BAB I

PENDAHULUAN

A. Latar Belakang

Dalam industry 4.0, teknologi berkembang dengan sangat cepat. Hal ini juga berdampak pada industry kesehatan atau medis. Dengan menggunakan teknologi. Saat ini para dokter dapat mengetahui penyakit-penyakit dalam, yang 10-15 tahun yang lalu masih sulit untuk dideteksi. Pada *paper* kali ini, kami akan membahas tentang pemanfaatan teknologi biometrika pada bidang kesehatan khususnya pada mata.

Teknologi biometrik adalah teknologi yang memanfaatkan bagian tubuh yang unik dari manusia untuk mengidentifikasi individu sehingga dapat membedakan satu individu dengan individu lainnya. Dengan menggunakan teknologi biometrik ini kita dapat mendeteksi penyakit yang terjadi pada mata. Dalam kasus ini kami membuat teknologi yang dapat mendeteksi penyakit mata dengan mudah melalui teknologi iris recognition.

Menurut data Riskesdas Kemenkes RI tahun 2013, per setiap 1,000 jiwa ada 1 orang penderita baru katarak. Kemudian menurut situs https://hellosehat.com/pusat-kesehatan/gangguan-mata-dan-penglihatan/penyakit-mata-di-indonesia/, “penyakit mata glaucoma di Indonesia menyumbang angka kebutaan sebanyak 13.4% di Indonesia. Hal tersebut terjadi dikarenakan banyaknya orang yang tidak mengetahui gejala-gejala penyakit mata sehingga mengabaikannya.

Pendeteksi iris manusia merupakan bagian dari teknologi biometrik yang memanfaatkan bagian iris manusia untuk mengidentifikasi. Iris mata adalah daerah berbentuk gelang pada mata yang dibatasi oleh pupil dan sclera (bagian putih dari mata). Iris dapat digunakan untuk mengidentifikasi penyakit mata karena iris merupakan bagian yang unik dari manusia dan pasti berbeda-beda.

Aplikasi yang kami buat ini bertujuan supaya mempermudah banyak orang untuk mengecek mata ketika mengalami keanehan pada mata sehingga dapat langsung mengetahui penyakit mata yang diderita yang kemudian bisa langsung dilakukan pengobatan untuk mengurangi kemungkinan mengalami kebutaan.

Pada project *artificial intelligence* ini, penulis membuat aplikasi *iris recognition* menggunakan *python* untuk keperluan kesehatan, yaitu mendeteksi penyakit mata yang dialami menggunakan iris recognition.

B. Rumusan Masalah

Rumusan masalah dari latar belakang di atas adalah bagaimana membuat suatu aplikasi yang dapat mendeteksi penyakit mata manusia dengan menggunakan *artificial intelligence*.

C. Tujuan & Manfaat

Tujuan yang ingin dicapai dalam pembuatan *project* ini adalah berhasil membuat aplikasi yang dapat mendeteksi iris mata manusia dan memiliki tingkat akurasi tinggi dalam menentukan jenis penyakit yang dialami.

Manfaat dari *project* ini adalah penulis dapat mengenal teknologi *artificial intelligence* dan mengetahui cara kerja teknologi biometrika khususnya *iris recognition*. Manfaat lainnya adalah penulis berharap aplikasi yang akan dibuat nantinya dapat digunakan untuk mempermudah mendeteksi penyakit dari mata manusia.

BAB II

LANDASAN TEORI

A. Metodologi

1. Classifier: Machine Learning

Machine learning adalah pengaplikasian dari AI yang memungkinkan sistem untuk belajar secara otomatis dan berkembang dari pengalaman tanpa harus diprogram secara manual. Machine learning berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar.

2. Algoritma

a. Fast Library for Approximate Nearest Neighbors (FLANN)

FLANN adalah library untuk melakukan pencarian cepat dengan melihat titik terdekat dari space yang ada. FLANN memiliki banyak algoritma yang digunakan untuk mencari titik terdekat dan sistem akan otomatis memilih algoritma yang terbaik dan optimal bergantung pada dataset

b. Scale-Invariant Feature Transform (SIFT)

SIFT adalah singkatan dari Scale Invariant Feature Transform adalah interest point descriptor yang populer yang banyak digunakan karena scale and rotation invariant characteristics. SIFT diciptakan oleh David Lowe dari University British Columbia pada 2004

Singkatnya, ada tiga tujuan yang diharapkan akan dicapai dengan menggunakan SIFT:

1) Untuk mengekstraksi fitur distinctive invariant yang dapat dicocokkan dengan benar terhadap fitur database besar, memberikan dasar untuk object and scene recognition
2) Mengekstraksi fitur yang berbeda dengan skala dan rotasi gambar
3) Mengekstrak fitur yang tangguh terhadap distorsi affine, perubahan sudut pandang 3D, dan noise.

BAB III

PEMBAHASAN

A. Dataset

Total dataset yang kami gunakan di aplikasi '*Dr Eyes Care*' ini adalah 1800 yang terdiri dari 600 dataset untuk katarak, 250 dataset untuk glukoma, 100 dataset untuk konjungtif, 100 dataset untuk trauma, dan 750 dataset untuk mata yang sehat. Sumber dataset ini adalah dari *database* orang kedokteran dan dokumentasi eksperimen mahasiswa

B. Komputasi

Langkah-langkah aplikasi ini berjalan adalah sebagai berikut:

1. Mendeteksi mata
2. *Training* dataset
3. Menerima *input*
4. Membandingkan *input* yang diterima dengan dataset yang sudah di *train*

C. Implementasi

1. Code

```
def FLANN():
        connectdb = sqlite3.connect("results.db")
        cursor = connectdb.cursor()

        img1 = cv2.imread('1.png', cv2.IMREAD_GRAYSCALE)
        imageA = cv2.resize(img1, (450, 237))
        database = os.listdir("db")

        for image in database:
            try:
                img2 = cv2.imread("db/" + image)

                imgprocess = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)

                imageB = cv2.resize(imgprocess, (450, 237))

                matcheslist = ""

                # Inisialisasi SIFT detector
```

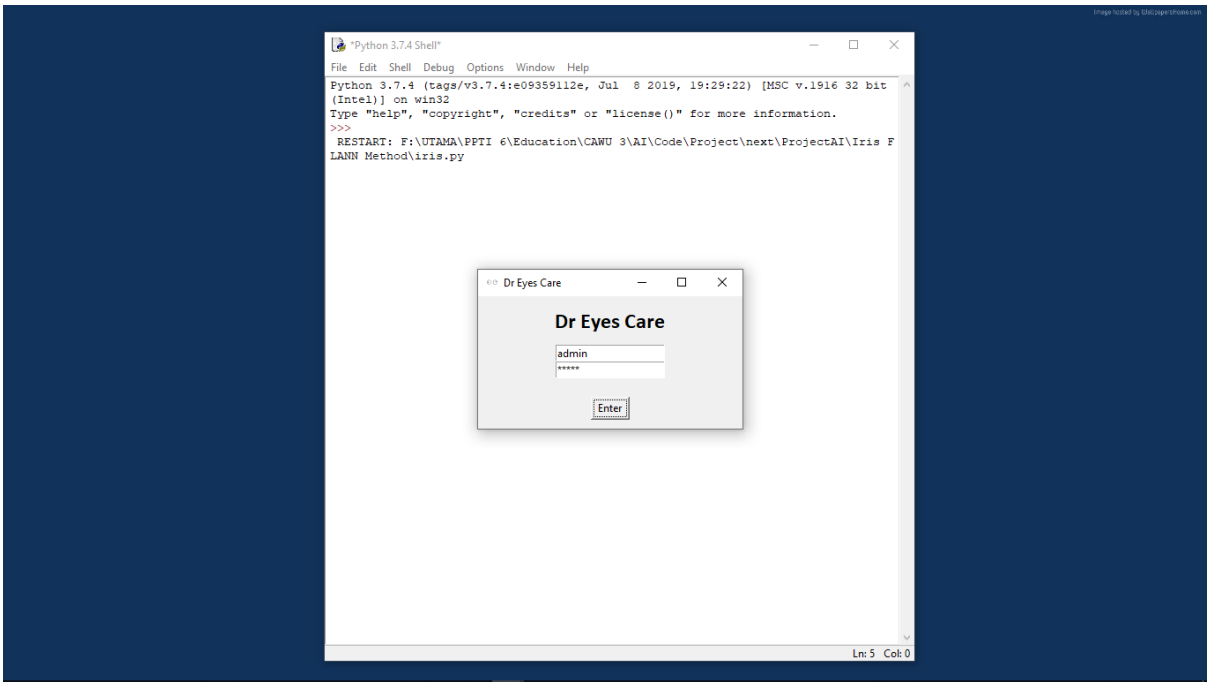
```
                    sift = cv2.xfeatures2d.SIFT_create()
                    # find the keypoints and descriptors with SIFT
                    kp1, des1 = sift.detectAndCompute(imageA, None)
                    kp2, des2 = sift.detectAndCompute(imageB, None)

                    # BFMatcher default params
                    bf = cv2.BFMatcher()
                    matches = bf.knnMatch(des1, des2, k=2)
                    # Apply rasio
                    good = []
                    for m, n in matches:
                        if m.distance < 0.75 * n.distance:
                            good.append([m])
                    # cv.drawMatchesKnn untuk list matches.

                    amount = len(good)
                    print('Comparing image with ' + image + " using FLANN method")

                    title = "Comparing"
                    fig = plt.figure(title)

                    cursor.execute("INSERT INTO flann (percentage, filename) VALUES (?,
?);", (amount, image))
                    connectdb.commit()

                except:
                    pass

            percentages = list(connectdb.cursor().execute("SELECT * FROM flann order by
percentage desc limit 10"))
            print(percentages[0])
            highest = percentages[0]

            # get number of matches
            highestperct = round(highest[0], 2)
            print(highestperct)

            # get file name of highest similarity
            filename = highest[1]
            print(filename)

            image1 = cv2.imread('1.png', cv2.IMREAD_GRAYSCALE)  # input image
            img1 = cv2.resize(image1, (450, 237))
            image2 = cv2.imread('db/' + filename, cv2.IMREAD_GRAYSCALE)  # closet image
```

```
            img2 = cv2.resize(image2, (450, 237))

            # Inisianilasi SIFT detector
            sift = cv2.xfeatures2d.SIFT_create()

            # find the keypoints and descriptors with SIFT
            keypoints1, destination1 = sift.detectAndCompute(img1, None)
            keypoints2, destination2 = sift.detectAndCompute(img2, None)

            # FLANN parameters
            FLANN_INDEX_KDTREE = 1
            index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
            search_params = dict(checks=50)   # or pass empty dictionary
            flann = cv2.FlannBasedMatcher(index_params, search_params)
            matches = flann.knnMatch(destination1, destination2, k=2)

            # Need to draw only good matches, so create a mask
            matchesMask = [[0, 0] for i in range(len(matches))]

            # ratio test as per Lowe's paper
            for i, (m, n) in enumerate(matches):
                if m.distance < 0.7*n.distance:
                    matchesMask[i] = [1,0]

            draw_params = dict(matchColor = (0, 255, 0),
                               singlePointColor = (255, 0, 0),
                               matchesMask = matchesMask,
                               flags = cv2.DrawMatchesFlags_DEFAULT)

            print(draw_params)
            print(len(matches))

            img3 = cv2.drawMatchesKnn(img1, keypoints1, img2, keypoints1, matches, None,
**draw_params)
            plt.imshow(img3)
            plt.suptitle("Amount of matches : " + str(highestperct) + "\n Similarity
Percentage : " + str(percent) + "%")
            disease = filename[:7]
            txt = " Result Scan: \n Best Matches: " + filename + "\n Status: " + disease
            plt.text(0.40, 0.20, txt, transform=fig.transFigure, size=11)
            plt.axis("off")

            plt.show()
```

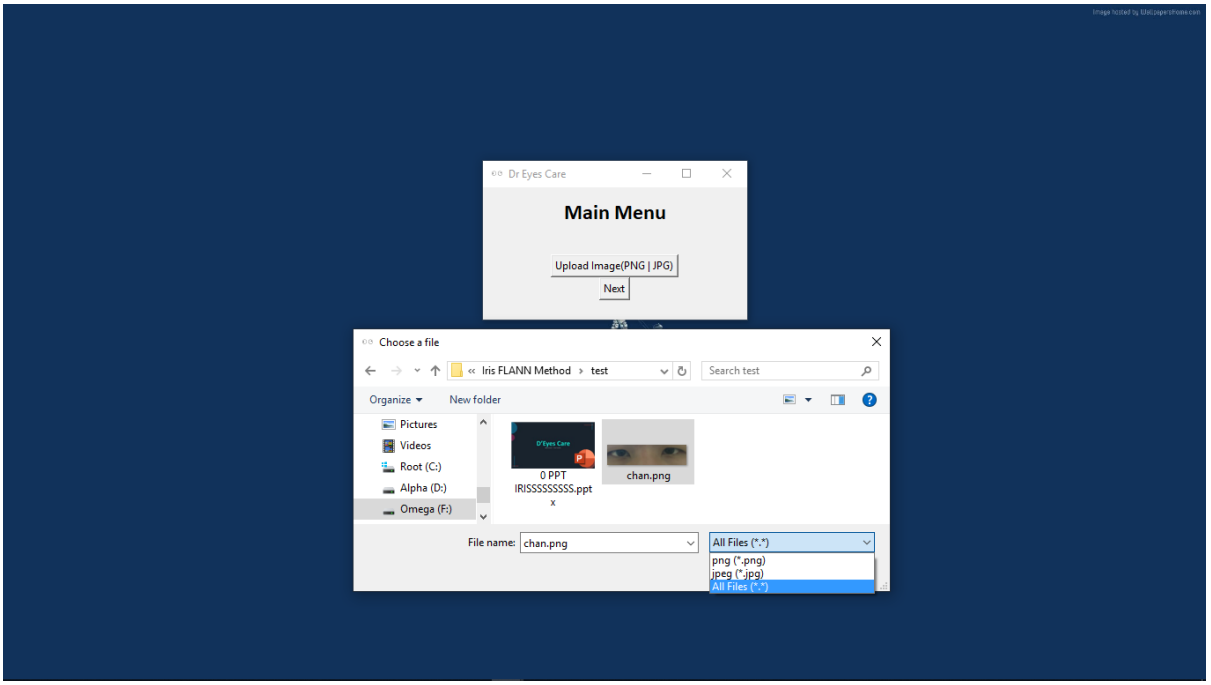
```
        cursor.execute("DELETE FROM flann")
        connectdb.commit()
```

2. Tampilan Aplikasi

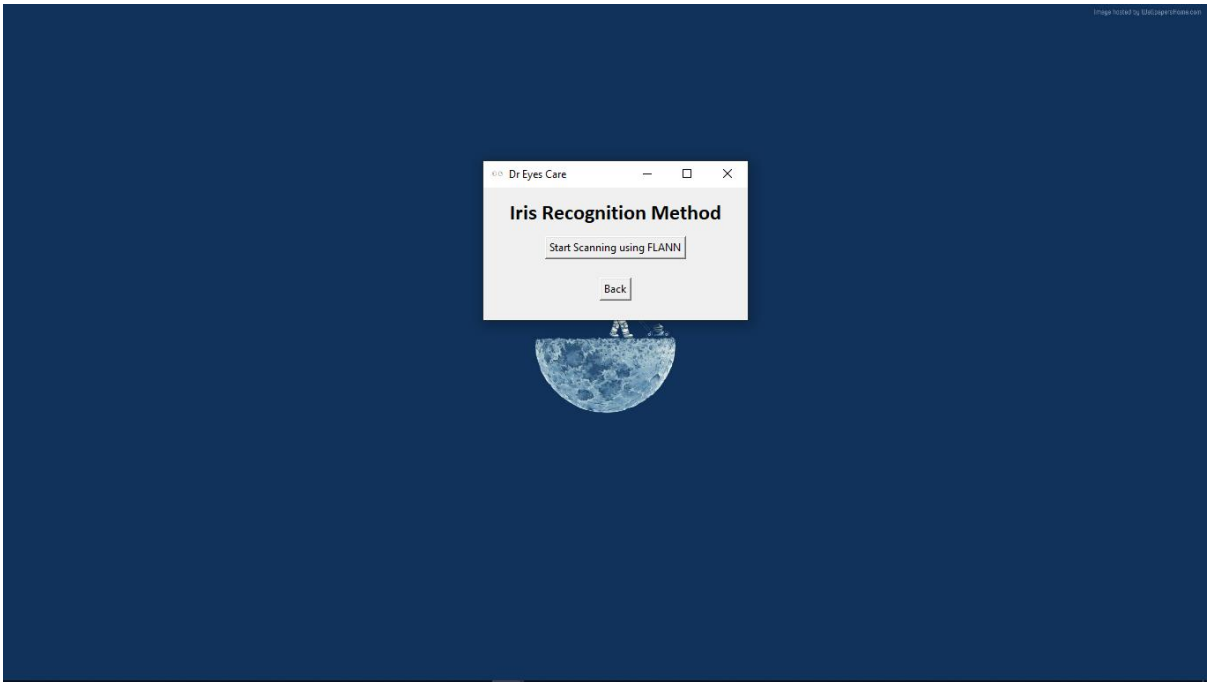
a. Tampilan *login*



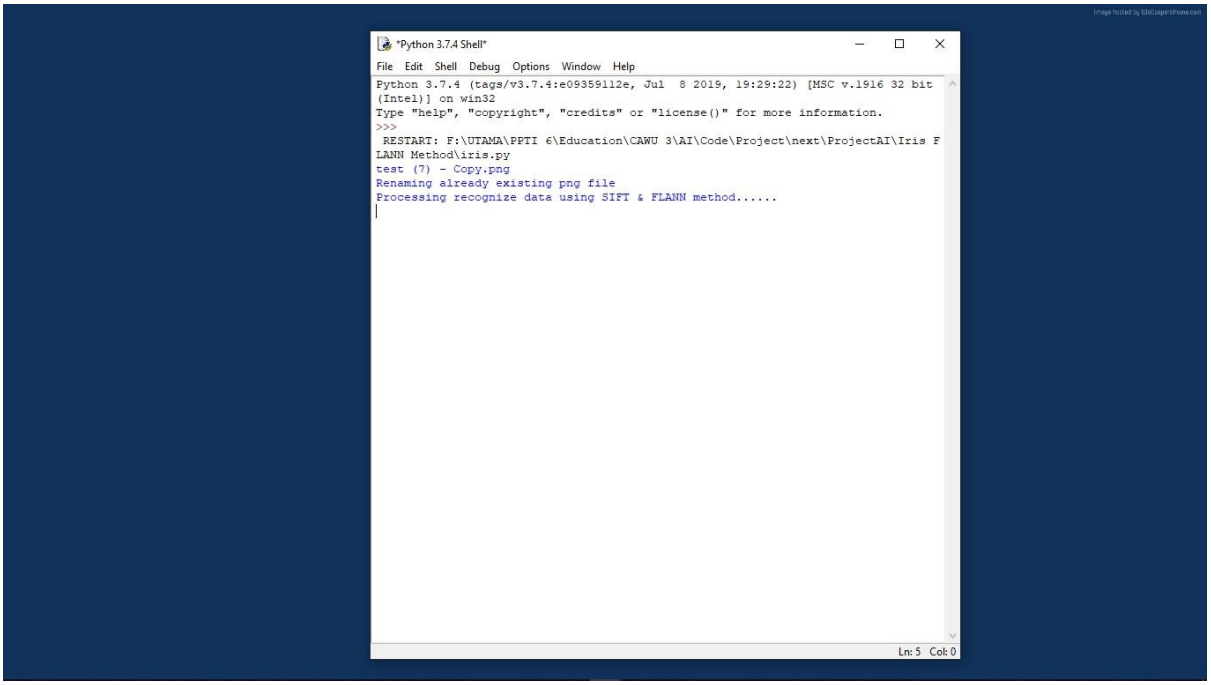
b. Tampilan *input* data atau *upload*



c. Tampilan untuk memulai *compare* data yang ingin dibandingkan dengan dataset

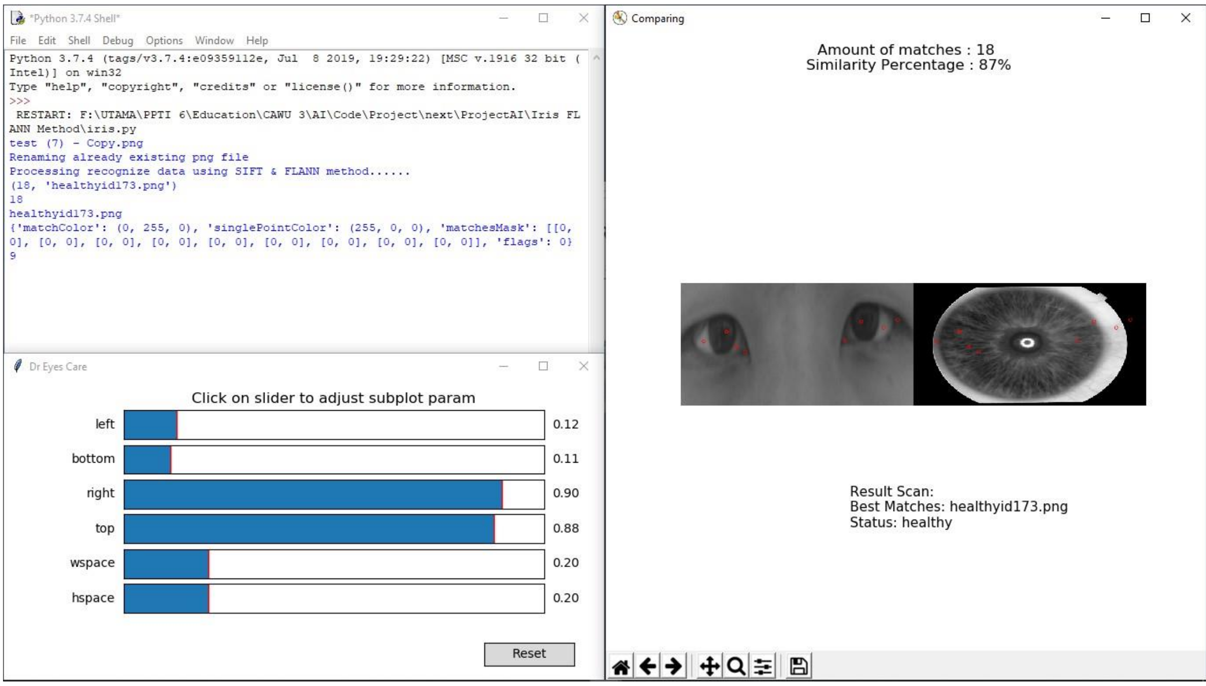


d. Tampilan saat membandingkan data

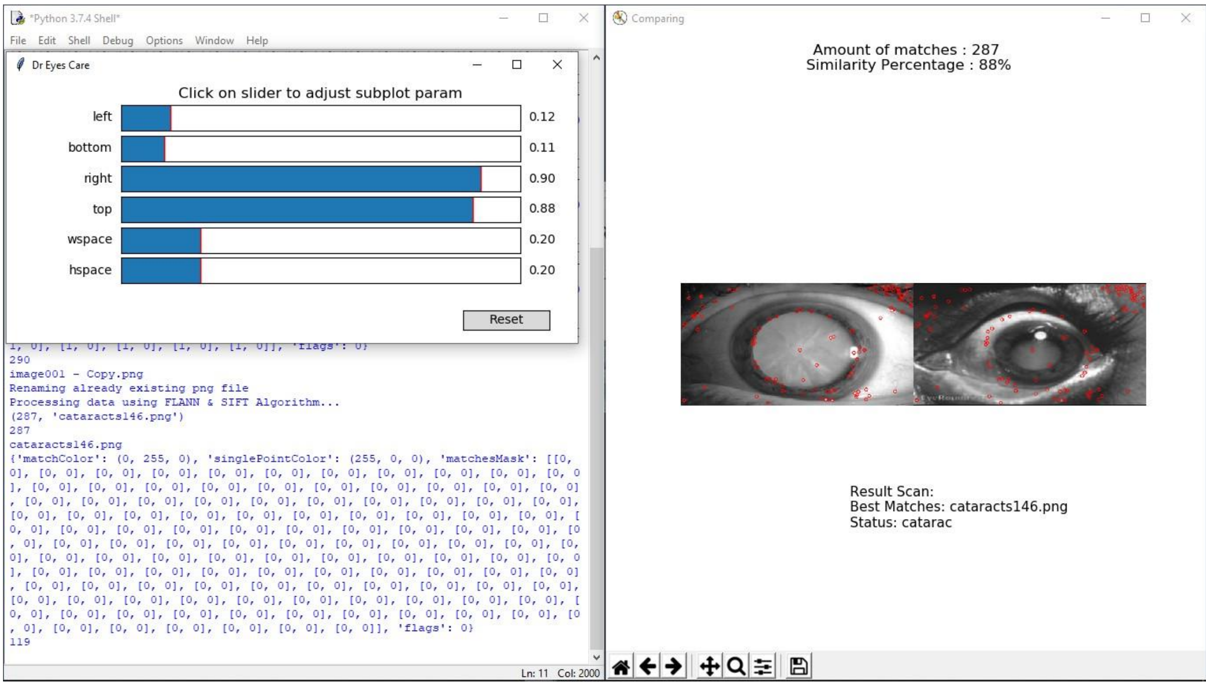


e. Tampilan *output*

1) Mata sehat



2) Mata katarak



BAB IV

PENUTUP

A. Kesimpulan

1. Pengalaman Belajar

Akurasi dari hasil yang didapatkan cukup tinggi karena banyaknya *data set* dan algoritma yang tepat.

2. Penemuan

a. Awalnya akurasi dibawah 10% karena data yang sedikit dan *learning* dari *artificial intelligence* yang belum maksimal.

b. Tetap dapat mendeteksi mata ketika mata terpejam

B. Saran

Berikut ini saran:

1. Menggunakan kamera langsung untuk memindai mata sehingga bisa dengan mudah menggunakan aplkasi ini.
2. Menambahkan fitur untuk memberikan saran pengobatan terhadap penyakit yang diderita user.
3. Menambahkan fitur *registration* supaya bisa digunakan oleh masyarakat luas.

DAFTAR PUSTAKA

Iswanto, Irene. 2017. *Feature Descriptor : SIFT (Scale Invariant Feature Transform) Part 1 : Introduction to SIFT*. https://socs.binus.ac.id/2017/06/13/feature-descriptor-sift-scale-invariant-feature-transform-part-1-introduction-to-sift/. (12 September 2019)

Marco, Daniel, Andrea, dan Luca. 2017. *What is Machine Learning? A definition*. https://www.expertsystem.com/machine-learning-definition/. (12 September 2019).

Puji, Aprinda. 2017. *5 Penyakit Mata yang Paling Umum di Indonesia*. https://hellosehat.com/pusat-kesehatan/gangguan-mata-dan-penglihatan/penyakit-mata-di-indonesia/. (12 September 2019).

https://www.cs.ubc.ca/research/flann/