

## Akurasi Sistem *Face Recognition OpenCV* Menggunakan *Raspberry Pi* Dengan Metode *Haar Cascade*

Raden Isum Suryani Maryati, Burhanuddin Tryatmojo

Universitas Paramadina, Jl. Gatot Subroto Kav. 97 Mampang Prapatan, Kota Jakarta Selatan, DKI Jakarta 12790

Universitas Paramadina, Jl. Gatot Subroto Kav. 97 Mampang Prapatan, Kota Jakarta Selatan, DKI Jakarta 12790

### INFORMASI ARTIKEL

#### Sejarah Artikel:

Diterima Redaksi: 28 Agustus 2019

Revisi Akhir: 24 September 2019

Diterbitkan Online: 30 September 2019

### KATA KUNCI

Akurasi

*Face Recognition*

*Raspberry Pi*

*OpenCV*

*Haar Cascade*

### KORESPONDENSI

No HP: 089621125869

E-mail: [radenisum@gmail.com](mailto:radenisum@gmail.com)

### A B S T R A C T

*Face recognition systems is developing very quickly with various methods and algorithms. Face recognition system is an identification system that is developed based on differences in facial features biometrics on body parts or even human behavior. Face is used by systems to recognize someone because faces have different shapes and textures. Information about a person's face can be extracted from the image that has been detected. However, detection system still needs to be developed to obtain better results. Based on that background, this research formulated how much accuracy is generated in the face recognition detection system using the Haar Cascade algorithm, using a webcam camera and Python as its programming language. In this study, Raspberry Pi is used as a Single Board Computer to run a face recognition program and Open Computer Vision (OpenCV) as a face recognition library. This system called FaceTrix and from the results of testing the accuracy of the system, the best accuracy of the system is at a distance of 40 cm and 60 cm which was 100% and distance of 80 cm which was 91%. Based on overall system testing, the detection system's accuracy was 97% with an average time of 1.779 FPS.*

## 1. PENDAHULUAN

Teknologi pengenalan wajah merupakan pendekatan pola untuk dapat mengidentifikasi seseorang dengan menggunakan pengolahan citra digital atau *image processing*. Teknologi pengenalan wajah makin banyak diaplikasikan di dalam sistem pengenalan biometrik, pencarian dan pengeindeksan *database* citra dan *video digital*, sistem keamanan, konferensi video maupun interaksi manusia dengan komputer. Pendeteksian wajah (*face detection*) juga merupakan salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*) [1].

Dengan penggunaan teknologi pengenalan wajah yang semakin banyak, diperlukan sistem pengolah citra (*image processing*) yang akurat untuk menghasilkan sebuah citra digital yang dapat mengidentifikasi seseorang dengan menganalisis pola berdasarkan tekstur dan bentuk wajah seseorang yang sebelumnya sudah tersimpan di dalam *database* atau sudah di *training* sebelumnya.

Saat ini telah terdapat berbagai macam sistem pengenalan wajah. Namun bagaimanapun juga masih diperlukan penelitian dan uji coba untuk terus mengembangkan dan mendapatkan hasil akurasi yang lebih baik. Berdasarkan pada latar belakang tersebut, pada penelitian ini penulis mengembangkan sistem FaceTrix yang berfokus pada tujuan untuk mengetahui tingkat akurasi sistem deteksi *face recognition* menggunakan metode *Haar Cascade* dengan melakukan uji coba sistem menggunakan *Single Board Computer Raspberry Pi* dengan menggunakan *library OpenCV*.

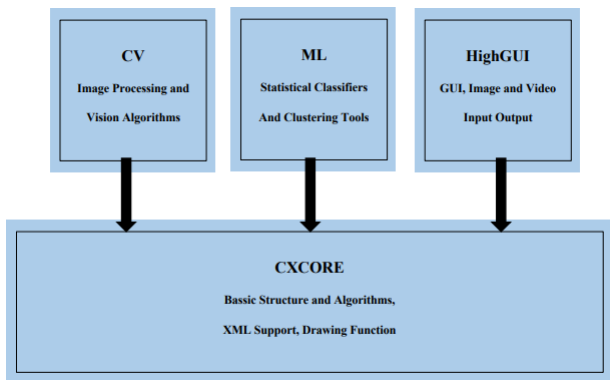
## 2. TINJAUAN PUSTAKA

### *Open Source Computer Vision Library (OpenCV)*

*OpenCV* merupakan salah satu *library* yang sering digunakan pada pengolahan citra *computer vision* yang memanfaatkan sebuah *Application Programming Interface (API)* dimana *OpenCV* memungkinkan komputer dapat melihat seperti manusia dengan *vision* tersebut. Komputer dapat mengambil keputusan, melakukan aksi dan mengenali terhadap suatu objek berdasarkan deteksi wajah.

*OpenCV* terdiri dari 5 *library* yaitu, *Computer Vision (CV)* sebagai algoritma *image processing* dan *vision*-nya, *Machine Learning (ML)*, *HighGUI* sebagai GUI, *Image dan Video I/O*,

CXCORE sebagai struktur data, support XML dan fungsi-fungsi grafis dan CvAux sebagai penolong *OpenCV*.



Gambar 1. Struktur dan Konten *OpenVC*

### Face Recognition

*Face Recognition* merupakan sebuah teknologi berbasis *Biometric Artificial Intelligence* (AI) yang dapat mengidentifikasi seseorang dengan menganalisis pola berdasarkan tekstur dan bentuk wajah seseorang yang sebelumnya sudah tersimpan di dalam *database* atau sudah dipelajari sebelumnya. *Face Recognition* digunakan dalam berbagai aplikasi yang dapat mengidentifikasi wajah manusia menggunakan gambar digital. Pendeteksian wajah atau *Face Detection* merupakan salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*Face Recognition*) [2].

Bidang-bidang penelitian yang juga erat kaitannya dengan pemrosesan wajah atau *face processing* adalah sebagai berikut:

- Autentikasi Wajah (*Face Authentication*).
- Lokalisasi Wajah (*Face Localization*).
- Penjejakan Wajah (*Face Tracking*).
- Pengenalan Ekspresi Wajah (*Facial Expression Recognition*).



Gambar 2. Blok Diagram Sistem Pengenalan Wajah [3].

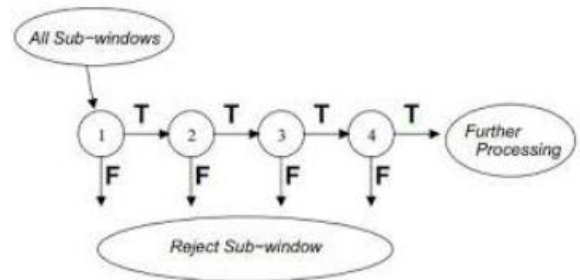
### Artificial Intelligence

Menurut Rich dan Knight, *artificial intelligence* atau kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia [4]. Dalam perkembangannya, kecerdasan buatan dikelompokkan dalam beberapa hal seperti *Natural Language Processing*, *Speech Recognition*, *Computer Vision*, dan lainnya.

### Haar Cascade

*Haar Cascade* merupakan metode yang digunakan dalam pendeteksian objek dimana metode ini menggabungkan empat kunci utama yaitu *Haar Feature Selection*, *Creating Integral Image*, *Adaboost Training* dan *Cascade Classifier*. Metode ini juga dikenal dengan metode Viola Jones karena Paul Viola dan Michael Jones merupakan tokoh pertama yang memperkenalkan

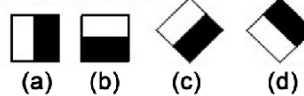
pendeteksian wajah pada tahun 2001 [5]. *Cascade Classifier* berfungsi sebagai rantai *stage classifier*, dimana setiap *stage classifier* digunakan untuk mendeteksi apakah di dalam *image sub window* terdapat objek yang ingin dideteksi.



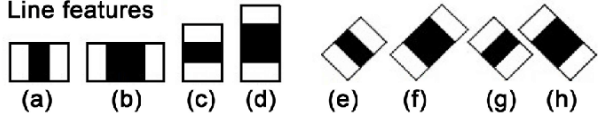
Gambar 3. Blok Diagram Sistem Pengenalan Wajah [5].

*Haar Feature* sendiri merupakan *rectangular features* yang memberikan indikasi secara spesifik pada sebuah gambar dengan tujuan untuk mengenali objek berdasarkan nilai sederhana dari fitur. *Haar Feature* memiliki kelebihan antara lain adalah komputasinya yang sangat cepat, karena bergantung pada jumlah pixel dalam persegi, bukan setiap pixel dari sebuah gambar [6]. *Haar Feature* ini merupakan dua dimensi, satu terang dan satu gelap yang terdiri dari gabungan kotak-kotak hitam dan putih seperti berikut ini:

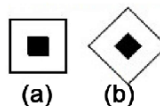
#### 1. Edge features



#### 2. Line features



#### 3. Center-surround features



Gambar 4. Macam-macam *Haar Feature* [6]

### Raspberry Pi

*Raspberry Pi* adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*Sistem-on-Chip*) ARM yang dikemas dan diintegrasikan di atas PCB. Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang. Dapat dikatakan bahwa *Raspberry Pi* merupakan sebuah modul *micro computer* yang akan di-*install* dengan sistem operasi Raspbian.

*Raspberry Pi* bersifat *open source* (berbasis Linux) dan bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama *Raspberry Pi* menggunakan Debian GNU/Linux dan bahasa pemrograman Python. Salah satu pengembang OS untuk *Raspberry Pi* telah meluncurkan sistem operasi yang dinamai Raspbian yang diklaim mampu memaksimalkan perangkat

Raspberry Pi. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS [7].



Gambar 5. Raspberry Pi 3 [8]

### Phyton

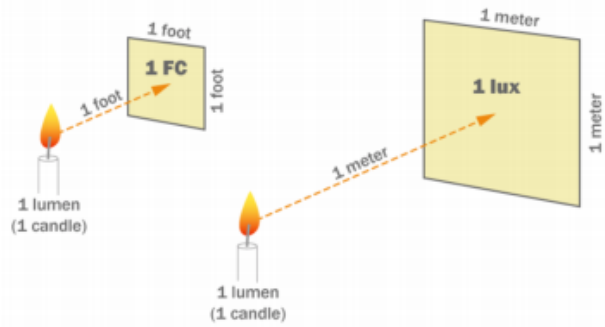
Bahasa pemrograman yang paling matang dalam menyediakan *library OpenCV* adalah bahasa pemrograman Python. Pada penelitian ini penulis menggunakan Python3 dalam uji coba sistem *Face Recognition*. Python3 merupakan *script language* yang berorientasi objek. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan bisa dijalankan melalui berbagai sistem operasi. Python juga merupakan bahasa pemrograman yang populer di bidang *data science* dan analisis karena bahasa pemrograman Python ini sangat mendukung terhadap *library* yang didalamnya menyediakan fungsi analisis data dan fungsi *machine learning*, *data preprocessing tools*, serta visualisasi data [9].

### Nano

*Nano* atau *GNU Nano* merupakan *desktop graphical editors* atau lebih sering disebut sebagai *command-line editors* yang terdapat pada *Raspberry Pi*, yang sudah ter-*install* secara otomatis *by default* pada sistem *Raspberry Pi*.

### 2.1 Lux Light Meter App

*Lux Lighting Apps* merupakan aplikasi pengukur cahaya sederhana untuk mengukur *lux* dan FC dengan menggunakan sensor cahaya pada perangkat Android. *Lux* merupakan satuan turunan SI dengan mengukur pencahayaan per satuan luas. *Illuminance* diukur dalam *foot candles* (FC) atau *lux*. 1 FC adalah jumlah cahaya yang menyentuh permukaan 1 ft<sup>2</sup> atau 1 kaki persegi ketika 1 lumen bersinar dari 1 kaki, setara dengan 1 lumen per kaki persegi. 1 *lux* adalah jumlah cahaya yang menyentuh permukaan 1m<sup>2</sup> ketika 1 lumen bersinar dari 1 meter - ini setara dengan 1 lumen per meter persegi. 10 *lux* kira-kira 1 FC.



Gambar 6. Tingkat Cahaya Dalam Bangunan [10].

## 3. METODOLOGI

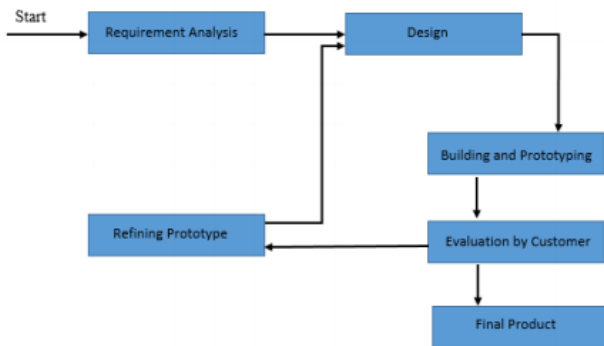
Dalam penelitian ini, metode penelitian yang penulis pilih adalah metode pendekatan kualitatif. Metode kualitatif merupakan metode penelitian yang lebih banyak menggunakan kualitas subjektif serta mencakup pengungkapan berdasarkan persepsi untuk memperoleh pemahaman fenomena sosial dan kemanusiaan.

Sebagai metode pengembangan sistem, penulis menggunakan *model prototyping*. *Prototyping* merupakan proses *iterative* dalam pengembangan sistem dimana *requirement* diubah ke dalam *working system* yang secara terus menerus diperbaiki melalui kerjasama antara pengguna dan analis. Proses *iterative* merupakan proses yang melibatkan hubungan kerja yang dekat antara perancang dan pengguna.. Adapun beberapa tahapan pada model *prototype* ini diantaranya sebagai berikut [11]:

- a. *Communication* merupakan tahap pertemuan yang dilakukan oleh pihak pengembang dan *stakeholder* dari pemakai perangkat lunak mengenai gambaran secara keseluruhan dari perangkat lunak yang akan dibangun. Mengidentifikasi kebutuhan yang sudah diketahui untuk kemudian membuat *quick plan* dari iterasi pertama. Pada tahap ini akan menghasilkan berbagai kebutuhan sistem atau *Requirement Analysis* yang akan digunakan pada tahap selanjutnya. Kebutuhan yang dihasilkan merupakan kebutuhan awal sistem dan dapat bertambah di iterasi berikutnya.
- b. Tahap *Quick Plan* menjelaskan tentang segala perencanaan yang bersifat teknis selama proses pengembangan perangkat lunak. Tahap ini akan memaparkan resiko yang akan didapatkan, pembagian pekerjaan dan perencanaan waktu agar proses yang dilakukan akan berjalan dengan baik. *Quick plan* akan menjelaskan hasil tahap *Communication* ke dalam format yang lebih rapih untuk selanjutnya diteruskan oleh *modeling quick design*.
- c. *Modelling Quick Design* adalah tahap dimana membuat gambaran besar dari perangkat lunak yang akan dibangun. Pemodelan dibuat untuk memberi gambaran kepada *stakeholder* yang berkaitan tentang aplikasi yang akan dibangun. *Modeling Quick Design* akan menghasilkan desain berupa sketsa atau *wireframe* dari antarmuka perangkat lunak.
- d. *Construction of Prototype* merupakan sebuah tahap pengimplementasian dari *Modeling Quick Design*,

dimana perangkat lunak dan selanjutnya akan dilanjutkan oleh pengkodean dari antarmuka yang digunakan.

- e. *Deployment Delivery and Feedback* merupakan tahap *delivery* dari perangkat lunak yang sudah selesai dibangun kepada stakeholder. Tahap ini akan menghasilkan feedback dari pengguna berdasarkan hasil testing yang dilakukan pengembang.



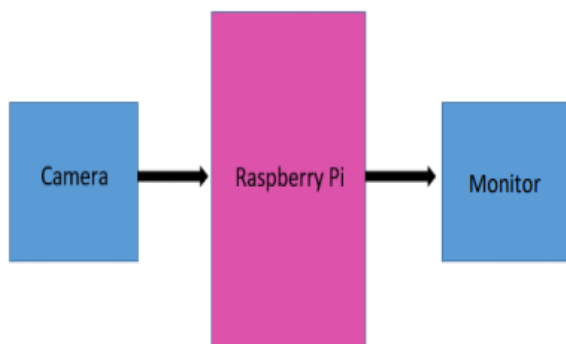
Gambar 7. Prototype Model [11]

#### 4. HASIL DAN PEMBAHASAN

##### Diagram Blok Sistem FaceTriX

Diagram blok menjelaskan tahapan yang terjadi pada sistem:

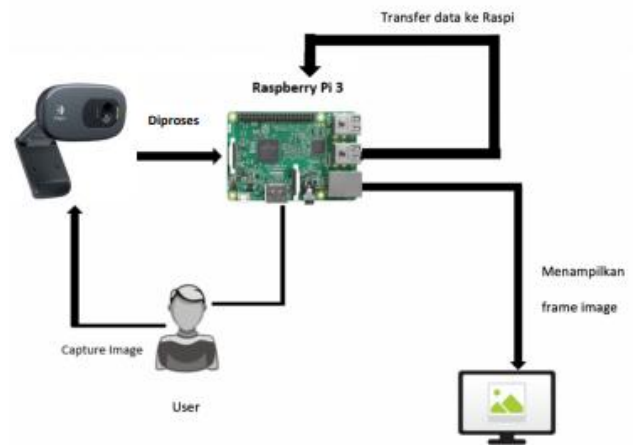
- Kamera akan mendeteksi *user*.
- Kamera melakukan pemindaian citra wajah.
- Data citra yang telah disimpan akan diproses ke Raspberry Pi.
- Raspberry Pi akan menampilkan frame ke monitor.



Gambar 8. Diagram Blok Sistem

##### Diagram Alur Sistem FaceTriX

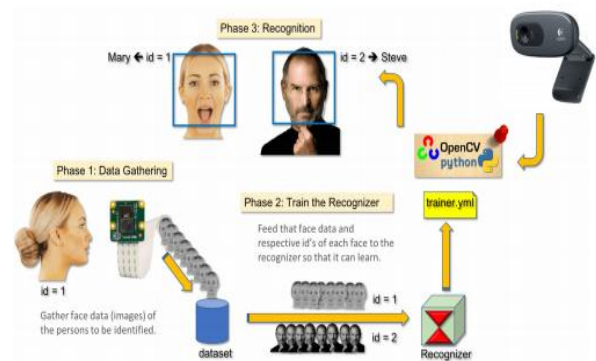
Diagram alur sistem ini dibuat untuk memudahkan penulis dalam melakukan uji coba sistem deteksi. Berikut adalah diagram alur sistem deteksi FaceTriX:



Gambar 9. Diagram Alur Sistem

Berdasarkan gambar diagram alur sistem diatas, dapat diketahui bahwa citra wajah akan dideteksi oleh kamera. Hasil deteksi itu nantinya akan dikirim dan diproses di *Raspberry Pi* pada folder *dataset*. Setelah *dataset* usai melakukan training, maka sistem sudah dapat digunakan karena sistem sudah mampu mengenali citra wajah yang sedang dideteksi.

##### Cara Kerja OpenCV Pada Sistem FaceTriX



Gambar 10. Cara Kerja OpenCV [12]

##### Tahap Persiapan Sistem FaceTriX

Setelah memahami alur sistem dan cara kerja *OpenCV*, penulis melanjutkan pada tahap persiapan sistem FaceTriX. Persiapan yang dilakukan yaitu:

- Memasang *OpenCV* dan *Python* pada *Raspberry Pi*.
- Setelah *OpenCV* dan *Python* telah terpasang pada *Raspberry Pi*, maka langkah selanjutnya adalah melakukan pengujian pada kamera. Pada kesempatan ini, penulis menggunakan *webcam* untuk mendeteksi citra.

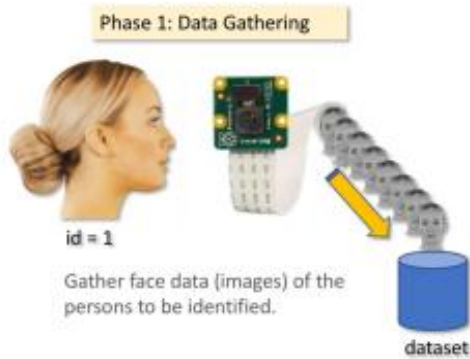
##### Proses Face Recognition

Setelah persiapan sistem FaceTriX dilakukan, penulis melakukan proses *face recognition*. Langkah yang dilakukan yaitu:

- Pengumpulan data dengan melakukan deteksi wajah secara aktual. Untuk mendeteksi citra pada *Face Recognition*, penulis menggunakan metode *Haar Cascade* sebagai cara untuk deteksi objeknya. Pengumpulan data dapat dilakukan dengan cara

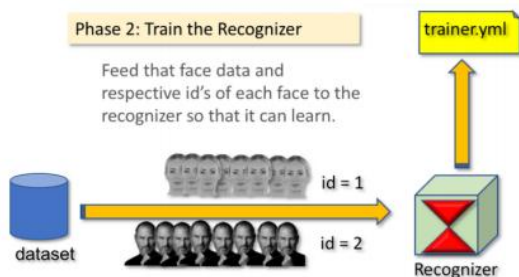


melakukan proses deteksi secara berulang sehingga data yang berada didalam folder *dataset* akan bertambah. Data tersebutlah yang nantinya akan di latih pada tahap selanjutnya. Data yang diambil dapat gunakan sesuai kebutuhan. Semakin banyak *dataset* yang dihasilkan dalam satu citra wajah, maka semakin mudah dan mempercepat proses pengenalan citra pada saat pendeteksian wajah



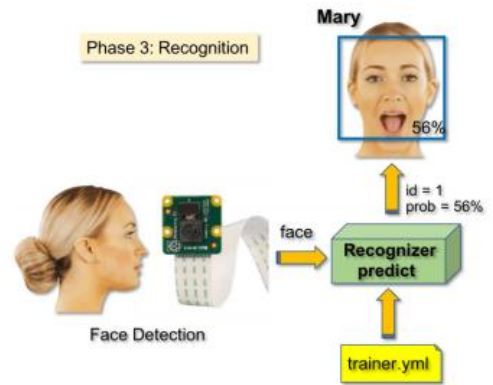
Gambar 11. *Data Gathering* [13]

- b. Setelah dilakukan pengumpulan data, penulis melakukan pelatihan citra pada sistem deteksi untuk dapat mengenali wajah.



Gambar 12. *Train The Recognizer* [13]

- c. Tahap selanjutnya adalah tahap pengenalan wajah. Tahap ini dilakukan jika proses pelatihan telah selesai dilakukan. Karena sebelumnya data hanya dicatat dan disimpan di dalam *dataset* dengan nama folder yang berbeda-beda. Pada saat pelatihan dijalankan dengan menggunakan metode *Haar Cascade*, maka citra dapat dengan mudah dikenali sebagai identitas masing-masing citra. Sehingga pada saat melakukan *Face Recogniton*, sistem telah mengenali wajah dari masing-masing citra yang dideteksi.



Gambar 13. *Recognition* [13]

**Pengujian Sistem FaceTrix**

Pada bab ini, penulis akan membahas pengujian sistem FaceTrix yang telah selesai dibangun. Untuk melakukan uji coba sistem penulis membutuhkan sampel yang akan dijadikan sebagai uji coba dalam pengujian sistem deteksi FaceTrix dengan ekspresi wajah bebas menghadap ke depan, dapat menggunakan atribut seperti kacamata, topi, kerudung dan atau tidak menggunakan atribut sekalipun. Pada penelitian ini penulis juga membatasi jumlah sampel sebanyak 10 orang.

Berikut penulis paparkan hasil pengujian sistem FaceTrix.

PENGUJIAN SISTEM						
NO	Jarak (cm)	Rata-rata FPS	Jumlah Citra Deteksi	Jumlah Deteksi		Total Rata-Rata Akurasi (%)
				Benar	Salah	
1	40 cm	1,50	10	10	0	1
2	60 cm	1,70	10	10	0	1
3	80 cm	2,067	10	9	1	0,91
Rata-rata Akurasi						0,97
Total Rata-rata FPS Keseluruhan						1,779

Berdasarkan hasil uji coba sistem, penulis mendapatkan hasil tingkat akurasi sebesar 0,97 atau setara dengan 97% dengan 3% tingkat kesalahan dalam deteksi sistem. Hal ini disebabkan pada saat mendeteksi, citra wajah tidak dapat dikenali secara jelas karena faktor objek yang terlalu banyak gerakan pada saat sistem sedang mendeteksi citra wajah. Selain itu, adanya perubahan cahaya pada saat pengenalan dan pendeteksian citra wajah juga sangat mempengaruhi sistem FaceTrix, sehingga objek dikenali sebagai objek lain dan tidak dapat dikenali secara pasti karena pencahayaan yang tidak merata. Untuk melakukan validasi pada objek yang *error*, penulis membutuhkan waktu yang cukup lama untuk sampai objek benar-benar dikenali sistem FaceTrix. Penulis melakukan pengenalan pada sistem deteksi lebih dari 3x pada objek yang *error*. Hasilnya adalah, sesekali sistem mengenali objek dengan benar tapi jika objek berubah posisi dan ekspresi maka objek dikenali sebagai objek lain lagi.

Dari tabel diatas, dapat dilihat pula rata-rata yang dihasilkan sistem deteksi FaceTrix pada saat melakukan pengujian wajah sebanyak 10 sample dengan pengujian jarak berbeda-beda. Pada jarak 40cm dan 60cm sistem masih dapat mendeteksi secara normal dan cepat dengan tingkat akurasi sebesar 100% dengan 0% tingkat kesalahannya. Namun, pada saat melakukan

pendeteksian pada jarak 80 cm sistem FaceTrix mendapatkan kendala pada pendeteksian citra wajah. Walaupun begitu, sistem tetap berjalan sebagaimana mestinya dengan tingkat akurasi yang didapati pada jarak 80cm adalah sebesar 91%.

### Hasil Uji Coba Sistem

Pada uji coba sistem FaceTrix, penulis melakukan uji coba sistem dengan ketentuan jarak minimal 40 cm dan maksimal 80 cm menghadap kamera dengan jumlah uji sample sebanyak 10 orang. Penulis membatasi jarak uji minimal sejauh 40 cm dan maksimal 80 cm yang diambil berdasarkan dari hasil uji coba tingkat akurasi sistem sebelumnya. Berikut adalah hasil uji coba sistem pada jarak 40 cm, 60 cm dan 80 cm:

Tabel 1. Hasil Uji Coba Sistem Pada Jarak 40 cm

PENGUJIAN SISTEM PADA JARAK 40 CM						
NO	Nama Citra Uji	Kecepatan Frame Deteksi (FPS)	Nilai Cahaya (Lux)	Tingkat Validasi		
				Terdeteksi	Tidak Terdeteksi	Unknown
1	M. Shiddiq Ghozali	0,2	13	Valid	-	-
2	Cynthia Evia Y.	1,9	13	Valid	-	-
3	Irwan Syarifudin	1,7	13	Valid	-	-
4	Kenindra M.	0,6	13	Valid	-	-
5	Dea Oktaviani	0,9	13	Valid	-	-
6	Yustiana Sari	3,4	13	Valid	-	-
7	Rafli Andika	0,2	13	Valid	-	-
8	M. Hanial	0,3	13	Valid	-	-
9	Garis Riski L.	0,5	13	Valid	-	-
10	Wulan Sari Gumilang	5,3	13	Valid	-	-
	<b>Rata-rata FPS</b>	1,50				
*) catatan :						

Tabel 2. Hasil Uji Coba Sistem Pada Jarak 60 cm

PENGUJIAN SISTEM PADA JARAK 60 CM						
NO	Nama Citra Uji	Kecepatan Frame Deteksi (FPS)	Nilai Cahaya (Lux)	Tingkat Validasi		
				Terdeteksi	Tidak Terdeteksi	Unknown
1	M. Shiddiq Ghozali	1,4	14	Valid	-	-
2	Cynthia Evia Y.	2,1	14	Valid	-	-
3	Irwan Syarifudin	3,4	14	Valid	-	-
4	Kenindra M.	0,6	14	Valid	-	-
5	Dea Oktaviani	0,3	14	Valid	-	-
6	Yustiana Sari	3,6	14	Valid	-	-
7	Rafli Andika	2,0	14	Valid	-	-
8	M. Hanial	2,0	14	Valid	-	-
9	Garis Riski L.	1,2	14	Valid	-	-
10	Wulan Sari Gumilang	1,0	14	Valid	-	-
	<b>Rata-rata FPS</b>	1,77				
*) catatan						

Tabel 3. Hasil Uji Coba Sistem Pada Jarak 80 cm

PENGUJIAN SISTEM PADA JARAK 80 CM						
NO	Nama Citra Uji	Kecepatan Frame Deteksi (FPS)	Nilai Cahaya (Lux)	Tingkat Validasi		
				Terdeteksi	Tidak Terdeteksi	Unknown
1	M. Shiddiq Ghozali	2,3	14	Valid	-	-
2	Cynthia Evia Y.	3,6	14	Valid	-	-
3	Irwan Syarifudin	1,43	14	Valid	-	-
4	Kenindra M.	1,26	14	Valid	-	-
5	Dea Oktaviani	1,85	14	Valid	-	-
6	Yustiana Sari	1	14	Valid	-	-
7	Rafli Andika	1,04	14	Valid	-	-
8	M. Hanial	2,19	14		Tidak valid *)	-
9	Garis Riski L.	2,6	14	Valid	-	-
10	Wulan Sari Gumilang	3,4	14	Valid	-	-
	<b>Rata-rata FPS</b>	2,067				
*) Pada saat pengujian, sistem yang terdeteksi adalah Irwan, Shiddiq dan Ken						

## 5. KESIMPULAN DAN SARAN

Berdasarkan pengujian sistem deteksi yang dilakukan penulis, penulis dapat menyimpulkan beberapa kesimpulan dari sistem deteksi Face Recognition OpenCV menggunakan metode Haar Cascade dan Raspberry Pi sebagai berikut:

- sistem dapat mendeteksi dan mengenali wajah seseorang dengan cepat menggunakan metode *Haar Cascade*.
- tingkat akurasi deteksi citra wajah terbaik pada saat dilakukan pendeteksian dalam keadaan jarak antar kamera dan wajah maksimal 80cm menghadap ke depan kamera dengan cahaya yang merata dan sama dengan pencahayaan yang pada saat pengambilan citra wajah pada proses training.
- akurasi sistem deteksi yang dihasilkan oleh sistem deteksi FaceTrix cukup bagus dengan tingkat akurasi rata-rata sebesar 97% dengan total rata-rata kecepatan keseluruhan sistem FaceTrix dalam satuan FPS adalah 1,779 pada jarak minimal 40cm dan jarak maksimal 80cm. Adapun citra wajah yang tidak dapat dikenali secara tepat karena adanya faktor cahaya yang tidak merata pada permukaan wajah dan posisi wajah tidak tegak ke depan menghadap kamera. Hal ini juga dapat terjadi jika objek yang dideteksi melakukan banyak gerakan pada saat proses pendeteksian berlangsung.
- spesifikasi kamera yang digunakan akan mempengaruhi akurasi dan kecepatan sistem deteksi. Semakin bagus dan tinggi spesifikasi kamera untuk mendeteksi citra wajah, maka semakin tinggi tingkat akurasinya.
- sistem mudah untuk melakukan pendeteksian dan pengenalan citra wajah seseorang berdasarkan banyaknya jumlah citra yang di *training* dan faktor pencahayaan yang stabil dan merata.

## DAFTAR PUSTAKA

- [1] S. Nugroho and A. Harjoko, "Sistem Pendeteksi Wajah Manusia Pada Citra Digital", J. Sains dan Siberatika, vol. 18, no. 1, 2005.
- [2] Research Gate, "Detecting Faces in Images: A Survey," 2002. [Online]. Available: [https://www.researchgate.net/publication/3193340\\_Detecting\\_Faces\\_in\\_Images\\_A\\_Survey](https://www.researchgate.net/publication/3193340_Detecting_Faces_in_Images_A_Survey).
- [3] Jurnal EECIS, "Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, Eigenface PCA & MySQL," 2013. [Online]. Available: <https://jurnaleeccis.ub.ac.id/index.php/eccis/article/download/223/197>
- [4] Akmal, "Lebih Dekat Dengan Industri 4.0", Sleman: Deepublish Publisher, 2019.
- [5] Universitas Muhammadiyah Malang, Pengenalan Wajah dengan Cascade Classifier.
- [6] W. S. Pambudi and B. M. N. Simorangkir, "Facetracker Menggunakan Menggunakan Metode Haar Like Feature dan PID Pada Model Simulasi", J. Teknologi dan Informatika, vol. 2, no. 2, 2012.
- [7] Eprints Polsri, "Rancang Bangun Sistem Audio Berbasis Raspberry Pi Via Bloethooth dengan Smartphone" 2016. [Online]. Available: <http://eprints.polsri.ac.id>
- [8] Amazon, "Raspberry Pi," 2018. [Online]. Available: <https://www.amazon.com/RSComponents-Raspberry-Pi-Motherboard/dp/B07BFH96M3>.
- [9] BINUS - Computer Science, "Penggunaan Python untuk Data Mining," 2018. [Online]. Available: <http://socs.binus.ac.id/2018/11/16/penggunaan-python-untuk-data-mining/>.
- [10] Fire Flier, [Online]. Available: <http://id.lamparaleds.com/news/how-much-lux-does-my-roomneed-18327451.html>
- [11] R. S. Pressman, in Software Engineering: A Practitioner Approach Seventh Edition, 2015.
- [12] OpenCV, "OpenCV," 2018. [Online]. Available: <https://opencv.org>.
- [13] Hackster.io, "Real-Time Face Recognition: An End to End Project", [Online]. Available: <https://www.hackster.io/mjrobot/real-time-face-recognitionan-end-to-end-project-a10826>.

## BIODATA PENULIS



### **Raden Isum Suryani Maryati**

Mahasiswa Jurusan Informatika, Fakultas Ilmu Rekayasa, Universitas Paramadina. Penulis dapat dihubungi pada alamat email radenisum@gmail.com



### **Burhanuddin Tryatmojo**

Mahasiswa Jurusan Informatika, Fakultas Ilmu Rekayasa, Universitas Paramadina. Penulis dapat dihubungi pada alamat email burhanuddin.tryatmojo@gmail.com