

Simulasi Pemanfaatan Aplikasi Sistem Deteksi Pengguna Masker Pada Ruangan Laboratorium

Islahul Anwar¹, Giri Wahyu Wiriasto²
anwar.islahul@gmail.com¹, giriwahyuwiriasto@unram.ac.id²

^{1,2}Universitas Mataram

Abstract: *During the post-pandemic or endemic period, the government continues to recommend the use of masks and maintenance of hygiene when engaging in activities outside the home. This extends to activities on the campus of the University of Mataram, particularly in the electrical engineering computer laboratory, where students are consistently encouraged to wear masks. To streamline activities in the laboratory, a Mask Detection System using the Haar Cascade Classifier (HCC) Method integrated with a Telegram Bot has been tested. The trial implementation aims to instill discipline among students in mask usage. This system was developed using Visual Studio Code and Python 3.10 tools, with image data input from a webcam. The HCC method is employed to detect students wearing masks or not in the laboratory, utilizing webcam-captured image data. Subsequently, the Telegram bot sends images if individuals are identified wearing masks. The study results demonstrate the system's capability to detect masks within a range of 30 cm to 120 cm. The warning feature, consisting of recorded images of students wearing masks sent via the Telegram bot, operates effectively.*

Keywords: *deteksi masker, haar cascade classifier, bot telegram, endemi covid-19*

Pendahuluan

Pada tahun 2019, seluruh dunia, termasuk Indonesia, menghadapi pandemi Covid-19. Saat ini, perlahan beralih ke kondisi endemik dengan berakhirnya kebijakan Pemberlakuan Pembatasan Kegiatan Masyarakat (PKKM) oleh pemerintah. Meskipun berada dalam kondisi endemik, ini tidak berarti virus telah hilang. Semua instansi publik tetap berhati-hati dan meneruskan kebiasaan menggunakan masker serta menjaga kebersihan dengan mencuci tangan. Penggunaan masker menjadi perlindungan pernafasan yang digunakan untuk melindungi individu dari menghirup zat-zat bahaya atau kontaminan yang mungkin berada di udara.

Kondisi saat itu mendorong banyak pihak untuk mengembangkan peralatan, terutama teknologi, guna memantau kondisi di ruang publik. Salah satunya adalah kami dari jurusan

Teknik Elektro Universitas Mataram. Beberapa kegiatan Praktek Kerja Lapangan mahasiswa terlibat dalam pengembangan perangkat lunak sederhana yang dapat memantau peserta mahasiswa di dalam laboratorium. Tujuannya adalah untuk mengedukasi dan mendisiplinkan mahasiswa dalam penggunaan masker. Perangkat lunak ini dilengkapi dengan sensor berupa webcam atau kamera web yang terpasang di beberapa sudut ruangan, digunakan untuk mendeteksi apabila terdapat mahasiswa yang tidak menggunakan masker di dalam ruangan dan memberi indikator peringatan bagi peserta yang tidak sedang menggunakan saat itu agar segera menggunakan masker. Perangkat lunak ini bekerja menggunakan metode *Haar Cascade Classifier* dan hasil monitoring akan langsung terkirim kepada admin melalui aplikasi *bot telegram*.

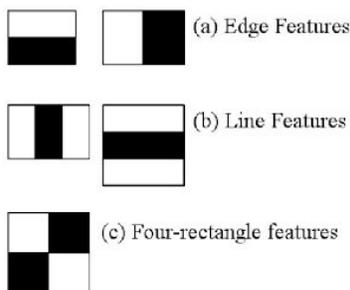
OpenCV (Opensource Computer Vision Library), adalah sebuah *library opensource* yang dikembangkan oleh intel yang fokus untuk menyederhanakan *programming* terkait citra digital. Di dalam OpenCV sudah mempunyai banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman filtering, dan berbagai jenis metode AI (*Artificial Intellegence*). Dan menyediakan berbagai algoritma sederhana terkait *Computer Vision* untuk *low level API* [2]. OpenCV merupakan *opensource computer vision library* untuk bahasa pemrograman C/C++, dan telah dikembangkan sehingga dapat digunakan Bahasa pemrograman lain salah satunya python.

Telegram Bot *Application Programming Interface (API)* atau API - robot Telegram adalah sebuah teknologi *opensource* yang disediakan oleh Telegram Messenger LLP untuk membangun aplikasi bot Telegram bagi para pengembang. Bot API ini merupakan *interface* berbasis HTTP untuk menghubungkan bot yang dikembangkan oleh para pengembang dengan sistem Telegram (<https://core.telegram.org/>).

Kelebihan dari Telegram ini adalah adanya landasan untuk menggunakan *Application Programming Interface (API)* untuk masyarakat luas (Cokrojoyo et al, 2017). Salah satu API yang disediakan adalah fitur bot. Bot Telegram adalah bot yang saat ini mulai banyak dipergunakan.

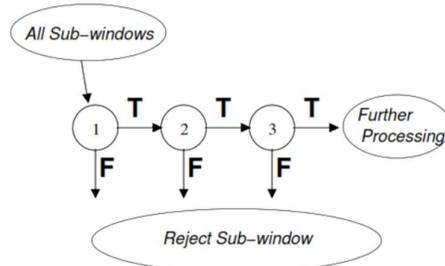
Haar Cascade Classifier (HCC) adalah algoritma pembelajaran mesin untuk mendeteksi objek. Algoritma ini adalah pendekatan berbasis pembelajaran mesin yang menggunakan fungsi *cascade*, dimana fungsi ini dilatih dari berbagai citra positif dan negatif. Citra positif merupakan citra yang memiliki objek 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 (Viola & Jones, 2001).



Gambar 1. Fitur *haar*

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 (Viola & Jones, 2001).



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 [8].

Metode

OpenCV digunakan untuk mengimplementasikan metode *Haar Cascade Classifier* agar sistem dapat mendeteksi penggunaan masker pada citra inputan dari *webcam* laptop. Data yang menjadi inputan merupakan data citra berwarna yang memiliki variasi jumlah objek wajah dan mulut. Kemudian citra berwarna tersebut akan diubah terlebih dahulu menjadi citra keabuan (*grayscale*) kemudian objek yang memiliki wajah akan terdeteksi. Kemudian apabila area mulut terdeteksi maka dianggap tidak menggunakan masker sebaliknya jika area mulut tidak terdeteksi maka dianggap menggunakan masker.



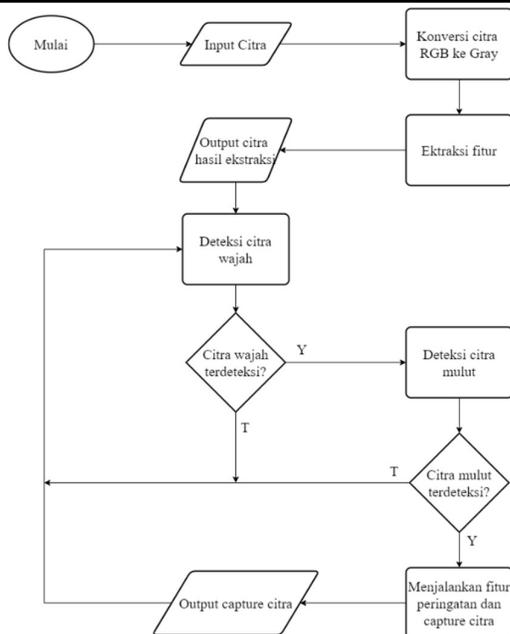
Gambar 3. Metode Pengembangan sistem dan pelaksanaan pengabdian

Metode *HCC* digunakan karena memiliki tingkat akurasi yang tinggi dan cepat dalam melakukan klasifikasi terhadap citra input yang diberikan. Tahapan yang dilakukan pada penelitian ini adalah melakukan beberapa kajian literatur yang berkaitan dengan metode *HCC* dan beberapa teori pendukung lainnya tentang pemrosesan data digital. Kemudian tahapan perancangan sistem dengan bahasa pemrograman python dengan beberapa modul *library* Python seperti *OpenCV*. Implementasi fitur *HCC* didapatkan dari modul *library* *OpenCV* yaitu *haarcascade_frontalface_default.xml* yang akan digunakan untuk klasifikasi wajah pada citra inputan. Sedangkan modul *library* yang digunakan untuk klasifikasi mulut untuk menentukan penggunaan masker adalah *haarcascade_smile.xml*.

Tahapan akhir dari penelitian ini adalah menganalisa serta menguji terhadap sistem yang telah dibuat untuk mendeteksi penggunaan masker dimana jika terdeteksi tanpa masker bot telegram akan memberikan notifikasi berupa hasil *capture* citra yang tidak menggunakan masker.

Gambaran Sistem

Dalam diagram alir sistem pada gambar 4 tampak kamera *webcam* meng-*capture image* berupa citra dengan kanal warna *red, green, blue* (RGB).



Gambar 4. Diagram alir sistem

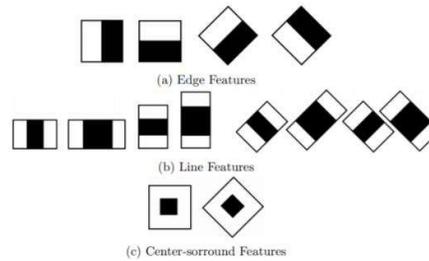
Kemudian citra input RGB dikonversi menjadi citra *grayscale* atau citra skala keabuan. Setelah didapatkan citra skala keabuan dilakukan ekstraksi fitur *haar cascade classifier* diantaranya *haar like feature*, *integral image*, *adaboost learning* dan *cascade classifier*. Setelah didapatkan output citra hasil ekstraksi dilakukan pendeteksian citra wajah. Jika citra wajah tidak terdeteksi maka sistem selesai atau tidak dilanjutkan. Sedangkan, jika citra wajah terdeteksi maka sistem dilanjutkan dengan mendeteksi hidung dan mulut. Jika citra mulut dan hidung tidak terdeteksi maka sistem tidak akan dilanjutkan atau selesai. Sedangkan, jika citra mulut dan hidung terdeteksi maka fitur peringatan dan *capture* citra dijalankan. Hasil dari *capture* kemudian dikirim oleh *bot telegram* ke grup telegram dan dilakukan pendeteksian citra wajah kembali.

Haar Like Feature

Haar like feature (HLF) merupakan fitur yang didasarkan pada algoritma *wavelet haar*. Setiap *HLF* terdiri dari gabungan kotak hitam dan putih. *Haar* adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Untuk dua dimensi, satu terang dan satu gelap selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik (Prathivi & Kurniawati, 2020).

Pengelompokan image/citra dilakukan berdasarkan nilai fitur, ini bertujuan untuk memisahkan citra yang tidak diperlukan, dalam kasus ini warna latar tidak ikut dihitung. Terdapat tiga jenis fitur berdasarkan jumlah persegi panjang (terang dan gelap) yang terdapat di dalamnya yaitu

pola dua (*edge features*), pola tiga (*line features*), dan pola empat persegi panjang (*four-rectangle features*) seperti gambar 5 (Anarki et al, 2021).



Gambar 5. Haar like feature

Dengan adanya fitur *haar* ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau *threshold*, maka dapat dikatakan bahwa difitur tersebut ada (Syarif & Wijanarto, 2015). Nilai dari *haar like feature* adalah perbedaan antara jumlah nilai piksel *greyscale level* dalam daerah kotak hitam dan daerah kotak putih.

Integral Image

Integral Image digunakan untuk menentukan ada atau tidaknya dari ratusan fitur *Haar* pada sebuah gambar dan pada skala yang berbeda secara efisien. Proses pencarian nilai fitur ini dilakukan secara interatif mulai dari ujung kiri atas hingga ujung kanan bawah (Hafsari & Isnani, 2021) dengan pergeseran sebesar x dan y . semakin kecil nilai x dan y maka semakin akurat deteksi citra tersebut. Nilai x dan y yang sering digunakan adalah 1.

Untuk menentukan ada tidaknya *Haar Feature* disetiap lokasi gambar digunakan teknik yang disebut *integral image*. Nilai dari integral pada masing-masing piksel adalah penjumlahan dari semua piksel diatasnya dan disebelah kirinya. Dimulai dari integral pada masing-masing piksel adalah jumlah dari semua piksel diatasnya dan disebelah kirinya. Dimulai dari kiri atas sampai kanan bawah, gambar bisa diintegrasikan sebagai operasi matematika perpiksel.

Adaboost Learning

Adaboost menggabungkan banyak *classifier* untuk membuat satu *classifier*. Masing-masing *classifier* menetapkan satu bobot dan gabungan dari bobot inilah yang akan membentuk satu *classifier* yang kuat. Viola dan jones menggabungkan serangkaian *adaboost classifier* sebagai rantai filter (Abidin, 2018). Dalam prakteknya tidak satupun fitur yang mampu melakukan pengklasifikasian dengan error yang kecil. Algoritma *adaboost* berfungsi untuk mencari fitur-

fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap terbesar antara wajah dan non-wajah.

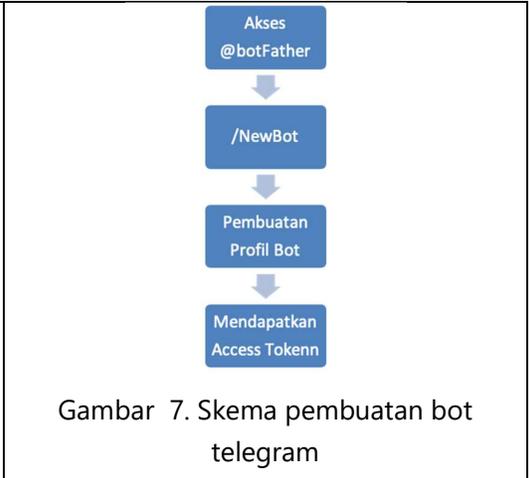
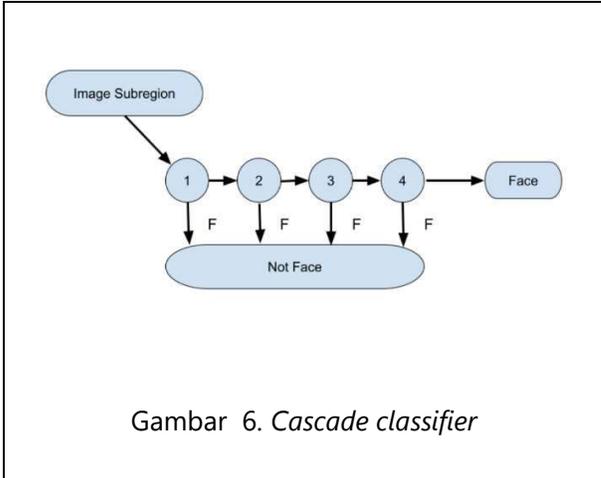
Cascade Classifier

Karakteristik dari haar cascade adalah adanya klasifikasi bertingkat (*cascade classifier*). Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra yang bukan wajah daripada menilai apakah tersebut berisi wajah.

Dari contoh potongan kode dibawah ini, dapat dijelaskan bahwa kumpulan data wajah sudah dimasukkan dalam file *xml* bernama *haarcascade_frontalface_default.xml*.

```
face = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

Pada gambar 6 dijelaskan bahwa setiap *subwindows* dibandingkan dengan setiap fitur di setiap *stage*. Jika tidak mencapai target maka *subwindows* akan bergerak ke *subwindows* berikutnya dan melakukan perhitungan yang sama dengan proses sebelumnya. Pada proses selanjutnya didapatkan hasil yaitu *subwindows* yang terdeteksi sebagai wajah dan berlanjut ke subcitra berikutnya sampai pada akhirnya didapatkan kandidat kuat yang terdeteksi sebagai wajah (Busran, 2022).



Pembuatan Bot Telegram

Pada gambar 7 dapat dijelaskan alur untuk membuat bot telegram yang dibantu dengan fasilitas yang telah disediakan oleh pihak telegram sendiri, dalam hal ini mengakses @botFather (Fahri, 2020). Setelah melakukan akses ke @botFather, selanjutnya mengetik perintah /newBot untuk membuat bot telegram baru. Bot akan membalas otomatis dengan memberikan perintah untuk mengisi nama dan *username* yang akan dibuat. Setelah

memberikan nama dan *username*, selanjutnya @botFather akan mengirimkan token akses bot yang telah dibuat.

Pembahasan

Simulasi Sistem Pendeteksian Ketika Tidak Memakai Masker



Gambar 8. Tampilan citra dari *webcam* ketika tidak memakai masker

Pada gambar 8 adalah tampilan ketika tidak memakai masker. Dimana pada gambar diatas di perlihatkan seseorang tidak menggunakan masker dan program tersebut mendeteksi seseorang tidak menggunakan masker sehingga program tersebut akan memotret *frame* dan mengirimnya melalui bot telegram dengan caption "Pegawai terdeteksi tanpa masker" seperti pada gambar 9.



Gambar 9. Tampilan pada bot telegram ketika tidak memakai masker

Simulasi sistem Pendeteksian Ketika Memakai Masker



(a)

(b)

Gambar 10 (a,b) Tampilan citra webcam dan bot telegram ketika memakai masker

Pada gambar 10 (a,b) merupakan tampilan ketika memakai masker. Dimana pada gambar diatas di perhatikan seseorang telah menggunakan masker dan program tersebut mendeteksi seseorang menggunakan masker dan program tersebut tidak akan memotret *frame* tersebut.

Simulasi Berdasarkan Jarak dan Sudut Wajah Terhadap Kamera

Pengujian ini dilakukan dengan dipengaruhi oleh 2 kondisi yaitu jarak dan sudut wajah terhadap kamera *webcam* dan diberikan pencahayaan dengan instensitas terang secara konstan, dengan variasi jarak dari 30 cm hingga lebih dari 120 cm. Pengujian dilakukan dengan 5 posisi sudut wajah terhadap kamera yaitu sebagai berikut:

1. Wajah tegak lurus dengan kamera
2. Wajah miring ke kanan 45°
3. Wajah miring ke kiri 45°
4. Wajah mendongak ke atas 15°
5. Wajah menunduk ke bawah 15°

Tabel 1. Simulasi Berdasarkan Jarak & Sudut

No.	Jarak	Sudut	Hasil	No.	Jarak	Sudut	Hasil
1.	30 cm	Tegak Lurus	✓	14.	120 cm	Miring Kiri 45°	✓
2.	60 cm	Tegak Lurus	✓	15.	>120 cm	Miring Kiri 45°	✗
3.	90 cm	Tegak Lurus	✓	16.	30 cm	Dongak 15°	✓
4.	120 cm	Tegak Lurus	✓	17.	60 cm	Dongak 15°	✓
5.	>120 cm	Tegak Lurus	✗	18.	90 cm	Dongak 15°	✓

6.	30 cm	Miring Kanan 45°	✓	19.	120 cm	Dongak 15°	✓
7.	60 cm	Miring Kanan 45°	✓	20.	>120 cm	Dongak 15°	✗
8.	90 cm	Miring Kanan 45°	✓	21.	30 cm	Nunduk 15°	✓
9.	120 cm	Miring Kanan 45°	✓	22.	60 cm	Nunduk 15°	✓
10.	>120 cm	Miring Kanan 45°	✗	23.	90 cm	Nunduk 15°	✓
11.	30 cm	Miring Kiri 45°	✓	24.	120 cm	Nunduk 15°	✓
12.	60 cm	Miring Kiri 45°	✓	25.	>120 cm	Nunduk 15°	✗
13.	90 cm	Miring Kiri 45°	✓				

Keterangan:

✓ = terdeteksi ; dan ✗ = tidak terdeteksi

Berdasarkan tabel simulasi 1, dapat disimpulkan sistem pendeteksi masker ini dapat bekerja secara optimum pada rentang jarak 30cm sampai dengan 120cm dengan diberikan pencahayaan dengan intensitas yang cukup terang.

Kesimpulan

Dalam simulasi sistem pendeteksi masker menggunakan metode HCC, beberapa temuan utama melibatkan integrasi sistem dengan bot Telegram yang telah digunakan pada skala terbatas di laboratorium komputer Teknik Elektro Universitas Mataram. Sistem berhasil berjalan dengan baik, mampu mengirimkan notifikasi peringatan melalui bot Telegram, dan menunjukkan kinerja optimal dalam kondisi pencahayaan ruangan yang memadai serta jarak antara webcam dan objek yang dideteksi berkisar antara 30 hingga 120 cm. Kesimpulan dari simulasi ini menegaskan keberhasilan implementasi sistem pendeteksi masker dengan metode HCC dalam lingkungan laboratorium, memberikan kontribusi positif terutama dalam upaya edukasi dan pengawasan disiplin penggunaan masker oleh mahasiswa.

Ucapan Terima Kasih

Diucapkan terimakasih kepada Laboratorium Komputer dan Jurusan Teknik Elektro Unram telah memberikan kesempatan agar dapat dilaksanakan simulasi secara terbatas sistem pendeteksi pengguna masker.

Daftar Pustaka

- Abidin, S. (2018). Deteksi wajah menggunakan metode Haar Cascade Classifier berbasis webcam pada Matlab. *Jurnal Teknologi Elekterika*, 15(1), 21–27.
- Anarki, G. A., Auliasari, K., & Orisa, M. (2021). Penerapan Metode Haar Cascade Pada Aplikasi Deteksi Masker. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(1), 179–186.
- Arifianto, T., & Sunardi, S. (2021). Penerapan Algoritma Viola-Jones Untuk Deteksi Masker Covid-19 Di Politeknik Perkeretaapian Indonesia Madiun. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(4), 2030–2040.
- Binus University. (2017). Introduction to Open CV. Retrieved from <https://binus.ac.id/malang/2017/10/introduction-to-open-cv/>
- Busran, B. (2022). PENDETEKSI PEMAKAIAN MASKER MENGGUNAKAN WEBCAM SEBAGAI ALTERNATIF PENCEGAHAN COVID-19 DI INSTITUT TEKNOLOGI PADANG. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 10(1), 29–34.
- Cokrojoyo, J. Andjarwirawan, & Noertjahyana, A. (2017). Pembuatan Bot Telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP. *Jurnal Infra*, 5(1), 224–227.
- Fahri, M. U. (2020). Implementasi Channel Bot Telegram (Real Time) COVID-19 di Kalimantan Barat dengan Memanfaatkan API. *Journal of Computer Science and Visual Communication Design*, 5(2), 77–84.
- Hafsari, R. T., & Isnani, S. R. (2021). RANCANG BANGUN ALAT PENDETEKSI WAJAH DAN PENDETEKSI SUHU TUBUH OTOMATIS GUNA MEMINIMALISIR PENYEBARAN COVID-19. Makassar.
- Prathivi, R., & Kurniawati, Y. (2020). Sistem Presensi Kelas Menggunakan Pengenalan Wajah Dengan Metode Haar Cascade Classifier. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 11(1), 135–142.
- Syarif, M., & Wijanarto, W. (2015). Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login Sistem. *Techno. com*, 14(4), 242–249.
- Telegram. (n.d.). Bot API. Retrieved from <https://core.telegram.org/bots/api>
- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 (Vol. 1, pp. I–I)*. IEEE.
- Warnilah, A. I., Sutisna, H., Jaya-Mulyana, A., Siti-Nuraeni, F., & Aninditya-Widianto, T. (2022). Program Aplikasi Pendeteksi Masker Dengan Menggunakan Algoritma Haarcascade. *EVOLUSI: Jurnal Sains dan Manajemen*, 10(1).

Yulina, S. (2021). Penerapan Haar Cascade Classifier dalam Mendeteksi Wajah dan Transformasi Citra Grayscale Menggunakan OpenCV. *Jurnal Komputer Terapan*, 7(1), 100–109.

Wirasto, G. W., Misbahuddin, M., A, L. S. I., Iqbal, M. S., Rachman, A., & Budiman, D. F. (2023). SIMULASI MULTIPENGGUNA SISTEM AKUNTANSI AKSYAA.COM BERBASIS WEB. *Prosiding PEPADU*, 5(1), 362–367