

LAPORAN

FINAL PROJECT PRAKTIKUM PENGOLAHAN CITRA

FACE RECOGNITION DENGAN METODE KNN

DAN LIBRARY FACE_RECOGNITION

DOSEN PENGAMPU : HERO YUDO MARTONO



Nama : Diana Fitri

NRP : 2103181005

2 D3 Teknik Informatika A

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

SURABAYA

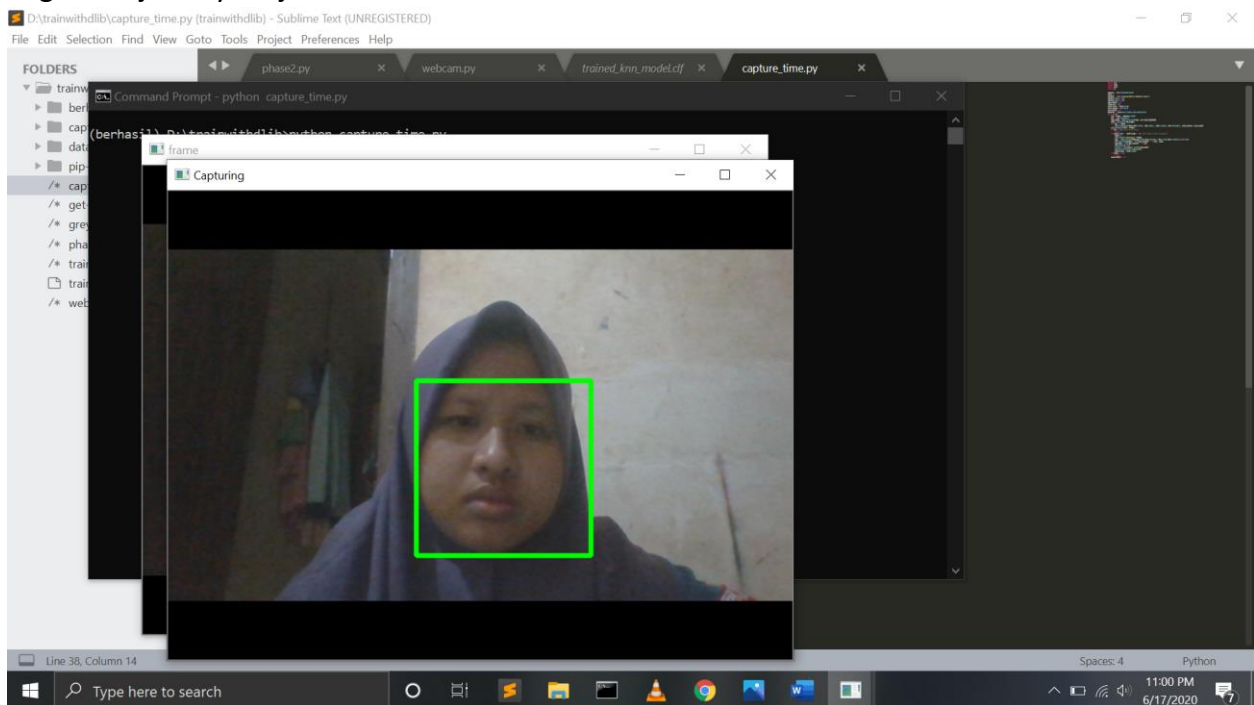
2019

FASE TESTING 01

Untuk fase testing 01 yang pertama kita akan membuat waktu capture untuk program yaitu mengcapture selang waktu 5 detik. Saya definisikan dengan code berikut yang ada di file source code saya yang bernama capture_time.py :

```
if time.time() - start_time >= 5: #kalo udah 3 detik dicapture
    total+=1
    cv2.imshow("Capturing", frame)
    img_ = cv2.resize(frame[det.top():det.bottom(), det.left():det.right()],(100,100))
    cv2.imwrite(newpath + str(img_counter) + ".jpg", img_)
    img_name = str(img_counter) + ".jpg"
    cv2.waitKey(1650)
    print("{} disimpan!".format(img_name))
    cv2.destroyAllWindows('Capturing')
    start_time = time.time()
if total == 3 :
    break
img_counter += 1
```

Dan hasilnya adalah sebagai berikut, wajah saya akan terdeteksi dengan bantuan Dlib face detection lalu akan dicapture dan disimpan di folder dataset/test yang mana disimpan hanya bagian wajah saya saja.



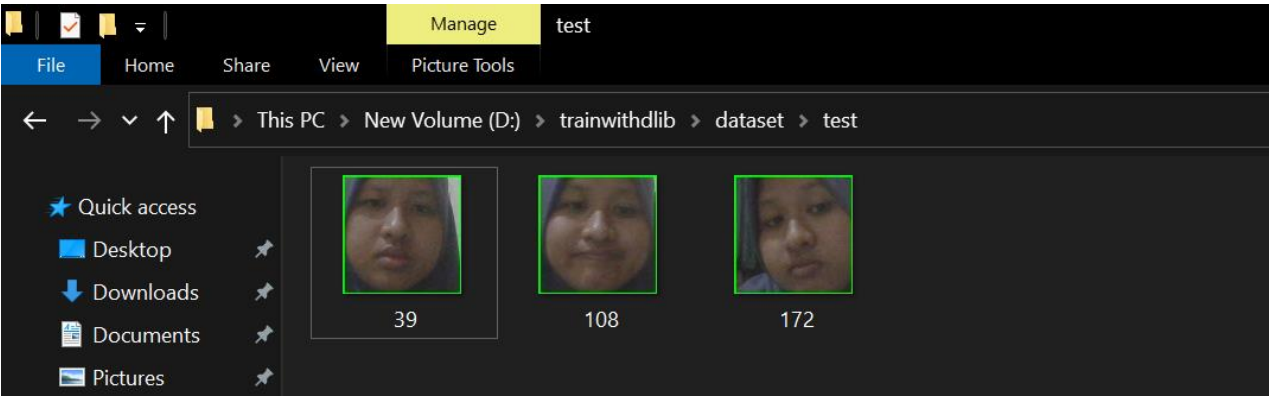
Nah setelah itu setelah 3 kali pengambilan atau dalam waktu 15 detik maka pengambilan gambar akan selesai

```
Command Prompt

(berhasil) D:\trainwithdlib>python capture_time.py
39.jpg disimpan!
108.jpg disimpan!
172.jpg disimpan!
[ WARN:1] global C:\projects\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (674) SourceReaderCB::~SourceReaderC
terminating async callback

(berhasil) D:\trainwithdlib>
```

Kemudian gambar akan tersimpan di dalam folder dataset/test dengan format .jpg



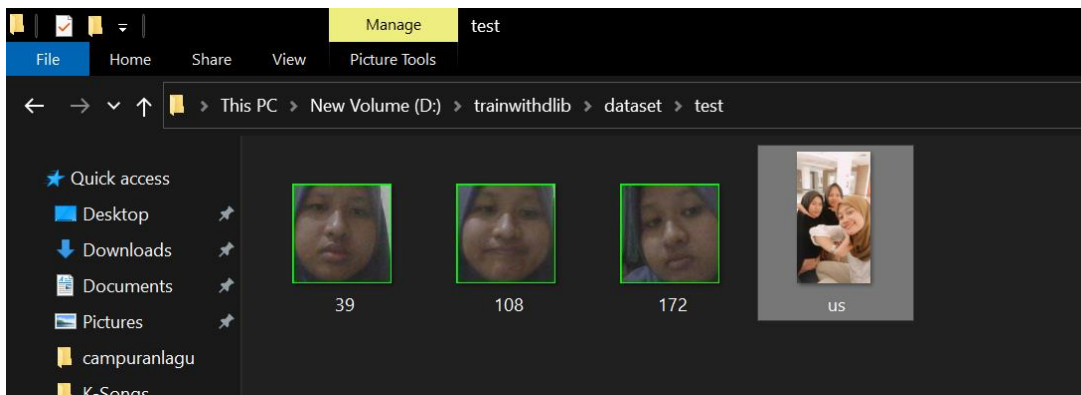
FASE TESTING 02

Pertama-tama kita akan mendeteksi apakah terdapat wajah di foto yang sudah kita dapatkan tadi, lalu bandingkan dengan fitur-fitur yang sudah diekstrak dari file-file di dataset dan juga kita beri label sesuai pengklasifikasian untuk gambar-gambar tersebut. Dan hasilnya adalah sebagai berikut dimana label NRP 2103181005 terdeteksi di gambar yang digunakan sebagai test.

```
(berhasil) D:\trainwithdlib>python tahap2.py
Proses Pembandingan ...
Pengenalan Selesai!
Mendeteksi wajah di 108.jpg
Mendeteksi wajah di 172.jpg
Mendeteksi wajah di 39.jpg
- Ketemu 2103181005 di (0, 7)

(berhasil) D:\trainwithdlib>
```

Kemudian kita coba dengan foto lain yang orangnya lebih dari satu yang mana wajah-wajahnya akan dikenali dari ekstraksi fitur dan labeling.



Dan hasilnya adalah yang terdeteksi adalah terdapat 3 wajah di dalam foto tersebut dengan label berupa NRP 2103181019, 2103181005 dan 2103181009

```
(berhasil) D:\trainwithdlib>python tahap2.py
Proses Pembandingan ...
Pengenalan Selesai!
Mendeteksi wajah di 108.jpg
Mendeteksi wajah di 172.jpg
Mendeteksi wajah di 39.jpg
- Ketemu 2103181005 di (0, 7)
Mendeteksi wajah di us.jpg
- Ketemu 2103181019 di (115, 460)
- Ketemu 2103181005 di (296, 397)
- Ketemu 2103181009 di (349, 528)

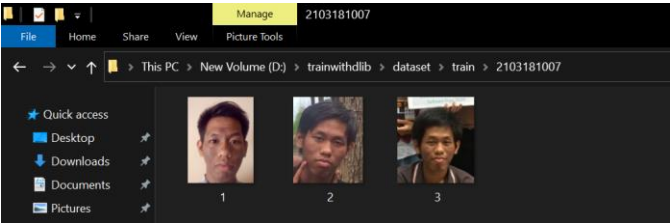
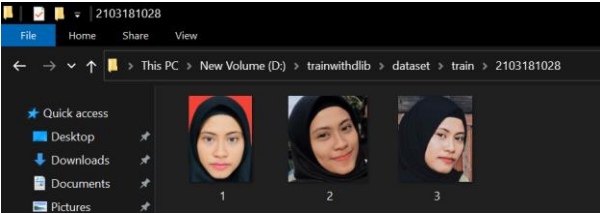
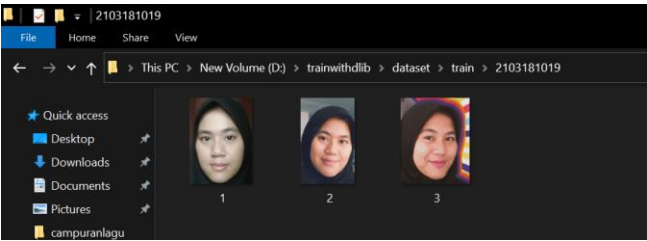
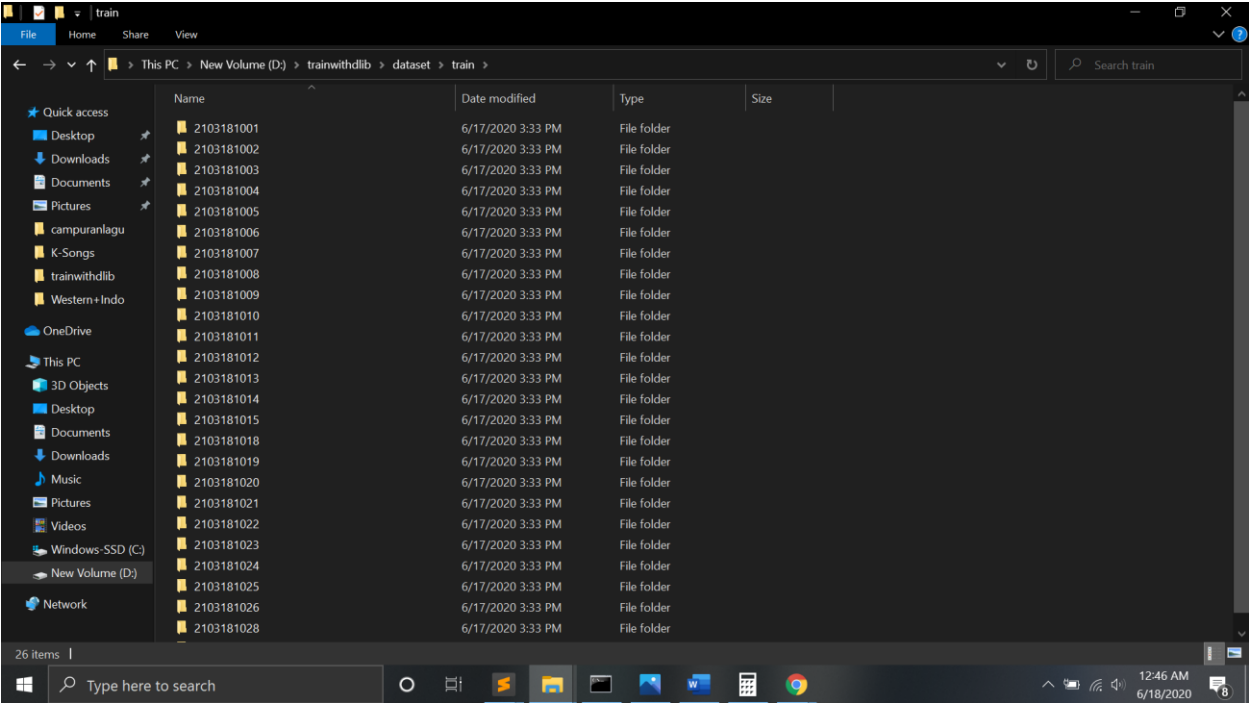
(berhasil) D:\trainwithdlib>
```

Dan hasilnya pun juga tertera di gambar dengan label sesuai dengan wajah masing-masing

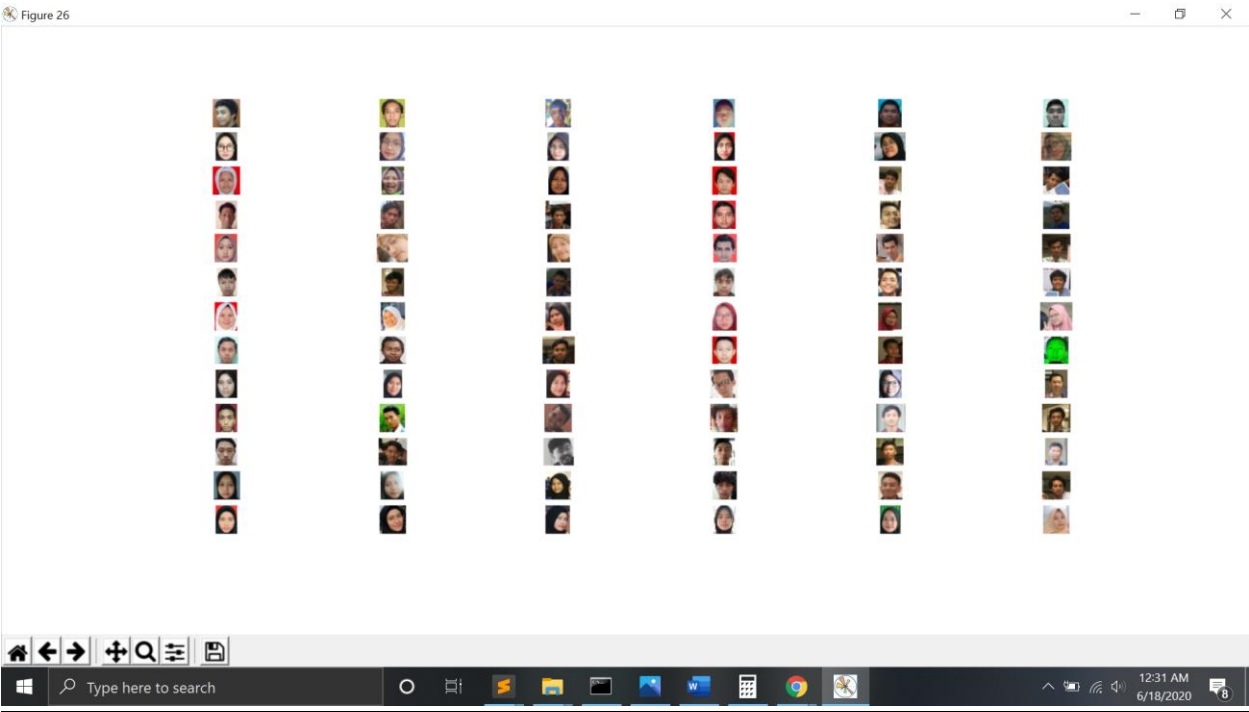


FASE TRAINING

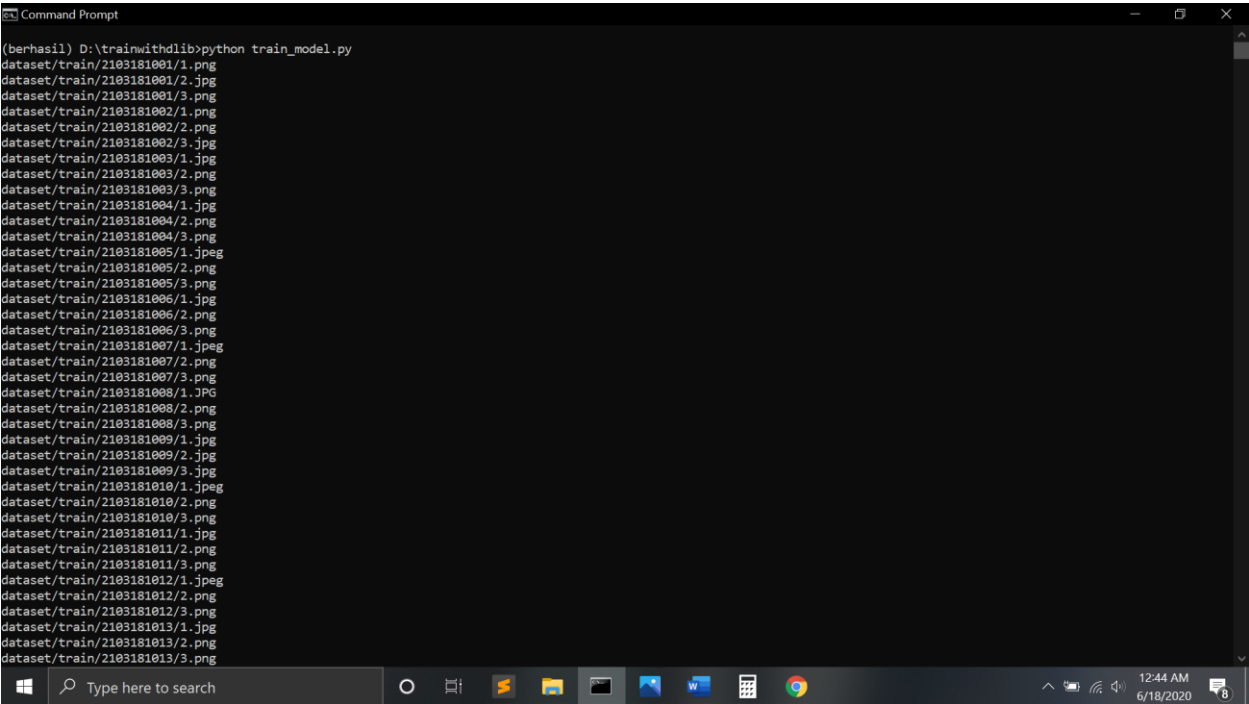
Yang terakhir adalah fase training. Dimana disini akan disimpan ekstraksi dari tiap gambar dan juga pelabelan untuk wajah yang ada di tiap subfolder dari train dataset. Berikut adalah dataset untuk train yang jumlahnya adalah sebanyak 26 folder atau dapat diartikan sebagai 26 orang yang mana tiap folder terdapat setidaknya 3 gambar untuk proses training. Sebenarnya akan lebih akurat jika banyak gambar yang digunakan namun karena keterbatasan data maka saya hanya memakai 3 gambar saja per folder.



Lalu ketika semua gambar ditampilkan akan sebanyak total 78 gambar seperti gambar berikut.



Dan ini adalah proses dari load semua gambar untuk ditampilkan dan ditrain



```
Command Prompt
dataset/train/2103181013/3.png
dataset/train/2103181014/1.jpg
dataset/train/2103181014/2.png
dataset/train/2103181014/3.png
dataset/train/2103181015/1.jpg
dataset/train/2103181015/2.png
dataset/train/2103181015/3.png
dataset/train/2103181018/1.jpg
dataset/train/2103181018/2.png
dataset/train/2103181018/3.png
dataset/train/2103181019/1.jpeg
dataset/train/2103181019/2.jpg
dataset/train/2103181019/3.jpg
dataset/train/2103181020/1.jpg
dataset/train/2103181020/2.png
dataset/train/2103181020/3.png
dataset/train/2103181021/1.jpg
dataset/train/2103181021/2.png
dataset/train/2103181021/3.png
dataset/train/2103181022/1.jpg
dataset/train/2103181022/2.png
dataset/train/2103181022/3.png
dataset/train/2103181023/1.jpg
dataset/train/2103181023/2.png
dataset/train/2103181023/3.png
dataset/train/2103181024/1.jpg
dataset/train/2103181024/2.png
dataset/train/2103181024/3.png
dataset/train/2103181025/1.jpeg
dataset/train/2103181025/2.png
dataset/train/2103181025/3.png
dataset/train/2103181026/1.jpg
dataset/train/2103181026/2.png
dataset/train/2103181026/3.png
dataset/train/2103181029/1.jpg
dataset/train/2103181029/2.png
dataset/train/2103181029/3.png
dataset/train/2103181030/1.jpg
dataset/train/2103181030/2.png
dataset/train/2103181030/3.png
Training dataset ...
```

Lalu kemudian akan dilakukan deteksi wajah dan ekstraksi fitur serta pelabelan untuk semua gambar seperti gambar berikut.

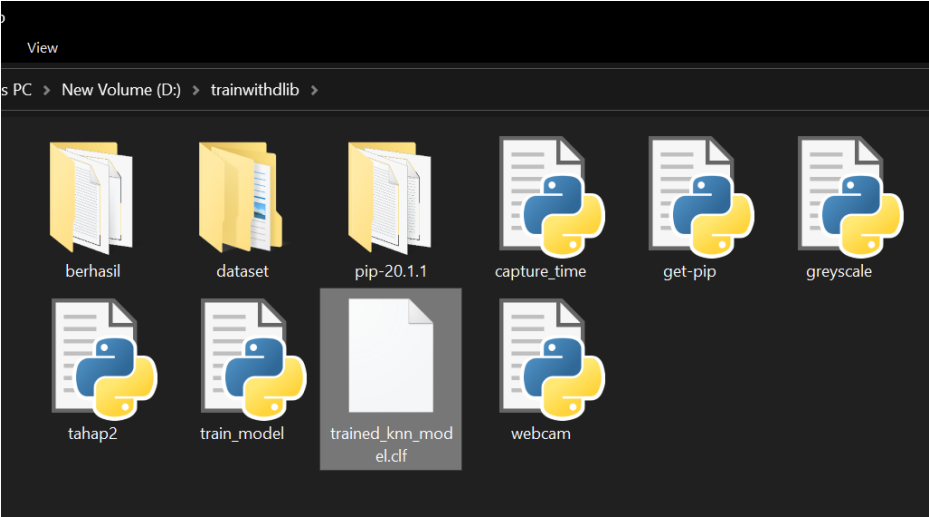
```
Command Prompt
Training dataset ...
Wajah terdeteksi untuk NRP : 2103181001
Wajah terdeteksi untuk NRP : 2103181001
Wajah terdeteksi untuk NRP : 2103181001
Wajah terdeteksi untuk NRP : 2103181002
Wajah terdeteksi untuk NRP : 2103181002
Wajah terdeteksi untuk NRP : 2103181002
Wajah terdeteksi untuk NRP : 2103181003
Wajah terdeteksi untuk NRP : 2103181003
Wajah terdeteksi untuk NRP : 2103181003
Wajah terdeteksi untuk NRP : 2103181004
Wajah terdeteksi untuk NRP : 2103181004
Wajah terdeteksi untuk NRP : 2103181004
Wajah terdeteksi untuk NRP : 2103181005
Wajah terdeteksi untuk NRP : 2103181005
Wajah terdeteksi untuk NRP : 2103181005
Wajah terdeteksi untuk NRP : 2103181006
Wajah terdeteksi untuk NRP : 2103181006
Wajah terdeteksi untuk NRP : 2103181006
Wajah terdeteksi untuk NRP : 2103181007
Wajah terdeteksi untuk NRP : 2103181007
Wajah terdeteksi untuk NRP : 2103181007
Wajah terdeteksi untuk NRP : 2103181008
Wajah terdeteksi untuk NRP : 2103181008
Wajah terdeteksi untuk NRP : 2103181008
Wajah terdeteksi untuk NRP : 2103181009
Wajah terdeteksi untuk NRP : 2103181009
Wajah terdeteksi untuk NRP : 2103181010
Wajah terdeteksi untuk NRP : 2103181010
Wajah terdeteksi untuk NRP : 2103181010
Wajah terdeteksi untuk NRP : 2103181011
Wajah terdeteksi untuk NRP : 2103181011
Wajah terdeteksi untuk NRP : 2103181011
Wajah terdeteksi untuk NRP : 2103181012
Wajah terdeteksi untuk NRP : 2103181012
Wajah terdeteksi untuk NRP : 2103181012
Wajah terdeteksi untuk NRP : 2103181013
Wajah terdeteksi untuk NRP : 2103181013
Wajah terdeteksi untuk NRP : 2103181013
Wajah terdeteksi untuk NRP : 2103181014
Wajah terdeteksi untuk NRP : 2103181014
```

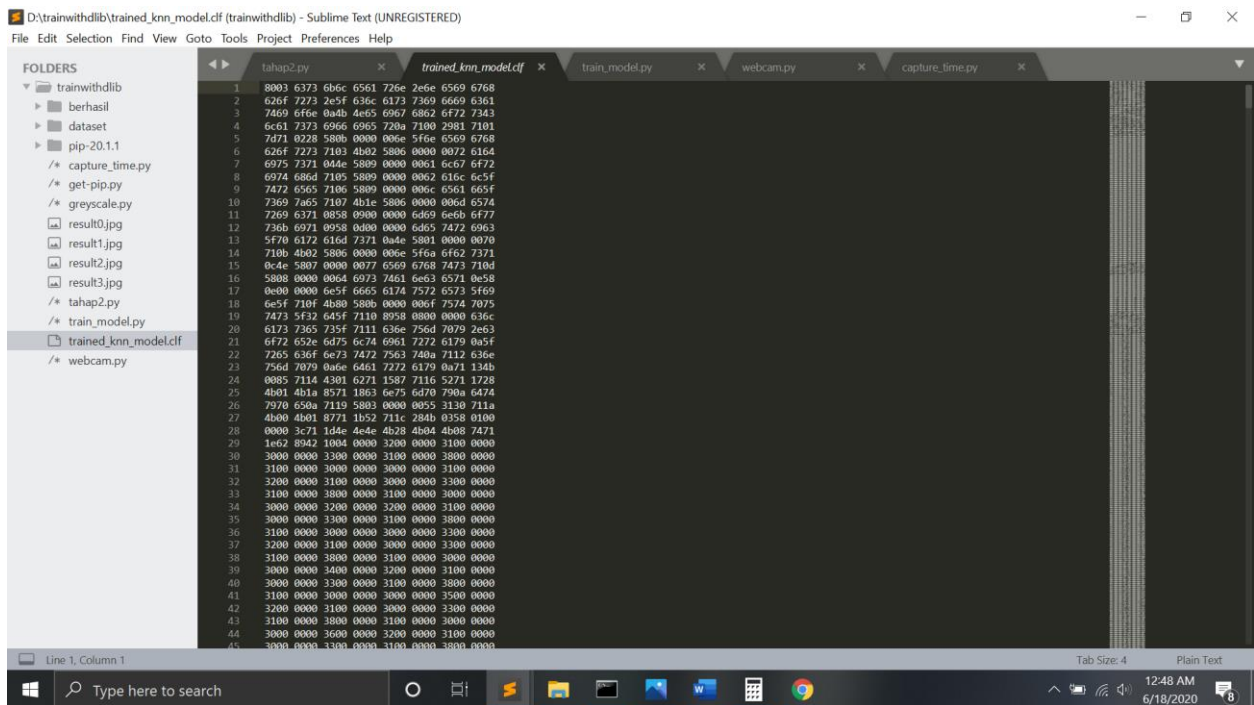


```
Command Prompt
Wajah terdeteksi untuk NRP : 2103181014
Wajah terdeteksi untuk NRP : 2103181014
Wajah terdeteksi untuk NRP : 2103181015
Wajah terdeteksi untuk NRP : 2103181015
Wajah terdeteksi untuk NRP : 2103181018
Wajah terdeteksi untuk NRP : 2103181018
Wajah terdeteksi untuk NRP : 2103181019
Wajah terdeteksi untuk NRP : 2103181019
Wajah terdeteksi untuk NRP : 2103181020
Wajah terdeteksi untuk NRP : 2103181020
Wajah terdeteksi untuk NRP : 2103181021
Wajah terdeteksi untuk NRP : 2103181021
Wajah terdeteksi untuk NRP : 2103181022
Wajah terdeteksi untuk NRP : 2103181022
Wajah terdeteksi untuk NRP : 2103181023
Wajah terdeteksi untuk NRP : 2103181023
Wajah terdeteksi untuk NRP : 2103181024
Wajah terdeteksi untuk NRP : 2103181024
Wajah terdeteksi untuk NRP : 2103181024
Wajah terdeteksi untuk NRP : 2103181025
Wajah terdeteksi untuk NRP : 2103181025
Wajah terdeteksi untuk NRP : 2103181026
Wajah terdeteksi untuk NRP : 2103181026
Wajah terdeteksi untuk NRP : 2103181026
Wajah terdeteksi untuk NRP : 2103181028
Wajah terdeteksi untuk NRP : 2103181028
Wajah terdeteksi untuk NRP : 2103181028
Wajah terdeteksi untuk NRP : 2103181030
Wajah terdeteksi untuk NRP : 2103181030
Training complete!

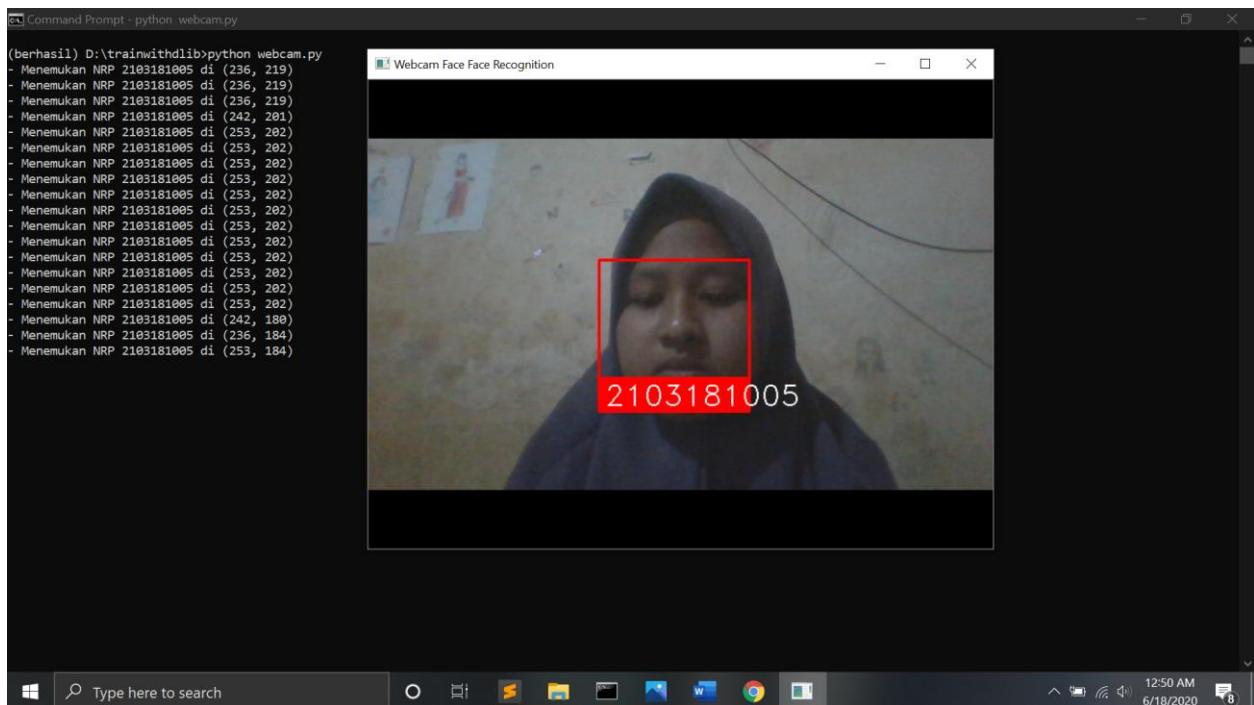
(berhasil) D:\trainwithdlib>
```

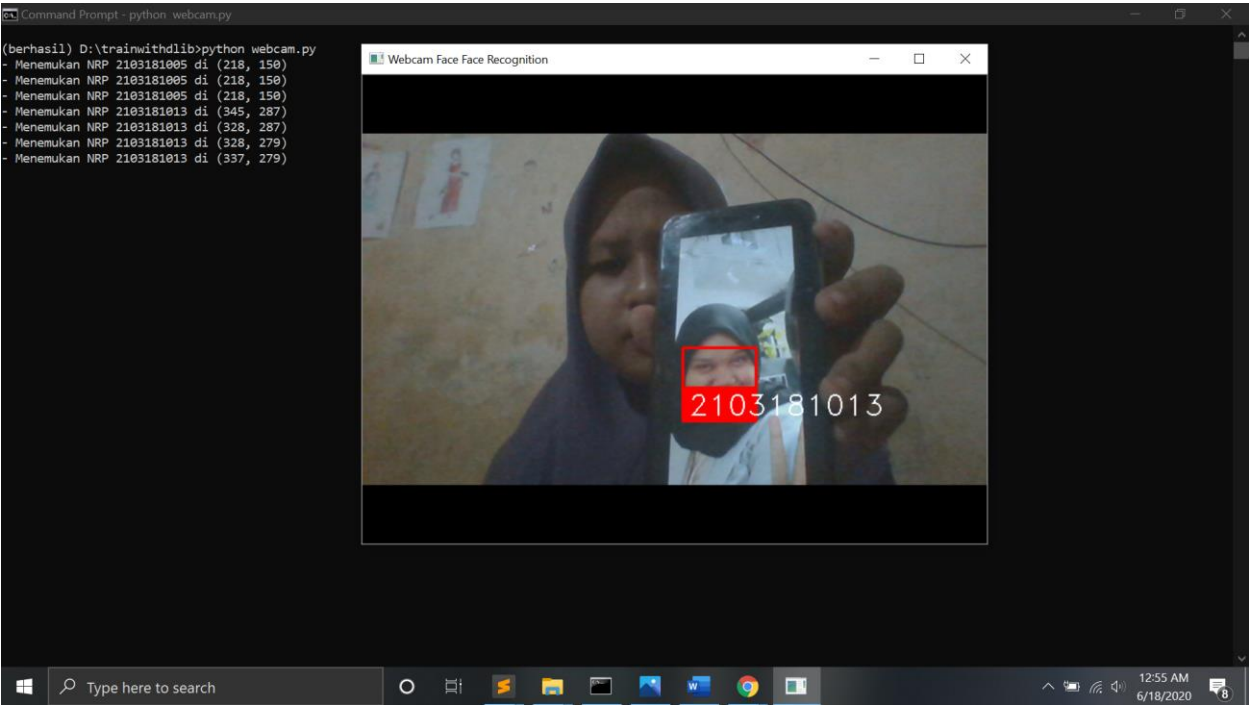
Nanti hasil ekstraksi dan pelabelan tersebut akan disimpan di file berekstensi .clf yang mana di dalamnya formatnya berupa numpy array.



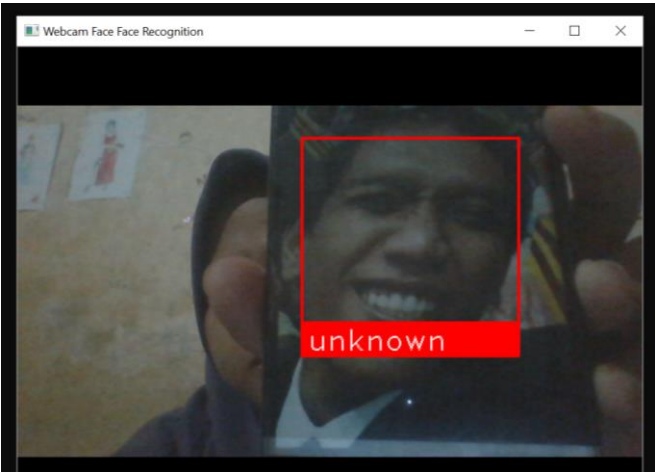


Lalu saya juga akan melakukan pengenalan wajah di webcam, dimana di webcam ditampilkan label sesuai dengan wajah yang dikenali. Dan di cmd tertulis wajah yang dikenali di frame dengan labelnya





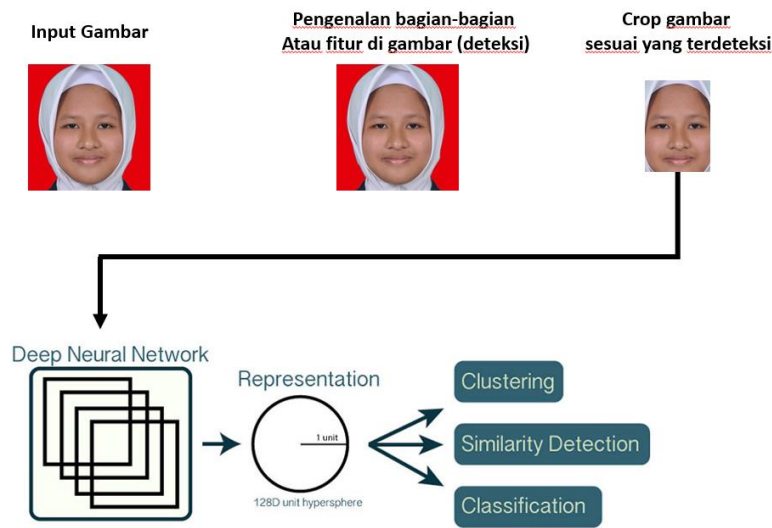
Lalu untuk wajah yang tidak terdapat di dalam dataset train maka tidak akan dikenali dan dilabeli dengan unknown atau tidak kenal wajah tersebut.



PENJELASAN METODE

1) Ekstraksi Fitur dengan Dlib (HOG) based module (face_recognition library)

Tahapan :



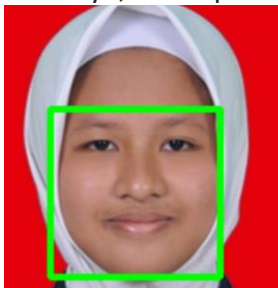
Penjelasan :

Langkah 1 : Memasukkan Gambar

- Kita dapat load gambar dengan fungsi bawaan library face_recognition yaitu load_image

Langkah 2: Bagaimana cara mendeteksi wajah?

- Pertama-tama, kita akan menghasilkan pola wajah berdasarkan algoritma HOG.
- Kami akan menemukan bagian dari gambar yang disederhanakan yang terlihat paling mirip dengan pola wajah HOG yang dikenal.
- Akhirnya, kotak pembatas ditarik di sekitar wajah yang terdeteksi.



Langkah 3 : Crop bagian wajah yang terdeteksi. Gambar wajah terpusat dilewatkan melalui jaringan saraf konvolusi yang dalam untuk mendapatkan 128 pengukuran yang merupakan 128 unit hypersphere dimensi.



Contoh Ekstraksi Fitur Gambar 1 :



```
Select Command Prompt
(berhasil) D:\trainwithdlib>python test.py
Gambar 1
: [-6.39770627e-02  5.13774604e-02  2.65656877e-03 -4.32051606e-02
-1.30075902e-01 -1.79791767e-02 -4.76131663e-02 -1.65215909e-01
 1.95848405e-01 -1.72331721e-01  1.84805706e-01 -6.88065365e-02
-1.50131136e-01 -8.41420144e-02 -1.05786789e-02  1.62740514e-01
-1.29565984e-01 -1.92972451e-01 -8.23013186e-02 -3.53694707e-02
 7.11165369e-03 -1.06752247e-01 -5.06719202e-03  4.34849337e-02
-9.65886712e-02 -3.65458071e-01 -6.97029382e-02 -1.18510135e-01
 2.18024477e-04 -8.06098655e-02 -2.35254727e-02  6.72791824e-02
-2.13678434e-01 -4.08974960e-02 -3.95615138e-02  6.19157329e-02
-3.63951735e-02 -8.28054175e-03  1.71294093e-01 -2.06870660e-02
-2.43288904e-01  3.24253412e-03  4.41590920e-02  1.67719215e-01
 1.50883958e-01  9.65268761e-02  4.24198732e-02 -1.33320376e-01
 6.46287948e-02 -1.54337376e-01  1.51827550e-02  1.16999194e-01
 3.37151662e-02  2.83284970e-02  2.85616852e-02 -1.31954804e-01
 1.98121089e-02  3.14108171e-02 -1.49258062e-01  5.52135073e-02
 1.32616654e-01 -8.19549263e-02 -1.32424459e-01 -7.03347325e-02
 3.15561295e-01  1.32548481e-01 -7.35233352e-02 -1.46751881e-01
 2.01364875e-01 -1.49408981e-01 -1.02174237e-01  1.91958882e-02
-8.13932642e-02 -1.96396887e-01 -2.43269995e-01  2.25936677e-02
 3.95019412e-01  1.54740348e-01 -1.90241382e-01  4.37595211e-02
-7.29277208e-02  3.86699103e-04  1.32210165e-01  1.07325599e-01
-1.84801668e-02  1.68418214e-02 -6.59161359e-02 -8.10240768e-03
 2.07862586e-01 -3.01655680e-02  9.49215237e-03  1.68350101e-01
-1.41009204e-02  3.66177298e-02 -2.61420999e-02 -1.75293311e-02
-8.33483487e-02  6.95055947e-02 -1.86543629e-01 -6.36711493e-02
 8.80475640e-02  3.33570428e-02  1.35253295e-02  6.12805337e-02
-1.32244244e-01  1.21894062e-01 -1.69082209e-02 -2.06194241e-02
 1.15252128e-02  4.23503965e-02 -1.01629846e-01 -6.75246865e-02
 1.29139259e-01 -2.17562169e-01  2.18126476e-01  1.67776570e-01
 9.67830643e-02  1.21060625e-01  1.50228560e-01  9.68206376e-02
 3.88659053e-02 -2.72835009e-02 -2.47991443e-01 -3.99903208e-02
 9.09908414e-02 -3.38508785e-02  6.74258620e-02  4.54427525e-02]
```

Contoh Ekstraksi Fitur Gambar 2 :



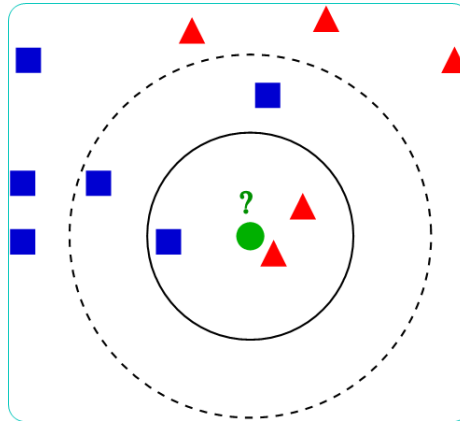
```
Gambar 2
: [-1.23455614e-01 5.29586971e-02 7.66732991e-02 -8.32161754e-02
-1.53302252e-01 4.30019461e-02 -1.03106804e-01 -8.96267593e-02
9.90389287e-02 -1.89064294e-01 1.46774203e-01 -7.14089209e-03
-1.80093631e-01 3.34508531e-02 -3.60667557e-02 1.61787897e-01
-1.35296941e-01 -9.04543847e-02 -3.30006517e-02 -6.23913389e-03
-2.33391998e-04 -5.27455145e-03 4.14091200e-02 7.45220706e-02
-1.17566824e-01 -3.82511735e-01 -9.59943458e-02 -8.77754316e-02
9.66743100e-03 -9.03999060e-03 -1.25709400e-01 3.05819083e-02
-1.21380188e-01 3.79191861e-02 2.08914317e-02 1.19418032e-01
-8.93696323e-02 -3.93170416e-02 1.77415401e-01 4.17363551e-03
-2.64324725e-01 -3.33140679e-02 4.01866287e-02 2.57282704e-01
2.14454383e-01 2.74008438e-02 6.32485151e-02 -4.54643853e-02
9.34552103e-02 -2.36420035e-01 5.62763326e-02 1.38507754e-01
4.68419939e-02 5.47205098e-02 1.24262106e-02 -1.02500990e-01
6.28463477e-02 9.35612619e-02 -8.42751339e-02 2.01489720e-02
8.21591690e-02 -1.23474352e-01 -3.26252077e-03 -8.19730666e-03
3.04538786e-01 1.27962023e-01 -1.25212505e-01 -1.59295902e-01
1.32635653e-01 -1.85660750e-01 -5.73672466e-02 -1.19383074e-02
-1.69896439e-01 -1.79091737e-01 -2.48387024e-01 2.02408973e-02
3.88218731e-01 2.10986659e-01 -1.33195549e-01 9.03011188e-02
-9.64886248e-02 -1.81018002e-02 1.21541828e-01 1.93303540e-01
-4.79592718e-02 4.17447947e-02 -4.47619297e-02 2.85374168e-02
2.34944671e-01 -9.80476439e-02 1.87646337e-02 2.08707035e-01
3.03352010e-02 4.24153320e-02 3.72453593e-02 5.58944559e-03
-1.16014212e-01 -7.35931424e-03 -1.38642609e-01 -2.92380936e-02
3.36383618e-02 -4.95419204e-02 4.90670316e-02 1.45295992e-01
-2.53316820e-01 1.86127961e-01 -5.67352995e-02 -2.99457554e-03
8.93648937e-02 2.87454966e-02 -1.12199858e-01 -1.25566781e-01
1.40353754e-01 -1.79474592e-01 1.26012817e-01 2.21762359e-01
1.00908726e-01 1.38993427e-01 1.75360441e-01 6.34850860e-02
4.34442870e-02 2.96002291e-02 -1.82754874e-01 -5.61263552e-03
1.05885357e-01 -5.36524393e-02 4.37784418e-02 1.98342763e-02]
```

Dapat kita lihat dari ekstrasi tadi bahwa ukuran dari fitur/dimensinya adalah 128 yang mana ini adalah ciri khas dari pengestrakan dengan library face_recognition yang menggunakan metode ekstraksi deep convolution neural network.

2) Penggunaan K-Nearest Neighbor untuk train dataset atau pelabelan

Setelah dilakukannya tahap pengekstraksian fitur pada data, maka tahap selanjutnya yang akan dilakukan adalah pengklasifikasian data. Pada penelitian ini metode klasifikasi yang digunakan adalah metode KNN (k-Nearest Neighbor). Penggunaan KNN dalam pengklasifikasian dalam metode pengenalan wajah karena training sangat cepat dan efektif untuk data training yang besar. Algoritma **k-Nearest Neighbor** adalah algoritma *supervised learning* dimana hasil dari instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori **k**-tetangga terdekat. Tujuan dari algoritma ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan *sample-sample* dari *training data*.

Algoritma *k-Nearest Neighbor* menggunakan *Neighborhood Classification* sebagai nilai prediksi dari nilai *instance* yang baru.



Lokasi dari gambar akan didapatkan dengan menggambar kotak pembatas di sekitar wajah dan menggunakan fungsi `face_encodings()` untuk mendapatkan 128 pengukuran.

```
face_bounding_boxes = face_recognition.face_locations(image)
face_encodings = face_recognition.face_encodings(image,
known_face_locations=face_bounding_boxes)
```

Gunakan model KNN untuk melatih classifier yang kemudian simpan model atau hasil train tersebut sehingga dapat menggunakannya tanpa train dataset lagi.

```
classifier = train("knn_examples/train",
model_save_path="trained_knn_model.clf", n_neighbors=2)
```

Jumlah tetangga terdekat adalah 2 yang berarti wajah yang tidak dikenal akan diuji berdasarkan berat dua tetangga terdekatnya. Jarak Euclidean digunakan dalam kasus ini.

```
closest_distances = knn_clf.kneighbors(faces_encodings, n_neighbors=2)
are_matches = [min(closest_distances[0][i][0],closest_distances[0][i][1]) <=
distance_threshold for i in range(len(X_face_locations))]
```

Lalu akan kita predict gambar yang ada di folder test tadi sesuai dengan model atau hasil dari dataset yang sudah kita train. Yang mana di dalam file tersebut terdapat hasil dari pelabelan dan juga pengolahan fitur dari gambar yang tadi sudah saya tunjukkan di bagian penjelasan fase training.

```
predictions = predict(full_file_path, model_path="trained_knn_model.clf")
```

