

PENGOLAHAN CITRA UNTUK PENGENALAN WAJAH (FACE RECOGNITION) MENGGUNAKAN OPENCV

Theresia Susim dan Cahyo Darujati

Universitas Narotama Surabaya Jawa Timur, Indonesia

Email: theresiasusim20@gmail.com dan cahyo.darujati@narotama.ac.id

INFO ARTIKEL

Diterima
12 Februari 2021
Direvisi
20 Februari 2021
Disetujui
15 Maret 2021

ABSTRACT

The face is a common object in research material for Computer Vision and Image Processing technology, the application of image processing and computer vision has the main task of making decisions about real physical objects obtained from devices or sensors . To distinguish one face ID from another, it takes several points to select image processing data , face recognition, face detection, face alignment and facial feature storage. Facial recognition algorithms use Eigenface and are implemented in OpenCv. Data based on an example of a face image is matched with the face image stored in the available database by measuring the level of equations of various kinds -kinds of points, image processing, face recognition, face detection, face alignment, face extraction, facial feature storage and face matching .The purpose of this research is only to apply facial recognition in the OpenCv library which is written using the Python programming language. The average face being tested as much as 5 face image unrecognizable and 2 were not saved because of factors more ambient light, the position of the face of near and far d ari fa c tors these factors result in the value of different accuracy according to the degree of success in recognize faces, with an average recognition rate of 85% after the results of the comparison in the comparison process proximity of about 81% for facial resemblance me n NOTICE eigenface PCA method can identify a person contained in the database and can not recognize the person who is not in the database.

Keywords:

*Image processing;
Eigenface;Face
recognition; OpenCv;
Python*

ABSTRAK

Wajah merupakan objek yang umum dalam materi penelitian teknologi *computer vision* dan *image processing*, penerapan pengolahan citra dan *computer vision* mempunyai tugas utama yaitu untuk membuat suatu keputusan tentang objek fisik nyata yang di dapat dari perangkat atau sensor. Untuk membedakan ID wajah yang satu dengan yang lainnya butuh beberapa point untuk memilih data pengolahan citra, pengenalan wajah, pendeteksi wajah, penyelarasan wajah dan

Kata Kunci :

Pengolahan citra;
Eigenface;
Face recognition;
OpenCv; Python

penyimpanan fitur wajah, algoritma pengenalan wajah menggunakan Eigenface dan diimplementasikan dalam OpenCv. Data berdasarkan sutau contoh citra wajah di cocokkan dengan citra wajah yang tersimpan dalam *database* yang tersedia dengan mengukur tingkat persamaan macam-macam *point, image processing, face recognition*, pendeteksi wajah, penyalarsan wajah, ekstraksi wajah, penyimpanan fitur wajah dan pencocokan wajah. Tujuan penelitian ini hanya untuk menerapkan pengenalan wajah (*face recognition*) pada *library* OpenCv yang di tulis menggunakan Bahasa pemrograman Python. Rata-rata wajah yang diuji sebanyak 5 citra wajah dapat dikenali dan 2 yang tidak tersimpan karena faktor pencahayaan yang lebih terang, posisi wajah dari jarak dekat dan jauh dari faktor-faktor ini menghasilkan nilai akurasi yang berbeda sesuai dengan dengan tingkat keberhasilan dalam mengenali wajah, dengan tingkat pengenalan rata-rata 85% setelah di proses perbandingan perbandingan hasil kedekatan sekitar 81% untuk kemiripan wajah menggunakan metode PCA Eigenface dapat mengenali seseorang yang terdapat pada database dan tidak dapat mengenali orang yang tidak terdapat dalam database.

Corresponden Author

Email: theresiasusim20@gmail.com

Artikel dengan akses terbuka dibawah lisensi



Pendahuluan

Saat ini telah banyak berkembang sistem yang memanfaatkan fitur pengenalan wajah diantaranya yaitu sistem akses keamanan maupun sistem kontrol, pengolahan citra dan *computer vision* merupakan sebuah penemuan dibidang komputer yang digunakan untuk menghasilkan suatu sistem yang hampir mendekati dengan sistem visual manusia pada umumnya. Pengolahan citra adalah suatu jenis teknologi untuk menyelesaikan masalah mengenai pemrosesan gambar, sedangkan *computer vision* mempunyai tugas untuk membuat suatu keputusan tentang objek fisik nyata yang didapat dari perangkat atau sensor, *computer vision* membuat komputer dapat mengenali suatu citra layaknya manusia, salah satunya pengenalan wajah (Baskoro, 2015). Penerapan pengolahan citra dan *computer vision* saat ini banyak di gunakan pada perusahaan atau lembaga untuk meningkatkan sistem keamanan berbasis data pada karakteristik tubuh atau perilaku yang disebut biometrik.

Wajah merupakan bentuk yang unik dari setiap manusia, untuk mengenali bentuk wajah dapat dilakukan dengan mengenali pola wajah (Ahuja & Chhabra, 2011). Pengenalan wajah adalah salah satu aplikasi yang paling berkembang dalam *computer vision* dan *pattern recognition* (pengenalan pola) (Zufar, 2016). Tujuan untuk mengenali

wajah manusia dari gambar atau video menggunakan *database* wajah yang tersimpan sehingga pengenalan wajah sendiri dapat dilakukan dengan berbagai cara, salah satunya dengan metode Viola Jones dan Eigenface, metode Viola Jones digunakan untuk proses pendeteksian yang merupakan hasil dari gabungan fitur- fitur khusus, sedangkan metode Eigenface merupakan algoritma berdasarkan Principal Component Analysis (PCA) yang mudah diimplementasikan dan dimulai dengan pemrosesan awal untuk menemukan hasil citra yang lebih baik setelah itu menghitung *eigenvector* dan *eigenvalue* dari citra wajah yang di *training*.

Oleh karena itu, dengan adanya penelitian ini yang berbentuk eksperimen rekayasa perangkat lunak yang luarnya berupa aplikasi dengan data dengan data dari penelitian berupa sampel citra yang di *capture* dari webcam atau foto yang terhubung dengan komputer. Citra wajah manusia yang diambil berbeda-beda dengan masing-masing mendapatkan perlakuan variasi yang sama yaitu kemiringan posisi citra wajah, jarak wajah, dan intensitas cahaya.

Berdasarkan jurnal penelitian yang dilakukan oleh (Muliawan et al., 2015) menunjukkan bahwa pemrosesan pengenalan wajah pada system absensi dengan menggunakan metode Eigenface pada OpenCV dapat dilakukan dengan cara input data pengguna dan data wajah beserta *password* dari masing-masing pengguna. “Metode *Face Recognition* untuk Identifikasi Personil Berdasar Citra Wajah” Dari jurnal penelitian yang dilakukan oleh (Kurniawan, 2014) membahas tentang rancang bangun *system face recognition online* pegawai Universitas Negeri Semarang dengan Python dan Opencv dapat terealisasi.

Penelitian sebelumnya yang dilakukan oleh (Harits, 2019) tentang “Penerapan Algoritma Viola Jones Pada Deteksi Wajah”. Penelitian ini menerapkan metode Viola Jones untuk pendeteksian citra wajah. Hasil penelitian ini adalah menampilkan kotak kuning dan tulisan “wajah” jika dalam gambar tersebut menampilkan gambar wajah. “Pengenalan Wajah Menggunakan Metode Viola-Jones dan Eigenface Dengan Variasi Posisi Wajah Berbasis Webcam”. Dari jurnal penelitian yang dilakukan oleh Aries Suharso Pada tahun 2016, menunjukkan tingkat akurasi deteksi citra wajah sebesar 88.89% dengan sudut rotasi variasi posisi citra wajah menghadap ke depan (frontal)-15⁰, kesamping kiri 30⁰, dan kanan 30⁰ dan terbukti pada nilai *threshold* 75 memiliki akurasi tertinggi dan jumlah dikenali yaitu sebesar 90.90%. Tujuan dari penelitian ini adalah untuk mempelajari dan memahami proses pengenalan wajah menggunakan OpenCV serta mencocokkan wajah manusia sesuai dengan *database* yang ada.

Metode Penelitian

Metode penelitian yang digunakan adalah metode pengumpulan data menggunakan pendekatan kualitatif dimana pengumpulan data yang dilakukan secara observasi dan *study literature*. Observasi adalah suatu proses penelitian secara langsung untuk melakukan analisis permasalahan-permasalahan pada citra untuk mendapatkan informasi yang menunjang pembuatan laporan ini (Larasati & Masripah, 2017). Sedangkan *study literature* adalah penelitian yang dilakukan dengan mengumpulkan data

informasi, dokumen-dokumen yang berhubungan dengan topik penelitian dari berbagai sumber seperti buku, jurnal dan sebagainya (Rukajat, 2018). Digunakan pula metode perancangan pengenalan wajah sebagai *block diagram software* untuk melakukan pengenalan dari sistem yang digunakan, dimana jalannya sistem dibagi menjadi 2, yaitu *block* pengambilan data dan *block* pengenalan wajah.

Hasil dan Pembahasan

Hasil yang didapatkan dari penelitian adalah:

1. Pengolahan Citra

Citra merupakan suatu representasi (gambar), kemiripan, dan imitasi dari suatu objek (Arisandi et al., 2018). Citra sebagai keluaran suatu sistem perekaman data dapat bersifat *optic* berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

2. Pengenalan Wajah

Pengenalan wajah adalah suatu metode pengenalan yang berorientasi pada wajah (Wuryandari & Afrianto, 2012). Pengenalan ini dapat dibagi menjadi dua bagian yaitu di kenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola yang sebelumnya disimpan di alam *database*, secara umum, system pengenalan wajah dibagi menjadi 2 jenis, yaitu *system Feature based* dan *image based*. Pada *system* pertama digunakan fitur yang disktraksi dari komponen citra wajah (mata, hidung, mulut, dll) dan kemudian hubungan antara fitur-fitur dimodelkan secara geometris. Sedangkan *system* kedua menggunakan informasi mentah dari piksel citra yang kemudian direpresentasikan dalam metode tertentu, yang kemudian digunakan untuk klasifikasi identitas citra.

3. Metode Eigenface

Metode Eigenface adalah hasil transformasi citra-citra wajah menjadi sebuah set fitur karakteristik wajah dalam bentuk *eigenvector* dari matriks kovarian citra-citra wajah tersebut (Fandiansyah et al., 2017). Dalam pendekatan dan pengenalan wajah dijelaskan bahwa Eigenface tidak melihat fitur-fitur tertentu pada wajah seperti mata, mulut, hidung, dan lain-lain untuk mendeteksi dan mengenali wajah, melainkan mengenal wajah dengan *eigenvector*. Pendekatan pendeteksian dan pengenalan wajah dengan Eigenface ini dibangun dengan tujuan agar proses pendeteksian dan pengenalan wajah menjadi cepat, simpel, dan akurat di dalam lingkungan yang terbatas seperti ruang kantor atau ruangan pada rumah. Konsep dari pendekatan dengan Eigenface ini ialah dengan mengekstrak informasi relevan dari citra wajah dengan cara *encoding*, kemudian membandingkan hasil *encoding* citra wajah tersebut dengan citra wajah yang telah di-*encode* sebelumnya, dalam pendekatan ini informasi yang di *encode* dari citra ialah *eigenvector* dari matriks kovarian citra-citra wajah.

4. Pendeteksi Wajah

Proses pendeteksi wajah ini bekerja dengan cara memeriksa citra yang dimasukan apakah memiliki citra wajah atau tidak, jika memiliki maka akan dilakukan pemisahan dengan cara memotong citra wajah dari latar belakang citra yang dimasukan. Jika masukan berbentuk video, proses yang dilakukan adalah proses *face tracking*. Secara umum, proses *face tracking* dan proses wajah mempunyai fungsi yang sama, perbedaaan terletak pada pendeteksiannya.

5. Penyelarasam Wajah

Pada proses pendeteksian wajah, citra wajah yang didapatkan masih berupa perkiraan kasar atau masih memiliki kualitas yang cukup buruk seperti ukuran yang berbeda dengan ukuran normal, faktor pencahayaan yang kurang atau lebih, kejelasan citra yang buruk dan sebagainya. Proses penyelarasam wajah merupakan proses yang bertujuan untuk menormalisasi wajah dari citra wajah yang didapatkan dari proses pendeteksian wajah (Wardana et al., 2016). Proses ini terdiri dari tahapan-tahapan sebagai berikut:

- a. *Grayscale* citra merupakan tahapan pertama dari proses penyelarasam, pada tahap ini terjadi pengkonversian citra warna RGB menjadi citra berwarna abu. Citra warna RGB terdiri dari 3 parameter warna yaitu merah (red), hijau (green) dan biru (blue), jika citra warna RGB ini dimasukan ke dalam proses ekstraksi, maka proses tersebut akan sulit untuk dilakukan karena citra RGB terdiri dari 3 parameter, oleh karena itu diperlukan penyamaan parameter yaitu dengan melakukan tahap *grayscale* ini.
 - b. Pemotongan Pada tahapan ini terjadi pemotongan citra yang memisahkan citra wajah dengan citra masukannya, tujuannya untuk mengambil citra yang hanya diperlukann untuk proses ekstraksi, dalam hal ini adalah citra wajah dan membuang citra lain yang tidak diperlukan. Dimensi citra yang dipotong disesuaikan dengan dimensi dari proses segmentasi atau pengkotakan objek wajah yang dilakukan pada proses pendeteksian wajah.
 - c. Resizing (tahap normalisasi dimensi citra), pada tahap *resizing* citra terjadi proses normalisasi dimensi citra wajah, yaitu proses pembesaran atau pengecilan dimensi citra wajah menjadi dimensi yang telah ditentukan. Tujuannya, untuk menyamakan dimensi wajah dari tiap citra yang dimasukan, sehingga pada proses ekstraksi citra nanti tidak ada perbedaaan dimensi dari matriks data citra wajah.
 - d. Equalizing (tahap koreksi tingkat kecerahan citra), tahap ini adalah tahapan terakhir dari proses penyelarasam yang tujuannya untuk memperjelas nilai histogram dari citra wajah hasil tahapan-tahapan sebelumnya.
- #### 6. Ekspresi Wajah

Ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu sampel wajah dari sampel wajah yang lain. Ekstraksi fitur juga merupakan proses yang berfungsi untuk mendapatkan informasi yang efektif dan

berguna untuk membedakan wajah-wajah orang yang berbeda dari citra wajah yang telah diselaraskan. Proses ini dilakukan menggunakan algoritma-algoritma ekstraksi seperti Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA) dan sebagainya. Informasi yang didapatkan dari ekstraksi fitur disebut vektor fitur, yaitu bentuk dasar pencarian citra berbasis konten yang menangkap properti citra seperti warna dan tekstur.

7. Pencocokan Wajah

Pencocokan fitur merupakan proses perbandingan fitur yang telah diekstrak dari citra uji dengan fitur citra wajah dari *database* yang sebelumnya telah melalui proses pelatihan citra.

8. OpenCV

OpenCV (Open Source Computer Vision Library) adalah sebuah *library open source* yang dikembangkan oleh *intel* yang fokus untuk menyederhanakan *programming* terkait citra digital (Fitrah, 2020). Di dalam OpenCV sudah mempunyai banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, deteksi wajah, *kalman filtering*, dan berbagai jenis metode AI (Artificial Intelligence) dan menyediakan berbagai algoritma sederhana terkait *computer vision* untuk *low level*, OpenCV merupakan *open source computer vision library* untuk bahasa pemrograman C/C++, dan telah dikembangkan ke Python, Java, Matlab.

OpenCV pertama kali diluncurkan secara resmi pada tahun 1999 oleh *Inter Research* sebagai lanjutan dari bagian proyek bertajuk aplikasi intensif berbasis CPU, *real-time ray tracing* dan tembok penampil 3D. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi di *Intel Russia*, dan juga tim pustaka performansi intel. OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, berikut ini adalah fitur utama dari OpenCV antara lain:

- a. *Image and video I/O* dengan antar muka ini kita dapat membaca data gambar dari *file*, atau dari umpan video langsung dan juga dapat menciptakan *file* gambar maupun video
- b. *Computer vision* secara umum dan pengolahan citra digital (untuk *low* dan *mid level* API) dengan antar muka ini kita dapat melakukan eksperimen uji coba dengan berbagai standar algoritma *computer vision*. Termasuk juga deteksi garis, tepi, pucuk, proyeksi elips, *image pyramid* untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai *transform* (*Fourier*, *cosine diskrit*, *distance transform*) dan lain lain.
- c. Modul *computer vision high level*
- d. Metode untuk AI dan *machine learning* aplikasi *computer vision* sering kali memerlukan *machine learning* atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV *machine learning*.

Pengenalan wajah atau *face recognition* adalah salah satu produk dari *image recognition*. *Face Recognition* memungkinkan perangkat untuk mengenali identitas seseorang atau objek hanya dengan gambar, ada beberapa bahasa pemrograman

yang dapat digunakan diantaranya:

a. Matlab

Matlab adalah bahasa pemrograman *independen* yang memiliki kerangka kerja sendiri (Kurniawati & Kusumawardhani, 2017). Bahasa pemrograman ini dilengkapi dengan berbagai alat yang dapat digunakan untuk menangani tugas-tugas pemrograman yang lebih teknis dengan mudah. Pengenalan citra dan pengolahan wajah adalah beberapa tugas yang dapat ditangani oleh Matlab. Matlab menawarkan berbagai alat bawaan untuk perhitungan matriks. Sangat penting untuk dicatat bahwa pengenalan gambar dan perhitungan matriks berjalan seiring. Beberapa alat yang tersedia di Matlab dapat melakukan tugas pemrosesan gambar yang kompleks seperti memotong, memutar, dan menutupi yang lain. Ada juga kode Matlab khusus untuk penerapan fitur pengenalan wajah, kode ini menggunakan *database* AT & T. Untuk alasan ini, terlebih dulu harus mengunduh *database* AT & T sebelum menggunakan kode ini aplikasi pengenalan wajah yang paling canggih dibuat menggunakan Matlab.

b. Python

Saat ini, Python dinilai sebagai bahasa pemrograman paling populer. Kesederhanaan dan keserbagunaannya adalah beberapa alasan mengapa banyak *programmer* lebih suka menggunakannya. Meskipun sederhana, Python adalah bahasa yang dapat diandalkan untuk melakukan tugas-tugas kompleks (Suryansah et al., 2020). Bahasa pemrograman ini dapat digunakan untuk membuat pemrosesan gambar dan fitur pengenalan, popularitas Python memang masuk akal. Bahasa ini dikemas dengan beberapa perpustakaan yang diperlukan untuk beberapa fungsi, di antaranya pengenalan gambar. Salah satu perpustakaan yang paling kuat dan efisien adalah perpustakaan *Scikit-Learn Machine Learning*. Selain pengenalan gambar, pustaka dapat digunakan untuk fungsi yang lebih cerdas seperti pengenalan wajah dan deteksi gerakan.

c. C/C++/C#

Bahasa pemrograman keluarga C memang benar-benar serba guna, mereka kuat dan dapat melakukan apa saja termasuk menciptakan fungsi pemrosesan gambar dan pengenalan. Bahasa pemrograman keluarga C akan memberikan dua opsi untuk membuat fitur pemrosesan gambar. Kita dapat memilih untuk mengkodekan segala sesuatu dari awal di mana harus menulis kode secara manual. Pilihan kedua adalah menggunakan pustaka yang ada yang dirancang khusus untuk bahasa pemrograman ini. Perpustakaan-perpustakaan ini termasuk OpenGL, EmguCV, OpenCV, dan banyak lainnya. Mereka memiliki fitur pengolah gambar cerdas untuk pengenalan gambar.

d. Java

Bahasa pemrograman Java Seperti C dan C ++ tidak dapat meremehkan kekuatannya. Bahasa ini cukup kuat untuk melakukan fungsi yang rumit, bahasa ini dapat digunakan untuk membuat aplikasi untuk pemrosesan gambar dan pengenalan gambar. Sebagai bahasa pemrograman mandiri, kita dapat membuat

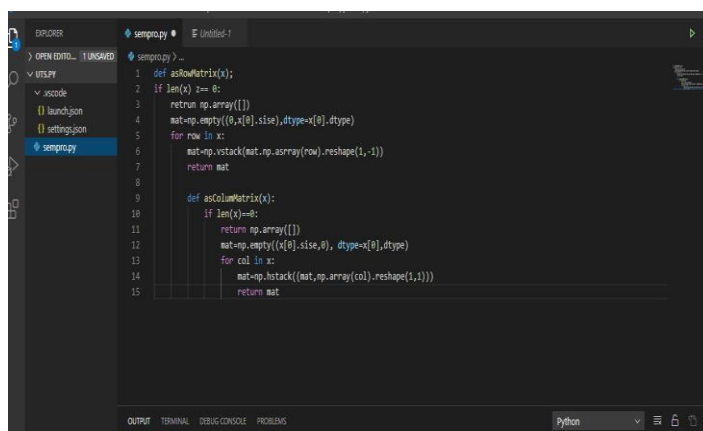
fungsi pemrosesan gambar dari awal. Keuntungan dari metode ini adalah kita memiliki kontrol penuh terhadap fungsi tersebut. Ada pustaka yang lengkap yang dapat dimasukkan ke dalam Java untuk tujuan menciptakan fitur pengenalan gambar. OpenCV adalah salah satu pustaka yang kompatibel dengan Java.

e. OpenCV

Open Source Computer Vision adalah alat canggih yang digunakan untuk pemrosesan *real-time* (Kusumanto et al., 2012). Ini juga merupakan alat *cross-platform* sehingga dapat diintegrasikan ke dalam bahasa pemrograman untuk tujuan menjalankan fungsi pengolahan gambar dan pengenalan. Ini dapat diintegrasikan dengan bahasa pemrograman C, C ++, Java, Python dan Android. OpenCV hadir dengan algoritma bebas paten yang dapat digunakan tanpa batasan hukum.

Dapat digunakan untuk tujuan komersial maupun akademis. OpenCV memiliki kelas *face recognition* khusus yang dapat digunakan untuk menguji kemampuan fitur pengenalan gambar tanpa kerumitan apa pun. Kelas disertai dengan dokumentasi kaya informasi yang akan menunjukkan bagaimana menerapkan fitur pengenalan gambar. Secara keseluruhan, OpenCV adalah alat untuk pengenalan gambar terlepas dari bahasa pemrograman yang digunakan.

Tahap ini meliputi tahap *Demonstration* yang telah dirancang ke dalam program.tahap ini akan diterapkan melalui gambar sebagai berikut:



```
1 def asRowMatrix(x):
2     if len(x) == 0:
3         return np.array([])
4     mat=np.empty((0,x[0].size),dtype=x[0].dtype)
5     for row in x:
6         mat=np.vstack((mat,np.asarray(row).reshape(1,-1)))
7     return mat
8
9
10
11 def asColumnMatrix(x):
12     if len(x)==0:
13         return np.array([])
14     mat=np.empty((x[0].size,0), dtype=x[0].dtype)
15     for col in x:
16         mat=np.hstack((mat,np.array(col).reshape(1,1)))
17     return mat
```

Gambar 1
Program Python

Python adalah bahasa pemrograman yang banyak dipakai di dunia komputer. Kemampuan Python untuk mengolah citra sangatlah terbatas, untuk itu perlu *import library* dari OpenCV, disinilah keterkaitan antara OpenCv dan Python. OpenCV bersama Python dimanfaatkan untuk mengolah *image* atau video (tumpukan *frame/image*) sesuai dengan tujuan masing-masing yang melibatkan kamera untuk menangkap gambar lalu diolah di komputer, terlihat pada Gambar 2.


```
import cv2

videoCam = cv2.VideoCapture(0)
face = cv2.CascadeClassifier("face-detect.xml")

def capture_images("Baukkan ID : ")
    sampleNum = 0
    while True:
        cond, frame = videoCam.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        muka = face.detectMultiScale(gray, 1.1, 3)
        for (x,y,w,h) in muka:
            sampleNum = sampleNum + 1
            cv2.imwrite("datacapture/muka-%02d-%02d-%02d.jpg" % (sampleNum, x, y), gray[y:y+h, x:x+w])
            cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 255, 0), 1)
            cv2.waitKey(100)
            roi_warna = frame[y:y+h, x:x+w]
            roi_gray = gray[y:y+h, x:x+w]

        cv2.imshow("Data capture", frame)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            cv2.waitKey()
            (sampleNum, img):
                pass

videoCam.release()
cv2.destroyAllWindows()
```

Gambar 2
Program Face Detection

Program *face detection* (pengenalan wajah) logika program pada *file capture* merupakan program untuk mendeteksi apakah ada wajah yang dapat dideteksi dari *frame* yang telah ada. Jika wajah telah terdeteksi pada *frame*, maka akan diambil gambarnya pada data *capture* sebanyak 200 kali sample dan disimpan dalam folder Data wajah. Berikut kode program dari *face detection* dapat dilihat pada Gambar 3.

```
import cv2
import numpy as np
import smtplib
import logging
import datetime
import sys
import time
import sleep
import twilio.rest

client

def detect():
    account_sid = 'AC2f2c9d779772973f6641125c5d1f6e'
    auth_token = '90230415e79f943678100efab0f9eaf'
    client = Client(account_sid, auth_token)
    message = client.messages.create(body="terdeteksi wajah " + str(id), from_="whatsapp:+6155238866", to="whatsapp:+6289686987951")
    print(message.sid)

def undetect():
    account_sid = 'AC2f2c9d779772973f6641125c5d1f6e'
    auth_token = '90230415e79f943678100efab0f9eaf'
    client = Client(account_sid, auth_token)
    message = client.messages.create(body="terdeteksi wajah tidak dikenal", from_="whatsapp:+6155238866", to="whatsapp:+6289686987951")
    print(message.sid)

def on(x):
    if (x%10==0):
        detect_sendnull()

def off(x):
    if (x%10==0):
        undetect_sendnull()

deoCam = cv2.VideoCapture(0)
ce = cv2.CascadeClassifier("face-detect.xml")
ne = cv2.CascadeClassifier("eye-detect.xml")
```

Gambar 3
Program rec.py

Program *face recognition* (pendeteksi wajah) setelah melakukan *training* pada program, maka program akan menghasilkan informasi apakah wajah dikenal atau tidak dikenal dengan nilai akurasi yang sesuai dengan hasil pendeteksi wajah.

Kesimpulan

Berdasarkan contoh implementasi diatas, dapat disimpulkan bahwa sistem pengenalan wajah menggunakan metode PCA eigenface dapat mengenali wajah

seseorang yang terdapat pada database dan tidak dapat mengenali wajah orang yang tidak terdapat pada database.

Analisa Hasil dari penulisan ini dibuat dengan memberikan kondisi 3 faktor yang dapat mempengaruhi hasil dari pendeteksian wajah . Faktor tersebut adalah Pencahayaan yang lebih terang, posisi wajah dari jarak dekat dan jauh. Dari faktor-faktor ini menghasilkan nilai akurasi yang berbeda sesuai dengan tingkat keberhasilan dalam mengenali wajah,

BIBLIOGRAFI

- Ahuja, M. S., & Chhabra, S. (2011). Effect of distance measures in PCA based face recognition. *International Journal of Enterprise Computing and Business Systems*, 1(2), 2230–8849.
- Arisandi, D., Syahputra, M. F., Putri, I. L., Purnamawati, S., Rahmat, R. F., & Sari, P. P. (2018). A real time mobile-based face recognition with fisherface methods. *Journal of Physics: Conference Series*, 978(1), 12038.
- Baskoro, H. A. (2015). *Visual sensor network dengan algoritma yang hemat energi*. Institut Teknologi Sepuluh Nopember.
- Fandiansyah, F., Sari, J. Y., & Ningrum, I. P. (2017). Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor. *Ultimatics: Jurnal Teknik Informatika*, 9(1), 1–9.
- Fitrah, G. F. R. (2020). *Rancang Bangun Aplikasi Rekomendasi Penyampaian Materi Pembelajaran Berdasarkan Deteksi Emosi Wajah*. Universitas Komputer Indonesia.
- Harits, A. A. (2019). *Rancang Bangun Sistem Keamanan Ruangan Menggunakan Deteksi Wajah Berbasis Raspberry PI*. Politeknik Negeri Sriwijaya.
- Kurniawan, L. M. (2014). Metode Face Recognition untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang. *Scientific Journal of Informatics*, 1(2), 210–220.
- Kurniawati, I. D., & Kusumawardhani, A. (2017). Implementasi Algoritma Canny Dalam Pengenalan Wajah Menggunakan Antarmuka GUI Matlab. *ResearchGate, Surabaya*.
- Kusumanto, R. D., Pambudi, W. S., & Tompunu, A. N. (2012). Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang. *Semantik*, 2(1).
- Larasati, H., & Masripah, S. (2017). Analisa dan perancangan sistem informasi pembelian grc dengan metode waterfall. *Jurnal Pilar Nusa Mandiri*, 13(2), 193–198.
- Muliawan, M. R., Irawan, B., & Brianorman, Y. (2015). Implementasi Pengenalan Wajah Dengan Metode Eigenface Pada Sistem Absensi. *Coding Jurnal Komputer Dan Aplikasi*, 3(1).
- Rukajat, A. (2018). *Pendekatan Penelitian Kualitatif (Qualitative Research Approach)*. Deepublish.
- Suryansah, A., Habibi, R., & Awangga, R. M. (2020). *Penggunaan Face Recognition untuk akses ruangan*. Kreatif.

- Wardana, I. P. P., Giriantari, I. A. D., & Sudarma, M. (2016). Aplikasi Verifikasi Wajah Untuk Absensi Pada Platform Android Dengan Menggunakan Algoritma Fisherface. *Majalah Ilmiah Teknologi Elektro*, 15(2), 45–52.
- Wuryandari, M. D., & Afrianto, I. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *Jurnal Komputer Dan Informatika (Komputa)*, 1(1), 1.
- Zufar, M. (2016). *Convolutional neural networks untuk pengenalan wajah secara real-time*. Institut Technology Sepuluh Nopember.