5])
    return result
```

```
[ ]
def draw_mnt(img, mnt_point):
    font = cv.FONT_HERSHEY_SIMPLEX
    for i, p in enumerate(mnt_point):
        center = (p['x'], p['y'])
        radian = p['rad']

        r = 6
        color = [255, 0, 0]
        border = 1

        x_new = p['x'] + int(r*2 * math.cos(radian))
        y_new = p['y'] + int(r*2 * math.sin(radian))

        cv.circle(img, center, r, color, border)
        cv.line(img, center, (x_new, y_new), color, border)
        cv.putText(img, str(i), center, font, 0.25, (0, 0, 0), 1, cv.LINE_AA)
    return img
```

```
[ ]
def group_minutiae(mnt_point, mnt_img):
    """
            Group mintiae by radius and radian
        """


    r = 6
    color = [255, 0, 0]
    border = -1

    length_mnt = len(mnt_point)
    mnt_group = []

    for i in range(length_mnt):
        mnt = {}
        mnt['index'] = i
        mnt['child'] = []

        xi = mnt_point[i]['x']
        yi = mnt_point[i]['y']
        radi = mnt_point[i]['rad']
        pi = [xi, yi]

        result = mnt_img.copy()

```

```
        for j in range(length_mnt):
            if i == j:
                continue

            xj = mnt_point[j]['x']
            yj = mnt_point[j]['y']
            radj = mnt_point[j]['rad']

            pj = [xj, yj]

            # print(i, j, distance(pi, pj), distance([radi], [radj], "L1"))

            if distance(pi, pj) <= radius_range and distance([radi], [radj], "L1") <= radian_range:
                mnt['child'].append(j)
                cv.circle(result, tuple(pj), r, color, border)

            # cv.imshow("Group"+str(i), result)

        mnt['density'] = len(mnt['child'])
        mnt_group.append(mnt)

    mnt_group = sorted(mnt_group, key=itemgetter('density'),reverse=True)
    return mnt_group
```

```
[84]  1 mnt_path = "/content/fingerPrint/FingerNet/Minutiae/F0024301.mnt"
      2
      3 mnt_point = read_mnt(mnt_path)
      4 img = cv.imread("/content/fingerPrint/images/F0024301.png")
      5 implot(img)
```

```
[82]  1 img_with_mnt = draw_mnt(img.copy(), mnt_point)
      2 implot(img_with_mnt)
```

```
[83]  1 mnt_group = group_minutiae(mnt_point, img_with_mnt)
      2
      3 for mnt in mnt_group[:10]:
      4   result = img_with_mnt.copy()
      5   i = mnt['index']
      6   child = mnt['child']
      7   mnt = mnt_point[i]
      8   r = 6
      9   p = [mnt['x'],mnt['y']]
     10   cv.circle(result, tuple(p), r, (0,0, 255), -1)
     11   cv.circle(result, tuple(p), radius_range, (0,0, 255), 1)
     12
     13   for i in child:
     14     mnt = mnt_point[i]
     15     p = [mnt['x'],mnt['y']]
     16     cv.circle(result, tuple(p), r, (255, 255, 0), -1)
     17
     18   implot(result)
     19
```

Result