
Sistem Absensi Berbasis Pengenalan Wajah

Andrew Febrian Miyata¹, Hartono Pranjoto^{2*}

^{1,2}Program Studi Teknik Elektro, Universitas Katolik Widya Mandala Surabaya, Jawa Timur
Email: ¹andrew.m@ukwms.ac.id, ^{2*}pranjoto@ukwms.ac.id

(Naskah masuk: 25 Okt 2021, direvisi: 25 Jan 2022, diterima: 7 Feb 2022)

Abstrak

Virus COVID-19 menyebar melalui *droplet* dari mulut dan hidung yang dapat menyebar melalui sentuhan. Sistem absensi yang biasa digunakan berbasis sentuhan seperti sidik jari maupun *Radio-frequency identification* (RFID). Untuk mengurangi penyebaran virus COVID-19 diperlukan absensi yang tidak berbasis sentuhan, karena itu dipilih sistem absensi menggunakan wajah. Untuk dapat mengenali wajah dibutuhkan metode guna mendeteksi keberadaan wajah, kemudian mendapatkan data *orthogonal* dari wajah. Data *orthogonal* dari 2 foto orang yang sama akan memiliki kesamaan 60-80%, sedangkan orang yang berbeda akan dibawah 60%. Program mempunyai akurasi 95% saat melihat orang yang terdaftar pada *database* namun akurasi berkurang saat melihat orang yang tidak terdapat pada *database* menjadi sebesar 80% saat dilakukan uji coba menggunakan 20 orang yang tidak terdaftar. Lebih susah bagi sistem untuk mengenali bahwa orang tersebut tidak terdaftar karena apabila batas bawah kecocokan ditingkatkan akan berpengaruh dengan kemampuan sistem mendeteksi wajah orang yang terdaftar. Untuk meningkatkan akurasi dapat dilakukan dengan cara memiliki *database* foto resolusi tinggi, pencahayaan mencerminkan ruangan yang digunakan untuk absensi, dan menggunakan gambar terbaru.

Kata Kunci: Kecerdasan Buatan, Pengenalan Wajah, Absensi, Visualisasi Citra Digital, *Euclidean Distance*.

Presence System Based on Face Recognition

Abstract

The COVID-19 virus can spread via droplet from mouth and nose. The most common attendance systems are touch-based because they read the fingerprints or Radio-frequency identification (RFID). One method of reducing the Covid-19 virus infection uses a piece of non-touch-attendance equipment, in this case, is the reading of the face. First, the system will read all of the pictures (pictures from files) of the attendance system then create orthogonal data. Comparison results of two orthogonal data with 60 – 80% similarity indicate that those pictures are from the same person. This method has 95% accuracy when comparing photographs of registered people, but then the accuracy decreases to 80% when comparing pictures, not in the database with 20 different people. Face detection for a person not in the database is lower since it can detect the closest orthogonal match. Higher resolution photographs can increase the comparison accuracy together with proper lighting. Registering a person in the database with the laster picture will also increase the accuracy.

Keywords: Artificial Intelligence, Face Recognition, Presence, Computer Vision, *Euclidean Distance*.

I. PENDAHULUAN

Pada bulan Desember tahun 2019 virus COVID-19 ditemukan di Wuhan, China. Setelah itu virus COVID-19 terus menyebar sampai ke seluruh dunia [1], [2]. Virus COVID-19 menyebar melalui *droplet* dari mulut atau hidung saat seseorang batuk, bersin, berbicara, maupun bernafas [3]. Seseorang juga dapat terinfeksi saat menyentuh permukaan

yang sudah terkontaminasi oleh virus kemudian menyentuh mata, hidung, atau mulut tanpa mencuci tangan terlebih dahulu [1].

Sebagian besar sistem absensi yang umum dilakukan sebelum masa pandemi berbasis sentuhan menggunakan sidik jari maupun kartu RFID [4]–[6]. Ada beberapa sistem absensi yang tidak berbasis sentuhan menggunakan *captive portal* atau aplikasi di ponsel pintar namun sistem ini mempunyai

kelemahan yaitu sangat mudah apabila ingin melakukan kecurangan contohnya titip absen.

Pada tahun 2022 kegiatan kantor dan pembelajaran akan memasuki era kenormalan baru [7]. Oleh karena itu diperlukan sistem absensi yang tidak berbasis sentuhan namun juga aman dari aksi kecurangan. Metode yang disarankan dalam tulisan ini merupakan sistem absensi berbasis wajah. Metode ini dipilih karena mendeteksi wajah tidak memerlukan sentuhan dan sistem masih memiliki harga yang terjangkau apabila dibandingkan dengan sistem berbasis mata atau iris. Kelemahan dari sistem berbasis pengenalan wajah adalah untuk mendapat hasil yang baik pencahayaan ruangan harus baik dan sudut pengambilan gambar oleh kamera harus sejajar dengan wajah. Selain itu apabila terdapat penutup pada wajah seperti masker akan membuat wajah tidak terdeteksi.

II. METODE

A. Perangkat Keras

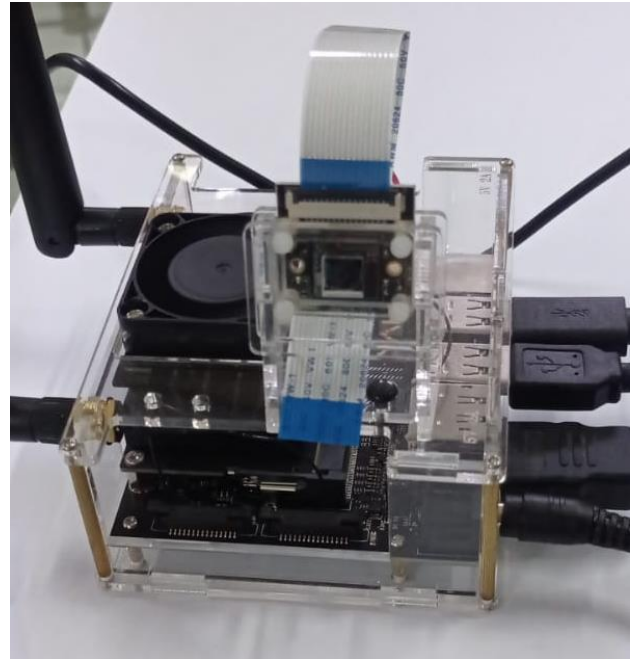
Untuk dapat melakukan absensi berbasis pengenalan wajah diperlukan kamera untuk mengambil gambar dan juga perangkat untuk memproses gambar yang sudah diambil. Terdapat beberapa kamera yang dapat digunakan oleh *Nvidia Jetson Nano*. Pertama dilihat dari antarmuka terdapat 2 macam *Camera Serial Interface* (CSI) dan *Universal Serial Bus* (USB). Kamera yang digunakan merupakan kamera USB. Kamera USB dipilih karena memiliki kabel yang relatif panjang dan juga memiliki variasi yang lebih banyak sehingga mudah untuk dilakukan penyesuaian dengan kebutuhan apakah ingin mendapatkan harga yang murah atau gambar dengan resolusi tinggi. Beberapa kamera USB juga memiliki keunggulan lainnya seperti pencahayaan bawaan seperti yang dapat dilihat pada Gambar 1 dan juga mudah dipasang.



Gambar 1. Kamera Dengan Pencahayaan Bawaan

Untuk memproses gambar digunakan *Nvidia Jetson Nano*. *Nvidia Jetson Nano* dipilih karena memiliki 128-core *Maxwell* GPU sehingga mampu memproses gambar dengan cepat, memiliki keluaran video *High-Definition Multimedia Interface* (HDMI) sekaligus *Embedded Display Port* (eDP) rendah daya,

ukuran yang relatif kecil seperti yang dapat dilihat pada Gambar 2, dan juga memiliki harga yang relatif terjangkau. Memori yang dimiliki juga relatif besar untuk *Single Board Computer* (SBC) yaitu sebesar 4 GB [8]. *Jetson Nano* juga sudah memiliki koneksi *Wi-Fi* sehingga apabila perangkat ingin dipindah, pemindahan dapat dengan mudah dilakukan.



Gambar 2. Nvidia Jetson Nano

Nvidia Jetson Nano juga memiliki fitur *remote desktop* sehingga apabila perlu melakukan penyesuaian seperti menambah wajah pegawai baru dapat dilakukan dari jarak jauh. Untuk mengendalikan *Nvidia Jetson Nano* dari jarak jauh dapat dilakukan menggunakan XRDP. XRDP merupakan versi *open-source* dari *Microsoft Remote Desktop Protocol* (RDP). Sebelum dapat menggunakan XRDP perlu mengatur *Nvidia Jetson Nano* untuk mengizinkan pengguna lain melihat tampilan dan mengontrol perangkat.

B. Deteksi Wajah

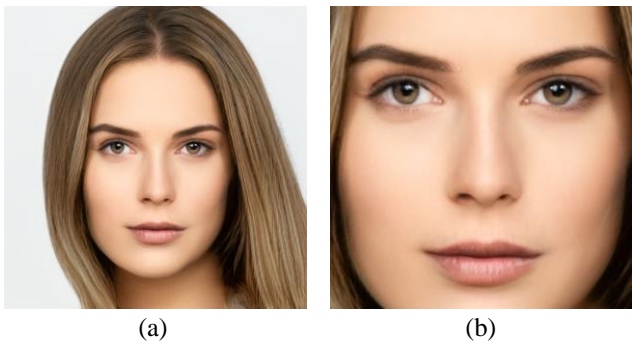
Sebelum mengenali wajah perlu diketahui terlebih dahulu apakah pada gambar tersebut terdapat wajah. Apabila tidak terdapat wajah dan gambar tersebut tetap dianalisis tentu akan menghabiskan sumber daya pada komputer. Selain itu dengan menganalisis wajah terlebih dahulu maka sistem dapat mendeteksi lebih dari satu wajah sekaligus.

Wajah dideteksi menggunakan deteksi mata, mulut, dan hidung. Awalnya program akan mencari mata terlebih dahulu karena mata merupakan objek yang paling mudah untuk dideteksi. Setelah mata terdeteksi maka program akan mencari mulut dan hidung kemudian wajah akan terdeteksi [9]–[11].

Deteksi wajah menggunakan kamera memiliki beberapa kelemahan. Untuk dapat mendeteksi wajah dengan baik diperlukan pencahayaan yang cukup, karena apabila gambar terlalu gelap program akan kesusahan mencari fitur pada wajah seperti mata. Wajah juga tidak boleh tertutup oleh objek

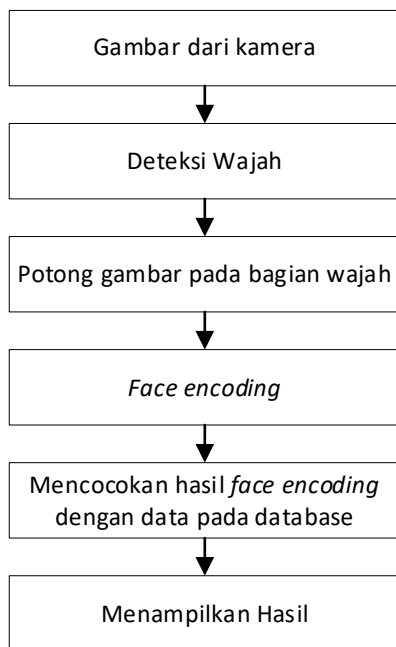
contohnya masker, karena apabila wajah tertutup oleh masker program tidak dapat mendeteksi hidung dan mulut sehingga wajah tidak terdeteksi. Orientasi wajah juga tidak boleh terputar, seperti orientasi wajah terputar atau miring, karena program diatur untuk dapat mendeteksi wajah pada orientasi lurus [12].

Setelah wajah terdeteksi maka bagian wajah akan dipotong seperti pada Gambar 3 untuk masing-masing wajah yang terdeteksi. Program dapat mendeteksi lebih dari 1 wajah pada gambar, program akan menyimpan hasil potongan gambar dalam *array* untuk masing-masing gambar dan tiap gambar akan dianalisis secara independen [13].



Gambar 3. Wajah Pada Saat: (a) Sebelum Dipotong, dan (b) Sesudah Dipotong

C. Pengenalan Wajah



Gambar 4. Proses Pengenalan Wajah

Proses pengenalan wajah dapat dilihat pada Gambar 4. Untuk dapat mengenali wajah diperlukan *database* nama dan wajah dari orang yang sudah dikenali. Idealnya pada *database*, satu subjek memiliki beberapa gambar dengan ekspresi dan sudut pengambilan yang bervariasi untuk meningkatkan

akurasi. Pada penelitian ini hanya digunakan satu gambar dari masing-masing subjek.

Saat program dijalankan maka program akan menghitung *face encoding* untuk tiap wajah pada *database*. *Face encoding* adalah representasi wajah yang melihat faktor penting pada gambar seperti warna dan ukuran mata, alis, jarak antara alis, bentuk hidung, bentuk bibir, dan lain sebagainya. Setiap gambar memiliki *array* yang menyimpan 128 variabel *float* yang setiap nilainya merepresentasikan komponen *ortogonal* dari wajah seperti yang dapat dilihat pada Tabel 1. Dua gambar berbeda dari orang yang sama akan memiliki *encoding* yang hampir sama, namun orang yang berbeda akan memiliki *encoding* yang berbeda.

Tabel 1. Hasil Face Encoding

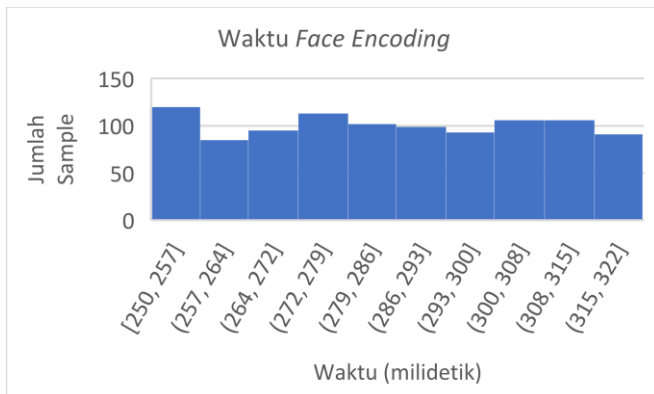
Index	Data
0	-0.11571178
1	0.09954529
2	0.07217095
3	-0.08318465
...	...
127	-0.0446847

Face encoding menyimpan data bagian penting wajah seperti mata, alis, bibir, hidung, dan dagu. *Face encoding* juga bisa merekonstruksi wajah menggunakan data kurva atau bentuk yang didapat dari *landmark* yang sudah didapat. Contoh dari *landmark* wajah dapat dilihat pada Tabel 2.

Tabel 2. Landmark Wajah

Landmark	Kordinat
<i>chin</i>	[(152, 84), (153, 92), (154, 101), (156, 109), (159, 116), (164, 122), (170, 128), (176, 131), (184, 132), (193, 130), (200, 126), (206, 121), (211, 114), (215, 107), (216, 98), (216, 89), (216, 81)]
<i>left_eyebrow</i>	[(155, 76), (158, 71), (164, 69), (170, 70), (176, 72)]
<i>right_eyebrow</i>	[(184, 71), (191, 68), (197, 67), (203, 68), (208, 72)]
<i>nose_bridge</i>	[(181, 80), (181, 84), (180, 88), (180, 93)]
<i>nose_tip</i>	[(175, 100), (178, 101), (181, 101), (185, 100), (189, 100)]
<i>left_eye</i>	[(161, 83), (164, 80), (169, 80), (172, 83), (169, 84), (164, 84)]
<i>right_eye</i>	[(190, 81), (194, 78), (198, 77), (202, 79), (199, 81), (194, 82)]
<i>top_lip</i>	[(171, 113), (175, 109), (179, 106), (182, 107), (185, 106), (190, 108), (194, 111), (192, 111), (186, 111), (182, 111), (179, 111), (173, 113)]
<i>bottom_lip</i>	[(194, 111), (190, 115), (186, 117), (182, 117), (179, 117), (175, 116), (171, 113), (173, 113), (179, 111), (182, 111), (186, 111), (192, 111)]

Proses *encoding* 1000 gambar dengan resolusi 140 x 200 *pixel* membutuhkan waktu sekitar 300 detik dengan rata-rata waktu yang dibutuhkan 0,3 detik per gambar seperti yang dapat dilihat pada Gambar 5. Hasil *encoding* dari gambar akan disimpan dalam bentuk *Pickle* agar saat program mau dijalankan ulang tidak perlu dilakukan *encoding* lagi.



Gambar 5. Waktu Face Encoding

Setelah memotong gambar dan mendapatkan hanya area wajah, selanjutnya dilakukan pencocokan wajah pada gambar dengan wajah pada *database*. Hasil *face encoding* dari tiap wajah yang terdeteksi akan dibandingkan dengan gambar pada *database* menggunakan *Euclidean distance*.

D. Euclidean Distance

Euclidean distance merupakan metode yang dapat digunakan untuk menghitung jarak terdekat antara 2 titik. Biasanya *Euclidean distance* digunakan ketika menghitung jarak antara 2 baris data yang memiliki nilai numerik seperti pada Tabel 1 [14], [15].

Euclidean distance antara 2 titik dapat dihitung dengan persamaan:

$$Ed = \sqrt{\sum_{i=0}^n ((x_2 - x_1)^2 + (y_2 - y_1)^2)} \quad (1)$$

$$Ed = \sum_{i=0}^n ((x_2 - x_1)^2 + (y_2 - y_1)^2) \quad (2)$$

Untuk mengurangi waktu proses penghitungan pada data dengan jumlah besar, umumnya akar dihilangkan dari persamaan 1 dan menjadi persamaan seperti yang dapat dilihat pada persamaan 2. Hasil perhitungan dengan atau tanpa akar memiliki hasil yang relatif proporsional, sehingga dapat digunakan secara efektif oleh *machine learning*. Karena data yang digunakan hanya sebesar 128 maka dipilih persamaan dengan akar [14].

Setelah mendapat data *landmark* wajah seperti yang didapat dari Tabel 2. Saat 2 wajah dari orang yang sama namun diambil dari sudut dan waktu yang berbeda maka akan dilihat bentuk mata kiri orang tersebut dan dibandingkan antara gambar pertama dan kedua. Menggunakan *Euclidean distance* akan dihasilkan nilai yang hampir sama karena bentuknya

sama. Hal ini akan berbeda apabila membandingkan orang yang berbeda karena bentuk mata orang tersebut tidak sama seperti gambar referensi.

E. Sistem Absensi

Daftar pegawai yang terdaftar terdiri dari *face encoding*, nomor pegawai, nama pegawai, jam masuk, dan jam pulang. Saat pegawai berdiri di depan kamera maka sistem akan mencocokkan orang tersebut dengan *face encoding* pada daftar pegawai [16], [17]. Apabila wajah tersebut ditemukan pada daftar maka akan muncul nama pegawai dan juga status “absen masuk” seperti yang dapat dilihat pada Gambar 6 atau “absen pulang” seperti yang dapat dilihat pada Gambar 7 pada tampilan.

Penentuan apakah pegawai masuk atau keluar ditentukan dari jadwal kerja pegawai tersebut disertai toleransi 1 jam. Apabila jadwal masuk pegawai jam 08.00 maka pegawai tersebut diberi rentang waktu dari 07.00 sampai dengan 09.00 untuk absen masuk, begitu pula untuk absen pulang.



Gambar 6. Contoh Tampilan Absen Masuk



Gambar 7. Contoh Tampilan Absen Pulang

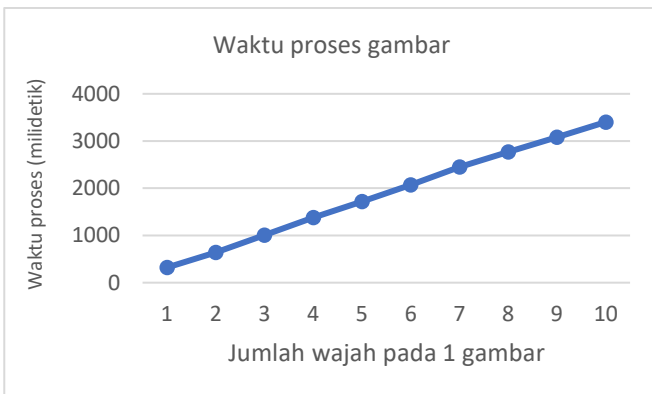
III. HASIL DAN PEMBAHASAN

A. Analisis Waktu Proses

Saat program membuat daftar pegawai, program perlu melakukan *face encoding* pada semua gambar pegawai. Waktu

proses untuk gambar dengan resolusi 140 x 200 *pixel* sekitar 0,3 detik per gambar. Dengan daftar pegawai sebanyak 1000 foto, setiap program dijalankan maka akan membutuhkan waktu sekitar 300 detik. Untuk menghemat waktu maka data *face encoding* disimpan dalam bentuk *Pickle* sehingga saat program dijalankan ulang tidak perlu melakukan ulang *face encoding* untuk daftar pegawai.

Setelah program berjalan maka program akan mencari keberadaan wajah pada gambar tangkapan kamera. Apabila terdapat 1 wajah pada hasil tangkapan kamera maka program akan melakukan *face encoding* pada wajah. Setelah mendapatkan nilai *face encoding* maka program akan melakukan *euclidean distance* antara wajah yang baru dianalisis dan wajah pada daftar. Waktu yang dibutuhkan apabila terdeteksi satu wajah pada hasil tangkapan kamera maka program membutuhkan waktu sekitar 0,4 detik. Namun apabila ditemukan 2 wajah, waktu yang dibutuhkan sekitar 0,8 detik. Saat dilakukan uji coba hingga 10 wajah pada 1 gambar kenaikan waktu linear seperti yang dapat dilihat pada Gambar 8.



Gambar 8. Waktu Yang Dibutuhkan Untuk Memproses Gambar

B. Analisis Pengaruh Cahaya ke Akurasi

Tabel 3. Hasil *Face Encoding*

No	Intensitas Cahaya (lux)	Tingkat Kecocokan (%)
1	200	67
2	358	65
3	386	65
4	391	63
5	500	62

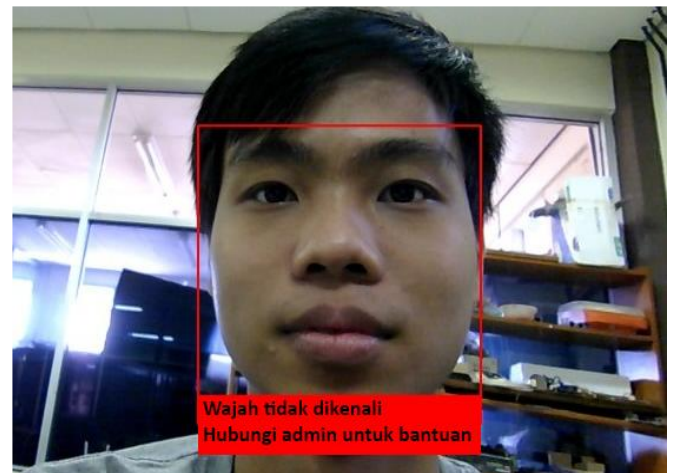
Kondisi pencahayaan dapat mempengaruhi tingkat kecocokan antara gambar referensi dan gambar yang ditangkap oleh kamera. Tingkat kecocokan paling tinggi saat ruangan lebih gelap karena gambar referensi diambil dengan ruangan dengan pencahayaan redup. Makin sesuai pencahayaan dibandingkan dengan kondisi saat gambar diambil akan membuat akurasi sistem makin baik. Meskipun tingkat kecocokan berkurang, namun orang tersebut masih

dikenali sebagai orang yang sama sehingga percobaan berhasil. Karena implementasi digunakan pada ruangan absensi yang pasti memiliki sumber cahaya, maka sistem tidak diuji coba saat ruangan tidak ada sumber cahaya.

C. Tingkat Kecocokan

Setelah program mendeteksi wajah maka wajah yang terdeteksi akan dicocokkan dengan daftar pada daftar pegawai. Wajah dari orang yang sama saat dicocokkan dengan *database* akan memiliki tingkat kecocokan sekitar 60-70% tergantung pada kondisi pencahayaan, sudut pengambilan gambar, dan waktu antara gambar diambil dengan waktu saat ini. Idealnya gambar untuk referensi memiliki tingkat pencahayaan sekaligus sudut pengambilan gambar yang sama dengan saat implementasi. Untuk perempuan pemakaian kosmetik juga akan mempengaruhi tingkat akurasi, sedangkan untuk laki-laki kumis dan janggut akan mempengaruhi.

Saat dilakukan pengujian dengan cara mengarahkan kamera pada layar monitor yang menampilkan gambar orang yang terdaftar pada *database*. Tingkat kecocokan yang terbaca 80-85% namun pada saat gambar yang sama diproses oleh program tingkat akurasi mendekati 100%. Hal ini dikarenakan pada saat kamera menangkap gambar pada monitor, tingkat cahaya terdapat perbedaan yang menyebabkan tingkat kecocokan menurun.



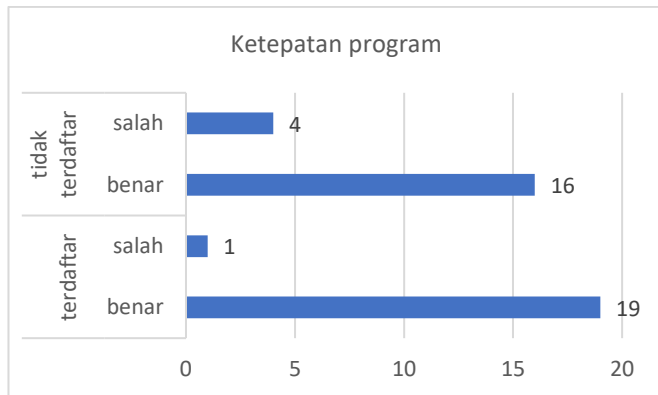
Gambar 9. Tampilan Saat Wajah Tidak Dikenali

Untuk menghindari kesalahan pembacaan maka perlu diberi batas bawah tingkat kecocokan. Jika batas tingkat kecocokan terlalu rendah, maka apabila ada orang yang tidak terdaftar berada di depan kamera, program akan mencocokkan dengan gambar yang terdekat. Apabila batas kecocokan terlalu tinggi maka saat ada perubahan pada orang tersebut seperti kosmetik maka orang tersebut akan dianggap tidak terdaftar. Untuk itu diambil tingkat kecocokan minimum 60% untuk menganggap orang tersebut terdaftar seperti yang terlihat pada Gambar 9.

D. Pengujian Dengan Subjek

Untuk memastikan performa sistem maka dilakukan pengujian dengan subjek. Dilakukan pengujian dengan 20 subjek yang terdaftar dan 20 subjek dengan kondisi tidak

terdaftar seperti yang terlihat pada Gambar 10. Hasil dari pengujian terdapat 1 orang terdaftar yang pada saat pengujian tidak terbaca hal tersebut dikarenakan foto yang digunakan sebagai referensi sudah 2 tahun lalu sehingga terdapat perbedaan pada wajah yang menyebabkan tingkat kemiripan di bawah 60%. Untuk mengatasi hal ini maka gambar pada gambar perlu di perbarui dengan interval lebih cepat yaitu setiap 1 tahun.



Gambar 10. Ketepatan Program Saat Mengenali Wajah

Saat dilakukan percobaan dengan cara menghapus data 20 subjek tersebut dari daftar, 4 orang terdeteksi sebagai orang terdaftar. Hal ini dikarenakan tingkat kecocokan sebesar 60% terhitung cukup rendah. Untuk mengatasi hal tersebut maka batas bawah dapat ditingkatkan ke 65%, namun hal ini dapat mengurangi tingkat kesuksesan saat orang yang terdaftar mencoba untuk melakukan absen.

Makin banyak jumlah wajah pada *database* maka makin tinggi kemungkinan bahwa wajah yang tidak terdaftar memiliki kesamaan dengan wajah yang tersimpan pada *database*. Oleh karena itu makin banyak wajah yang tersimpan menyebabkan akurasi program saat mengenali wajah yang tidak tersimpan di *database* menurun.

Karena lebih susah untuk mendeteksi bahwa seseorang yang tidak ada di daftar, terutama jika terdapat banyak gambar pada *database*, yang memungkinkan seseorang yang tidak terdaftar akan cocok dengan wajah di dalam *database* lebih tinggi. Maka sebaiknya absensi diletakkan pada ruangan yang hanya boleh dimasuki oleh staf, dengan demikian wajah yang akan dicocokkan sudah ada pada *database* dan akan memudahkan untuk menentukan wajah siapa.

IV. KESIMPULAN

Dengan menggunakan absensi berbasis pengenalan wajah maka dapat dilakukan absensi tanpa melakukan sentuhan sehingga dapat mengurangi penyebaran virus Covid-19. Sistem dapat mengenali seseorang dalam waktu 0,5 detik sehingga tidak membuat orang merasa proses absensi memakan banyak waktu.

Tingkat kecocokan pada saat wajah seseorang berada pada depan kamera dan dibandingkan dengan gambar pada *database* bervariasi di antara 60-80% sehingga apabila

kecocokan seseorang di bawah 60% dianggap tidak terdaftar. Semakin banyak jumlah sampel pada *database* akan meningkatkan kemungkinan seseorang terdeteksi terdaftar meskipun sebenarnya tidak terdaftar.

Referensi yang baik pada *database* merupakan gambar yang memiliki pencahayaan sama dengan kondisi ruangan sistem akan digunakan. Gambar juga sebaiknya diambil dalam 1 tahun terakhir sehingga perubahan pada wajah tidak terlalu signifikan, untuk mengamati efektifitasnya akan diamati perubahan akurasi dalam waktu 1 tahun kedepan.

REFERENSI

- [1] CDC, "Basics of COVID-19," 2021. <https://www.cdc.gov/coronavirus/2019-ncov/your-health/about-covid-19/basics-covid-19.html>.
- [2] T. P. Velavan and C. G. Meyer, "The COVID-19 epidemic," *Trop. Med. Int. Heal.*, vol. 25, no. 3, pp. 278–280, Mar. 2020, doi: 10.1111/tmi.13383.
- [3] L. Yang *et al.*, "COVID-19: immunopathogenesis and Immunotherapeutics," *Signal Transduct. Target. Ther.*, vol. 5, no. 1, p. 128, Dec. 2020, doi: 10.1038/s41392-020-00243-2.
- [4] M. Arironang, I. D. Hutahaean, H. Sipayung, and I. H. Tambunan, "Implementation of Fingerprint Recognition Using Convolutional Neural Network and RFID Authentication Protocol on Attendance Machine," *ACM Int. Conf. Proceeding Ser.*, pp. 151–156, 2020, doi: 10.1145/3397391.3397449.
- [5] A. M. Bangun, "Implementasi Kebijakan Sistem Absensi Elektronik Sidik Jari (Fingerprint) dalam Meningkatkan Kedisiplinan Pegawai pada Dinas Pekerjaan Umum dan Penataan Ruang Kabupaten Karo," 2020.
- [6] Md. M. Rahman, T. I. MishuI, Md. S. Islam, and Md. S. Akanda, "Implement Fingerprint Authentication for Employee Automation System," *Int. J. Innov. Res. Inf. Secur. Issue*, vol. 4, no. 9, 2017, doi: 10.26562/IJIRIS.2017.SPIS10080.
- [7] H. Limanseto, "Pemerintah Siapkan New Normal, Herd Immunity Jadi Syarat Utama," *Kementerian Koordinator Bidang Perekonomian*, 2021. <https://ekon.go.id/publikasi/detail/3351/pemerintah-siapkan-new-normal-herd-immunity-jadi-syarat-utama>.
- [8] "NVIDIA Jetson Nano Developer Kit Specification," 2020.
- [9] A. Priadana and M. Habibi, "Face Detection using Haar Cascades to Filter Selfie Face Image on Instagram," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIIT)*, Mar. 2019, pp. 6–9, doi: 10.1109/ICAIIIT.2019.8834526.
- [10] T. Mantoro, M. A. Ayu, and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," in *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, May 2018, pp. 1–5, doi: 10.1109/ICMCS.2018.8525935.

-
- [11] E. Rudinskaya and R. Paringer, "Face Detection Accuracy Study Based on Race and Gender Factor Using Haar Cascades," *CEUR Workshop Proc.*, vol. 2667, pp. 238–242, 2020.
- [12] S. Dev and T. Patnaik, "Student Attendance System using Face Recognition," *Proc. - Int. Conf. Smart Electron. Commun. ICOSEC 2020*, no. Icosec, pp. 90–96, 2020, doi: 10.1109/ICOSEC49089.2020.9215441.
- [13] R. Agarwal, R. Jain, R. Regunathan, and C. S. Pavan Kumar, "Automatic Attendance System Using Face Recognition Technique," *Adv. Intell. Syst. Comput.*, vol. 828, pp. 525–533, 2019, doi: 10.1007/978-981-13-1610-4_53.
- [14] R. Ahdid, S. Safi, and B. Manaut, "Euclidean and Geodesic Distance Between a Facial Feature Points in Two-dimensional Face Recognition System," *Int. Arab J. Inf. Technol.*, vol. 14, no. 4A Special Issue, pp. 565–571, 2017.
- [15] M. Y. Florestiyanto, A. H. Pratomo, and N. I. Sari, "Penguatan Ketepatan Pengenalan Wajah Viola-Jones Dengan Pelacakan," *Teknika*, vol. 9, no. 1, pp. 31–37, 2020, doi: 10.34148/teknika.v9i1.241.
- [16] D. Prangchumpol, "Face Recognition for Attendance Management System Using Multiple Sensors," *J. Phys. Conf. Ser.*, vol. 1335, no. 1, 2019, doi: 10.1088/1742-6596/1335/1/012011.
- [17] A. J. E. Cañete, "OpenCV Real-time Face Recognition Attendance System to Online-School Attendances," vol. 6, no. 5, pp. 725–728, 2021.