

Pengenalan Wajah Dengan *Face-Api.js* Berbasis CNN dan Geolokasi Menggunakan *Equirectangular Approximation*

Bimo Akbar Fadli^{1*}, Edy Winarno²

Teknologi Informasi dan Industri, Universitas Stikubank, Semarang, Indonesia

*e-mail *Corresponding Author*: bimoakbarfadli@mhs.unisbank.ac.id

Abstract

Computer vision usag on an attendance system role is to make sure that the attendance cannot be done by another person. Besides that, the attendance system could also use geolocation to determine the attendee location and the supposed attendance location. These process takes a considerable amount of server resource. To reduce the load of a server, a client side process could be used. Face recognition is done by using face-api.js which is a javascript module build on top of tensorflow.js core that implements some CNN (Convolutional Neural Networks) to detect, recognize, and extract feature of a face, that is optimized for website and mobile device. For the calculation of geolocation coordinate an equirectangular approximation formula is used. The accuracy of face recognition ranges between 93% to 97% and the geolocation calculation takes around 0.008300000003ms to complete.

Key words: *Face Detection; Geolocation; Tensorflow; Attendance*

Abstrak

Penggunaan computer vision pada sistem absensi berperan sebagai pengaman untuk memastikan bahwa absensi tidak dapat diwakilkan oleh orang lain. Selain itu pada sistem absensi juga dapat ditambahkan sistem geolokasi untuk menentukan jarak pelaku dari jarak lokasi yang sudah ditentukan. Proses-proses tersebut memakan sumber daya besar pada server. Dengan mendelegasikan proses tersebut kepada klien, beban serer akan berkurang secara signifikan. Pengenalan wajah dilakukan dengan menerapkan face-api.js yang merupakan sebuah modul javascript yang dibangun diatas tensorflow.js core, yang mengimplementasikan beberapa CNN (*Convolutional Neural Networks*) untuk melakukan deteksi, pengenalan, dan deteksi fitur-fitur wajah, yang dioptimalkan untuk web dan perangkat mobile. Untuk geolokasi pengukuran jarak koordinat, digunakan formula *equirectangular approximation*. Dari penerapan pengenalan wajah dan geolokasi tersebut didapatkan akurasi pengenalan wajah berkisar antara 93% hingga 97% dan geolokasi menggunakan *equirectangular approximation* berjalan dalam waktu rata-rata 0.008300000003ms.

Kata kunci: *Pengenalan wajah; Geolokasi; Tensorflow; Presensi*

1. Pendahuluan

Computer Vision merupakan salah satu bidang kecerdasan buatan yang semakin marak seiring berkembangnya teknologi informasi di masa modern ini. Definisi dari *Computer Vision* itu sendiri adalah kemampuan sebuah komputer untuk melihat dengan sudut pandang yang sama dengan manusia [1]. Semakin dekat sebuah komputer dengan sudut pandang manusia maka dapat dikatakan semakin mutakhir *Computer Vision* yang diterapkan [2]. Dikarenakan pengaplikasiannya yang beragam dan manfaat-manfaat yang dihasilkan sangat luas [3], *Computer Vision* menjadi salah satu bidang dari teknologi informasi yang sangat sering digunakan untuk membantu dan mempermudah memecahkan masalah-masalah yang ada saat ini [4].

Salah satu pengaplikasian *Computer Vision* adalah pembuatan sistem pengenalan wajah [5], dimana suatu citra digital wajah diproses guna pengenalan atau pencocokan dengan citra digital wajah sumber yang digunakan sebagai pembanding [6]. Pengenalan wajah

merupakan salah satu metode identifikasi biometrik yang paling penting, dan semakin menarik perhatian dikarenakan banyaknya penggunaan pengenalan wajah di berbagai bidang [7].

Suatu sistem pengenalan wajah dapat digunakan untuk membantu pada Pengelolaan Sumber Daya Manusia (SDM) lebih tepatnya untuk pengolahan data absensi menggunakan pengenalan wajah [9]. Dengan menggunakan sistem pengenalan wajah, masalah titip absen akan berkurang secara drastis [10], dikarenakan SDM yang akan melakukan absensi diharuskan menyamakan wajahnya dengan wajah pembanding yang sudah tersimpan di sistem [11]. Selain itu diperlukan juga adanya penanggulangan kecurangan absensi diluar lokasi yang sudah ditentukan. Untuk menghalangi kecurangan absensi diluar lokasi dapat dilakukan perhitungan jarak antara dua titik koordinat, yaitu koordinat lokasi perangkat yang digunakan untuk melakukan absensi dan koordinat yang sudah ditentukan pada sistem sebagai titik absensi.

Tentunya proses pengenalan wajah dan perhitungan jarak memakan sumber daya server yang cukup signifikan. Terlebih ketika dalam satu server terjadi ratusan bahkan ribuan data absensi yang diproses tiap harinya. Dengan mendelegasikan tugas pendeteksian wajah dan perhitungan jarak koordinat kepada sisi klien, beban yang ditanggung kepada server akan berkurang secara signifikan. Untuk itu diperlukan suatu sistem proses ringan yang mampu berjalan dari sisi klien tanpa membebani server.

Dalam proses pengenalan wajah akan digunakan face-api.js yang merupakan sebuah modul javascript yang dibangun diatas tensorflow.js core, yang mengimplementasikan beberapa CNN (*Convolutional Neural Networks*) untuk melakukan deteksi wajah, pengenalan wajah, dan deteksi fitur-fitur wajah, yang dioptimalkan untuk web dan perangkat mobile [8]. Dengan begitu, face-api.js dapat digunakan melalui semua perangkat yang memiliki peramban web dan dapat mengakses website, dengan kompatibilitas yang luas tanpa memerlukan sumber daya yang besar. Sedangkan untuk perhitungan jarak koordinat akan digunakan rumus *equirectangular approximation* [12] dengan mengaplikasikan teori pythagoras kedalam equirectangular projection dari dua titik koordinat yaitu titik koordinat perangkat yang digunakan untuk absensi dan titik koordinat lokasi absen seharusnya. Berbeda dengan metode haversine yang menggunakan tujuh fungsi trigonometri, equirectangular approximation hanya menggunakan satu fungsi trigonometri membuat komputasi metode ini lebih ringan dengan kesalahan akurasi yang kecil [13][14], membuat fungsi ini cocok digunakan dari sisi klien.

Di dalam penelitian ini akan dibahas upaya meminimalisir kecurangan yang dilakukan pada kegiatan absensi dengan mengimplementasikan pengenalan wajah untuk mengatasi permasalahan penitipan absen dan pengukuran jarak koordinat untuk mengatasi permasalahan kegiatan absen di luar area lokasi yang ditentukan, dimana kedua proses tersebut dilakukan pada sisi klien untuk mengurangi beban pada server.

2. Tinjauan Pustaka

Penelitian yang dilakukan oleh Yadav et al, (2021), membahas mengenai isu-isu yang ada pada metode absensi yang dilakukan dalam dunia pendidikan, terutama pada sekolah dan universitas. Metode-metode yang dibahas adalah Google Forms, Melalui chatbox, dan secara verbal (manual). Sedangkan isu-isu dari masing-masing metode tersebut adalah siswa dapat mengabsenkan siswa lain menggunakan id email siswa yang bersangkutan pada *Google Form*, guru harus melakukan rekap apabila absensi dilakukan melalui chatbox, dan absensi secara verbal sangat tidak efisien dikarenakan memakan waktu cukup lama. Oleh karena itu penelitian ini ditujukan untuk mengatasi isu-isu yang telah disebutkan dengan menggunakan pengenalan wajah[1].

Penelitian oleh Irawati (2016), yang menjelaskan penggunaan equirectangular approximation untuk menghitung jarak chemical store di Kota Medan dan menggunakan hasil perhitungan tersebut untuk menentukan lokasi chemical store terdekat dari lokasi pengguna[9].

Penelitian yang dilakukan oleh Morar et al, (2020) mengenai pola waktu respon ambulan di Timișoara. Penelitian ini bertujuan untuk menganalisa kinerja bidang kesehatan di Timișoara dengan membandingkan waktu respon dengan jarak antara pusat penugasan ambulan dan tempat tujuan respon penugasan menggunakan equirectangular formula. Dari penelitian ini dihasilkan kesimpulan bahwa perbedaan waktu respon ambulan sangat terlihat pada pola Utara-Selatan dimana terjadi perbedaan waktu hingga 2.82 menit pada zona yang bersebelahan dan hingga 3.19 menit untuk zona yang berseberangan. Diketahui pada penelitian ini bahwa perbedaan waktu tersebut sangat dipengaruhi oleh sungai yang menjadi penghalang alami dalam respon ambulan di Timișoara[12].

Sandoval (2019) melakukan penelitian menggunakan face-api.js untuk melakukan deteksi dan pengenalan wajah, memperkirakan emosi, jenis kelamin, hingga umur seseorang. Tujuan dari penelitian ini adalah untuk menemukan bidang-bidang dimana pengaplikasian teknologi ini dapat berguna[7].

Penelitian yang dilakukan oleh Asadul et al (2012), berisi analisa dari pola pergerakan taksi perkotaan. Salah satu analisis yang dilakukan adalah berkaitan dengan perhitungan jarak geografik, dimana yang dulunya menggunakan Euclidean distance dari dua titik koordinat yang mana mengabaikan bahwa bumi berbentuk bulat, pada penelitian ini dibandingkan dua metode baru untuk menghitung jarak geografis dua titik koordinat, yaitu menggunakan formula equirectangular dan hukum bulat kosinus. Dari penelitian ini didapat kesimpulan bahwa hukum bulat kosinus lebih akurat, sedangkan formula equirectangular lebih cepat dalam pengeksesian kalkulasi. Karena data yang dibutuhkan dalam penelitian ini adalah rata-rata dari data yang terkumpul, kecepatan eksekusi kalkulasi lebih diutamakan sehingga formula equirectangular approximation dipilih sebagai metode penghitungan jarak geografis[14].

Lukas et al. (2016), melakukan penelitian untuk mengotomatisasi absensi siswa menggunakan pengenalan wajah menggunakan RBFN (Radial Basis Function Network). Dari penelitian tersebut berhasil dilakukan multi pengenalan wajah siswa yang kemudian digunakan sebagai data absensi siswa dalam suatu kelas. Hasil dari pengenalan wajah yang dilakukan masih terdapat kekeliruan dimana wajah siswa dikenali sebagai siswa lain [9].

Penelitian yang dilakukan oleh Yang dan Han (2016), melakukan pengenalan wajah berdasarkan Real-Time Video Processing. Pengenalan wajah tersebut dilakukan guna mengurangi kecurangan yang terjadi pada kegiatan absensi yang dilakukan oleh mahasiswa suatu universitas. Dari penelitian tersebut dihasilkan akurasi pengenalan wajah yang berkisar pada 82%. Dengan begitu kecurangan absensi dapat diminimalisir hingga 13% [11].

Penelitian tentang deteksi wajah untuk absensi dengan notifikasi GSM yang dilakukan oleh Okokpujie et al (2020), menghasilkan kesimpulan bahwa deteksi wajah yang dilakukan memiliki tingkat kesuksesan yang bervariasi dan terpengaruh oleh faktor-faktor seperti: pencahayaan, sudut citra digital, dan ekspresi wajah. Dengan ekspresi wajah yang bervariasi tingkat kesuksesan pengenalan wajah berada pada kisaran 70,83% sedangkan dengan ekspresi wajah yang sama, tingkat kesuksesan menurun pada kisaran 54,17% [15].

Sunaryono dkk. (2021) melakukan penelitian untuk membangun suatu sistem pengenalan wajah berbasis android menggunakan LDA (Linear Discriminant Analysis). Gambar wajah yang digunakan untuk pembandingan disimpan sebanyak 10 kali dengan ukuran 244x244 pixel dan disimpan ke dalam database. Gambar wajah tersebut dicocokkan dengan gambar dari kamera depan android. Dalam penelitian ini, tingkat akurasi pengenalan wajah adalah sebesar 82% [10].

Winarno dkk. (2019) melakukan penelitian mengenai pengenalan wajah dengan menggunakan metode CNN-PCA (Convolutional Neural Network - Principal Component Analysis). Dimana CNN digunakan sebagai pendeteksi wajah dan berfungsi untuk merekonstruksi citra wajah 2D menjadi model 3D, dan PCA digunakan untuk melakukan feature extraction dari wajah yang menjadi input. Dari penelitian ini dihasilkan akurasi 90% hingga 98% yang lebih tinggi dibandingkan hanya dengan menggunakan PCA dimana akurasi berkisar antara 90% hingga 96% [3].

State of the art penelitian ini adalah penggunaan face-api.js yang berfungsi sebagai pencocok wajah dan digabungkan dengan menggunakan equirectangular approximation yang berfungsi sebagai perhitungan jarak koordinat perangkat dan koordinat lokasi absensi yang diharapkan dapat mengurangi kecurangan absensi karyawan. Selain itu, dikarenakan sumber daya yang diperlukan oleh face-api.js dan equirectangular approximation yang ringan, memungkinkan proses pencocokan wajah dan perhitungan koordinat dilakukan dari sisi klien, yang berdampak pada muatan server yang lebih ringan.

3. Metodologi

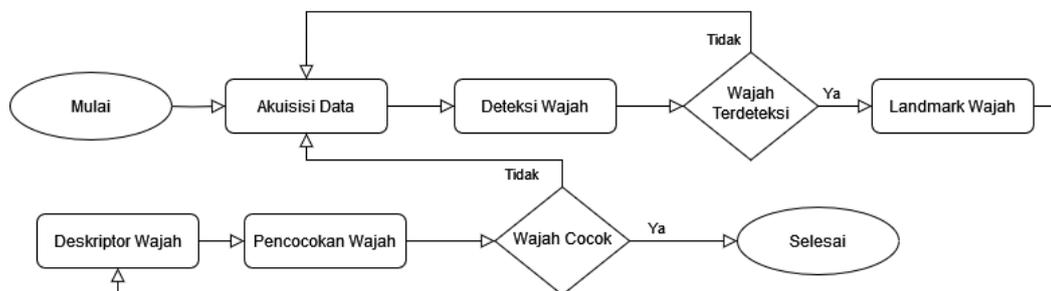
Metode yang digunakan dalam penelitian ini dibahas dalam dua bagian, yaitu metode pengenalan wajah dan metode perhitungan jarak koordinat geografis. Dalam metode pengenalan wajah, digunakan face-api.js sebagai modul javascript yang digunakan untuk mencocokkan dan mengenali wajah. Sementara pada metode perhitungan jarak koordinat digunakan formula equirectangular approximation untuk menyelesaikan perhitungan jarak.

3.1 Pengenalan Wajah

Pengertian deteksi wajah yaitu merupakan suatu sistem dalam teknologi komputer yang ditujukan untuk melakukan deteksi wajah dari sebuah citra digital dengan memanfaatkan artificial intelligence (AI). Suatu face detection dapat diaplikasikan kedalam berbagai hal, salah satunya adalah deteksi wajah atau pengenalan wajah dimana suatu wajah hasil deteksi wajah dibandingkan dengan wajah lain dari deteksi wajah sebagai citra sumber. Apabila hasil dari perbandingan tersebut memenuhi kriteria maka suatu pengenalan wajah akan dianggap berhasil.

Dalam pengenalan wajah akan digunakan face-api.js sebagai modul untuk melakukan proses pengenalan wajah. Face-api.js merupakan sebuah modul javascript yang dibangun diatas tensorflow.js core, yang mengimplementasikan beberapa CNN (Convolutional Neural Networks) untuk melakukan deteksi wajah, pengenalan wajah, dan deteksi fitur-fitur wajah, yang dioptimalkan untuk web dan perangkat mobile (Vincent Mühler, 2018). Dengan begitu, face-api.js dapat digunakan melalui semua perangkat yang memiliki peramban web dan dapat mengakses website, dengan kompatibilitas yang luas.

Pengenalan wajah menggunakan face-api.js dilakukan melalui beberapa proses. Proses proses tersebut yaitu akuisisi data, pengenalan wajah, landmark wajah, deskriptor wajah, dan pencocokan wajah seperti pada Gambar 1.



Gambar 1. Alur pengenalan wajah

Proses akuisisi, data menggunakan interface webcam pada HTML, deteksi wajah menggunakan pre-trained model, feature extraction dan face descriptor dimana didalamnya termasuk pendeteksian 68 titik landmark wajah serta pengkomputasian 148 feature vector menggunakan pre-trained model, dan pencocokan wajah yang dilakukan dengan melakukan perhitungan euclidean distance pada feature vector wajah referensi dan hasil komputasi feature vector langkah sebelumnya.



Gambar 2. Wajah Terdeteksi

1) Akuisisi Data

Terdapat dua tahap akuisisi data untuk proses pengenalan wajah yaitu akuisisi data foto profil yang akan digunakan sebagai sumber pencocokan wajah dan akuisisi data webcam (Gambar 2) yang akan digunakan sebagai pembandingan pencocokan wajah.



Gambar 3. Foto profil

Akuisisi foto profil (Gambar 3) dilakukan dengan mengambil url foto karyawan yang sudah tersimpan dalam database. Url tersebut kemudian dimasukkan sebagai nilai dari atribut src pada elemen HTML .

Akuisisi data kedua dilakukan menggunakan fitur webcam yang terintegrasi dengan HTML dan JavaScript. Dalam frame-frame yang diambil melalui webcam akan dilakukan proses pengenalan wajah dengan menggunakan frame secara bergiliran tiap perulangan setelah wajah terdeteksi dan telah dilakukannya pencocokan wajah atau setelah proses pendeteksian selesai dan wajah tidak ditemukan. Hal ini dilakukan agar sistem tidak terbebani apabila harus melakukan proses-proses pencocokan wajah dari tiap frame yang digenerasi oleh webcam, melainkan hanya melakukan proses pencocokan pada frame pertama dan frame selanjutnya setelah pencocokan wajah selesai dijalankan pada langkah sebelumnya. Dengan begitu sistem tetap mampu berjalan secara optimal menggunakan resource seminimal mungkin.

2) Deteksi Wajah

Pada proses deteksi wajah, dilakukan perubahan ukuran citra yang diambil melalui frame dalam webcam menjadi maksimal 512px untuk salah satu nilai dimensi citra tertinggi dan melakukan penyesuaian skala untuk dimensi citra terendah. Setelah dilakukan perubahan ukuran, citra diproses menjadi nilai pixel tensor dengan menggunakan fungsi `tf.browser.fromPixels()`. Fungsi tersebut mengkonversi gambar menjadi array matrix dengan nilai 0-250 untuk tiap pixel yang merepresentasikan gambar dari suatu elemen HTML (dalam hal ini, frame dari webcam) yang kemudian akan diolah untuk deteksi wajah.

Dari gambar yang telah diolah menjadi matrix, dilakukan pendeteksian wajah dengan menggunakan model tensorflow SSDMobileNetV1 yang merupakan sebuah pre-trained model yang menggabungkan Single Shot MultiBox Detector (SSD) framework dengan MobileNetV1 convolutional neural network. Telah dilakukan pre-training terhadap model SSDMobileNetV1 yang akan digunakan terhadap dataset WIDERFACE. Hasil keluaran dari model SSDMobileNetV1 adalah titik x dan y serta tinggi dan lebar dari objek yang terdeteksi pada gambar

3) Feature Extraction

Dari hasil deteksi wajah pada langkah sebelumnya, akan dilakukan proses kalkulasi ekstraksi fitur 68 titik landmark wajah menggunakan model `face_landmark_68_model` yang merupakan sebuah pre-trained model yang berjalan diatas tensorflow. Model tersebut sudah dilatih menggunakan 35 ribu dataset gambar wajah yang telah diberi label untuk 68 titik landmark

wajah. Sintaks yang digunakan untuk kalkulasi adalah `withFaceLandmarks()` dimana sintaks tersebut akan menghasilkan array 68 titik koordinat x dan y dari masing-masing landmark wajah yang meliputi alis kanan, alis kiri, mata kanan, mata kiri, hidung, mulut, dan outline wajah.

4) Face Descriptor

Proses selanjutnya adalah generasi face descriptor, yang dilakukan menggunakan model `face_recognition_model`. Model tersebut menggunakan arsitektur seperti ResNet-34 yang diimplementasikan untuk mengkomputasi face descriptor (sebuah feature vector dengan 128 nilai) dari sebuah gambar wajah, yang digunakan untuk mendeskripsikan karakteristik wajah seseorang. Fungsi yang digunakan untuk melakukan kalkulasi face descriptor adalah `withFaceDescriptor()` yang akan menghasilkan array berisi nilai-nilai bertipe float32 yang digunakan untuk mendeskripsikan wajah.

5) Pencocokan Wajah

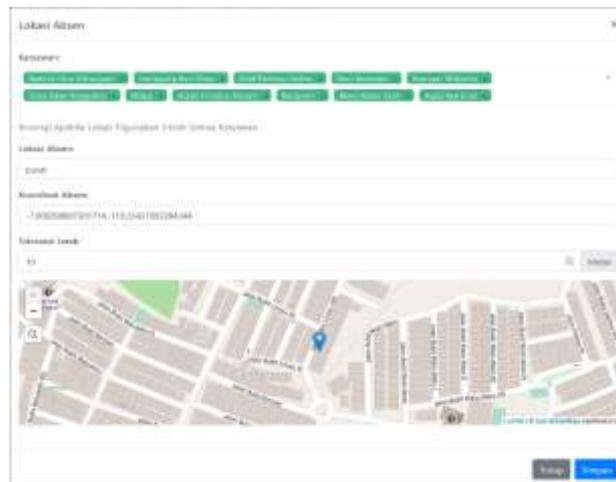
Pencocokan wajah dilakukan dengan mengaplikasikan formula euclidean distance terhadap dua value face descriptor dari gambar karyawan pada foto profil dan webcam.

$$\sqrt{(q_1 - p_1)^2 + \dots + (q_n - p_n)^2} \quad (1)$$

Formula (1) merupakan formula yang digunakan pada euclidean distance adalah. Dengan menerapkan formula tersebut, akan didapatkan threshold dari perbandingan wajah foto profil dan wajah pada webcam. Dari threshold tersebut ditentukan nilai yang tepat untuk digunakan sebagai nilai maksimal perbedaan deskripsi wajah. Sehingga apabila nilai hasil kalkulasi formula euclidean distance melebihi threshold yang sudah ditentukan, maka wajah akan dianggap berbeda.

3.2 Perhitungan jarak koordinat geografis

1) Penyimpanan Koordinat Kantor



Gambar 4. Input Koordinat Kantor

Penyimpanan koordinat kantor (Gambar 4) dilakukan oleh bagian SDM dengan memasukkan daftar karyawan, nama lokasi absen, koordinat kantor, dan toleransi jarak absensi. Data tersebut disimpan kedalam database sebagai data koordinat yang akan digunakan untuk menghitung jarak koordinat perangkat dari koordinat kantor.

2) Akuisisi Koordinat Perangkat

Akuisisi koordinat perangkat dilakukan dengan menggunakan fungsi javascript `navigator.geolocation.getCurrentPosition()`. Dengan menjalankan fungsi tersebut akan didapatkan data koordinat longitude dan latitude dari sensor GPS pada perangkat. Koordinat kemudian disimpan dalam variabel javascript yang kemudian dikalkulasikan dengan data koordinat kantor yang telah disimpan dalam database.

3) Perhitungan Jarak

Dari kedua koordinat yang telah disimpan, diterapkan formula equirectangular approximation yang akan menghasilkan jarak kedua koordinat.

Penerapan metode equirectangular approximation kedalam perhitungan jarak antara dua titik koordinat mengasumsikan bahwa Bumi adalah objek datar, dan menggunakan proyeksi equirectangular untuk memetakan koordinat lintang dan bujur ke dalam sistem koordinat kartesian.

Pada metode equirectangular, koordinat lintang dan bujur dari dua titik di permukaan Bumi direpresentasikan dalam derajat desimal. Perbedaan bujur antara kedua titik dinyatakan dalam derajat ($\Delta\lambda$), sedangkan perbedaan lintang antara kedua titik dinyatakan dalam derajat ($\Delta\phi$).

Dalam metode ini, digunakan rumus jarak Euclidean pada sistem koordinat kartesian untuk menghitung jarak linear antara dua titik.

$$\begin{aligned}
 \phi &= \textit{latitude} \\
 \lambda &= \textit{longitude} \\
 R &= \textit{radius bumi} \\
 x &= \Delta\lambda \times \cos(\phi m) \\
 y &= \Delta\phi \\
 d &= R \times \sqrt{x^2 + y^2} \tag{2}
 \end{aligned}$$

Dalam (2), digunakan nilai kosinus dari rata-rata lintang untuk memperhitungkan perbedaan skala dalam perhitungan jarak pada lintang. Ini diperlukan karena jarak sepanjang garis lintang akan berkurang seiring meningkatnya jarak dari garis khatulistiwa.

Setelah menghitung jarak menggunakan rumus di atas, akan didapatkan jarak linier antara dua titik koordinat yang diaproksimasi dengan menggunakan pendekatan equirectangular.

Dari kedua koordinat yang telah disimpan, diterapkan formula equirectangular approximation yang akan menghasilkan jarak kedua koordinat. Contoh dari perhitungan jarak koordinat menggunakan equirectangular approximation adalah seperti pada (3).

$$\begin{aligned}
 \phi 1 &= -7.000407499503508 \\
 \lambda 1 &= 110.33437895133723 \\
 \phi 2 &= -7.0004317 \\
 \lambda 2 &= 110.3343664 \\
 R &= 6371000 \\
 \Delta\lambda &= ((\lambda 2 * \pi/180) - (\lambda 1 * \pi/180)) \\
 \phi m &= (\phi 1 + \phi 2)/2 \\
 x &= \Delta\lambda \times \cos(\phi m) \\
 y &= ((\phi 2 * \pi/180) - (\phi 1 * \pi/180)) \\
 d &= 6371000 \times \sqrt{x^2 + y^2} = 3.03 \tag{3}
 \end{aligned}$$

Dalam kalkulasi tersebut, R merupakan diameter bumi dalam satuan meter, yang dilakukan untuk mengkalkulasi jarak dua koordinat yaitu (-7.000407499503508, 110.33437895133723) sebagai koordinat kantor dan (-7.0004317, 110.3343664) sebagai koordinat perangkat. Hasil dari kalkulasi equirectangular approximation dari kedua koordinat tersebut adalah 3.0265863352430693 meter. Apabila jarak koordinat kurang dari sama dengan toleransi yang disimpan dalam database, maka perangkat dianggap berada dalam lokasi kantor.

4. Hasil dan Pembahasan

4.1 Penentuan Threshold Pengenalan Wajah

Penentuan threshold diperlukan untuk menemukan ambang batas yang tepat digunakan dalam pencocokan wajah dengan tujuan sehingga wajah yang dicocokkan dengan pembanding hanya dapat terdeteksi kecocokan wajah apabila kedua wajah tersebut merupakan wajah orang yang sama.



Gambar 5. Wajah Sampel

Penentuan threshold dilakukan dengan melakukan proses pencocokan lima sampel wajah pada Gambar 5 dengan masing-masing wajah dilakukan seratus kali proses pencocokan. Dari hasil proses pencocokan tersebut didapatkan threshold yang kemudian dibulatkan ke bawah dalam satu digit desimal dan ditemukan nilai paling rendah, nilai paling tinggi, dan modus dari hasil proses pencocokan.

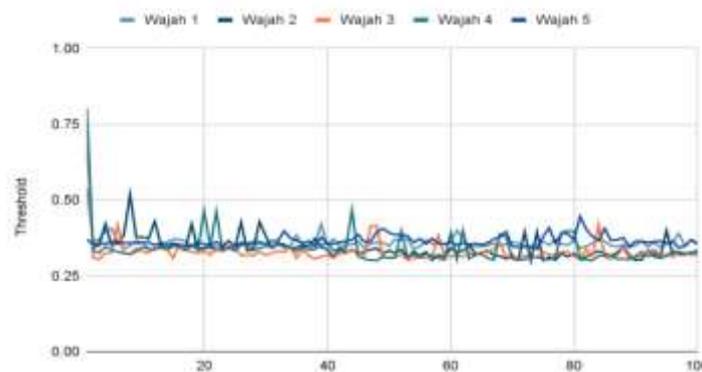
Tabel 1. Hasil proses pencocokan

W	PROFILE 1			PROFILE 2			PROFILE 3			PROFILE 4			PROFILE 5		
	R	S	T	R	S	T	R	S	T	R	S	T	R	S	T
W1	0.3	0.3	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.5	0.5	0.7	0.5	0.6	0.7
W2	0.5	0.6	0.7	0.3	0.3	0.7	0.5	0.6	0.8	0.5	0.5	0.7	0.6	0.7	0.7
W3	0.6	0.7	0.7	0.5	0.6	0.6	0.3	0.3	0.7	0.4	0.4	0.7	0.6	0.6	0.7
W4	0.5	0.6	0.7	0.5	0.5	0.6	0.5	0.5	0.7	0.3	0.3	0.8	0.5	0.5	0.7
W5	0.5	0.5	0.7	0.6	0.6	0.8	0.6	0.7	0.7	0.6	0.6	0.7	0.3	0.3	0.4

Berdasarkan hasil tersebut (Tabel 1), dimana R = nilai terendah, S = nilai paling sering keluar, dan T = nilai tertinggi, dapat diambil keputusan bahwa 0.3 merupakan threshold yang tepat sebagai ambang batas pencocokan wajah dimana hanya wajah orang yang sama yang akan terdeteksi kecocokannya.

4.2 Analisis Keakuratan Pengenalan Wajah

Berdasarkan threshold yang sudah ditentukan, dapat dihitung keakuratan pengenalan wajah dengan membandingkan data dibawah threshold dengan data keseluruhan (Gambar 6).



Gambar 6. Grafik threshold pengenalan wajah

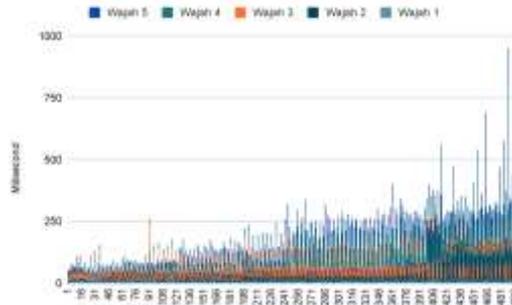
Perbandingan tersebut dapat ditampilkan dalam bentuk sebagai persentase keberhasilan pengenalan wajah (Tabel 2).

Tabel 2. Akurasi pengenalan wajah

Wajah					
1	2	3	4	5	Rata-rata
97%	93%	95%	96%	95%	95.2%

4.3 Waktu pemrosesan pengenalan wajah

Untuk pengujian waktu pemrosesan pengenalan wajah, durasi pemrosesan dihitung sejak fungsi dijalankan hingga wajah dari webcam selesai dicocokkan dengan wajah dari profile.

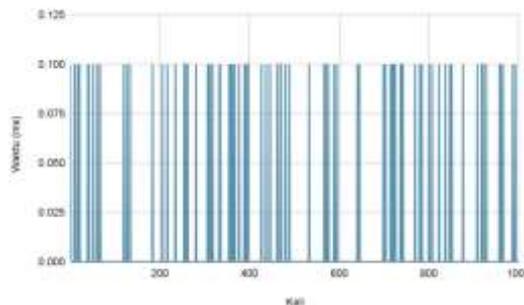


Gambar 7. Grafik durasi pengenalan wajah

Berdasarkan percobaan yang telah dilakukan (Gambar 7), waktu pemrosesan rata-rata adalah 35.5228 ms.

4.4 Kecepatan perhitungan jarak koordinat

Analisis kecepatan perhitungan jarak koordinat dilakukan dengan melakukan kalkulasi formula equirectangular approximation terhadap seribu pasang koordinat acak yang dilakukan sebanyak seribu kali. Dari perhitungan tersebut dapat ditarik data kecepatan perhitungan jarak koordinat sebagai berikut.



Gambar 8. Grafik kecepatan perhitungan jarak

Berdasarkan data yang sudah terkumpulkan (Gambar 8) dapat ditentukan nilai maksimum kalkulasi selama 0.1000000089 ms, nilai minimum kalkulasi selama 0 ms, dan nilai rata-rata kalkulasi selama 0.008300000003 ms. Dengan begitu, dapat disimpulkan bahwa perhitungan koordinat menggunakan equirectangular approximation memerlukan waktu kurang dari 0.2 ms yang dapat dikategorikan sebagai waktu yang sangat cepat dalam proses kalkulasi pada server.

Dari proses-proses dan hasil keluaran yang sudah disajikan berupa akurasi dan waktu ditemukan bahwa temuan dalam penelitian ini sejajar dengan penelitian-penelitian terdahulu [1][7][12][13] dimana proses pencocokan wajah dan perhitungan koordinat dapat berjalan dengan waktu yang relatif cepat dan hasil yang akurat meskipun proses-prosesnya berjalan pada sisi klien. Hal ini menunjukkan bahwa proses-proses yang digunakan dalam penelitian ini dapat mengurangi beban server dengan mendelegasikan proses pencocokan wajah dan perhitungan jarak koordinat kepada sisi klien.

5. Simpulan

Berdasarkan pembahasan yang telah dipaparkan dapat diambil beberapa kesimpulan bahwa threshold yang tepat digunakan dalam deteksi wajah adalah 0.3. Dimana wajah yang berbeda tidak akan mendapat nilai threshold dibawah 0.4, sehingga hanya wajah yang sama yang akan terdeteksi. Dengan menggunakan threshold tersebut, akurasi deteksi wajah yang didapat berkisar antara 93% hingga 97% dengan rata-rata 95.2%, yang memakan waktu proses kalkulasi dengan rata-rata 38.5228 ms. Sedangkan dalam perhitungan jarak koordinat menggunakan equirectangular approximation, waktu rata-rata yang dibutuhkan untuk kalkulasi jarak adalah 0.008300000003 ms dengan nilai maksimum 0.1000000089 ms dan nilai minimum 0 ms.

Beberapa usulan juga dapat disampaikan berdasarkan penelitian ini untuk penelitian-penelitian yang akan datang bahwa penggunaan face recognition masih dalam tahap dasar dan masih bisa dilabui menggunakan foto statis. Diharapkan adanya metode untuk mencegah penggunaan foto statis. Dan pada kalkulasi jarak, koordinat yang diambil oleh perangkat dapat diganti secara manual menggunakan perangkat lunak pihak ketiga. Diharapkan penambahan validasi koordinat asli perangkat.

Daftar Referensi

- [1] D. Yadav, S. Maniar, K. Sukhani, and K. Devadkar, In-Browser Attendance System using Face Recognition and Serverless Edge Computing. Institute of Electrical and Electronics Engineers Inc., 2021.
- [2] A. Hermawan, L. Lianata, Junaedi, and A. R. K. Maranto, 'Implementasi Machine Learning Sebagai Pengenal Nominal Uang Rupiah dengan Metode YOLOv3', SATIN - Sains dan Teknologi Informasi, vol. 8, no. 1, pp. 12–22, 2022.
- [3] E. Winarno, I. H. A. Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati, and M. T. Anwar, Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-Time Camera. Institute of Electrical and Electronics Engineers Inc., 2019.
- [4] I. E. Hendarawan, 'Vehicle Tire Crack Classification Using ResNet50 Architecture', SATIN - Sains dan Teknologi Informasi, vol. 9, no. 1, Dec. 2022.
- [5] D. Frenza and R. Mukhaiyar, 'Aplikasi Pengenalan Wajah Menggunakan Metode Adaptive Resonance Theory (ART)', Ranah Research : Journal Of Multidisciplinary Research and Development, vol. 3, no. 3, pp. 147–153, 2021.
- [6] N. Wayan Wardani and M. Yoka Fathoni, 'Perancangan Absensi Berbasis Face Recognition Pada Desa Sokaraja Lor Menggunakan Platform Android', Jurnal Teknik Informatika dan Sistem Informasi, vol. 8, no. 1, pp. 91–104, 2021.
- [7] C. Gabriel and E. Sandoval, 'Multiple Face Detection and Recognition System Design Applying Multiple Face Detection and Recognition System Design Applying Deep Learning in Web Browsers using JavaScript Deep Learning in Web Browsers using JavaScript Multiple Face Detection and Recognition System Design Applying Deep Learning in Web Browsers using JavaScript'. 2019.
- [8] Vincent Mühler, 'face-api.js — JavaScript API for Face Recognition in the Browser with tensorflow.js', ITNEXT. 2018.
- [9] S. Lukas, A. R. Mitra, R. I. Desanti, and D. Krisnadi, Student attendance system in classroom using face recognition technique. Institute of Electrical and Electronics Engineers Inc., 2016.
- [10] D. Sunaryono, J. Siswantoro, and R. Anggoro, 'An android based course attendance system using face recognition', Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 3, pp. 304–312, 2021.
- [11] H. Yang and X. Han, 'Face Recognition Attendance System Based on Real-Time Video Processing', IEEE Access, vol. 8, pp. 159143–159150, 2020.
- [12] A. M. Morar, L. Efleih-Hassan, and D. Lungeanu, 'Apparent Patterns in Ambulance Response Time in Timișoara', 2020.
- [13] S. Shaikh, A. Matono, and K.-S. Kim, A Distance-Window Based Real-Time Processing of Spatial Data Streams. 2019.
- [14] M. Asadul Hoque, X. Hong, and B. Dixon, 'Analysis of Mobility Patterns for Urban Taxi Cabs', Applied Medical Informatics, vol. 42, no. 1, pp. 28–35, 2020.
- [15] K. Okokpujie, E. Noma-Osaghae, S. John, K.-A. Grace, and I. Okokpujie, A face recognition attendance system with GSM notification. 2017.