

Penerapan Metode *Haar-Cascade* Dan LBPH Untuk *Face Detection* dan *Recognition*

Deva Safara Alfani^{1*}, Abdul Rochman², Mutiara Firdaus³, Nendi Setiawan⁴, Perani Rosyani⁵

¹⁻⁵Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46,
Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}safaradeva65@gmail.com, ²rochmanabdul414@gmail.com, ³firdausmutiara600@gmail.com,
⁴nendisetiawan297@gmail.com, ⁵dosen00837@unpam.ac.id

(* : coressponding author)

Abstrak– Salah satu topik yang banyak dibicarakan dalam pengolahan citra dan kecerdasan buatan saat ini adalah pengenalan wajah. Dua teknik yang banyak digunakan untuk deteksi dan pengenalan wajah yang diterapkan dalam banyak studi seperti Local Binary Patterns Histograms (LBPH) untuk pengenalan wajah dan Haar Cascade untuk deteksi wajah, sistem berbasis machine learning Haar Cascade mendeteksi wajah dalam gambar. Metode ini memiliki akurasi deteksi wajah real-time yang sangat baik. Setelah wajah teridentifikasi, pendekatan LBPH digunakan untuk mengenalinya. Untuk menghitung histogram dari pola biner ini, pendekatan pada awalnya menganalisis pola lokal dalam gambar wajah. Lalu, LBPH digunakan untuk mengenali wajah dengan pencahayaan dan ekspresi yang bervariasi. Kami menggunakan berbagai dataset wajah untuk mengevaluasi sistem ini dalam studi kami, dan akurasi serta kecepatan pemrosesan sistem ini cukup memadai.

Kata Kunci: *Face Detection; Face Recognition; Haar Cascade Classifier; OpenCV; Pengolahan Citra*

Abstract– One of the most talked-about topics in image processing and artificial intelligence today is facial recognition. Two widely used techniques for face detection and recognition applied in many studies are Local Binary Patterns Histograms (LBPH) for face recognition and Haar Cascade for face detection. The machine learning-based system Haar Cascade detects faces in images. This method has excellent real-time face detection accuracy. After a face has been identified, the LBPH approach is used to recognize it. To compute the histograms of these binary patterns, the approach initially analyzes local patterns in the facial image. Then, LBPH is used to recognize faces with varying lighting and expressions. We used various face datasets to evaluate this system in our study, and the accuracy and processing speed of this system were sufficient.

Keywords: *Face Detection; Face Recognition; Haar Cascade Classifier; OpenCV; Image Processing*

1. PENDAHULUAN

Sebagai makhluk istimewa, manusia dapat dibedakan dari makhluk lain berdasarkan ciri khasnya sendiri yang merupakan tanda untuk mengenali manusia tersebut. Identifikasi adalah proses pengenalan individu berdasarkan barang milik mereka, pengetahuan kode (seperti PIN atau sandi), atau kombinasi dari keduanya. Biometrik mengacu pada karakteristik alami seseorang, seperti wajah, iris mata, dan sidik jari. Manusia dapat dibedakan dari hewan lain berdasarkan kualitas istimewa karena merupakan makhluk unik. Metode mengidentifikasi seseorang melibatkan pemeriksaan kartu identitas, mencari tahu kode-kode (PIN atau sandi), atau menggunakan kombinasi keduanya. Biometrik adalah atribut alami individu, seperti sidik jari, iris, dan wajah. Fitur wajah yang diperoleh diklasifikasikan menggunakan metode LBPH.

Pada masa kini, penggunaan teknologi identifikasi wajah semakin meluas dalam berbagai aplikasi seperti *video conference*, hubungan antara manusia dan komputer, sistem pencarian, indeks database gambar dan *digital video*, sistem keamanan kontrol akses area terbatas, serta sistem pengenalan biometrik yang dapat dikombinasikan dengan fitur biometrik lain seperti sidik jari dan suara. Untuk mengidentifikasi wajah yang akurat, algoritma pengenalan wajah membandingkan gambar wajah yang diinput dengan wajah yang sudah ada di database. Sementara itu, *facial authentication* akan memeriksa apakah wajah cocok dengan data wajah yang telah dimasukkan sebelumnya atau merupakan wajah yang otentik. Pengolahan wajah mencakup pelacakan wajah, yang mengidentifikasi posisi wajah secara

langsung dalam video, deteksi wajah, yang mengenali wajah dengan asumsi hanya satu wajah dalam gambar, dan identifikasi ekspresi wajah, yang menentukan emosi manusia.

Untuk membuat sistem yang dapat membedakan wajah setiap pengguna, penulis studi ini menggabungkan pendekatan biometrik dengan pengenalan wajah. Kamera laptop digunakan untuk pengambilan wajah guna melakukan deteksi dan pengenalan wajah. Sistem memerlukan sampel pengenalan wajah untuk membangun dataset. Dataset ini dikumpulkan dengan mengumpulkan sampel wajah dari pengambilan gambar wajah. Tahap selanjutnya adalah pelatihan data, yang memungkinkan sistem untuk memvalidasi wajah dan mengidentifikasi pengguna untuk deteksi wajah melalui penggunaan fungsi Haar Cascade Classifier. Penelitian ini menggunakan ekstraksi fitur dengan algoritma Local Binary Pattern Histogram (LBPH) dari library OpenCV yang telah diinstal sebelumnya untuk membedakan wajah setiap pengguna. Untuk mengidentifikasi wajah yang telah dilatih dalam dataset, sistem mencocokkan wajah selama pengujian data. Bergantung pada wajah yang diidentifikasi oleh sistem, nama pengguna akan ditampilkan.

2. METODE PENELITIAN

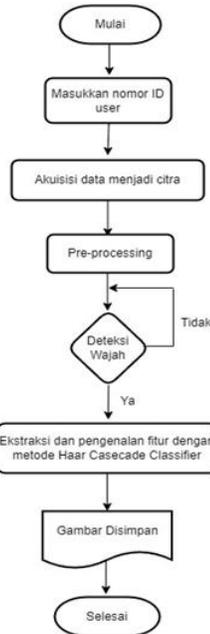
2.1 Arsitektur Sistem

Pada arsitektur sistem yang digunakan, langkah awal adalah memulai program sampai kamera laptop menyala adalah proses penyiapan sistem pengenalan wajah. Menggunakan ID yang dimasukkan oleh pengguna, kamera akan mengambil gambar wajah pengguna, kemudian data akan disimpan dan dikumpulkan dalam kumpulan data (dataset) dalam bentuk gambar dengan nama yang dimulai dengan "_1". Sistem kemudian akan melalui prosedur pelatihan untuk mempelajari cara mengenali gambar wajah pengguna menggunakan ID tersebut. Gambar yang telah dilatih dalam dataset akan disimpan dengan ekstensi .yml dan nama yang diawali dengan "_1". Setelah memperoleh data pelatihan, pengenalan wajah secara real-time akan dilakukan.

Menggunakan karakteristik dari fitur Haar, kamera akan merekam objek berupa wajah. Untuk mengidentifikasi apakah objek di kamera adalah wajah atau bukan, sistem akan membandingkan fitur Haar dalam program dengan objek tersebut. Sistem telah membaca data pelatihan dari _1.yml untuk menentukan wajah pengguna yang telah dikenali berdasarkan nama yang telah dialokasikan sebelumnya. Selanjutnya, sistem akan melakukan prediksi menggunakan fitur LBPH.

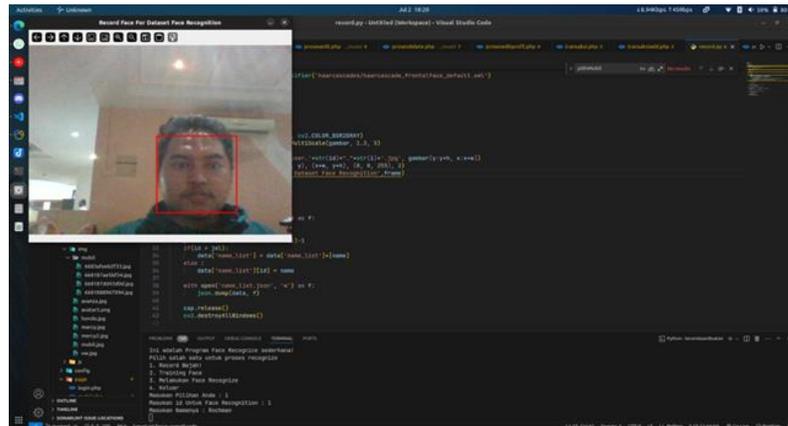
2.2 Pengambilan Data Set Dengan Metode Haar Cascade

Pendekatan Haar Cascade Classifier digunakan dalam metode pengenalan wajah yang digunakan dalam studi ini. Pada dasarnya, fungsi matematika yang menyerupai kotak disebut sebagai Haar Wavelet. Awalnya, pemrosesan gambar dilakukan dengan memeriksa nilai RGB setiap piksel, namun pendekatan ini terbukti tidak berhasil. Pendekatan Haar Cascade Classifier digunakan dalam prosedur deteksi wajah pada penelitian ini. Metode Viola Jones merupakan metode yang paling banyak digunakan untuk mendeteksi citra. Pada dasarnya, istilah "Haar" mengacu kepada fungsi matematika berbentuk kotak yang disebut "Haar Wavelet." Awalnya, pemrosesan gambar dilakukan dengan mengamati nilai RGB tiap pixel, tetapi pendekatan itu tidak berhasil. Kemudian, peneliti Viola dan Jones menciptakan teknik pemrosesan gambar yang menghasilkan karakteristik seperti Haar. Gambar diproses menggunakan fitur seperti Haar dalam kotak-kotak, dengan beberapa piksel di setiap kotak. Setelah itu, kotak-kotak ini diproses untuk menghasilkan nilai perbedaan yang menunjukkan apakah ada area gelap atau terang. Pemrosesan gambar selanjutnya akan dibangun berdasarkan nilai-nilai ini. Viola-Jones digunakan sebagai algoritma deteksi karena tingkat deteksi yang tinggi, cepat dan tangguh untuk deteksi wajah yang 15 kali lebih cepat daripada teknik apa pun pada saat rilis dengan akurasi 95%.



Gambar 1. Flowchart Pengambilan Dataset

Pengguna diharuskan menginput *ID number* yang digunakan untuk mengidentifikasi pengguna dari pengguna lain saat mencoba mengumpulkan dataset. Wajah yang difoto disimpan dalam skala abu-abu dan disimpan dalam berkas .jpg di folder 'dataset'. Untuk tujuan pengumpulan dataset ini, setiap pengguna harus mengambil 100 foto. Kamera otomatis akan hidup dan menunjukkan ilustrasi dari tahapan pengambilan dataset setelah *ID number* pengguna dimasukkan.



Gambar 2. Proses Pengambilan Dataset

Foto-foto yang dikumpulkan awalnya melalui proses *pre-processing* sebelum disimpan. Langkah *pre-processing* bertujuan untuk membuat data lebih kompatibel dengan pustaka yang akan digunakan. *Pra-processing* digunakan dalam penelitian ini untuk menghasilkan foto yang lebih baik.

Dalam banyak waktu, pada kamera menghasilkan *grain* karena pencahayaan yang terlalu terang atau redup, penyalarsan wajah, reduksi kebisingan pada latar belakang pendeteksi, dan pengubahan dimensi. Proses persiapan dan peningkatan kualitas gambar diperlukan untuk mencegah kebisingan yang dapat menyebabkan gambar kurang akurat.

Studi ini menggunakan proses cropping, resizing, dan greyscaling sebagai bagian dari tahap pre-processing. Dengan menggunakan cropping, Anda dapat menghapus bagian-bagian tertentu dari sebuah gambar. Dimensi gambar diubah melalui resizing, yang dapat meningkatkan atau mengurangi resolusi gambar. Terakhir, greyscaling mengubah foto-foto RGB menjadi grayscale dari mode RGB asli (merah, hijau, dan biru) dalam penelitian ini. Gambar-gambar ini memiliki resolusi 100 x 100 piksel dalam format .jpg. Modifikasi-modifikasi ini dimaksudkan untuk mempercepat deteksi dengan meningkatkan kecepatan komputasi. Teknik ini akan menghasilkan hasil yang akan disimpan dalam sebuah dataset.



Gambar 3. Isi Pada Folder Dataset

2.3 Melatih Dataset Menggunakan Python Imaging Library

Pada saat melatih dataset yang harus dilakukan adalah mengumpulkan dan menyiapkan gambar wajah yang ada dalam dataset. Kemudian, gunakan PIL untuk membuka dan memproses setiap gambar, seperti mengubah ukuran atau formatnya, sebelum Anda dapat melatih dataset wajah dan menyimpannya ke file YML menggunakan modul Python Imaging Library (PIL). Selanjutnya, menggunakan teknik deteksi wajah dengan OpenCV untuk mengekstrak fitur dari gambar yang telah diproses. Setelah fitur diekstraksi, gunakan dataset yang telah diproses untuk melatih model pengenalan wajah seperti LBPH (Local Binary Patterns Histograms). Terakhir, simpan model terlatih ke file YML untuk penggunaan berikutnya.

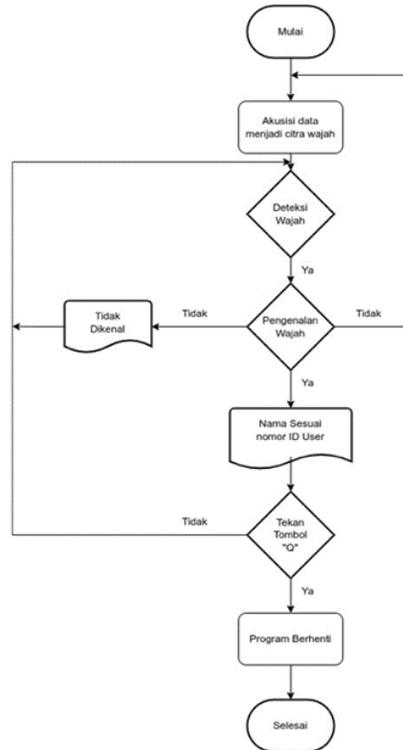
```
kecerdasanBuatan > train.py > getImagesWithLabels
1 import cv2, os
2 import numpy as np
3 from PIL import Image
4
5 def runTrain():
6     recognizer = cv2.face.LBPHFaceRecognizer.create()
7     wajah, ids = getImagesWithLabels('dataset/img')
8     recognizer.train(wajah, np.array(ids))
9     recognizer.save('dataset/training.xml')
10    print("Berhasil mentrain image")
11
12 def getImagesWithLabels(path):
13    detector = cv2.CascadeClassifier("haarcascades/haarcascade_frontalface_default.xml")
14    lokasiGambar = [os.path.join(path, f) for f in os.listdir(path)]
15    print(lokasiGambar)
16    sampelWajah = []
17    ids = []
18
19    for lokGambar in lokasiGambar:
20
21        pilGambar = Image.open(lokGambar).convert('L')
22        arrayGambar = np.array(pilGambar, 'uint8')
23        arrayGambar = cv2.resize(arrayGambar, (100,100), interpolation=cv2.INTER_AREA)
24        equalized_image = cv2.equalizeHist(arrayGambar)
25        arrayGambar = cv2.normalize(arrayGambar, equalized_image, 0, 255, cv2.NORM_MINMAX)
26        id = int(os.path.splitext(lokGambar)[-1].split('.')[1])
27        wajah = detector.detectMultiScale(arrayGambar)
28
29        for (x,y,w,h) in wajah:
30            sampelWajah.append(arrayGambar [y:y+h,x:x+w])
31            ids.append(id)
32
33    return sampelWajah, ids
```

Gambar 4. Source Code Untuk Memasukkan Dataset ke .xml

2.4 Face Recognition Dengan LBPH

Proses pengenalan wajah dengan pendekatan LBPH (Local Binary Patterns Histograms) dimulai dengan identifikasi wajah dari foto. Teknik ini mengkodekan setiap piksel wajah menjadi angka biner yang memiliki hubungan dengan piksel di sekitarnya. Hasilnya adalah set angka biner yang menunjukkan tekstur lokal wajah. Histogram yang menunjukkan distribusi pola biner di seluruh wajah kemudian dibuat dari angka-angka ini. Metode LBPH dapat mengidentifikasi wajah yang paling mirip dengan membandingkan histogram wajah yang tidak dikenal dengan histogram wajah yang ada dalam database.

Setelah histogram dibuat, LBPH menggunakan model pelatihan untuk mengasosiasikan setiap histogram dengan label yang sesuai. Kemudian, dengan membandingkan histogramnya dengan histogram yang telah disimpan dalam model, model ini dapat digunakan untuk menentukan identitas wajah baru. Karena fokusnya pada pola tekstur lokal, LBPH dapat mengenali wajah meskipun ada variasi pencahayaan, sudut pandang, dan ekspresi wajah. Metode ini populer untuk aplikasi real-time dan perangkat dengan sumber daya terbatas karena relatif mudah dan cepat dibandingkan dengan teknik pengenalan wajah lainnya.



Gambar 5. Flowchart Proses Pengenalan Wajah

Jika wajah terdeteksi, nilainya akan menjadi satu, dan gambar persegi warna biru akan tampil untuk menunjukkan bagian wajah yang telah terdeteksi. Setelah itu, LBPH akan memanggil hasil dari *training* pada file format .yml untuk melakukan prediksi dan membandingkannya dengan wajah di webcam. Jika wajah user dikenali, gambar persegi panjang berwarna biru akan muncul di layar bersama dengan tulisan *user name* sesuai identitas. Jika data wajah user yang tidak diketahui terdeteksi, gambar persegi panjang berwarna biru akan muncul di layar bersama dengan keterangan "tidak diketahui". Untuk menghentikan proses pengenalan wajah, tekan "Q" di keyboard. Program akan berhenti.

3. ANALISA DAN PEMBAHASAN

Tujuan dari pengujian ini adalah untuk menilai tingkat keberhasilan sistem dalam mengidentifikasi wajah yang telah terlatih. Wajah yang telah dilatih dihadapkan ke *webcam* dalam jangkauan 25 hingga 125 cm. Program kemudian dijalankan sepuluh kali dan dalam beberapa detik terdeteksi dan mengenali wajah. Ini dilakukan untuk setiap user ID yang telah di-*training*.

Tabel 1. Pengujian Deteksi Wajah Pada User ID

No	Kode User ID	Hasil Pengenalan	Jumlah Pengujian	Hasil Analisa	
				Dikenali	Tidak Dikenali
1	User1		100	85	15
2	User2		100	80	20
3	User3		100	75	25
4	User4		100	82	18

Rumus berikut dapat digunakan untuk menghitung hasil presentase keberhasilan sistem dalam mengenali wajah user yang telah ditraining dan disimpan ke dalam dataset.

$$\text{Pengujian (Tabel 1)} = \frac{\text{Jumlah } n \text{ dikenali}}{\text{Total uji}} \times 100\%$$

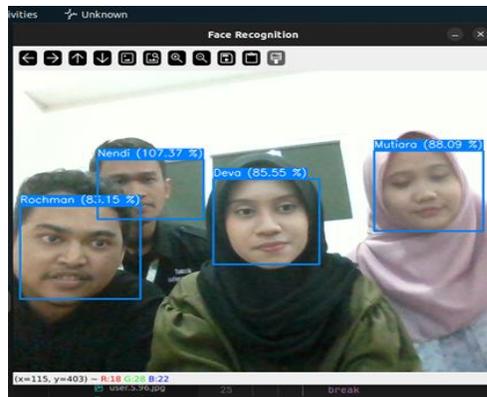
Maka, untuk hasil uji hitung pengenalan wajah yaitu :

$$\begin{aligned} \text{Pengujian (Tabel 1)} &= \frac{85+80+75+82}{400} \times 100\% \\ &= \frac{322}{400} \times 100\% \\ &= 80,5\% \end{aligned}$$

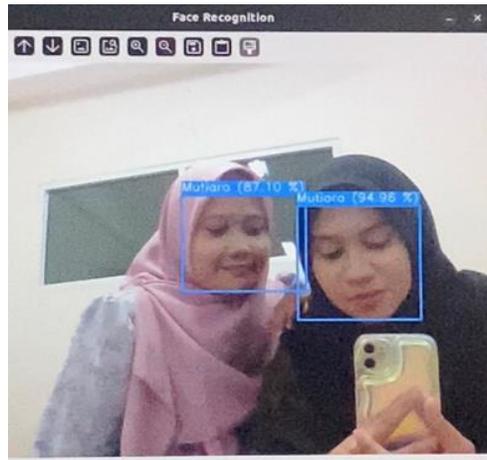
Keberhasilan sistem dalam melakukan *authentication* wajah pengguna sebesar 80,5 persen. Namun, pada kasus beberapa pengujian tidak berhasil. Proses analisis lebih lanjut diperlukan untuk mengetahui alasan mengapa pengujian pengenalan objek wajah user tidak berhasil. Maka dari itu, penulis melakukan pendalaman dengan melihat hasil uji dan mengevaluasi faktor-faktor yang menyebabkan kegagalan tes. Kesimpulan yang penulis berikan adalah:

1. Mengatur tingkat pencahayaan selama pengujian. Untuk mengidentifikasi wajah, pencahayaan harus sebanding dengan tingkat pencahayaan saat sedang mengumpulkan dataset, sehingga wajah dapat diidentifikasi dengan benar. Dengan ini, bagian wajah dan mata dapat terlihat dengan jelas.
2. Pengujian algoritma pendeteksi wajah untuk gambar yang diambil dari kamera pada jarak di atas 125 cm menunjukkan bahwa sistem tidaklah dapat mendeteksi wajah dengan benar. Tidak dapat melakukan pengenalan wajah dalam jangkauan tersebut, tetapi adanya hal lain mengapa algoritma tidak dapat mendeteksi wajah. Karakteristik *face object* dan *brightness* yang tidak cukup dapat menyebabkan objek wajah kurang jelas.
3. Posisi wajah yang tidak fokus dan tidak dalam posisi lurus terhadap *webcam* membuat sulitnya untuk membedakan wajah. Selain itu, mimik wajah berbeda dengan *sample* suatu gambar dalam dataset menyebabkan wajah tidak dikenali.
4. Terdapat *conf* (Confidence) atau skor kepercayaan diri dalam mengenali wajah user, jika lebih dari atau sama dengan 80 dicari *namelist* yang kecocokannya paling tinggi (*similarity score* besar atau *distance similarity* nya kecil). Sedangkan jika skornya kurang dari 50 akan muncul label *unknown* atau tidak dikenali.

Studi ini kemudian melakukan percobaan lanjutan dengan menggunakan empat wajah dalam satu frame. Empat subjek percobaan ditempatkan pada jangkauan 30 cm hingga 170 cm dari *webcam*, dengan seluruh wajah subjek percobaan berada dalam frame yang sama. Dari percobaan tersebut, didapatkan hasil percobaan yang menunjukkan bahwa ada beberapa kesalahan, tetapi sistem masih dapat mengenali wajah user.



Gambar 6. Hasil Percobaan Empat Wajah Dalam Satu Frame



Gambar 7. Kesalahan Dalam mengenali Dua Wajah

4. KESIMPULAN

Penerapan metode Haar-Cascade dan LBPH untuk face detection dan recognition menunjukkan bahwa meskipun kombinasi kedua teknik ini mampu mendeteksi dan mengenali wajah secara cepat, akurasinya mencapai 80,5 persen tetapi masih kurang optimal dan beberapa kali menghasilkan kesalahan pengenalan. Haar-Cascade, meskipun efisien dalam mendeteksi wajah dalam berbagai kondisi pencahayaan dan latar belakang, terkadang gagal mendeteksi wajah dengan tepat, terutama pada sudut pandang yang ekstrem atau ekspresi wajah yang bervariasi. Di sisi lain, LBPH juga menunjukkan keterbatasan dalam mengenali wajah dengan akurasi tinggi ketika terdapat perubahan signifikan pada penampilan subjek. Namun, sistem ini dapat mengenali sampai empat wajah sekaligus dalam satu frame. Oleh karena itu, diperlukan peningkatan pada algoritma atau penggunaan metode tambahan untuk mencapai performa yang lebih andal dan akurat.

REFERENCES

- Rahim, Abdur "Face Recognition using Local Binary Patterns (LBP)." *Global Journal of Computer Science and Technology Graphics & Vision*, Pabna University of Science and Technology Bangladesh, 2013.
- A. Fauzan, M. Novamizanti Ledy S.Si., and S. T. M. T. Fuadah Yunendah Nur, "Perancangan Sistem Deteksi Wajah Untuk Presensi Kehadiran Menggunakan Metode LBPH (Local Binary Pattern Histogram) Berbasis Android", *e-Proceeding of Engineering : Vol.5, No.3*, 2018.
- P. Choirina and R. A. Asmara, "Deteksi Jenis Kelamin Berdasarkan Citra Wajah Jarak Jauh Dengan Metode Haar Cascade Classifier," *J. Inform. Polinema*, vol. 2, no. 4, p. 164, 2016, doi: 10.33795/jip.v2i4.77.
- A. Fahrurrozi and R. Kosasih, "Face Recognition Using Local Binary Pattern Combined With PCA For Images Under Various Expression and Illumination," in *Proceeding on International Workshop on Academic Collaboration 2017*, 2017, no. May, pp. 1-7.
- L. Wang and A. A. Siddique, "Facial recognition system using LBPH face recognizer for anti-theft and surveillance application based on drone technology," *Meas. Control*, vol. 53, no. 7, pp. 1070-1077, 2020, doi: 10.1177/0020294020932344.
- Budiman, A., Fabian, Yupiter, R. A., Achmad, S., & Kurniawan, A. (2023). Student attendance with face recognition (LBPH or CNN): Systematic literature review. *Procedia Computer Science*, 216, 31-38.

- Malhotra, S., Aggarwal, V., Mangal, H., Nagrath, P., & Jain, R. (2021). Comparison between attendance system implemented through haar cascade classifier and face recognition library. IOP Conference Series: Materials Science and Engineering, 1022(1), 012045.
- Thary Al-Ghraiiri, A. H., Abdulwahhab Mohammed, A., & Sameen, E. Z. (2022). Face detection and recognition with 180 degree rotation based on principal component analysis algorithm. IAES International Journal of Artificial Intelligence (IJ-AI), 11(2), 593.
- Ahsan, M. M., Li, Y., Zhang, J., Ahad, M. T., & Gupta, K. D. (2021). Evaluating the Performance of Eigenface, Fisherface, and Local Binary Pattern Histogram-Based Facial Recognition Methods under Various Weather Conditions. Technologies, 9(2).
- Bahit, M., Utami, N. P., Candra, H. K., & Al Madhani, H. (2022). Performance Analysis of The Haar Cascade Classification Method In Performing Face Detection Based on Opencv. The International Conference on Computer Science and Engineering Technology Proceeding (ICCSET), 1(1), 31–37.