

IMPLEMENTASI SISTEM PENGENALAN WAJAH UNTUK KEAMANAN AKSES BERBASIS UBUNTU MENGGUNAKAN PYTHON

Fahrizal¹, Andri Maulana², Sari Dermawan Aritonang³,

¹ Sistem Informasi, Universitas Bina Sarana Informatika

² Sistem Informasi, STMIK Nusamandiri

³ Teknologi Informasi, Universitas Bina Sarana Informatika

^{1,2,3} Jl. Kamal Raya No.18, RT.6/RW.3, Cengkareng Barat, Kecamatan Cengkareng, Kota Jakarta Barat,
Daerah Khusus Ibukota Jakarta 11730

email: fahrizal.fzl@bsi.ac.id, andry.ayz@bsi.ac.id, fsariarios43@gmail.com

Article history

Received June 09, 2021

Revised June 20, 2021

Accepted June 27, 2021

Available online June 30, 2021

Keywords

Computer Vision,

OpenCv,

Face recognition,

Python,

Ubuntu

Abstract

Security is one of the most important needs for human beings in both the building and the house. For the development of security technology used face recognition. Face recognition is a system that identifies facial features that are capable of detecting familiar faces and unknown Faces. In this research is implemented with computer vision where the computer can see and understand so that it is information from an Image or video. This computer can also mimic the ability of human intelligence. To classify a face object using OpenCv with Haar Cascade Classifier algorithm and uses Python programming language. Application used face recognition program is PyCharm Community 2018 version 3 with Linux operating system Ubuntu 18.04.2 LTS version. The results showed that the accuracy of face recognition depends on the analysis of OpenCv and the classification of Cascade for computer vision process.

Riwayat

Diterima 09 Juni 2021

Revisi 20 Juni 2021

Disetujui 27 Juni 2021

Terbit 30 Juni 2021

Kata Kunci

Computer Vision,

OpenCv,

Face recognition,

Python,

Ubuntu

Abstrak

Keamanan merupakan salah satu kebutuhan yang sangat penting bagi manusia baik di gedung maupun rumah. Untuk pengembangan teknologi keamanan menggunakan face recognition. Face recognition adalah suatu sistem yang mengidentifikasi fitur wajah yang mampu mendeteksi wajah yang dikenal dan wajah tidak dikenal. Pada penelitian ini diimplementasikan dengan komputer vision yang mana komputer dapat melihat dan memahami sehingga mengekstrak informasi dari sebuah gambar atau video. Komputer ini juga dapat meniru kemampuan kecerdasan manusia. Untuk mengklasifikasi sebuah objek wajah, menggunakan OpenCv dengan algoritma Haar Cascade Classifier dan menggunakan bahasa pemrograman python. Aplikasi yang digunakan program face recognition adalah PyCharm Community 2018 Versi 3 dengan sistem operasi Linux Ubuntu versi 18.04.2 LTS. Hasil penelitian menunjukkan bahwa akurasi dari face recognition tergantung pada analisa OpenCv dan klasifikasi cascade untuk proses computer vision.

PENDAHULUAN

Pelacakan dan pengenalan wajah manusia merupakan salah satu bidang yang cukup berkembang saat ini, dimana sistem ini diterapkan dalam bidang keamanan (*security system*) seperti izin akses masuk ruangan, pengawasan lokasi (*surveillance*), maupun pencarian identitas individu pada database

kepolisian, karena diterapkan dalam kasus keamanan, dibutuhkan sistem yang handal terhadap beberapa kondisi, seperti : pengaruh latar belakang gambar dan pose wajah.

Menurut (Purwanto, Dirgantoro, & Jati, 2015) mengemukakan bahwa "Rasa aman merupakan salah satu kebutuhan yang sangat penting bagi manusia dalam menghadapi era

teknologi saat ini. Keingintahuan pada keadaan atau kondisi hal yang dimiliki juga menjadi faktor penting bagi manusia untuk terus mengembangkan sistem keamanan gedung maupun rumah. Sistem dapat mendeteksi objek berupa wajah dari masukan citra. Sistem akan memerlukan masukan citra dari sebuah kamera keamanan. Setelah objek berupa wajah terdeteksi, sistem akan melakukan pencocokan wajah dengan gambar wajah yang terdapat pada database sistem. Dari pengolahan data, sistem akan menghasilkan logika yang digunakan pada proses selanjutnya dalam sistem keseluruhan. Sistem merupakan penerapan dari *Computer Vision* dalam sistem keamanan gedung atau rumah. Sistem akan mengambil citra dari masukan berupa video”.

Mengenali wajah dalam sebuah gambar atau foto, bagi manusia merupakan hal yang mudah tetapi tidak demikian untuk komputer, agar dapat mengenali wajah manusia diperlukan perlakuan khusus sehingga ketika diberi *input* sebuah gambar atau foto, komputer dapat mendeteksi apakah dalam gambar tersebut terdapat sebuah gambar wajah atau tidak dan mengenali wajah tersebut. Sistem ini juga membutuhkan peran sistem kecerdasan buatan agar komputer dapat memiliki keputusan yang tepat dalam mengidentifikasi wajah.

Dalam tulisan ini akan dibahas tentang sistem pengenalan wajah (*face recognition*), yaitu pengenalan wajah dengan citra banyak wajah sehingga diharapkan dapat menjadi suatu sistem keamanan yang handal sebagai pengganti sistem *fingerpint* untuk meminimalkan sentuhan untuk mencegah penyebaran COVID19 yang sedang menjadi wabah pandemi global saat ini. Diharapkan dengan metode *face recognition*, pengenalan banyak wajah ini dapat memperoleh tingkat akurasi tinggi.

Maksud dalam tulisan ini dapat dirumuskan sebagai berikut :

1. Merancang dan mengimplementasikan sistem pengenalan wajah dari tiap sumber dalam banyak wajah.
2. Mendapatkan ciri yang membedakan wajah manusia dengan metode *face recognition*.

3. Menganalisa *face recognition* dengan menggunakan metode OpenCV.

LITERATUR

Sebagai referensi dalam penelitian ini memperoleh literatur dari beberapa buku dan penelitian sebelumnya yaitu:

1. Jurnal dengan judul “Sistem Pengenalan Wajah Secara *Real-Time* dengan Adaboost, *Eigenface* PCA & MySQL” yang ditulis oleh Dodit Suprianto dkk, dalam penelitian pada jurnal tersebut pengenalan wajah dilakukan melalui tahap *face detection*, *feature extraction* dan *face recognition*, selanjutnya dicocokkan dengan data profil yang tersimpan di dalam database. Pendeteksian wajah menggunakan metode Adaboost, pengenalan wajah menggunakan metode *Eigenface* PCA dan database MySQL untuk menyimpan informasi profile. Penggunaan metode tersebut untuk pengenalan wajah pada kondisi *real time* dengan perbedaan jarak antara sensor dan wajah, posisi wajah, intensitas cahaya yang mengenai wajah, mimik muka dan atribut wajah dalam penelitian ini memberikan tingkat keberhasilan sebesar 80% dalam mengidentifikasi wajah.

2. Jurnal berjudul “Metode *Face recognition* untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang” yang ditulis oleh Luthfi Maslichul Kurniawan, pada penelitian tersebut dikembangkanlah sebuah sistem *face recognition* yang dirancang bangun menggunakan bahasa pemrograman Python dan pustaka OpenCV. Hasil dari rancang bangun ini adalah sistem *face recognition* yang mampu berjalan secara otomatis di komputer server untuk membaca basis data presensi, mengolah foto-foto yang tersimpan pada basis data tersebut, mendeteksi wajah pada foto-foto yang diolah kemudian menampilkan hasilnya pada tabel basis data presensi untuk diolah dalam bentuk skor deteksi wajah yang tampil di rekapitulasi presensi online pegawai.

3. Jurnal berjudul “Implementasi *Face recognition* Pada Absensi Kehadiran Mahasiswa Menggunakan Metode *Haar Cascade Classifier*” yang ditulis oleh Munawir

Munawir, Liza Fitria, Muhammad Hermansyah. Metode yang digunakan dalam penelitian ini adalah *Haar Cascade Classifier*. Terdapat 125 data training wajah dari 25 orang mahasiswa yang telah diinputkan kedalam sistem. Pengujian dilakukan dengan cara pengenalan satu wajah dan banyak wajah (*multiple face recognition*) sekaligus. Hasil pengujian menunjukkan bahwa tingkat akurasi pengenalan pada satu wajah dengan 25 data testing wajah didapatkan nilai akurasi sebesar 76%, sedangkan pengenalan pada banyak wajah didapatkan nilai akurasi sebesar 33.3%.

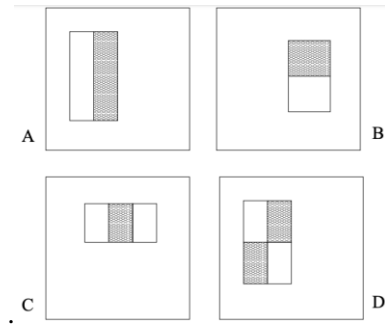
4. Buku berjudul “*Applied Natural language processing with Python. Implementing Machine learning and Deep Learning Algorithms for Natural language processing* yang disusun oleh Beysolow II, T. dalam buku tersebut membahas tentang pemanfaatan kemampuan AI untuk *Natural language processing* (NLP), yang dapat melakukan tugas-tugas seperti pemeriksa ejaan, peringkasan teks, klasifikasi dokumen, dan generasi bahasa alami. Dalam buku tersebut, mempelajari keterampilan untuk menerapkan metode ini dalam infrastruktur yang lebih besar untuk mengganti kode yang ada atau membuat algoritma baru. Penerapan *Natural language processing* menggunakan Python dimulai dengan meninjau konsep-konsep *machine learning* yang diperlukan sebelum membahas berbagai masalah NLP. Setelah membaca buku tersebut, Pembacanya akan memiliki keterampilan untuk menerapkan konsep-konsep ini di lingkungan sendiri.

METODE PENELITIAN

Agar lebih terarah, metodologi yang digunakan dalam penelitian ini yaitu: algoritma *Haar Cascade Classifier* yang diusulkan oleh Paul Viola dan Michael Jhon pada jurnalnya yang berjudul “Rapid Object Detection using a Boosted Cascade of Simple Features” yang terdiri atas:

Haar feature selection

Yaitu dengan cara menentukan ada atau tidaknya haar feature

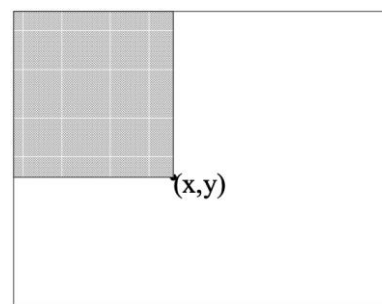


Gambar 1. contoh *haar feature detection windows*

Gambar 1 menunjukkan contoh feature persegi panjang yang ditampilkan relatif terhadap detection window terlampir. Jumlah piksel yang terletak pada persegi panjang putih dikurangi dari jumlah piksel dalam persegi panjang abu-abu. fitur dua persegi panjang ditunjukkan pada (A) dan (B). gambar (C) menunjukkan fitur tiga persegi panjang, dan (D) fitur empat persegi panjang Jumlah piksel yang terletak di dalam putih.

Creating integral Images

yaitu membuat gambar integral dari gambar keseluruhan hasil dari deteksi ada atau tidaknya haar feature. dengan menggunakan rumus *integral Image*.



Gambar 2 nilai dari *integral Image* pada titik (x,y) adalah jumlah dari keseluruhan piksel ujung atas ke kiri.

$$ii(x,y) = \sum_{x^l \leq x, y^l \leq y}^n i(x^l, y^l)$$

Dimana $ii(x,y)$ adalah *integral Image* dan $i(x^l, y^l)$ adalah gambar asli

Adaboost training

Yaitu melakukan *training* terhadap komputer untuk pendeteksian object, dalam hal ini yaitu object wajah, adapun proses Adaboost *training* ini terdiri atas *feature discussion, learning discussion, learning result, training a cascade of classifiers*.

HASIL DAN PEMBAHASAN

Penelitian ini merupakan implementasi aplikasi dalam bentuk eksperimen. Adapun langkah awal yang diperlukan adalah membuat sebuah aplikasi yang mampu membedakan wajah untuk mendapatkan nilai keakuratan. Nilai keakuratan tersebut didapat dari kemampuan pemrosesan sebuah data oleh komputer *vision*. Pada bagian ini akan diuraikan jenis dan cara mendapatkan jenis dan cara mendapatkan data, serta cara pemrosesan data dan kerangka penelitian.

Tinjauan Kasus

Sistem keamanan dengan menggunakan CCTV (*Closed Circuit Television*) yang menyiratkan sinyal buatan tertutup atau tersembunyi, berbeda dengan sinyal siaran TV biasa. CCTV digunakan untuk keamanan dan juga digunakan secara luas di berbagai lokasi. Dalam peninjauan kasus ini yang perlu diperhatikan yaitu bagaimana deteksi wajah yang terekam oleh kamera CCTV hanya mampu melihat gambar sekilas dan merekam gerak gerik yang sudah direkam tetapi tidak mampu mengenali lebih akurat gambar yang ada di kamera.

Spesifikasi Rancangan Program

Dalam spesifikasi rancangan program ini memberikan tentang gambaran mengenai program yang akan dibuat, yang terdiri dari spesifikasi bentuk masukan, spesifikasi bentuk keluaran, spesifikasi file, spesifikasi program dan *flowchart*. untuk lebih jelas spesifikasi rancangan program ini akan dijelaskan kedalam beberapa sub pembahasan.

Spesifikasi Rancangan Program

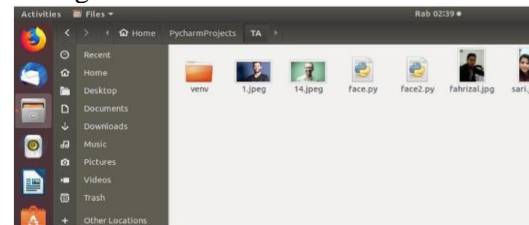
Dalam spesifikasi rancangan program ini memberikan tentang gambaran mengenai program yang akan dibuat, yang terdiri dari spesifikasi bentuk masukan, spesifikasi bentuk

keluaran, spesifikasi file, spesifikasi program dan *flowchart*. untuk lebih jelas spesifikasi rancangan program ini akan dijelaskan kedalam beberapa sub

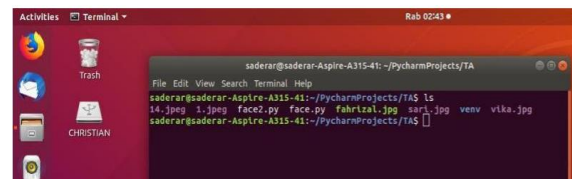
1. Nama Direktori : TA

Fungsi : Sebagai *interface* awal dan mengeksekusi fungsi– fungsi pada aplikasi *face recognition*.

Media : Terminal atau Nautilus file manager.



Gambar 3 Interface awal



Sumber : Dokumen Pribadi

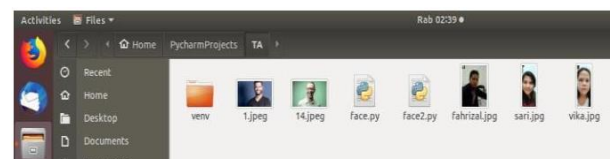
Gambar 4 Interface awal dalam bentuk CLI
(*Command Line Interface*)

Spesifikasi Bentuk Masukan

Spesifikasi bentuk Masukan merupakan rangkaian data yang masuk kedalam sistem dan diproses oleh algoritma *Machine learning* sehingga menghasilkan sesuatu *Output*. Adapun spesifikasi bentuk masukan sebagai berikut :

Nama Direktori : TA

Fungsi : Sebagai label dari kumpulan gambar atau foto yang berupa data wajah yang akan di kenali (*Known Face*)



Gambar 5 Kumpulan gambar yang di kenal (*Known Face*)

Spesifikasi Bentuk Keluaran

spesifikasi bentuk masukan. Adapun bentuk keluaran ditunjukkan pada tabel berikut:

Spesifikasi bentuk keluaran adalah dokumen yang dihasilkan dari proses

Tabel 1 Spesifikasi bentuk keluaran

N o	Nama	Fungsi	Tujuan	Media	Frekuensi
1	<i>Image processing ke Face detection</i>	Tahapan awal dalam pemrosesan gambar.	<i>User</i>	Terminal	melakukan training pada dataset yang disiapkan.
2	<i>Output Face Recognize pada Image</i>	Sebagai tahap pengujian dari dataset yang telah diproses oleh computer vision.	<i>User</i>	Terminal	Melakukan pengujian pada dataset yang telah diproses.
3	<i>Output Face Recognize pada Video secara Real time</i>	Sebagai tahap pengujian dari dataset yang telah di proses oleh computer vision	<i>User</i>	Terminal	Melakukan pengujian terhadap dataset yang telah diproses.
4	<i>Output Face Recognize pada Video secara Real time</i>	Sebagai tahap pengujian dari dataset yang telah di proses oleh computer vision	<i>User</i>	Terminal	Melakukan pengujian terhadap dataset yang telah diproses
5	<i>Output Face Recognize pada Video secara Realtime</i>	Sebagai tahap pengujian dari dataset yang telah di proses oleh computer vision	<i>User</i>	Terminal	Melakukan pengujian terhadap dataset yang telah diproses.

Spesifikasi File

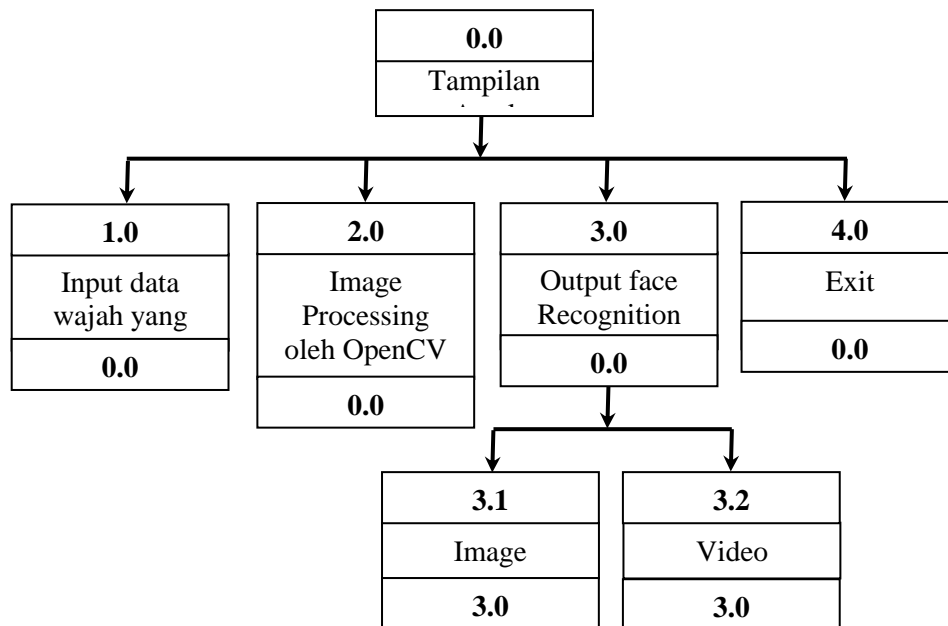
- Spesifikasi File face.py
 Nama File : face.py
 Akronim : *Face recognition on Image*
 Fungsi : *Face recognition* pada sebuah gambar atau foto
 Tipe File : *Output File*
 Akses File : Random
 Media : Solid State Drive (SSD)
 Software : PyCharm Community edition 2018 Versi 3.
- Spesifikasi File face2.py
 Nama File : face2.py

- Akronim : *Face recognition on real time video*
 Fungsi : *Face recognition* pada sebuah video secara *real time*.
 Tipe File : *Output File*
 Akses File : Solid State Drive (SSD)
 Software : PyCharm Community Edition 2018 Versi 3

HIPO (Hierarchy Input Process Output)

Dalam spesifikasi rancangan program diperlukan suatu struktur guna memahami

fungsi dari sistem, *input* ke suatu program dan *Output* yang dihasilkan.



Gambar 6 HIPO (Hierarchy *Input Process Output*)

Spesifikasi Program

Spesifikasi program ini menjelaskan nama program dan fungsi masing masing program yang sebelumnya telah digambarkan dalam bentuk HIPO.

1. Spesifikasi program *Output Face recognition on Image*

Nama Program : *Output Face recognition on Image*

Akronim : recognize.py

Fungsi : *Face recognition* pada gambar atau foto

Bahasa Program : Python

Proses : Masukan foto atau gambar yang akan diuji ke dalam folder atau direktori "Images" Eksekusi file python face.py untuk melakukan *Face recognition on Image* dengan tambahan parameter di terminal dan nama file tersebut beserta format

2. Spesifikasi program *Output Face recognition on real time video*

Nama Program : *Output Face recognition on real time video*

Akronim : recognize_video.py

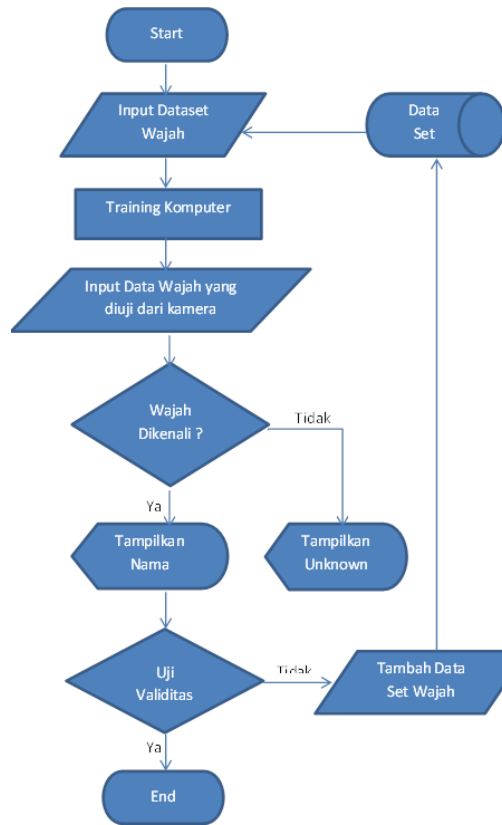
Fungsi : *Face recognition* pada sebuah video secara *real time*

Bahasa Program : Python

Proses : Eksekusi file python face.py untuk melakukan *Face recognition on real time video* dengan tambahan parameter di terminal

Flowchart

Sedangkan diagram alir / *flowchart* program yang dibuat seperti gambar IV.5 di bawah ini.



Gambar .7 Flowchart Program

Sarana Pendukung Program

a. Perangkat Keras (Hardware)

Perangkat keras (*hardware*) adalah perangkat atau *tools* yang diperlukan untuk proses mengimplementasi aplikasi yang telah dibuat. Adapun perangkat keras (*hardware*) yang diperlukan oleh aplikasi *Face recognition* ini antara lain :

Laptop

Spesifikasi perangkat keras (*hardware*) yang digunakan untuk membuat aplikasi ini adalah :

- a. Menggunakan *Laptop Processor intel inside core i3*
- b. RAM (*Random Access Memory*) 4GB DDR4 .
- c. VGA (*Video Graphic Adapter*) AMD Radeon R5

b. Perangkat Lunak (Software)

Perangkat lunak (*software*) adalah suatu aplikasi yang digunakan dalam proses pengimplementasian sistem atau aplikasi yang akan dibuat. Adapun perangkat lunak (*software*) yang diperlukan oleh aplikasi *Face recognition* dalam pembuatannya antara lain :

Sistem Operasi

- a. Linux Ubuntu dengan *Desktop Environment GNOME Versi 18.04.2 LTS (Long Term Support) Codename Bionic Beaver.*
- b. Kernel v5.1.16
- c. PyCharm *Community Edition 2018 Versi 3*
- d. Nautilus *File Manager.*

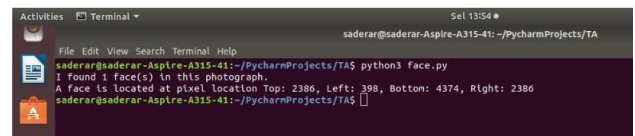
Tabel 2. Pengujian *Black Box Testing*

No	Skenario	Test Case	Hasil yang diinginkan	Keterangan
1	Tampilan hasil proses <i>Output Face detection</i> pada <i>Image</i>	Dengan mengetikkan nama file <i>Image</i> di CLI Menampilkan jumlah wajah yang dideteksi dan ditampilkan pada CLI	Program menampilkan jumlah wajah yang dideteksi dan ditampilkan pada CLI	berhasil

2	Hasil pengujian <i>Output Face detection</i> pada <i>Image</i>	Memasukkan <i>Image</i>	Program menampilkan gambar wajah tanpa <i>background (auto crop)</i>	berhasil
3	Tampilan hasil proses <i>Output Face recognition</i> pada Video	Mengarahkan kamera pada wajah yang telah didaftarkan pada program untuk dikenali (<i>Known Face</i>)	Program mengenali wajah dan menampilkan nama yang benar pada wajah tersebut secara realtime	berhasil
4	Hasil pengujian <i>Output Face recognition</i> pada video dengan wajah yang dikenal	Mengarahkan kamera pada lebih dari satu wajah yang telah didaftarkan pada program untuk dikenali (<i>Known Face</i>)	Program mengenali wajah dan menampilkan nama yang benar pada wajah tersebut secara <i>real time</i>	berhasil
5	Hasil pengujian <i>Output Face recognition</i> pada video dengan wajah yang dikenal dan tidak dikenal	Mengarahkan video pada wajah yang dikenal dan tidak dikenal	Program menampilkan nama yang benar pada wajah yang dikenal dan menampilkan <i>Unknown</i> pada wajah yang tidak dikenal	berhasil

Dari hasil pengujian diatas maka didapatkan hasil dengan menggunakan program ini cukup akurat dengan persentase 80% dengan kecepatan proses paling lambat sekitar 1 detik, dapat mendeteksi wajah pada file *Image*, bahkan dapat mendeteksi lebih dari satu wajah namun belum dapat membedakan antara gambar wajah dengan wajah asli dari kamera langsung. Selain itu pengujian belum dilakukan untuk membedakan wajah saudara kembar identik.

Program juga dapat mengenali kembali wajah-wajah yang telah didaftarkan pada sistem dan menampilkan identitas nama dari wajah yang tampil tersebut dengan benar. Dan mengenali juga wajah yang belum terdaftar pada sistem sebagai wajah yang tidak dikenal/ *unknown Face*. Hasil-hasil pengujian tersebut terlihat pada gambar-gambar lampiran dibawah ini.



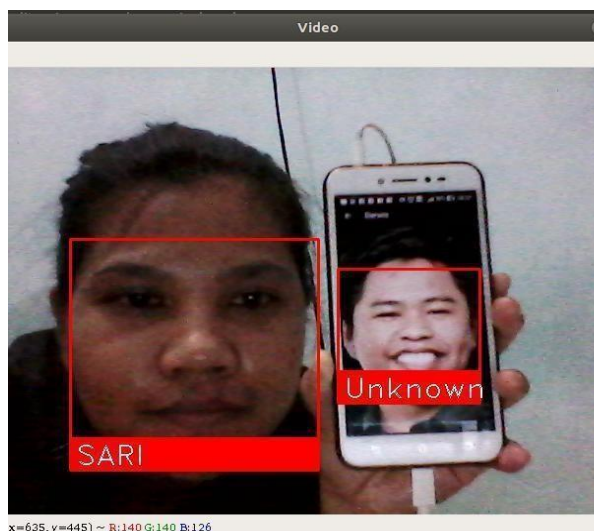
Gambar 8. Tampilan hasil proses *Output Face detection* pada *Image* di CLI



Gambar 9. Hasil pengujian *Output Face detection* pada *Image*



Gambar 10. Tampilan hasil proses *Output Face recognition* pada Video



Gambar 11. Hasil pengujian *Output Face recognition* pada video dengan wajah yang dikenal dan tidak dikenal

KESIMPULAN

Berdasarkan pembahasan dan pengujian program *face recognition*, maka menghasilkan kesimpulan sebagai berikut:

1. Semakin banyak gambar yang diambil sebagai data training dapat mempengaruhi ketajaman analisa komputer dalam mengenali objek yang diuji.
2. Kualitas kamera dapat mempengaruhi nilai keakuratan komputer dalam mengenali suatu objek.
3. System *Face recognition* sangat membantu keamanan di sebuah ruangan perkantoran sekolah maupun

gedung. Dan untuk pengembangan selanjutnya diharapkan dapat digunakan juga sebagai bagian dari sistem presensi otomatis menggunakan kamera dan pendeteksi wajah ini yang terhubung dengan basis data sistem presensi.

REFERENSI

- II, T. B. (2018). *Applied Natural language processing with Python: Implementing Machine learning and Deep Learning Algorithms for Natural Language Processing*, 158. Retrieved from <https://books.google.iq/books?hl=en&lr=&id=FDBuDwAAQB-AJ&oi=fnd&pg=PR5&dq=Beysolow+II,+T.,+2018.+Applied+Natural+Language+Processing+with+Python:+Implementing+Machine+Learning+and+Deep+Learning+Algorithms+for+Natural+Language+Processing.+Apr ess&ots=P5cG-Z7>
- Kurniawan, L. M. (2015). Metode *Face recognition* untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang. *Scientific Journal of Informatics*, 1(2), 210–220. <https://doi.org/10.15294/sji.v1i2.4027>
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1(February). <https://doi.org/10.1109/cvpr.2001.990517>

- Purwanto, P., Dirgantoro, B., & Jati, A. N. (2015). Implementasi Face Identification Dan *Face recognition* Pada Kamera Pengawas Sebagai Pendeteksi Bahaya. *EProceedings of Engineering*, 2(1), 718–724. Retrieved from <https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/2045>
- Suprianto, D. (2013). Sistem Pengenalan Wajah Secara Real-Time. *Sistem Pengenalan Wajah Secara Real-Time Dengan Adaboost, Eigenface PCA & MySQL*, 7(2), 179–184.
- Wang, Y.-Q. (2014). An Analysis of the Viola-Jones *Face detection* Algorithm. *Image processing On Line*, 4, 128–148. <https://doi.org/10.5201/ipol.2014.104>
- Xu, M. (2012). Robust object detection with real-time fusion of multiview foreground silhouettes. *Optical Engineering*, 51(4), 047202. <https://doi.org/10.1117/1.oe.51.4.047202>