



## Penerapan *Haar Cascade Classifier* dalam Mendeteksi Wajah dan Transformasi Citra *Grayscale* Menggunakan OpenCV

Syefrida Yulina<sup>1</sup>

<sup>1</sup>Politeknik Caltex Riau, email: syefrida@pcr.ac.id

### Abstrak

Aplikasi pendeteksi wajah pada citra digital sangat diperlukan dalam proses pengenalan wajah seseorang. Aplikasi ini banyak dimanfaatkan pada berbagai disiplin ilmu, salah satunya ilmu komputer visi (computer vision) seperti sistem pengenalan biometrik, sistem pencarian, dan sistem keamanan. Komputer visi ini merupakan gabungan dari ilmu artificial intelligent dan machine learning. Komputer visi ini bertujuan agar sebuah komputer dapat memahami citra dan video. Banyak penelitian terdahulu yang telah mengembangkan aplikasi deteksi wajah dengan berbagai algoritma tertentu dengan bahasa pemrograman tertentu. Pendeteksian sebuah objek merupakan bagian terpenting didalam komputer visi. Untuk mendeteksi sebuah objek diperlukan algoritma. Penentuan lokasi wajah yang akurat masih merupakan tugas yang menantang bagi para peneliti. Lokasi wajah merupakan langkah utama dalam komputer visi untuk menemukan area wajah pada citra input. Open Source Computer Vision Library (OpenCV) adalah sebuah software yang bersifat open-source yang berisikan library yang mendukung pendeteksian objek yang mudah diimpor kedalam bahasa pemrograman Java. Haar cascade classifier adalah salah satu algoritma yang digunakan untuk mendeteksi objek. Algoritma ini dapat mendeteksi sebuah objek secara cepat dengan mengambil jumlah piksel dalam bentuk persegi pada sebuah citra gambar. Pada penelitian ini, membahas tentang aplikasi deteksi wajah pada citra digital menggunakan Haar Cascade Classifier dan transformasi citra menjadi citra keabuan/grayscale menggunakan library OpenCV. Hasil pada penelitian ini memiliki akurasi 100% pada citra inputan yang memiliki objek pada posisi frontal.

**Kata kunci:** Deteksi Wajah, Haar Cascade Classifier, Komputer Visi, OpenCV

### Abstract

Face detection applications on digital images are very necessary in the process of face recognizing. This application is widely used in various disciplines, one of them is computer vision such as biometric recognition systems, search systems, and security systems. Computer vision is a combination of artificial intelligence and machine learning. It can gain informations from image and video by using computer algorithms. Many previous studies have developed face detection applications with various algorithms with certain programming languages. The detection of an object is the most important part in computer vision. Determining an accurate face location is still a challenging task for researchers. The location of the face is the main step in computer vision to find the face part in the input image. Open Source Computer Vision Library (OpenCV) is software that allows open-source library containing supporting object

*detection that is easily accessed into the Java programming language. Haar cascade classifier is one of the algorithms used for object detection. This algorithm can convert an object quickly by taking the number of images in a square shape on an image. In this study, discussing the application of face detection in digital images using the Haar Cascade Classifier and the transformation of images into gray / grayscale images using the OpenCV library. The results in this study have 100% accuracy in input images that have objects in the frontal position*

**Keywords:** Computer Vision, Face Detection, Haar Cascade Classifier, OpenCV

---

## 1. Pendahuluan

Komputer visi merupakan suatu cara komputer untuk mengumpulkan dan menginterpretasikan informasi visual/citra yang diperlukan untuk menyelesaikan tugas tertentu. Pada umumnya sebuah citra digital pertama kali akan diproses untuk lebih ditingkatkan kualitas citranya, misalnya menghilangkan *noise*. Lalu citra tersebut diproses pada tingkatan yang lebih tinggi, misalnya deteksi pola dan bentuk agar didapatkan informasi tentang apa yang ada pada citra itu. Deteksi objek merupakan metode yang diperlukan untuk menemukan dan mengidentifikasi keberadaan objek-objek pada citra digital tertentu. Penentuan lokasi wajah yang akurat masih merupakan tugas yang menantang bagi para peneliti. Lokasi wajah merupakan langkah utama dalam komputer visi untuk menemukan area wajah pada citra input.

Berbagai penelitian terdahulu yang membahas tentang metode pendeteksian wajah telah banyak dilakukan, seperti algoritma MRC (*Maximal Rejection Classifier*) untuk citra berwarna [1], *component analysis* dan *neural network classification* [2] dan *template matching* [3] berdasarkan segmentasi kulit manusia, metode *Faster R-CNN* [4], dan metode Viola-Jones [5]. Metode Viola-Jones banyak digunakan oleh para peneliti untuk mendeteksi lokasi wajah dan objek dalam sebuah citra. Penerapan fitur *Haar* dengan kontribusi metode Viola-Jones memiliki tingkatan kecepatan yang tinggi dalam mendeteksi wajah.

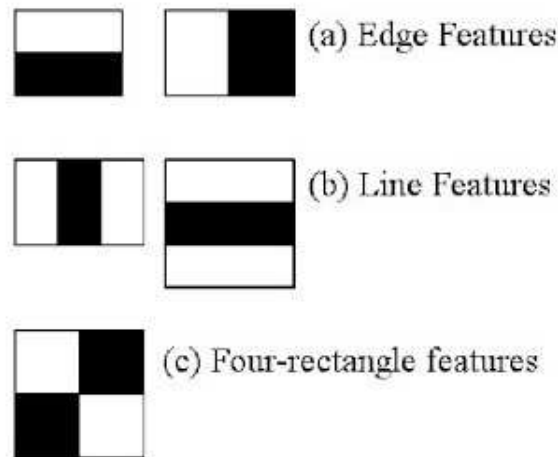
Oleh sebab itu, pada penelitian ini berfokus kepada pemanfaatan fitur *Haar Cascade Classifier* untuk klasifikasi wajah secara otomatis berdasarkan lokasi wajah pada citra berwarna. Adapun lokasi yang ditentukan berdasarkan posisi *frontal* dan mendekati *frontal* pada wajah. Biasanya sebuah wajah dimodelkan dalam *pose* yang dicirikan oleh tiga sudut rotasi: *pitch*, *roll*, dan *yaw*. Dengan kepala manusia menghadap kamera, *yaw* adalah sudut menggerakkan kepala ke kiri dan ke kanan (rotasi mengelilingi sumbu Y); *pitch* adalah gerakan kepala ke atas dan ke bawah (rotasi sekitar sumbu X); dan *roll* adalah sudut kemiringan (rotasi di sekitar sumbu Z). Sudut rotasi kepala relative terhadap kamera yaitu seperti: *frontal* =  $\pm 0^\circ \psi$ , mendekati *frontal* =  $\pm 90^\circ \psi$ . Pada implementasi untuk metode ini menggunakan library OpenCV, pada tahapan awal adalah melakukan transformasi citra kedalam citra *grayscale* dan kemudian menyediakan fitur *Haar* untuk klasifikasi wajah.

## 2. Landasan Teori

### 2.1 Haar Cascade Classifier

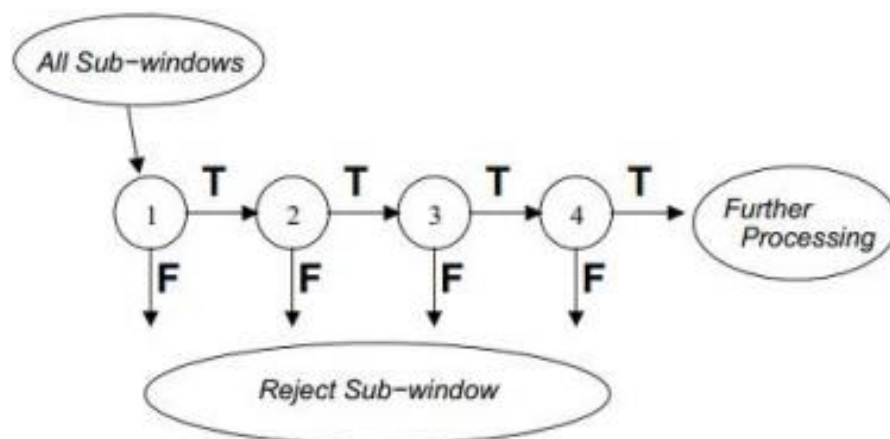
*Haar Cascade* adalah algoritma pembelajaran mesin untuk mendeteksi objek yang diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001. Algoritma ini adalah pendekatan berbasis pembelajaran mesin yang menggunakan fungsi *cascade*, dimana fungsi ini dilatih dari berbagai citra positif dan negatif. Citra positif merupakan citra yang memiliki objek yang akan dideteksi, sedangkan citra negatif merupakan citra yang tidak memiliki objek

deteksi. Sehingga fungsi ini dapat digunakan untuk mendeteksi objek pada citra yang lain. Saat ini, OpenCV sudah memberikan *library* untuk algoritma *Haar Cascade* yang sudah dilatih sebelumnya, serta sudah dikategorikan kedalam beberapa kategori seperti wajah, mata, dan sebagainya, tergantung pada gambar yang telah dilatih. *Haar Cascade* mengekstraksi fitur dari gambar menggunakan sebuah “*filter*” mirip dengan konsep kernel konvolusional. *Filter* ini disebut fitur *Haar* dan terlihat seperti gambar 1.



Gambar 1 Fitur Haar. Sumber: [8]

*Filter* ini akan memeriksa satu bagian pada satu waktu. Kemudian untuk tiap bagian, semua intensitas piksel pada bagian hitam dan putih akan dijumlahkan, lalu menghitung selisih dari tiap nilai yang dijumlahkan. Nilai tersebut merupakan nilai fitur yang diekstraksi. Karakteristik dari algoritma ini adalah adanya klasifikasi bertingkat. Klasifikasi ini terdiri dari beberapa tingkatan dimana tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Gambar 2 menggambarkan alur kerja dari klasifikasi bertingkat [6]



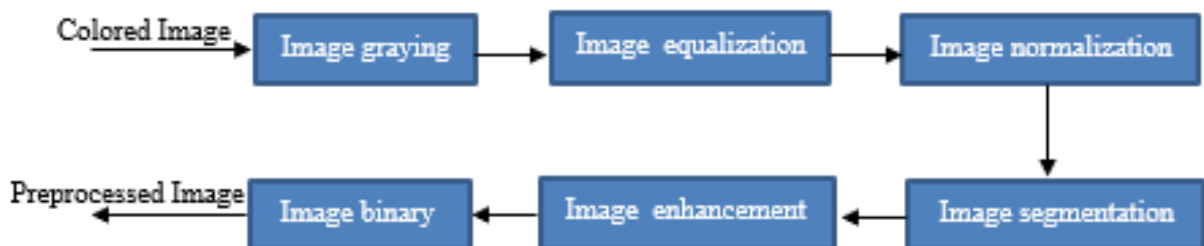
Gambar 2 Klasifikasi Bertingkat Fitur Haar

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Hasil dari klasifikasi pertama ini berupa T (*True*) untuk gambar yang memenuhi fitur *Haar* tertentu dan F (*False*) bila tidak. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Hasil dari klasifikasi kedua berupa T (*True*) untuk gambar yang memenuhi proses *integral image* dan F (*False*) bila tidak. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan

menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%.

## 2.2 Teknologi Pemrosesan Citra pada Pengenalan Wajah

Pada proses deteksi wajah, pemrosesan sebuah citra (*pre-processing*) sangat diperlukan. Terdapat beberapa tahapan dalam *preprocessing* citra pada pengenalan wajah [7]. Salah satu tahapan tersebut adalah citra berwarna akan diubah menjadi citra keabuan (*image graying*) dimana beberapa informasi yang tidak diperlukan akan dikurangi.



Gambar 3 Tahapan preprocessing citra pada citra wajah [10]

Pada gambar 3 tahapan *preprocessing* citra diawali dengan beberapa kumpulan data citra berwarna akan diubah menjadi citra keabuan (*image graying*). Kemudian dilanjutkan dengan beberapa rangkaian pemrosesan seperti: *image equalization*, *image normalization*, *image segmentation* dan *image binaryzation*, untuk meningkatkan kualitas dari sebuah citra.

## 3. Metode Penelitian

OpenCV digunakan untuk implementasi algoritma *Haar Cascade Classifier* agar sistem dapat mendeteksi wajah pada citra digital. Data yang menjadi inputan adalah data citra berwarna yang memiliki variasi jumlah objek wajah. Citra berwarna tersebut akan diubah terlebih dahulu menjadi citra keabuan (*grayscale*) kemudian objek yang memiliki wajah akan terdeteksi. Gambar 4 merupakan metode dari penelitian ini



Gambar 4 Metodologi Penelitian

Algoritma *Haar Cascade Classifier* digunakan karena memiliki tingkat akurasi yang tinggi dan cepat dalam melakukan klasifikasi terhadap sebuah citra digital. Tahapan yang dilakukan pada penelitian ini adalah melakukan beberapa kajian literatur terkait algoritma *Haar Cascade Classifier* dan teori pendukung tentang pemrosesan citra digital. Teknik pemrosesan citra yang diterapkan pada penelitian ini adalah transformasi citra kedalam bentuk citra keabuan/*grayscale*.

Kemudian tahapan pengumpulan data citra yang akan dijadikan sebagai citra inputan berupa citra berwarna dengan ukuran gambar berbeda serta posisi wajah pada citra yang

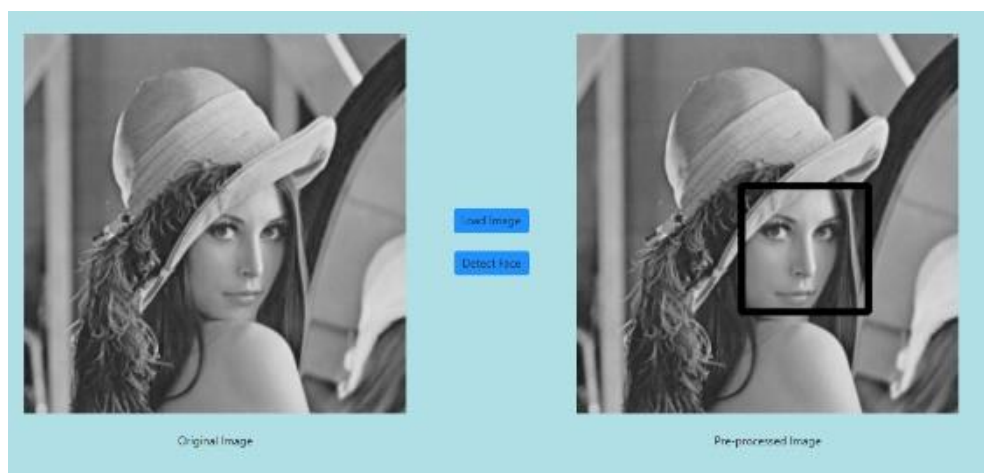
berbeda pula. Data citra ini diambil dari berbagai ukuran, posisi wajah, jarak objek wajah serta ukuran wajah yang terkandung didalam citra tersebut.

Tahapan selanjutnya adalah mengembangkan sebuah sistem pendeteksi wajah dengan melakukan impor *library* OpenCV dengan menggunakan OpenCV versi 3.4 dan bahasa pemrograman Java. Implementasi fitur *Haar* didapatkan dari *library* OpenCV yaitu `haarcascade_frontalface_alt2.xml` yang akan digunakan untuk klasifikasi wajah pada sebuah citra inputan. Sedangkan *library* yang digunakan untuk mengubah citra berwarna menjadi citra keabuan adalah `ImgCodecs.CV_LOAD_IMAGE_GRAYSCALE`

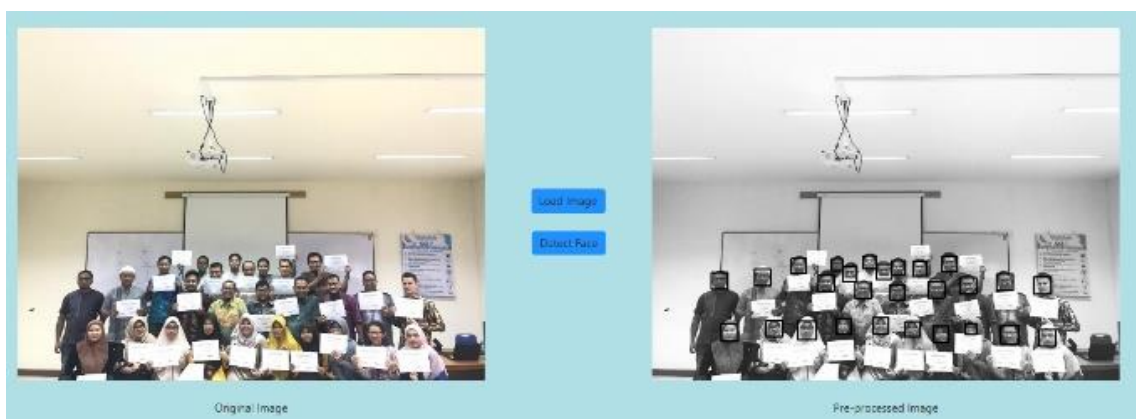
Tahapan akhir dari penelitian ini adalah analisa dan pengujian terhadap citra inputan. Setelah sistem berhasil dibangun, maka data citra yang menjadi inputan akan diproses untuk mendapatkan hasil akurasi dari deteksi wajah.

#### 4. Hasil dan Pembahasan

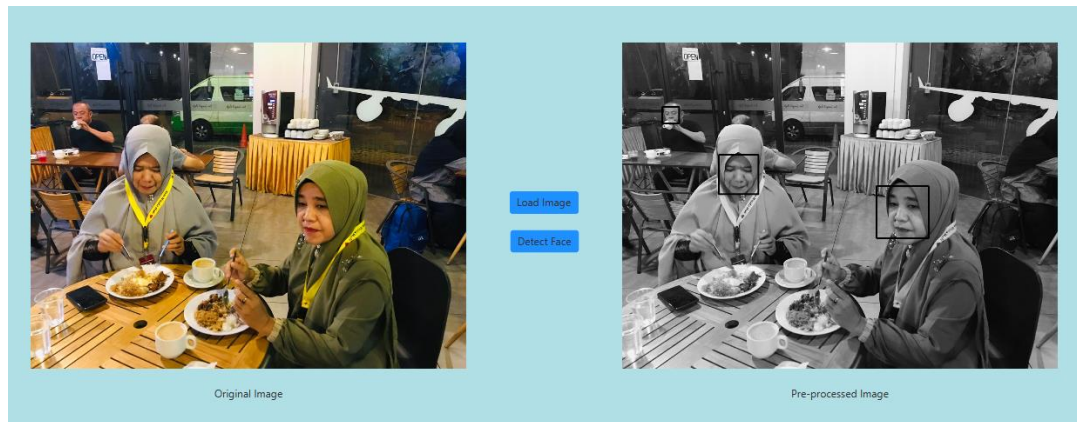
Inputan dari sistem ini merupakan citra berwarna dengan jumlah objek wajah dan posisi wajah yang bervariasi agar dapat diproses untuk menghasilkan klasifikasi wajah. Citra sampel pada penelitian ini adalah sebanyak 50 buah citra yang merupakan citra dengan tampilan *frontal* atau mendekati *frontal* wajah. Untuk akurasi dari metode dihitung dari jumlah wajah yang terdeteksi dengan posisi *frontal*/mendekati *frontal*, seperti dalam gambar 5. Kemudian jumlah kesalahan deteksi wajah pada citra input, dapat dilihat pada gambar 6.



(a)



(b)



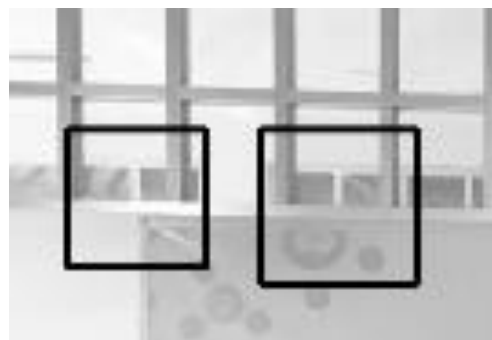
(c)

**Gambar 5 Deteksi Wajah pada Citra**

Dapat dilihat pada Gambar 5 bahwa fitur *Haar* dapat mendeteksi wajah pada citra berwarna yang kemudian diubah menjadi citra keabuan. Sebuah citra inputan berupa citra berwarna akan di *preprocessing* menjadi citra keabuan, kemudian citra keabuan tersebut dilakukan proses deteksi wajah. Pada Gambar 5(a) merupakan data uji untuk inputan berupa citra keabuan dengan posisi wajah mendekati *frontal* dan wajah telah berhasil terdeteksi, sedangkan pada Gambar 5(b) merupakan data uji untuk inputan citra berwarna dengan posisi wajah *frontal* dengan jarak wajah yang jauh dan data ini berhasil mendeteksi wajah, kemudian pada Gambar 5(c) inputan citra berwarna dengan posisi wajah mendekati *frontal* dan juga telah berhasil mendeteksi wajah. Wajah yang telah terdeteksi diperlihatkan pada bagian kotak persegi diposisi wajah.



(a)



(b)



(c)



(d)

**Gambar 6 Objek bukan Wajah yang Terdeteksi oleh Sistem**

Pada Gambar 6 menampilkan hasil kesalahan deteksi pada posisi bukan wajah. Terdapat beberapa kesalahan deteksi wajah pada metode ini, dimana sebuah objek yang bukan wajah dapat terdeteksi. Pada Gambar 6(a) terdapat kesalahan deteksi wajah dimana memperlihatkan tiga buah objek yang terdeteksi: dua buah objek wajah dan satu objek bukan wajah. Pada Gambar 6(b) menampilkan dua buah deteksi objek bukan wajah. Pada Gambar 6(c) menghasilkan dua buah deteksi objek: satu objek wajah dan satu objek bukan wajah, dimana seharusnya terdapat dua buah objek wajah. Pada Gambar 6(d) memberikan keluaran system berupa empat buah deteksi objek: 3 objek wajah dan satu objek bukan wajah. Objek wajah/bukan wajah yang telah terdeteksi digambarkan dengan kotak persegi dibagian wajah/bukan wajah.

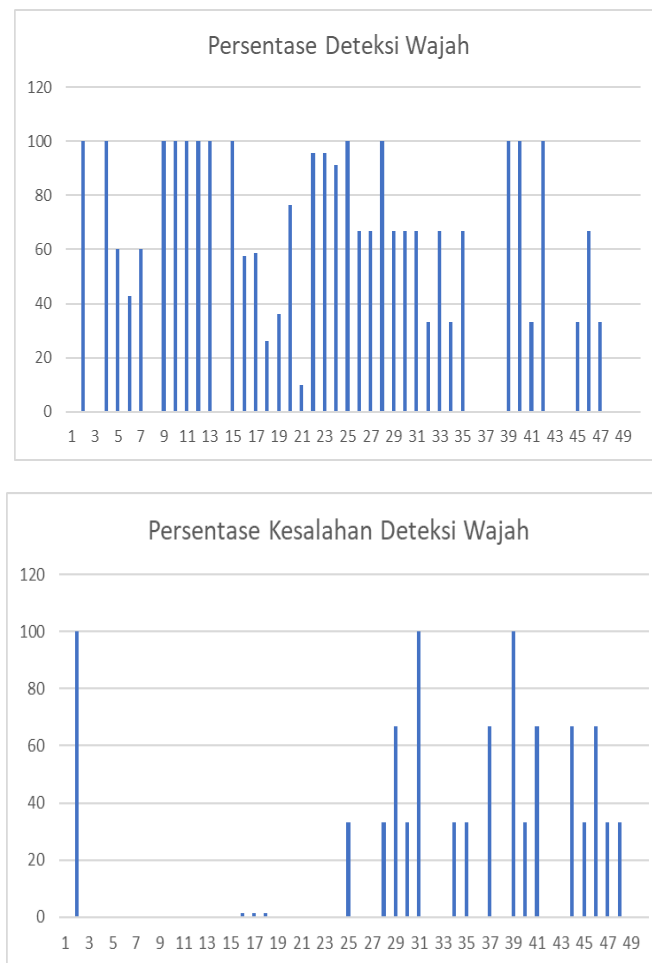
Oleh sebab itu untuk mendapatkan akurasi pada penerapan metode ini, pada Tabel 1 merupakan detail hasil pengujian dari beberapa citra inputan dengan variasi jumlah objek wajah dan posisi wajah. Gambar 7 menampilkan data pengujian berupa grafik.

**Tabel 1. Data Pengujian Deteksi Wajah**

Citra ke-	Jumlah Wajah Asli	Jumlah Wajah Terdeteksi	Rata-Rata Deteksi Wajah( %)	Jumlah Bukan Wajah Terdeteksi	Rata-Rata Kesalahan Deteksi Wajah (%)
1	1	0	0	0	0
2	1	1	100	1	100
3	0	0	0	0	0
4	1	1	100	0	0
5	5	3	60	0	0
6	7	3	43	0	0
7	10	6	60	0	0
8	1	0	0	0	0
9	1	1	100	0	0
10	1	1	100	0	0
11	1	1	100	0	0
12	1	1	100	0	0
13	1	1	100	0	0

Citra ke-	Jumlah Wajah Asli	Jumlah Wajah Terdeteksi	Rata-Rata Deteksi Wajah( %)	Jumlah Bukan Wajah Terdeteksi	Rata-Rata Kesalahan Deteksi Wajah (%)
14	1	0	0	0	0
15	1	1	<b>100</b>	0	0
16	80	46	58	1	1
17	80	47	59	1	1
18	80	21	26	1	1
19	80	29	36	0	0
20	34	26	76	0	0
21	80	8	10	0	0
22	23	22	96	0	0
23	23	22	96	0	0
24	23	21	91	0	0
25	3	3	<b>100</b>	1	33
26	3	2	67	0	0
27	3	2	67	0	0
28	3	3	<b>100</b>	1	33
29	3	2	67	2	67
30	3	2	67	1	33
31	3	2	67	3	100
32	3	1	33	0	0
33	3	2	67	0	0
34	3	1	33	1	33
35	3	2	67	1	33
36	0	0	0	1	0
37	3	0	0	2	67
38	3	0	0	0	0
39	2	2	<b>100</b>	2	100
40	3	3	<b>100</b>	1	33
41	3	1	33	2	67
42	3	3	<b>100</b>	0	0
43	3	0	0	0	0
44	3	0	0	2	67
45	3	1	33	1	33
46	3	2	67	2	67
47	3	1	33	1	33
48	3	0	0	1	33
49	0	0	0	0	0
50	0	0	0	0	0





**Gambar 7 Grafik Akurasi Sistem Deteksi Wajah**

Pada hasil pengujian, persentasi keberhasilan sebesar 100% untuk deteksi wajah adalah pada posisi wajah *frontal*. Akurasi dari metode ini diukur dari posisi wajah. Pada citra inputan yang akan dideteksi harus memiliki fitur wajah pada posisi *frontal*. Dengan analisis citra inputan yang berbeda dengan posisi wajah tidak *frontal*, maka hasil klasifikasi akan sangat rendah bahkan wajah tidak dapat terdeteksi sama sekali. Hasil analisis ini didukung oleh penelitian sebelumnya [5].

## 5. Kesimpulan dan Saran

Berdasarkan hasil pengujian pada deteksi wajah menggunakan metode *Haar Cascade Classifier*, maka didapatkan total akurasi adalah 100%, jika citra inputan memiliki objek wajah dengan posisi *frontal*, sedangkan akurasi dibawah 50% didapatkan dari citra inputan yang memiliki objek wajah *frontal*/tidak *frontal* sama sekali. Hasil ini dipengaruhi oleh posisi wajah yang ada pada citra inputan. Sedangkan untuk kesalahan deteksi, didapatkan hasil sebesar 24%. Pada sistem ini juga telah berhasil mengubah citra inputan berwarna menjadi citra keabuan dengan menggunakan *library* OpenCV. Posisi wajah sangat dipengaruhi dalam mendeteksi wajah. Sistem ini tidak dapat mendeteksi wajah yang memiliki posisi tidak *frontal*.

### Daftar Pustaka

- [1] G. Chern, G. Paul, and S. Jared, "Face Detection," 2003. [Online]. Available: [https://web.stanford.edu/class/ee368/Project\\_03/Project/reports/ee368group01.pdf](https://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group01.pdf).
- [2] W. Lu and S. Sun, "Face Detection in Color Images," 2003. [Online]. Available: [https://web.stanford.edu/class/ee368/Project\\_03/Project/reports/ee368group03.pdf](https://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group03.pdf).
- [3] S. Leahy, "Face Detection on Similar Color Photographs," 2003. [Online]. Available: [https://web.stanford.edu/class/ee368/Project\\_03/Project/reports/ee368group17.pdf](https://web.stanford.edu/class/ee368/Project_03/Project/reports/ee368group17.pdf).
- [4] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," in *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASLAGUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, 2017, doi: 10.1109/FG.2017.82.
- [5] R. Padilla, C. C. Filho, and M. Costa, "Evaluation of haar cascade classifiers designed for face detection," *J. WASET*, 2012.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, doi: 10.1109/cvpr.2001.990517.
- [7] C. HaiYan and L. JunHong, "The Image Process Techonologies in Face Recognition," in *The 2nd International Conference on Information Science and Engineering, Hangzhou, 2010*, pp. 4151-4154, 2010.
- [8] "Face Detection using Haar Cascades." [Online]. Available: [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_objdetect/py\\_face\\_detection/py\\_face\\_detection.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html).
- [9] N. R. Syambas and U. H. Purwanto, "Image processing and face detection analysis on face verification based on the age stages," in *2012 7th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2012*, 2012, doi: 10.1109/TSSA.2012.6366070.
- [10] F. Roman, "Digital Image Processing Techniques for Face Recognition," 2013. [Online]. Available: <https://www.math.ksu.edu/research/i-center/reports/fernandoroman.pdf>.