

# Pengembangan Sistem *Doorcam* Berbasis Mini PC Raspberry Pi dengan Menggunakan *Face Recognition*

S I Lestaringati, M Y Shiddiq

**ABSTRACT** -- *Doorcam* is a tool with a system that functions as a notification for homeowners. By using the Raspberry Pi Camera Module, residents of the house can find out who is visiting guests from anywhere. This system can provide information regarding visitors who come to visit. Communication between residents and guests can also be done by using a microphone and speaker via Telegram. The development of the Raspberry pi mini PC based doorcam system will deny the intelligence that can support the faces of people who come to visit by utilizing the OpenCV library in the Python programming language. The system will send information on the name of the guest who has arrived if it has been identified or provides another notification if the guest is not yet known by the system. In this case, the occupants of the house (users) can add data or update people's data that has been recognized by the system. In addition, this study also aims to improve the image and sound quality of existing systems..

**ABSTRAK** -- *Doorcam* adalah sebuah alat dengan sistem yang berfungsi sebagai notifikasi tamu kepada pemilik rumah. Dengan menggunakan *Raspberry Pi Camera Module*, maka penghuni rumah dapat mengetahui tamu yang berkunjung ke rumah dari manapun. Sistem ini dapat memberikan informasi terkait pengunjung yang datang bertamu. Komunikasi antara penghuni rumah dan tamu juga dapat dilakukan dengan menggunakan *microphone* dan *speaker* melalui Telegram. Pengembangan sistem *doorcam* berbasis mini PC Raspberry pi akan disematkan sebuah kecerdasan yang dapat mengenali wajah orang yang datang bertamu dengan memanfaatkan *library* OpenCV dalam bahasa pemrograman Python. Sistem akan mengirimkan informasi nama tamu yang datang jika telah dikenali ataupun memberikan notifikasi yang lain jika tamu belum dikenal oleh sistem. Dalam hal ini, penghuni rumah (*user*) dapat menambahkan data ataupun memperbaharui data orang yang telah dikenali oleh sistem. Selain itu, penelitian ini juga bertujuan untuk meningkatkan kualitas gambar dan suara dari sistem yang telah ada.

## Pendahuluan

Sistem *Doorcam* berbasis Mini PC Raspberry Pi yang mulai digarap pada tahun 2017, oleh Bayu Hendra Nugroho memungkinkan tamu dapat memberikan notifikasi kepada pemilik rumah melalui aplikasi Telegram dengan menekan sebuah tombol. *Doorcam* dapat berjalan dengan baik untuk sekedar mengirim gambar. Setelah itu, mengalami pengembangan yang dilakukan oleh Silvia Gusti Anggraeni pada tahun 2018 dimana memungkinkan tamu untuk mengirim pesan gambar dan

pesan suara melalui aplikasi Bot Telegram. Tampilan GUI juga ditambahkan untuk memberikan kesan yang lebih baik.

Pengembangan sistem *Doorcam* sejak awal sampai hari ini selalu memanfaatkan kamera untuk proses pengambilan gambar. Namun, bukan berarti sistem *Doorcam* telah berjalan dengan baik tanpa perlu improvisasi. Sistem yang telah ada hanya dapat menampilkan gambar dengan *framerate* yang sangat kecil yaitu 2 *fps* (*frame per Second*), artinya sistem

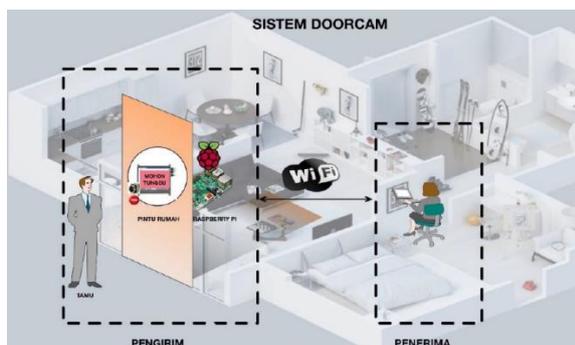
menampilkan gambar pada LCD memberikan kesan yang sangat lambat.

Maka dari itu, pengembangan kali ini akan menitikberatkan pada fungsi kamera. Selain berfungsi sebagai alat pemrosesan gambar, kamera juga dapat dimanfaatkan untuk sistem keamanan dan pengenalan objek tertentu. Kamera telah dapat melakukan proses pengenalan wajah untuk dimanfaatkan sebagai kebutuhan keamanan. Sistem *Face Recognition* (Pengenalan wajah) merupakan salah satu pemanfaatan dari fungsi utama kamera. Bahkan beberapa korporasi multinasional menghapuskan sistem keamanan sidik jari yang kemudian digantikan dengan *Face ID* (Identitas wajah).

Dari sinilah pengembangan kali ini akan dilakukan, yaitu dengan menggarap sistem pengenalan wajah yang selanjutnya akan diimplementasikan pada sistem *Doorcam* berbasis Mini PC Raspberry Pi. Selain itu, sistem lama akan ditingkatkan sehingga memungkinkan untuk menampilkan gambar sampai dengan 10 *fps*. Pada jurnal ini, akan dibahas lebih dalam tentang proses pendeteksian dan pengenalan wajah serta bagaimana penerapannya pada sistem *Doorcam* yang telah ada sebelumnya.

### Metode dan Bahan

Gambar 1 adalah gambaran umum dari sistem *Doorcam*. Terlihat ada dua orang yang diasumsikan sebagai tamu dan yang lain sebagai pemilik rumah.



Gambar 1. Gambaran Umum Sistem

Untuk menerapkan sistem *face recognition* pada sistem diperlukan banyak komponen yang dapat diolah sehingga membuat sistem berjalan sesuai dengan yang diharapkan. Kebutuhan-kebutuhan ini terangkum dalam dua komponen utama yaitu *Hardware* dan *Software*.

Tabel 1. Komponen Kebutuhan *Hardware*

Komponen	Spesifikasi	Jumlah
Raspberry Pi versi 3	RAM 1 GB Ethernet Port	1
Pi-Camera Module	Resolution 5 MP Supports 1080p at 30 fps video	1
LCD for Raspberry Pi	7" IPS Screen Five-points touch	1

Tabel 2. Komponen Kebutuhan *Software*

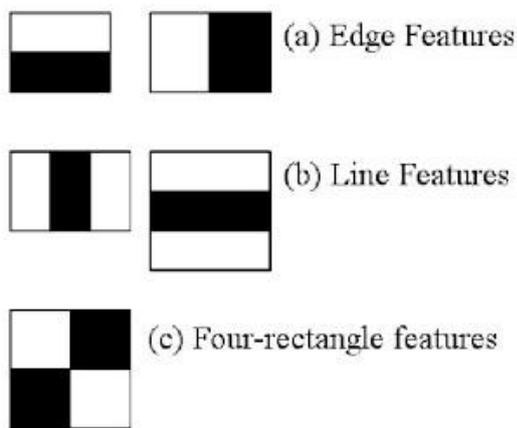
No	Tipe Software	Nama Software
1	Sistem Operasi Raspberry Pi	Raspbian Jessie
2	Bahasa Pemrograman	Python v3.4
3	Komunikasi Data	BotTelegram

Terlihat pada tabel 2 bahwa bahasa pemrograman yang dipakai untuk membangun sistem *face recognition* ini adalah Python v3.4, yang mana sebelumnya menggunakan v2.6.

Setelah memastikan semua kebutuhan dan bahan telah terpenuhi, langkah selanjutnya adalah memahami bagaimana sistem pengenalan wajah ini berjalan.

Metode yang digunakan pada pemrosesan wajah menggunakan metode *Haar Cascade Classifier*. Metode ini akan diterapkan pada modul *face recognition* yang akan dibuat. Sebelumnya akan dijelaskan tentang apa itu *Haar cascade*.

Di sini kita akan bekerja dengan deteksi wajah. Awalnya, algoritma membutuhkan banyak gambar positif (gambar wajah) dan gambar negatif (gambar tanpa wajah) untuk melatih pengklasifikasi. Maka kita perlu mengekstrak fitur darinya. Untuk ini, fitur Haar yang ditunjukkan pada gambar di bawah ini digunakan. Mereka seperti kernel konvolusional kami. Setiap fitur adalah nilai tunggal yang diperoleh dengan mengurangi jumlah piksel di bawah kotak putih dari jumlah piksel di bawah kotak hitam.

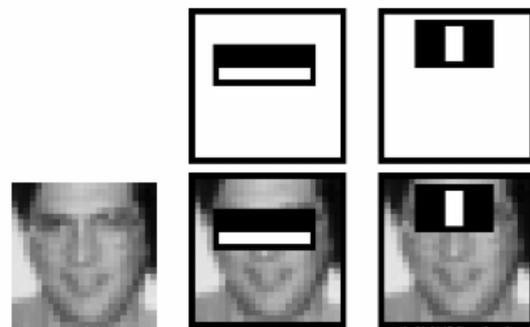


Gambar 2. *Haar-cascade* hitam putih

Sekarang, semua ukuran dan lokasi yang mungkin dari masing-masing kernel digunakan untuk menghitung banyak fitur. (Bayangkan saja berapa banyak perhitungan yang dibutuhkan? Bahkan kotak berukuran 24x24 menghasilkan lebih dari 1.600 fitur). Untuk setiap perhitungan fitur, kita perlu menemukan jumlah piksel di bawah persegi panjang putih dan hitam. Untuk mengatasi ini, mereka memperkenalkan gambar integral. Seberapa besar pun gambar, itu mengurangi

perhitungan untuk piksel yang diberikan untuk operasi yang hanya melibatkan empat piksel.

Tetapi di antara semua fitur yang kami hitung, sebagian besar tidak relevan. Sebagai contoh, perhatikan gambar 3. Baris atas menunjukkan dua fitur bagus. Fitur pertama yang dipilih tampaknya berfokus pada properti bahwa daerah mata seringkali lebih gelap daripada daerah hidung dan pipi. Fitur kedua yang dipilih bergantung pada properti bahwa mata lebih gelap daripada pangkal hidung. Tetapi jendela yang sama diterapkan pada pipi atau tempat lain tidak relevan.



Gambar 3. *Haar-Cascade* Pada Wajah

Untuk ini, diterapkan setiap fitur pada semua gambar pelatihan. Untuk setiap fitur, menemukan ambang terbaik yang akan mengklasifikasikan wajah ke positif dan negatif. Jelas, akan ada kesalahan atau kesalahan klasifikasi. Kami memilih fitur dengan tingkat kesalahan minimum, yang berarti mereka adalah fitur yang paling akurat mengklasifikasikan gambar wajah dan non-wajah. (Prosesnya tidak sederhana ini. Setiap gambar diberi bobot yang sama di awal. Setelah setiap klasifikasi, bobot gambar yang salah diklasifikasikan ditingkatkan. Kemudian proses yang sama dilakukan. Tingkat kesalahan baru dihitung. Juga bobot baru. proses dilanjutkan hingga akurasi atau tingkat kesalahan yang diperlukan tercapai atau sejumlah fitur yang diperlukan ditemukan).

Penggolong akhir adalah jumlah terbobot dari penggolong lemah ini. Disebut lemah karena itu sendiri tidak dapat mengklasifikasikan gambar, tetapi bersama-sama dengan yang lain membentuk penggolong yang kuat. Makalah itu mengatakan bahkan 200 fitur menyediakan deteksi dengan akurasi 95%. Pengaturan akhir mereka memiliki sekitar 6000 fitur. (Bayangkan pengurangan dari 160000+ fitur ke 6000 fitur.

Jadi sekarang Anda mengambil gambar. Ambil setiap jendela 24x24. Terapkan 6000 fitur untuk itu. Periksa apakah wajah atau tidak. Wow .. Bukankah ini sedikit tidak efisien dan memakan waktu? Ya itu. Penulis punya solusi yang bagus untuk itu.

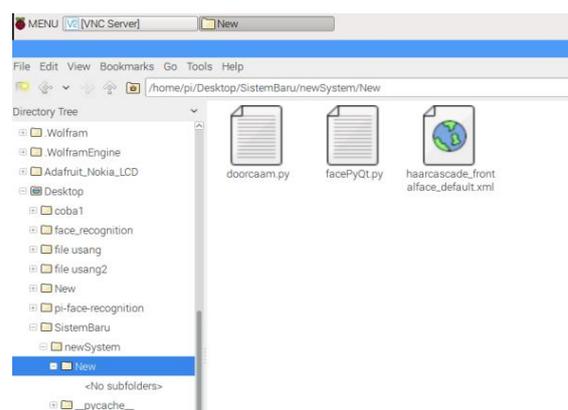
Dalam sebuah gambar, sebagian besar gambar adalah daerah non-wajah. Jadi itu adalah ide yang lebih baik untuk memiliki metode sederhana untuk memeriksa apakah jendela bukan daerah wajah. Jika tidak, buang saja dalam satu tembakan, dan jangan diproses lagi. Alih-alih, fokuslah pada daerah di mana ada wajah. Dengan cara ini, kami menghabiskan lebih banyak waktu memeriksa kemungkinan daerah wajah.

Untuk ini mereka memperkenalkan konsep Cascade of Classifiers. Alih-alih menerapkan semua 6000 fitur pada sebuah jendela, fitur-fitur tersebut dikelompokkan ke dalam beberapa tahapan pengklasifikasi dan diterapkan satu per satu. (Biasanya beberapa tahap pertama akan mengandung fitur yang jauh lebih sedikit). Jika jendela gagal pada tahap pertama, buanglah. Kami tidak mempertimbangkan fitur yang tersisa di dalamnya. Jika lewat, terapkan fitur tahap kedua dan lanjutkan prosesnya. Jendela yang melewati semua tahapan adalah daerah wajah.

Setelah memahami konsep teori di atas, kini akan diterapkan pada sebuah modul

openCV pada pemrograman Python. Untuk mengidentifikasi sebuah wajah, harus dibuat beberapa modul diantaranya adalah modul GUI (*Graphic User Interface*) yang berfungsi sebagai tampilan antarmuka sistem dengan pemakai (dalam hal ini adalah tamu), modul OpenCV untuk melakukan pemrosesan gambar, modul *haar-cascade-classifier* dalam format .xml yang berguna sebagai identifikasi wajah sebagaimana yang telah dijelaskan pada pembahasan teori di atas, serta modul *face recognition* yang akan dipakai untuk membandingkan gambar yang disimpan dengan gambar yang diambil dari Pi-Camera.

Untuk memudahkan pemrosesan beberapa modul akan dibuat dalam sebuah folder yang sama.



Gambar 4. Folder Modul

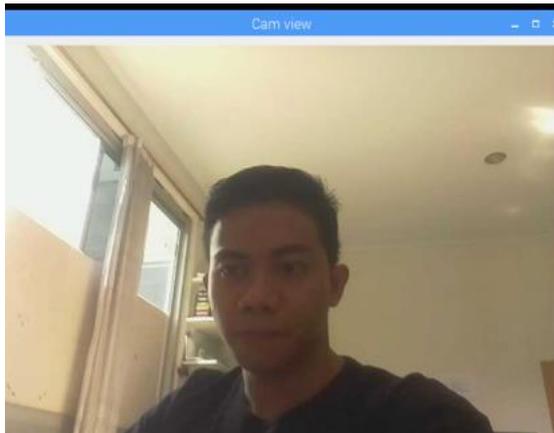
Jika semua telah dilakukan, maka akan dimulai proses eksekusi sistem pada terminal Raspberry Pi. Apabila sistem berjalan dengan baik maka proses akan menghasilkan nilai yang sesuai untuk mendeteksi wajah.

## Pembahasan dan Pengujian

Hal pertama yang dilakukan adalah mengonfigurasi semua sistem pada Raspberry Pi dengan menggunakan terminal. Pastikan waktu sesuai dengan

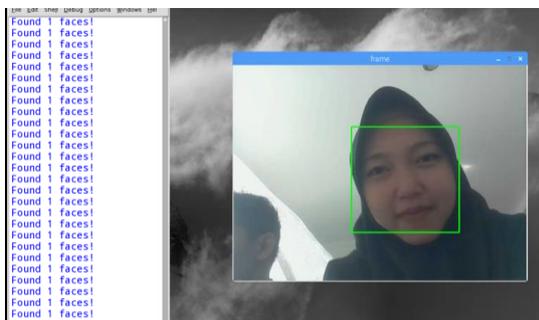
waktu global untuk menghindari kegagalan pada saat proses *update* sedang berlangsung. Kemudian memulai proses pengodean pada modul-modul yang sudah disiapkan sebelumnya.

Untuk memastikan sistem telah berjalan dengan baik maka diperlukan pengujian beberapa fungsi.

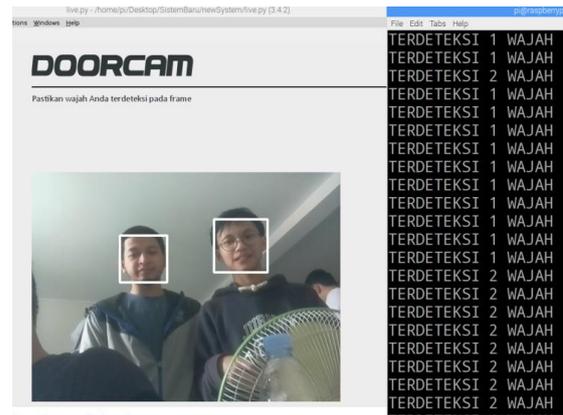


Gambar 5. Pengujian Pi-Camera

Pada gambar 5, dilakukan pengujian kualitas gambar dengan Pi-Camera v1.3 untuk memastikan sistem dapat menangkap gambar yang jernih sehingga memudahkan proses pendeteksian wajah. Jika gambar telah dirasa pas dan cukup untuk melakukan *image processing*, akan dimulai tahap pengodean pada modul pendeteksian wajah.

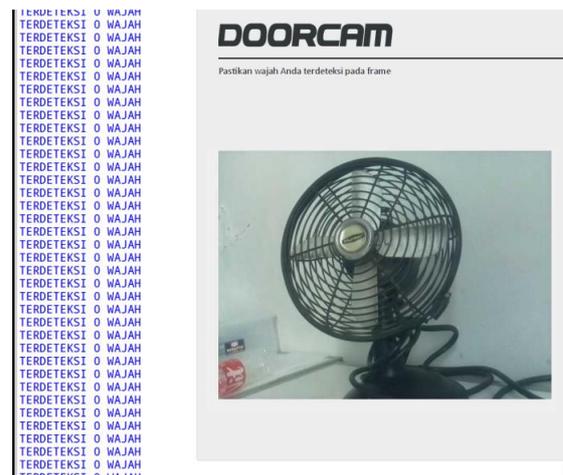


Gambar 6. Pengujian Ke-1 Deteksi Wajah



Gambar 7. Pengujian Ke-2 Deteksi Wajah

Setelah melakukan pengujian pada wajah maka dilakukan pengujian terhadap objek lain selain wajah untuk memastikan sistem dapat mendeteksi dengan benar.



Gambar 8. Pengujian Objek selain Wajah

Setelah melakukan beberapa percobaan yang melibatkan wajah dan objek lain barulah didapatkan hasil pengujian apakah sistem telah bisa membaca wajah atau bukan. Dalam percobaan yang benar-benar aktual sebenarnya terjadi beberapa kesalahan kecil dimana sistem membaca objek yang bukan wajah namun dianggap wajah. Meskipun begitu hal ini dapat diabaikan dikarenakan kesalahan yang terjadi masih sangat kecil. Hasil pengujian dirangkum dalam tabel 3.

Tabel 3. Hasil Pengujian Pendeteksi Wajah

Nama Objek	Tipe Objek		Hasil Pengujian
	Wajah	Bukan	
Muka 1	√		Berhasil
Botol		√	Berhasil
Muka 2	√		Berhasil
Muka 3	√		Berhasil
Komputer		√	Berhasil
Buku		√	Berhasil

Adapun pengujian sistem *face recognition* pada sistem *Doorcam* yang lama, masih belum didapatkan hasil yang diinginkan. Hal ini disebabkan bahasa yang dipakai untuk membangun sistem lama masih menggunakan bahasa pemrograman Python versi 2.

### Kesimpulan

Hasil dari pengujian yang dilakukan menyimpulkan bahwa sistem *face recognition* yang dibangun telah dapat berjalan dengan baik, namun belum dapat diimplementasikan pada sistem lama karena perbedaan versi pada bahasa pemrograman. Selain itu, penyebab lainnya adalah aplikasi dan modul yang dipakai untuk membuat *Graphic User Interface* (GUI) juga berbeda. Pada sistem lama menggunakan modul Tkinter, sedangkan yang baru menggunakan aplikasi PyQt5 yang dapat berjalan pada Python 3. Maka untuk dapat diaplikasikan, sistem yang lama harus dibangun ulang (*rebuild*) dengan menggunakan bahasa yang sama dengan sistem *Doorcam* yang baru.

### Daftar Pustaka

- [1] V. Paul and J. J. Michael , "Robust real-time face detection," *International Journal of Computer Vision*, vol. II, no. 57, pp. 137-154, 2004.
- [2] S. G. Anggraeni, "Pengembangan Sistem Doorcam Berbasis Mini PC Raspberry Pi," *Jurnal Teknik Komputer Unikom*, pp. 16-20, 2018.
- [3] S. I. Lestaringati, H. Irmayanti and H. Hawari, "Perancangan Sistem Pengambilan Gambar Video Berdasarkan Pergerakan Benda Menggunakan Metode Background Subtraction," *Komputika*, vol. VIII, no. 1, pp. 21-21, 2019.