



## Sistem Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Berbasis OpenCV

Firdaus<sup>1</sup>, Edi Winarno<sup>2</sup>, Rauzan Habas Addawy<sup>3</sup>, Topan Fadilla Malik<sup>4</sup>, Perani Rosyani<sup>5</sup>

Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl Raya Puspitek No. 46 buaran, Serpong, Kota Tangerang Selatan. Provinsi Banten, Indonesia, 15310

Email: [1firdausbinasrin@gmail.com](mailto:firdausbinasrin@gmail.com), [2rauzanhabas28@gmail.com](mailto:rauzanhabas28@gmail.com), [3edie.winz@gmail.com](mailto:edie.winz@gmail.com), [4pangyourbae@gmail.com](mailto:pangyourbae@gmail.com), [5dosen00837@unpam.ac.id](mailto:dosen00837@unpam.ac.id).

**ABSTRAK** - Pengenalan wajah merupakan teknologi biometrik yang memanfaatkan analisis pengolahan citra untuk mengidentifikasi individu. Proses ini melibatkan dua tahap utama: deteksi dan identifikasi wajah. Meskipun kedua proses ini sebelumnya dikembangkan secara terpisah, melalui sistem pengenalan wajah, identitas seseorang dapat dengan mudah dikenali hanya dengan menggunakan kamera. Algoritma Local Binary Pattern Histogram (LBPH) adalah salah satu saca yang efektif diterapkan dalam analisis karakter wajah untuk deteksi dan pengenalan. LBPH bekerja dengan mengubah tekstur wajah menjadi nilai biner yang mewakili piksel-piksel wajah dalam pola lingkaran dengan pusat sebagai acuan. Jarak antara nilai biner ini disebut neighbors. OpenCV, sebagai modul pengolahan citra yang kuat, digunakan untuk membuat, mengubah, dan memodifikasi citra digital, termasuk dalam pengolahan citra wajah. Hasil uji coba citra wajah menunjukkan bahwa sistem pengenalan wajah yang dikembangkan memiliki akurasi yang sangat sebesar. Hasil ini menunjukkan potensi besar dari sistem berbasis LBPH dan OpenCV dalam aplikasi pengenalan wajah yang efisien dan akurat.

Kunci : Identifikasi Wajah, Algoritma Haar, LBPH.

**ABSTRACT** - Facial recognition is a biometric technology that utilizes image processing analysis to identify individuals. This process involves two main stages: face detection and identification. Although these two processes were previously developed separately, through facial recognition systems, a person's identity can be easily recognized just by using a camera. The Local Binary Pattern Histogram (LBPH) algorithm is one method that is effectively applied in facial character analysis for detection and recognition. LBPH works by converting facial texture into binary values that represent facial pixels in a circular pattern with the center as a reference. The distance between these binary values is called neighbors. OpenCV, as a powerful image processing module, is used to create, change and modify digital images, including facial image processing. The results of facial image trials show that the facial recognition system developed has very high accuracy. These results demonstrate the great potential of LBPH and OpenCV-based systems in efficient and accurate facial recognition applications.

Key: Face Identification, Haar Algorithm, LBPH.

## 1. PENDAHULUAN

Sistem identifikasi wajah telah menjadi salah satu teknologi biometrik yang paling diminati dan banyak digunakan di berbagai sektor. Teknologi ini memanfaatkan analisis pengolahan citra untuk mengidentifikasi identitas seseorang. Terdapat dua aplikasi utama dari sistem pengenalan wajah, Proses yang dilakukan adalah verifikasi dan identifikasi. Verifikasi dilakukan untuk memastikan kebenaran data yang melibatkan proses pencocokan data wajah baru dengan data yang sudah ada di basis data (one-to-one), dan hasilnya biasanya berupa konfirmasi true atau false. Identifikasi, di sisi lain, melibatkan proses mengenali seseorang dengan membandingkan data wajah dengan semua informasi yang tersimpan dalam basis data (one-to-many), memberikan hasil berdasarkan tingkat kemiripan atau kedekatan.

Dalam era teknologi yang semakin maju, kebutuhan akan Sistem pengenalan wajah secara real-time semakin penting. Inovasi teknologi di masa depan dituntut untuk lebih cerdas, cepat, efisien, dan praktis. Dalam konteks ini, pengenalan wajah memiliki peran penting dalam menjaga kerahasiaan dan keamanan data pribadi. Proses identifikasi wajah dapat memastikan akses yang aman dan pelayanan yang lebih baik, menjadikannya komponen integral dalam berbagai aplikasi keamanan dan kenyamanan sehari-hari.



Pengenalan wajah adalah teknik biometrik yang memungkinkan komputer atau perangkat otentifikasi untuk mengidentifikasi wajah manusia. Implementasi teknologi ini melibatkan penggunaan kamera atau webcam untuk menangkap citra wajah secara real-time, kemudian mencocokkan citra tersebut dengan dataset wajah yang ada. Namun, pengenalan wajah tidak bebas dari tantangan. Variasi ekspresi wajah, kondisi pencahayaan yang berubah-ubah, dan jarak antara kamera dan subjek adalah beberapa faktor yang dapat mempengaruhi akurasi sistem.

Penelitian ini bertujuan untuk mengembangkan sistem pengenalan wajah yang berfungsi dengan baik dan efisien dengan menggunakan metode Haar Cascade dan Lokal Binary Pattern Histograms. Haar Cascade dimodifikasi untuk deteksi wajah, sedangkan LBPH digunakan untuk menentukan ciri-ciri wajah. Metode Haar Cascade dipilih karena kemudahannya dalam pengenalan wajah dan ketersediaan dukungan dari pengembang, yang memudahkan proses implementasi. Metode LBPH, di sisi lain, digunakan karena kemampuannya dalam mengidentifikasi tekstur gambar dengan menganalisis kerapatan piksel, yang dapat meningkatkan akurasi prediksi wajah.

## 2. METODE PENELITIAN

### 2.1 implementasi metode

#### 2.1.1 Haar Cascade

Haar Cascade adalah sebuah teknik yang digunakan untuk deteksi objek dalam gambar berbasis fitur. Teknik ini awalnya dikembangkan untuk mendeteksi wajah dalam gambar, tetapi sekarang telah diterapkan untuk deteksi berbagai jenis objek lainnya. Dengan memanfaatkan metode Haar Cascade, deteksi wajah dalam gambar dapat diimplementasikan dengan efisiensi dan dapat diandalkan. Pendekatan ini merupakan langkah awal yang sangat penting dalam berbagai aplikasi untuk pengenalan wajah dan deteksi objek lainnya. Metode Haar Cascade mengizinkan identifikasi objek dengan mengenali fitur-fitur yang telah dipelajari, memungkinkan sistem untuk secara efektif mengenali wajah dalam berbagai kondisi pencahayaan dan ekspresi.

Sistem pengenalan yang diterapkan pada platform berbasis desktop menggunakan kamera webcam untuk identifikasi. Proses dimulai dengan pengambilan sampel wajah baru, kemudian dilanjutkan dengan pengambilan sampel wajah baru dan dilanjutkan dengan identifikasi. Admin melakukan pengisian biodata kandidat sebelum data diunggah, dan sistem akan memunculkan pop up untuk pengambilan sampel wajah. Deteksi wajah dilakukan dengan fitur Haar Cascade frontal face yang menangkap wajah dari arah depan. Haar Cascade merupakan file xml yang berisikan data objek yang telah dilatih atau dikenali sebelumnya, digunakan untuk mendeteksi objek seperti wajah yang ditangkap oleh kamera.

Proses pengambilan sampel melibatkan penangkapan wajah sebanyak 30 kali, dengan satu kali penyimpanan setiap kali sistem berhasil mendeteksi wajah. Setelah pengambilan wajah selesai, pop up muncul untuk melakukan pelatihan data sampel wajah yang telah diambil agar dapat dikenali oleh sistem. Pada tahap ini, metode LBPH digunakan. Sistem komputer hanya mengenali kode dalam bentuk biner, sehingga data sampel wajah dikonversi dalam bentuk biner menggunakan metode ini. Tahap ini dikenal sebagai ekstraksi data wajah menjadi biner oleh metode LBPH. Setelah proses pelatihan selesai, tahap identifikasi wajah dapat dilakukan.

Pengolahan data melibatkan pengolahan citra yang kemudian diproses menjadi data citra yang dapat digunakan untuk melatih dan pengujian. Data dikumpulkan dengan mengambil citra wajah melalui kamera streaming secara real time, kemudian disimpan dalam folder dan diberi label untuk setiap data citra wajah tersebut. Objek penelitian yang digunakan adalah citra wajah laki-laki dan Perempuan dengan rasio 17:13 dengan total sekitar 30 data citra wajah digital, satu objek mendekati citra wajah seseorang dan satu orang berupa sketsa wajah murni.

#### 2.1.2 Metode LBPH



Metode LBPH (Local Binary Patterns Histograms) adalah teknik pengenalan wajah yang menggunakan pola biner lokal untuk mengidentifikasi tekstur dalam gambar. Prosesnya dimulai dengan membagi gambar wajah menjadi sel-sel kecil. Setiap piksel dalam sel dibandingkan dengan delapan tetangga sekitarnya; jika nilai piksel tetangga sama atau lebih besar dari nilai piksel tengah, diberi nilai 1, jika tidak, diberi nilai 0. Nilai-nilai biner ini digabungkan menjadi kode biner yang dikonversi ke bentuk desimal. Histogram lokal yang menggambarkan frekuensi nilai desimal dihitung untuk setiap sel, dan histogram-histogram ini digabungkan untuk membentuk fitur deskriptif wajah. Fitur deskriptif dari gambar wajah baru kemudian dibandingkan dengan fitur dari gambar-gambar wajah yang telah dikenal sebelumnya untuk mengidentifikasi atau memverifikasi identitas wajah tersebut. Metode LBPH efektif dalam berbagai kondisi pencahayaan dan sudut pandang, serta memiliki proses yang relatif sederhana dan cepat.

a. Proses Ekstraksi Wajah dengan Metode LBPH.

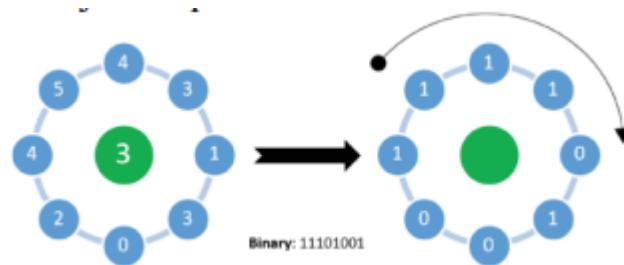
Proses ekstraksi wajah dengan metode LBPH (Local Binary Patterns Histograms) dimulai dengan membagi gambar wajah menjadi beberapa sel kecil. Pada setiap sel, setiap piksel dibandingkan dengan delapan tetangganya. Jika nilai piksel tetangga sama atau lebih besar dari nilai piksel tengah, maka piksel tetangga tersebut diberi nilai 1; jika tidak, diberi nilai 0. Hasil perbandingan ini membentuk pola biner untuk setiap piksel. Pola biner ini kemudian dikonversi ke dalam bentuk desimal, sehingga menghasilkan nilai desimal untuk setiap piksel dalam sel.

Selanjutnya, histogram lokal yang berisi frekuensi nilai desimal dihitung untuk setiap sel. Histogram-histogram ini digabungkan untuk membentuk fitur deskriptif dari wajah. Fitur deskriptif ini kemudian digunakan untuk mengidentifikasi atau memverifikasi identitas wajah dengan membandingkannya dengan fitur dari gambar wajah yang telah dikenal sebelumnya. Metode LBPH terkenal karena kemampuannya bekerja dengan baik dalam berbagai kondisi pencahayaan dan sudut pandang, serta prosesnya yang relatif sederhana dan cepat.

b. Implementasi Menggunakan OpenCV

Berikut adalah langkah-langkah yang diambil dalam implementasi kode untuk memahami dan menerapkan pengenalan wajah menggunakan OpenCV:

1. Mengumpulkan Dataset Wajah  
Kumpulkan gambar wajah yang akan digunakan sebagai dataset. Simpan gambar-gambar ini dalam struktur folder yang sesuai di Firebase.
2. Menggunakan Haar Cascade untuk Deteksi Wajah  
Gunakan metode Haar Cascade untuk mendeteksi wajah dari gambar atau video stream.
3. Melatih Model LBPH untuk Pengenalan Wajah  
Setelah mendeteksi wajah, langkah berikutnya adalah melatih model LBPH dengan gambar wajah yang sudah dikumpulkan.
4. Mengimplementasikan Pengenalan Wajah Real-Time
5. Integrasi dengan Firebase  
Integrasikan sistem dengan Firebase untuk menyimpan dan mengakses dataset wajah secara efisien. Gunakan pustaka firebase-admin untuk berinteraksi dengan Firebase.



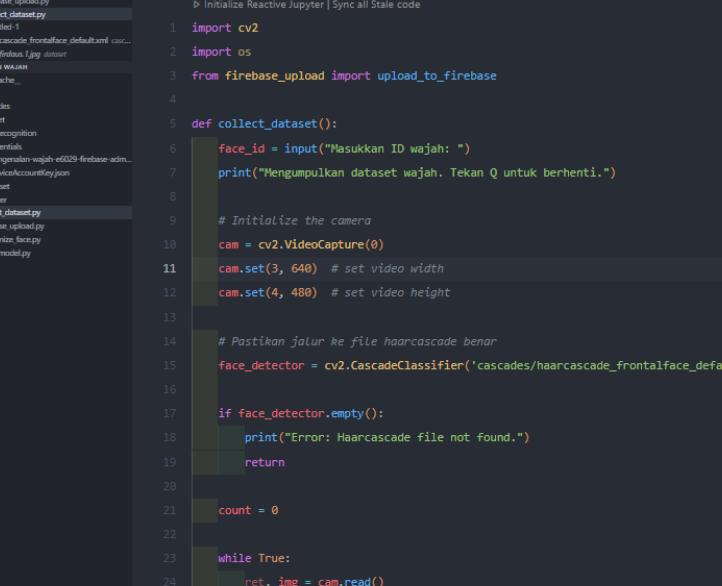
Gambar 1

### Keterangan :

- Pilih Piksel Pusat Pada gambar kiri, piksel pusat (dalam lingkaran hijau dengan nilai 3) dipilih untuk dihitung nilai biner di sekitarnya.
- Bandingkan dengan Piksel Tetangga Setiap piksel di sekitar piksel pusat dibandingkan dengan nilai piksel pusat. Jika nilai piksel tetangga lebih besar atau sama dengan nilai piksel pusat, maka piksel tetangga diberi nilai 1, sebaliknya diberi nilai 0. Pada gambar, ini menghasilkan pola biner di sekitar piksel pusat.
- Membentuk Nilai Biner Nilai biner dari piksel tetangga kemudian diurutkan dalam bentuk lingkaran, dimulai dari posisi yang ditentukan (biasanya kiri atas), dan bergerak searah jarum jam. Dalam contoh ini, hasilnya adalah 11101001.
- Menghitung Nilai Desimal Nilai biner yang dihasilkan dikonversi menjadi nilai desimal. Nilai ini digunakan untuk merepresentasikan tekstur di area sekitar piksel pusat dalam bentuk histogram.

### 2.3 Implementasi kode

Berikut source kode yang kami gunakan dalam pembuatan project ini:



The screenshot shows a Jupyter Notebook interface with the following details:

- File Explorer:** On the left, it shows the project structure. The main folder is "PENGENALAN WAJAH", which contains "pycache", "venv", "cascades", "dataset", and "face\_recognition". "face\_recognition" contains "credentials", "haarcascade\_frontalface\_default.xml", "serviceAccountKey.json", "dataset", and "trainer". "dataset" contains "collect\_dataset.py", "firebase\_upload.py", "recognize\_face.py", and "train\_model.py".
- Code Editor:** The main area displays Python code for a "collect\_dataset" function. The code uses OpenCV (cv2) and the Firebase Python client library to capture video from a camera, convert it to grayscale, and use a Haar cascade classifier to detect faces. It also uploads detected faces to a Firebase storage.
- Toolbar:** At the top, there are standard Jupyter Notebook toolbar icons for File, Edit, Selection, View, Go, Run, and Help.
- Status Bar:** At the bottom, it shows "TIMELINE" and "Select folder".

```
import cv2
import os
from firebase_upload import upload_to_firebase

def collect_dataset():
    face_id = input("Masukkan ID wajah: ")
    print("Mengumpulkan dataset wajah. Tekan Q untuk berhenti.")

    # Initialize the camera
    cam = cv2.VideoCapture(0)
    cam.set(3, 640) # set video width
    cam.set(4, 480) # set video height

    # Pastikan jalur ke file haarcascade benar
    face_detector = cv2.CascadeClassifier('cascades/haarcascade_frontalface_default.xml')

    if face_detector.empty():
        print("Error: Haarcascade file not found.")
        return

    count = 0

    while True:
        ret, img = cam.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_detector.detectMultiScale(gray, 1.3, 5)

        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
            cv2.putText(img, str(count), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

            # Save the face
            face = img[y:y+h, x:x+w]
            cv2.imwrite(f'dataset/{face_id}_{count}.jpg', face)

            count += 1

        cv2.imshow('Video', img)

        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cam.release()
    cv2.destroyAllWindows()
```

Gambar 2. implementasi kode



### 3. ANALISA DAN PEMBAHASAN

Dalam proses sistem pengenalan wajah, Anda dapat melakukan langkah-langkah berikut untuk mendapatkan hasil yang lebih baik:

1. Pengumpulan Dataset Wajah (collect\_dataset.py)

Tujuan mengumpulkan sampel gambar wajah untuk melatih sistem pengenalan.

Langkah-langkah:

- Input ID Wajah Pengguna diminta untuk memasukkan ID yang akan digunakan untuk mengidentifikasi wajah.
- Inisialisasi Kamera diaktifkan dan diatur resolusinya.
- Deteksi Wajah menggunakan Haar Cascade untuk mendeteksi wajah dalam frame kamera.
- Penyimpanan Dataset setiap wajah yang terdeteksi disimpan sebagai gambar dengan format user.<face\_id>.<count>.jpg di dalam folder dataset/.
- Pengunggahan ke Firebase setiap gambar yang disimpan diunggah ke Firebase Storage untuk penyimpanan yang aman dan terkelola.

2. Pelatihan Recognizer (train\_recognizer.py)

Tujuan melatih model pengenalan wajah menggunakan dataset yang telah dikumpulkan.

Langkah-langkah:

- Inisialisasi Recognizer membuat objek recognizer menggunakan metode LBPH dari OpenCV.
- Baca Dataset memuat semua gambar wajah dari folder dataset/ untuk pelatihan.
- Pelatihan Recognizer memproses setiap gambar, mengekstrak fitur wajah, dan melatih model pengenalan menggunakan recognizer.train.
- Simpan Model model yang dilatih disimpan sebagai file trainer.yml di dalam folder trainer/.
- Pengunggahan ke Firebase model yang dilatih diunggah ke Firebase Storage untuk kemudahan akses dan keamanan.

3. Pengenalan Wajah Real-Time (recognize\_face.py)

Tujuan mengenali wajah secara real-time menggunakan kamera.

Langkah-langkah:

- Inisialisasi Recognizer dan Detektor Wajah memuat model pengenalan wajah (trainer/trainer.yml) dan Haar Cascade untuk deteksi wajah.
- Ambil Data Video mengaktifkan kamera dan mengambil frame secara berkelanjutan.
- Deteksi Wajah menggunakan Haar Cascade untuk mendeteksi wajah dalam setiap frame.
- Pengenalan Wajah memproses setiap wajah yang terdeteksi, menerapkan model pengenalan untuk mengidentifikasi wajah yang dikenali.
- Tampilkan Hasil menampilkan frame video dengan kotak di sekitar wajah yang terdeteksi dan label nama jika dikenali.
- Interaksi Pengguna program berjalan dalam loop sampai pengguna menekan tombol 'q' untuk keluar.

Proses pengenalan citra wajah berhasil ketika 30 gambar telah diambil, yang kemudian disimpan dalam folder bernama "dataset" Setelah memproses semua citra, pengujian dilakukan

menggunakan metode LBPH. Hasil pengujian mencakup tiga jenis citra, (1) citra wajah asli, (2) citra wajah digital dari foto smartphone, dan (3) sketsa citra wajah berbentuk oval



Gambar 3. dataset kumpulan citra wajah

#### 4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan tentang sistem pengenalan wajah menggunakan metode Local Binary Pattern Histogram (LBPH) berbasis OpenCV, dapat disimpulkan beberapa hal penting:

1. Efektivitas Metode LBPH LBPH terbukti efektif dalam mengidentifikasi wajah berdasarkan tekstur gambar. Metode ini mengubah tekstur wajah menjadi representasi biner yang mampu meningkatkan akurasi dalam pengenalan identitas.
2. Penggunaan Haar Cascade Penggunaan Haar Cascade sebagai teknik deteksi wajah memungkinkan sistem untuk dengan cepat dan akurat mendeteksi keberadaan wajah dalam gambar atau video.
3. Implementasi dan Pengujian Proses implementasi sistem dilakukan dengan mengumpulkan dataset wajah, melatih model menggunakan LBPH, dan mengintegrasikan sistem untuk pengenalan wajah real-time. Pengujian sistem melibatkan berbagai kondisi citra wajah untuk menguji keandalan dan akurasi sistem.
4. Keunggulan LBPH Dibandingkan dengan metode lain seperti Eigenface, LBPH menunjukkan keunggulan dalam efisiensi dan akurasi, terutama dalam menghadapi variasi ekspresi wajah dan kondisi pencahayaan yang berbeda.
5. Aplikasi Potensial Teknologi ini memiliki aplikasi potensial yang luas dalam berbagai bidang seperti keamanan, pengawasan, dan pengelolaan data pribadi dengan menggunakan pengenalan wajah sebagai alat biometrik yang aman dan efisien.

#### REFERENCES

Setiawan, Fajar, and Dewi Agushinta. "Sistem Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Pada Firebase Berbasis Open CV." *Prosiding Seminar SeNTIK*. Vol. 4. No. 1. 2020.

Pratama, Tutus Lusni, Made Ayu Dusea Widyadara, and Julian Sahertian. "Perbandingan Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Dan Eigenface Untuk Presensi." *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*. Vol. 5. No. 1. 2021.

Tarigan, Ita Augustina, and Asep Kurniawan. "Prototipe Pendekripsi dan Pengenalan Wajah Berbasis Web Menggunakan Algoritma Local Binary Pattern Histogram untuk Absensi." *MULTINETICS* 8.1 (2022): 77-86.

Abidin, Suhepy. "Deteksi wajah menggunakan metode Haar Cascade Classifier berbasis webcam pada Matlab." *J. Teknol. Elekterika* 15.1 (2018): 21.

Anarki, Galang Aprilian, Karina Auliasari, and Mira Orisa. "Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker." *JATI (Jurnal Mahasiswa Teknik Informatika)* 5.1 (2021): 179-186.



**JRIIN: Jurnal Riset Informatika dan Inovasi**

**Volume 1, No. 11 April 2024**

**ISSN 3025-0919 (media online)**

**Hal 1199-1205**

Kenda, Patris. "Sistem Presensi Berbasis Wajah Dengan Metode Haar Cascade." *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi* 1.2 (2021): 419-429.

Ahmad, Fathul Luthfillah, Anan Nugroho, and Alfa Faridh Suni. "Deteksi pemakai masker menggunakan metode haar cascade sebagai pencegahan covid 19." *Edu Elektrika Journal* 10.1 (2021): 13-18.

Indra, Evta, et al. "Desain Dan Implementasi Sistem Absensi Mahasiswa Berdasarkan Fitur Pengenalan Wajah Dengan Menggunakan Metode Haar-Like Feature: Sistem Informasi." *Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP)* 2.2 (2019): 363-370.

Prathivi, Rastri, and Yunita Kurniawati. "Sistem Presensi Kelas Menggunakan Pengenalan Wajah Dengan Metode Haar Cascade Classifier." *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer* 11.1 (2020): 135-142.