

IMPLEMENTASI FACE DETECTION DAN RECOGNITION MENGUNAKAN PYTHON DENGAN NUMPY DAN OPENCV MENGUNAKAN METODE HAAR-CASCADE DAN LBPH (LOCAL BINARY PATTERN HISTOGRAM)

Susitanto M Taib^a, Sakinah Sudin^b, Abdul Haris Muhammad^c

^aMahasiswa Program Studi Teknik Informatika,

^{bc}Dosen Teknik Informatika Universitas Muhammadiyah Maluku Utara, Ternate, Indonesia

email: susyantoa281@gmail.com^a, sakinahsudin80@yahoo.co.id^b, agry.arisandi@gmail.com^{bc}

ABSTRAK

Pengenalan citra wajah manusia merupakan salah satu teknologi penting yang terus berkembang pada bidang computer vision dengan penerapannya dalam sistem pengenalan biometrik, Sistem pencarian, pengindeksan pada database video digital, sistem keamanan kontrol akses area terbatas, konferensi video, interaksi manusia dengan komputer dan lain sebagainya. Penelitian ini menerapkan metode *Haar-Cascade* untuk membangun sistem deteksi wajah dengan bahasa pemrograman Python. Metode *Haar-Cascade* merupakan salah satu metode deteksi wajah dengan tingkat akurasi yang tinggi dan komputasi yang cepat. Dari hasil pengujian deteksi wajah; python; *haar-cascade*. Kemudian juga menggunakan metode *LBPH(Local Binary Pattern Histogram)* Pengenalan wajah merupakan proses lanjutan dari proses pendeteksian wajah. Di dalam pendeteksian wajah yaitu mendeteksi bagian wajah dari seseorang, wajah tersebut bisa didapatkan dari gambar maupun video. Dengan memanfaatkan hasil *training* dari *haar cascade*. Kemudian hasil dari proses ini dikombinasikan dengan proses *Image Matching* dengan algoritma *Local Binary Pattern Histogram*.

Kata Kunci : *Face Detection, Recognition, Python, Haar-Cascade, LBPH*

1. PENDAHULUAN

Mengenali wajah dalam sebuah gambar atau foto, bagi manusia merupakan hal yang mudah. Tetapi tidak demikian untuk komputer, agar dapat mengenali wajah manusia diperlukan khusus sehingga ketika diberi *input* sebuah gambar atau foto, komputer dapat mendeteksi apakah dalam sebuah gambar tersebut terdapat sebuah gambar wajah atau tidak mengenali wajah tersebut.

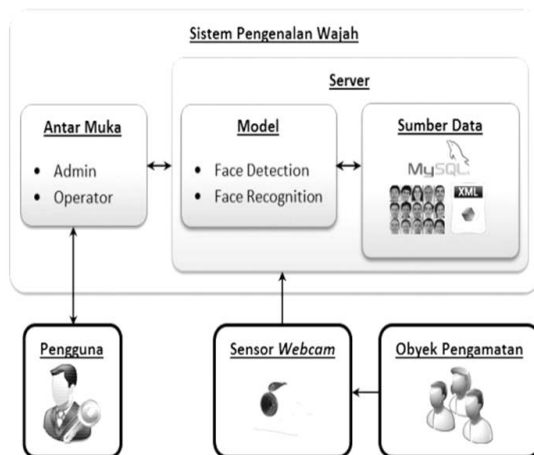
System keamanan pada suatu gedung lebih banyak menggunakan *system* manual sebagai bentuk pengawasannya, seperti dengan menggunakan kunci manual. *System* keamanan seperti ini masih belum *efektif* karena selama ini banyak orang yang tidak berkepentingan bisa masuk ke dalam ruangan tersebut, sehingga informasi - informasi penting dapat dicuri dengan mudah. Oleh karena itu diperlukan teknologi untuk mengamankan informasi

penting tersebut, seperti penggunaan pengenalan wajah atau *face recognition* sebagai media pengamanannya. *Face recognition* adalah suatu teknologi yang memungkinkan pengguna untuk mengidentifikasi wajah seseorang melalui gambar digital. Teknologi ini adalah perkembangan dari teknologi *image processing* [1]. *Face recognition* berfungsi juga untuk mengenali wajah seseorang untuk dikenali. Dalam *face recognition* terdapat dua aktivitas yang harus dilakukan yaitu mencari data training serta data testing, untuk pengujian gambar yang akan diuji. *Face recognition* memiliki banyak jenis *algoritma* yang dapat digunakan untuk mendeteksi dan mengenali wajah seseorang. Salah satu *algoritma* dalam pengenalan wajah adalah *algoritma eigenface*. Konsep dari *eigenface* adalah serangkaian *eigenvector* yang digunakan untuk mengenali wajah manusia dalam suatu *computer vision* [2]. *Algoritma* ini

mempunyai keakuratan yang cukup baik dibanding *metode face recognition* yang lain. Metode *eigenface* merupakan pengenalan wajah berdasarkan *Principle Component Analysis*, pada *eigenface* gambar dicapture dan disimpan pada database untuk menjadi data *training* yang kemudian akan dibandingkan pada data *sample*. Pada tugas akhir ini memanfaatkan karakteristik wajah seseorang untuk dijadikan hak akses membuka kunci pintu, menggantikan kunci fisik manual.

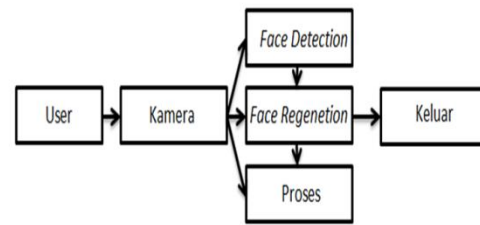
2. METODOLOGI PENELITIAN

Model Algoritma yang digunakan dalam pengenalan wajah untuk melakukan *face detection* dan *Face Recognition* yang berhubungan dengan sumber data berupa dataset wajah FERET (berupa file *XML*) saat dilakukan pendeteksian wajah, data profile yang disimpan dalam *database MySQL* dan berkas wajah berupa citra wajah berformat *JPEG*. Adapun kerangka dalam pengenalan waja sebagai berikut:



Gambar 1. Kerangka pengenalan wajah

Pada umumnya sistem deteksi wajah (*face detectetion*) dan pengenalan wajah (*face regenetion*) saat ini dapat dilihat konsepnya sebagai berikut :



Gambar 2. Sistem Deteksi dan Mengenali Wajah

3. PEMBAHASAN

1. Rancangan Penelitian

Dalam melaksanakan penelitian ini, tentunya dibutuhkan rancangan-rancangan penelitian yang akan dilakukan. Penelitian dirancang agar lebih terstruktur pengerjaannya dari awal memulai penelitian sampai akhir selesainya penelitian. Selain itu, rancangan penelitian ini terdiri dari langkah-langkah atau tahapan-tahapan yang akan dilakukan. Langkah-langkah tersebut antar lain:

Memiliki sejumlah *Refrensi*, sebagai bahan pertimbangan dalam membangun sebuah sistem. Adapun perancangan ini menggunakan bahasa *pemrograman python*.

2. Perancangan Sistem

Perancangan sistem merupakan tahap selanjutnya setelah analisa sistem, mendapat gambaran yang jelas tentang apa yang di kerjakan pada sistem, maka dilanjutkan dengan memikirkan bagaimana membentuk sisten tersebut dalam sebuah perancangan sistem *face detection* dan *face recognition* dengan menggunakan pemrograman *Python*, *openCV* dan *Numpy*.

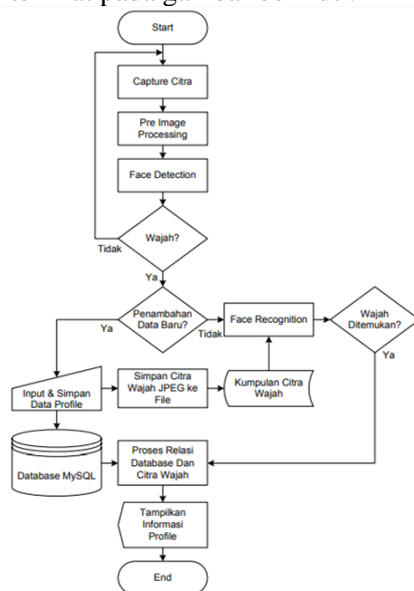
3. Analisis

Analisis dilakukan untuk mengurai secara mendalam terhadap perancangan yang akan dilakukan. Pada perancangan deteksi wajah ini juga dilakukan analisis seperti analisis kebutuhan dan analisis interaksi.

Analisis kebutuhan merupakan

langkah awal untuk menentukan sebuah desain sistem yang akan dilakukan. Dalam perancangan deteksi wajah ini, dibutuhkan banyak informasi mengenai pendeteksian wajah seperti teknik serta peralatan yang digunakan untuk merancang sebuah aplikasi deteksi wajah.

Untuk mengawali suatu perancangan *face detectetion* dan *face regenetion* pada tahapan ini dilakukan sebuah gambaran secara umum alur sistem secara keseluruhan, sebagaimana terlihat pada gambar berikut :



Gambar 3. Alur Kerja Pengenalan Wajah

4. Perangkat Penelitian

a. Perangkat Keras

Perangkat keras yang digunakan pada saat pembuata sistem adalah seperangkat komputer dengan spesifikasi antara lain :

1. Platform : Laptop
2. Prosesor : Intel(R) Core(TM) i3-700U CPU @ 2.40GHz 2.40GHz
3. RAM : 4,00 GB
4. HDD : 500GB
5. VGA : ATI Radeon HD 5450
6. Webcam: USB 2.0 Camera Standart

b. Perangkat Lunak

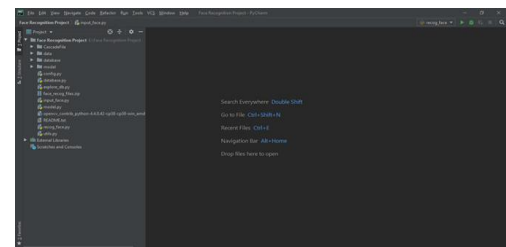
Selain perangkat keras yang digunakan untuk mendukung sistem, digunakan juga perangkat-perangkat lunak antara lain:

1. Sistem Operasi : Windows 10 64 bit
2. Python 3.8.3
3. Pycharm Community edition 2020.1.2x64
4. CMD
5. Numpy 1.18.5
6. OpenCV 4.2.0.34

5. Implementasi Sistem

Sistem kemudian dibuat dengan menggunakan perangkat-perangkat yang telah disebutkan di atas. Sistem dibuat bukanlah sistem informasi besar yang menengani banyak operasi dan alur yang kompleks.

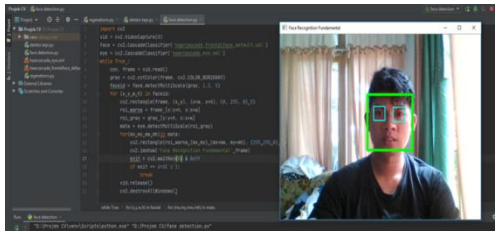
a. Pembuatan sistem face detection dan recognition sistem face detection dan recognition dibuat menggunakan pycram yang merupakan salah satu libray python 3.8.3 dengan menggunakan ini akan memudahkan dalam pembuatan program untuk membuat tampilan. Berikut ini adalah tampilan pycram dalam membuat program :



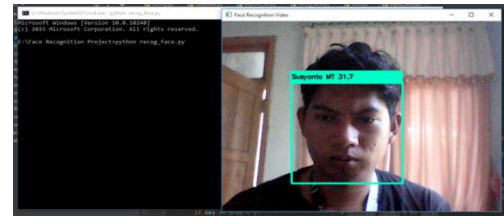
Gambar 4. Tampilan Pycram Community edition 2020.1.2x64

b. Tampilan face detection

Tampilan face detection adalah mendeteksi wajah dengan menggunakan webcam yang berada dalam sistem komputer. Berikut ini adalah Tampilan face detection:



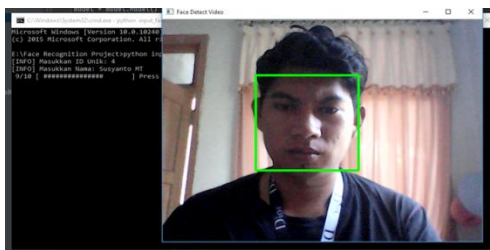
Gambar 5. Tampilan *face detection*



Gambar 8. Tampilan *face recognition*

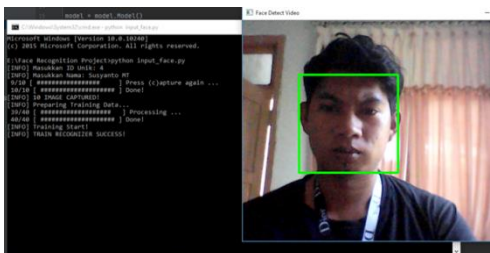
c. Tampilan *face recognition*

Face recognition adalah tampilan untuk mengenali wajah, langkah-langka untuk mengenali pemilik wajah yang tampil pada *camera webcam* yang berada dalam *sistem* yaitu, menginput wajah dan informasi ID dan nama dari wajah tersebut. Selanjut akan dilakukan *capture* data wajah :



Gambar 6. Tampilan *face detection*

Setelah wajah di *capture* data-data tersebut akan dilakukan *training* untuk pengujian data wajah yang sudah di *capture*.



Gambar 7. Tampilan *training* data wajah

d. Tahapan *Face recognition* atau Pengenalan Wajah

Tahapan selanjutnya dari kumpulan data wajah yang dilakukan *training* kemudian proses untuk mengenali wajah-wajah tersebut.

e. Implementasi *face detection* dan *face recognition*

Penerapan *face detection* dan *face recognition* dimulai dengan menyanding 2 data wajah dan ID-nya masing-masing untuk melakukan pengujian mengenali wajah-wajah yang sudah dilakukan *training*.

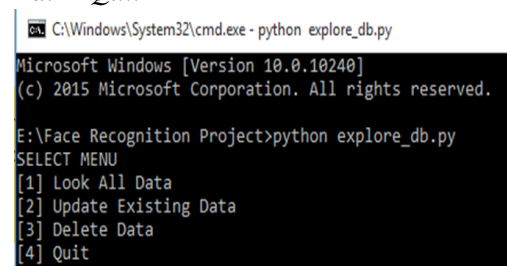


Gambar 9. *face detection* pada 2 wajah

f. Database

Untuk melihat face yang tersimpan di *database* "*python explore_db.py*" terdapat menu *face recognition* yaitu:

- Look All Data
- Update Existing Data
- Delete Data
- Quit



Gambar 10. Tampilan *DataBase*

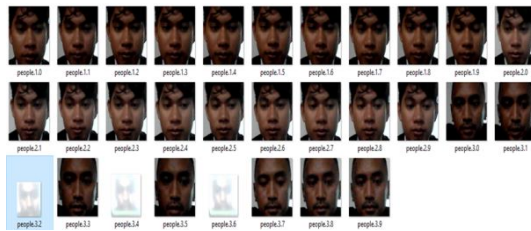
6. Pemrograman *face detection* dan *recognition*

Deteksi wajah merupakan salah satu hal yang sangat menarik untuk dilakukan, terlebih lagi dalam pengolahan *citra*.

Dengan menggunakan algoritma *Haar cascade* untuk mendeteksi dan *LBPH* untuk mengenali wajah.

7. Analisis terhadap *face detection* dan *face recognition*

Sejumlah data set yang dilakukan sebagai langkah awal dalam mendeteksi dan mengenali wajah disertai dengan keterangan sangat berpengaruh terhadap keberhasilan sistem *detection*. Factor lain yang berpengaruh terhadap jumlah data set adalah factor pencahayaan, factor jarak serta factor posisi objek wajah tersebut. Adapun sejumlah data set yang dilakukan sebagai data training adalah sebagai berikut:



Gambar 11. Tampilan *DataSet*

Selanjutnya factor lain yang dapat berpengaruh terhadap hasil *face detection* dan *face recognition* adalah pencahayaan terhadap objek tersebut dapat dilihat pada tabel berikut :

Tabel 1. Hasil Faktor Pencahayaan Terang dan Gelap

Faktor	Nilai	
Pencahayaan	Terang	1. Tidak dikenali sebagai pengguna awal nilai akurasi 28%
		2. Tidak dikenali sebagai pengguna awal nilai akurasi 36%
		3. Tidak dikenali sebagai pengguna awal nilai akurasi 20%
		4. Tidak dikenali sebagai pengguna awal nilai akurasi 18%
		5. Tidak dikenali sebagai pengguna awal nilai akurasi 22%
	Gelap	1. Tidak dikenali sebagai pengguna awal nilai akurasi 57%
		2. Tidak dikenali sebagai pengguna awal nilai akurasi 55%
		3. Tidak dikenali sebagai pengguna awal nilai akurasi 45%
		4. Tidak dikenali sebagai pengguna awal nilai akurasi 51%
		5. Tidak dikenali sebagai pengguna awal nilai akurasi 22%

Dari tabel diatas menunjukkan bahwa pencahayaan terhadap sejumlah objek wajah tidak dapat dikenali baik dalam pencahayaan terang maupun gelap. Pada pencahayaan terang, wajah tidak dapat dikenali mencapai 36% nilai akurasinya sedangkan pada kondisi

pencahayaan gelap, wajah tidak dapat dikenali mencapai 57%. Objek data set tersebut dijadikan data testing sebagai pengguna awal pada *face detection*.

Berikut ini adalah hasil analisis faktor jarak terhadap wajah dekat dan jauh dengan melakukan percobaan sebanyak 5 kali pada data set wajah yang telah tersimpan.

Pada tabel jarak wajah, dapat dilihat bahwa pengujian jarak dekat 20-30 cm dapat dikenali dengan nilai akurasi mencapai 70%. Sedangkan jarak jauh 31-60 cm masih dapat dikenali dengan nilai akurasi deteksi mencapai 73%.

Tabel 2. Hasil Faktor Jarak Wajah Dekat dan Jauh

Faktor	Nilai	
Jarak Wajah	Dekat (20-30 cm)	1. Wajah dikenali nilai akurasi 70%
		2. Wajah dikenali nilai akurasi 67%
		3. Wajah dikenali nilai akurasi 70%
		4. Wajah dikenali nilai akurasi 65%
		5. Wajah dikenali nilai akurasi 68%
	Jauh(31-60cm)	1. Wajah dikenali nilai akurasi 60%
		2. Wajah dikenali nilai akurasi 60%
		3. Wajah dikenali nilai akurasi 60%
		4. Wajah dikenali nilai akurasi 73%
		5. Wajah dikenali nilai akurasi 72%

Berikut ini adalah tabel hasil analisa faktor posisi wajah hadap atas, depan dan bawah, dengan melakukan percobaan sebanyak 5 kali pada wajah orang yang telah tersimpan pada data set data wajah.

Tabel 3. Hasil Faktor Posisi Wajah Hadap Atas, Depan dan Bawah

Faktor	Nilai	
Posisi Wajah	Atas	1. Tidak dikenali sebagai pengguna awal nilai akurasi 57%
		2. Tidak dikenali sebagai pengguna awal nilai akurasi 55%
		3. Tidak dikenali sebagai pengguna awal nilai akurasi 45%
		4. Tidak dikenali sebagai pengguna awal nilai akurasi 51%
		5. Tidak dikenali sebagai pengguna awal nilai akurasi 52%
	Depan	1. Wajah dikenali nilai akurasi 69%
		2. Wajah dikenali nilai akurasi 71%
		3. Wajah dikenali nilai akurasi 68%
		4. Wajah dikenali nilai akurasi 70%
		5. Wajah dikenali nilai akurasi 67%
	Bawah	1. Wajah dikenali nilai akurasi 69%
		2. Wajah dikenali nilai akurasi 65%
		3. Wajah dikenali nilai akurasi 68%
		4. Wajah dikenali nilai akurasi 64%
		5. Wajah dikenali nilai akurasi 67%

Hasil pengujian pada posisi wajah menghadap depan dan bawah dengan 5 sample yang sama dapat dikenali wajah pengguna awal,

dengan nilai akurasi lebih tinggi hingga 71% dan pada posisi wajah hadap atas tidak di kenali sebagai pengguna awal karena nilai akurasi dibawah 60% atau hanya 57%. Rata-rata nilai akurasi pada posisi wajah hadap depan adalah 69% dan rata-rata nilai akurasi pada posisi wajah hadap bawah adalah 66,6%.

4. KESIMPULAN

Berdasarkan pembahasan dan hasil uji coba *face detection* dan *face recognition* melalui kamera *webcam* pada sistem komputer maka di simpulkan sebagai berikut :

1. Proses pengenalan wajah sangat tidak optimal terhadap perubahan cahaya, jarak, atribut wajah, dan perubahan yang terlalu ekstrim. Jika *citra* yang digunakan sebagai data *training* maupun data *test face* memiliki intensitas cahaya yang berbeda dan tidak berada pada posisi yang sama maka proses tersebut tidak dapat memberikan hasil yang akurat.
2. *Metode haar cascade classifier* sangat ideal digunakan untuk deteksi wajah secara *realtime* yang di *capture* melalui *webcam laptop* .
3. *Metode Haar-Cascade* proses pendeteksian wajah dilakukan dengan mengklasifikasikan sebuah gambar setelah sebelumnya sebuah pengklasifikasi dibentuk dari data set.
4. *Metode recognition* wajah menggunakan *metode LBPH(Local Binary Pattern Histogram)* sistem pengenalan wajah menggunakan *algoritma Local Binary Pattern Histograms* untuk pengolahan fitur-fitur wajah dan *Haar Cascade* untuk pengambilan *frame* wajah.
5. Hasil uji sistem mendeteksi wajah dan pengenalan wajah dapat mendeteksi *objek* wajah secara *real time*.
6. Konsep deteksi nilai *T (true)* dapat mendeteksi wajah dengan berbagai posisi wajah yang benar.
7. Hasil nilai pada wajah adalah jarak data wajah yang tersimpan dengan jarak wajah yang terdeteksi.
8. Pada pengujian secara *realtime* jika *citra* wajah terhalang oleh objek lain maka *citra* wajah tersebut tidak akan terdeteksi .

REFERENSI

1. Asti Riani Putri, 2016, “*Pengolahan Citra dengan Menggunakan Webcam pada Kendaraan Bergerak di Jalan Raya*”, JIPI (Jurnal Ilmiah Pendidikan Informatika) Volume 1, Nomor1, Tahun 2016: 1-6.
2. Al-Aidid, S., & Pamungkas, D. (2018). *Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram*. Jurnal Rekayasa ElektriKa, 14(1), 62– 67. <https://doi.org/10.17529/jre.v14i1.9799>
3. Anggraini, N., Rozy, N. F., & Lazuardy, R. A. (2013). *Facial Recognition System For Fatigue Detection Using Intel Realsense Technology* Departement of Informatics Engineering Faculty of Science and Technology State Islamic University of Syarif Hidayatullah Jakarta. (may).
4. Bruno, L. (2019). *Aplikasi Pendeteksian Ras Kucing dengan Mendeteksi Wajah Kucing dengan Metode Viola-Jones berbasis Android* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
5. Emami, S., & Suci, V. P. (2012). *Facial Recognition using OpenCV*. Journal of Mobile, Embedded and Distributed Systems, 4(1), 38–43. Retrieved from <http://www.jmeds.eu/index.php/jmeds/article/view/57>
6. Ivanjul.com. (2018). *Fungsi Membaca, Menampilkan, dan Menyimpan Gambar OpenCV Python*. Retrieved December 21, 2019, from <https://www.ivanjul.com/fungsi-membaca-menampilkan-dan-menyimpan-gambar-opencv-python>
7. Iqbal, Muhammad, 2009, *Dasar Pemrograman Citra Menggunakan MATLAB*, Institut Pertanian Bogor, [ebook] (diunduh pada tanggal 12 Juni 2014).
8. Petkov N., Wieling, M.B., *Gabor Filter for Image Processing and Computer Vision*, University of Groningen, Department of Computing Science, [online] (diakses pada tanggal 23 Juli 2014)
9. Satriyanto, Edi. (2010). *Clustering*. Diakses pada 9 Februari 2015, dari <http://lecturer.eepis-its.edu/~kangedi>.