

**PENGEMBANGAN SISTEM PENGENALAN WAJAH TERINTEGRASI
PADA ROBOT RESEPSIONIS MENGGUNAKAN METODE DEEP
LEARNING BERBASIS CONVOLUTIONAL NEURAL NETWORK**



TESIS

Oleh :

ENDANG DARMAWAN YUDI

222420033

ENTERPRISE IT INFRASTRUCTURE

**PROGRAM STUDI TEKNIK INFORMATIKA – S2
PROGRAM PASCASARJANA
UNIVERSITAS BINA DARMA
PALEMBANG
2026**

**PENGEMBANGAN SISTEM PENGENALAN WAJAH TERINTEGRASI
PADA ROBOT RESEPSIONIS MENGGUNAKAN METODE DEEP
LEARNING BERBASIS CONVOLUTIONAL NEURAL NETWORK**

Tesis ini diajukan sebagai salah satu syarat
untuk memperoleh gelar



MAGISTER TEKNIK INFORMATIKA

Oleh :

ENDANG DARMAWAN YUDI

222420033

ENTERPRISE IT INFRASTRUCTURE

PROGRAM STUDI TEKNIK INFORMATIKA – S2

PROGRAM PASCASARJANA

UNIVERSITAS BINA DARMA

PALEMBANG

2026

Halaman Pengesahan Pembimbing Tesis

Judul Tesis: PENGEMBANGAN SISTEM PENGENALAN WAJAH PADA ROBOT
RESEPSIONIS MENGGUNAKAN METODE DEEP LEARNING
BERBASIS CONVOLUTIONAL NEURAL NETWORK

Oleh ENDANG DARMAWAN YUDI, NIM 222420033, Tesis ini telah disetujui dan
disahkan oleh Pembimbing Program Studi Teknik Informatika – S2 konsentrasi
ENTERPRISE IT INFRASTRUCTURE, Program Pascasarjana Universitas Bina Darma
pada 4 Maret 2026 dan telah dinyatakan LULUS.

Palembang, 4 Maret 2026
Mengetahui,
Program Studi Teknik Informatika – S2
Universitas Bina Darma
Ketua,


Universitas Bina Darma
Magister Teknik Informatika

.....
Dr. Usman Ependi, S.Kom., M.Kom.

Pembimbing,



.....
Dr. Yesi Novaria Kunang, S. T., M. Kom

Halaman Pengesahan Penguji Tesis

Judul Tesis: PENGEMBANGAN SISTEM PENGENALAN WAJAH PADA ROBOT
RESEPSIONIS MENGGUNAKAN METODE DEEP LEARNING
BERBASIS CONVOLUTIONAL NEURAL NETWORK

Oleh ENDANG DARMAWAN YUDI , NIM 222420033, Tesis ini telah disetujui dan
disahkan oleh Tim Penguji Program Studi Teknik Informatika – S2 konsentrasi
ENTERPRISE IT INFRASTRUCTURE, Program Pascasarjana Universitas Bina Darma
pada 4 Maret 2026 dan telah dinyatakan LULUS.

Palembang, 4 Maret 2026

Mengetahui,

Program Pascasarjana

Universitas Bina Darma

Direktur,


PROGRAM PASCASARJANA

.....
Prof. Dr. Ir. Achmad Syarifudin, M.Sc.

Penguji I,



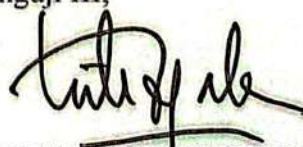
.....
Dr. Yesi Novaria Kunang, S. T., M. Kom

Penguji II,


.....

Dr. Usman Ependi, S. Kom., M. Kom

Penguji III,


.....

M. Izman Herdiansyah, S. T., M. M., Ph. D

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : ENDANG DARMAWAN YUDI

NIM : 222420033

Dengan ini menyatakan bahwa:

1. Karya tulis Saya (Tesis, Skripsi, Tugas Akhir) ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik baik (Magister, Sarjana, dan Ahli Madya) di Universitas Bina Darma;
2. Karya tulis ini murni gagasan, rumusan dan penelitian Saya sendiri dengan arahan tim pembimbing;
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dikutip dengan mencantumkan nama pengarang dan memasukkan ke dalam daftar pustaka;
4. Karena yakin dengan keaslian karya tulis ini, Saya menyatakan bersedia Tesis/Skripsi/Tugas Akhir, yang Saya hasilkan di unggah ke internet;
5. Surat Pernyataan ini Saya tulis dengan sungguh-sungguh dan apabila terdapat penyimpangan atau ketidakbenaran dalam pernyataan ini, maka Saya bersedia menerima sanksi dengan aturan yang berlaku di perguruan tinggi ini.

Demikian Surat Pernyataan ini saya buat agar dapat dipergunakan sebagaimana mestinya.

Palembang, 4 Maret 2026
Yang Membuat Pernyataan,



ENDANG DARMAWAN YUDI
NIM: 222420033

ABSTRAK

Penelitian ini bertujuan mengembangkan sistem pengenalan wajah yang terintegrasi pada robot resepsionis menggunakan metode Deep Learning berbasis Convolutional Neural Network (CNN) untuk meningkatkan kualitas interaksi antara manusia dan robot secara otomatis dan real-time. Sistem dirancang melalui beberapa tahapan yaitu deteksi wajah menggunakan metode Histogram of Oriented Gradients (HOG), ekstraksi fitur wajah menggunakan arsitektur ResNet pada CNN untuk menghasilkan representasi numerik berupa face embedding berdimensi 128, serta proses identifikasi menggunakan metode Euclidean Distance untuk menentukan tingkat kemiripan wajah. Sistem diimplementasikan pada komputer sebagai pusat pemrosesan dan diintegrasikan dengan mikrokontroler Arduino untuk mengendalikan pergerakan robot menggunakan motor servo serta memberikan respons otomatis berupa salam, proses check-in dan check-out, serta keluaran suara berdasarkan hasil pengenalan wajah. Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi dan mengenali wajah secara stabil dengan performa real-time. Visualisasi embedding menggunakan metode t-SNE menunjukkan terbentuknya kluster fitur wajah yang konsisten untuk setiap individu. Selain itu, pengujian performa menunjukkan rata-rata waktu pemrosesan sistem sebesar sekitar 2,4 detik per frame dengan proses ekstraksi embedding sekitar 0,37 detik per citra. Hasil tersebut menunjukkan bahwa integrasi metode HOG dan CNN mampu menghasilkan sistem pengenalan wajah yang akurat dan responsif, sehingga robot resepsionis dapat memberikan layanan yang lebih cerdas, personal, dan otomatis dalam mendukung sistem pelayanan berbasis kecerdasan buatan.

Kata kunci: Pengenalan Wajah, Deep Learning, Convolutional Neural Network, ResNet, Euclidean Distance, Robot Resepsionis, Face Embedding.

ABSTRACT

This study aims to develop a face recognition system integrated into a receptionist robot using a Deep Learning approach based on Convolutional Neural Networks (CNN) to improve the quality of human–robot interaction in an automatic and real-time manner. The system is designed through several stages, including face detection using the Histogram of Oriented Gradients (HOG) method, facial feature extraction using a ResNet-based CNN architecture to generate numerical representations in the form of 128-dimensional face embeddings, and an identification process using the Euclidean Distance method to determine the similarity between facial features. The system is implemented on a computer as the main processing unit and integrated with an Arduino microcontroller to control the robot's movements using servo motors and provide automatic responses such as greetings, check-in and check-out processes, and voice output based on the face recognition results. The experimental results show that the system is capable of detecting and recognizing faces consistently with stable real-time performance. Visualization of facial embeddings using the t-SNE method demonstrates the formation of consistent clusters for each individual. In addition, performance testing indicates that the average processing time of the system is approximately 2.4 seconds per frame, while the embedding extraction process requires about 0.37 seconds per image. These results indicate that the integration of HOG and CNN methods can produce an accurate and responsive face recognition system, enabling the receptionist robot to provide more intelligent, personalized, and automated services to support artificial intelligence-based hospitality systems.

Keywords: *Face Recognition, Deep Learning, Convolutional Neural Network, ResNet, Euclidean Distance, Receptionist Robot, Face Embedding.*

MOTTO DAN PERSEMBAHAN

MOTTO :

"Inovasi lahir dari keberanian untuk mencoba hal-hal baru, sementara kemajuan tercipta melalui ketekunan, kesabaran, dan komitmen untuk terus menyempurnakan setiap proses yang dijalani. Setiap langkah kecil dalam proses belajar merupakan bagian dari perjalanan panjang menuju pemahaman yang lebih luas, di mana pengetahuan tidak hanya menjadi alat untuk berkembang, tetapi juga sarana untuk memberikan kontribusi positif bagi masyarakat dan dunia."

PERSEMBAHAN:

Dengan penuh rasa syukur kepada Tuhan Yang Maha Esa, karya ilmiah ini penulis persembahkan kepada:

- Ayah dan Ibu tercinta, yang senantiasa memberikan kasih sayang, doa, dukungan moral, serta pengorbanan yang tidak ternilai sepanjang perjalanan hidup dan pendidikan penulis.
- Istri tercinta, yang selalu memberikan semangat, kesabaran, serta dukungan penuh dalam setiap proses yang dilalui selama penyusunan karya ilmiah ini.
- Dosen pembimbing, yang dengan penuh kesabaran, dedikasi, dan bimbingan akademis telah memberikan arahan, ilmu, serta motivasi sehingga penelitian ini dapat diselesaikan dengan baik.
- Para pendidik dan dosen, yang telah menanamkan ilmu pengetahuan, wawasan, serta nilai-nilai akademik yang menjadi bekal berharga dalam perjalanan intelektual penulis.
- Sahabat dan rekan seperjuangan, yang telah memberikan dukungan, motivasi, serta kebersamaan dalam menghadapi berbagai tantangan selama proses studi dan penelitian.
- Almamater tercinta, sebagai tempat penulis menimba ilmu, mengembangkan pemikiran, serta membentuk karakter akademis yang menjadi fondasi dalam perjalanan keilmuan di masa depan.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga tesis berjudul “*Pengembangan Sistem Pengenalan Wajah pada Robot Resepsionis Menggunakan Metode Deep learning Berbasis Convolutional Neural Network*” ini dapat diselesaikan dengan baik sebagai salah satu syarat memperoleh gelar akademik. Penelitian ini mengembangkan sistem pengenalan wajah berbasis *Deep learning* menggunakan arsitektur CNN dengan metode *face embedding* dan Euclidean Distance yang diintegrasikan pada Robot Resepsionis untuk mendukung interaksi otomatis yang akurat dan real-time.

Penulis menyadari bahwa penyusunan tesis ini tidak terlepas dari dukungan, bimbingan, serta doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. **Prof. Dr. Ir. Achmad Syarifudin, M.Sc.** Selaku Direktur Pascasarjana Universitas Bina Darma Palembang.
2. **Dr. Usman Ependi, M.Kom.** Selaku ketua program Studi Magister Informatika Universitas Bina Darma Palembang dan penguji, terima kasih atas masukan dan koreksi berharga yang memperkaya penelitian ini.
3. **Dr. Yesi Novaria Kunang, S.T., M.Kom.** Selaku pembimbing utama yang telah memberikan bimbingan, arahan, dan dukungan selama proses penyusunan tesis ini.
4. **M. Izman Herdiansyah, S. T., M. M., Ph. D.** Selaku Penguji atas masukan dan koreksi berharga yang memperkaya penelitian ini.
5. **Kepala Sekolah dan Staf SMK Negeri 8 Palembang**, atas izin dan bantuan selama pelaksanaan penelitian.
6. **Rekan-rekan guru SMK Negeri 8 Palembang dan para sahabat**, yang telah memberikan motivasi dan dukungan moral.

Penulis menyadari bahwa tesis ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan

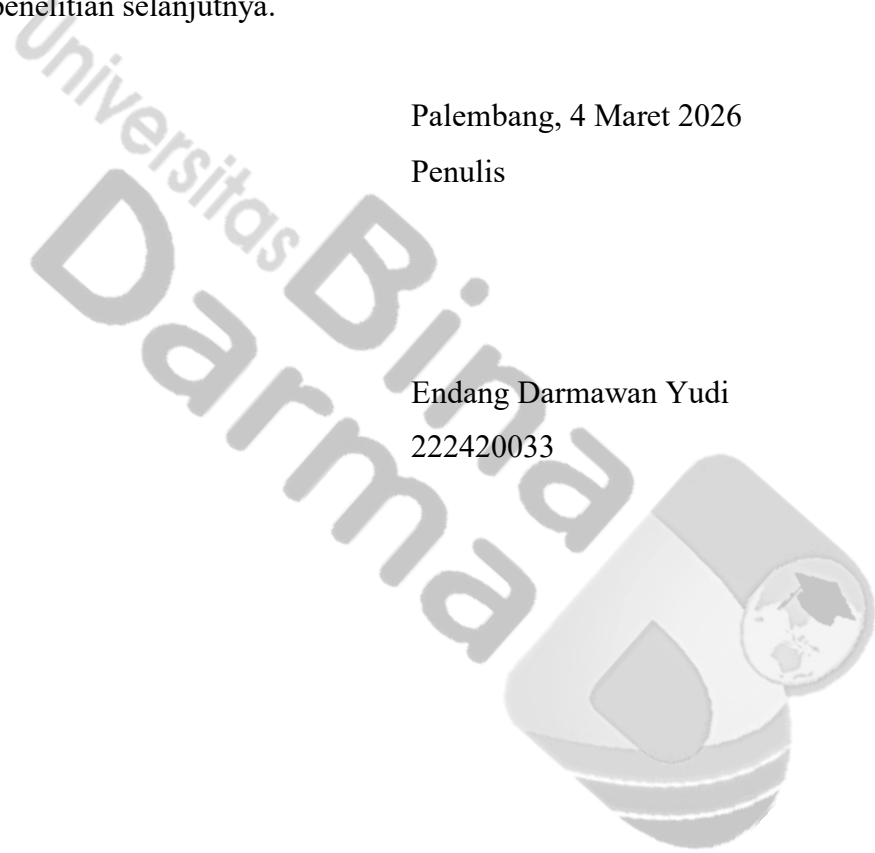
demi penyempurnaan karya ini di masa yang akan datang. Semoga penelitian ini dapat memberikan manfaat serta kontribusi dalam pengembangan teknologi. Akhir kata, semoga tesis ini dapat memberikan manfaat bagi pembaca dan menjadi referensi untuk penelitian selanjutnya.

Palembang, 4 Maret 2026

Penulis

Endang Darmawan Yudi

222420033



DAFTAR ISI

COVER TESIS.....	i
HALAMAN DEPAN	ii
HALAMAN PENGESAHAN PEMBIMBING TESIS	iii
HALAMAN PENGESAHAN PENGUJI TESIS.....	iv
SURAT PERNYATAAN.....	v
ABSTRAK (BAHASA INDONESIA)	vi
ABSTRACT (BAHASA INGGRIS)	vii
MOTTO DAN HALAMAN PERSEMBAHAN.....	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat.....	4
BAB II.....	6
TINJAUAN PUSTAKA DAN LANDASAN TEORI	7
2.1 Tinjauan Pustaka	7
2.2 Landasan Teori	10
2.2.1 Pengolahan Citra Digital	12
2.4 Face Detection.....	14
2.5 <i>Histogram of Oriented Gradients</i> (HOG)	15
2.5.1 Perhitungan Gradien pada <i>Histogram of Oriented Gradients</i>	16
2.5.2 Cascade Classifier.....	17
2.6 Pengenalan Wajah	19
2.7 Euclidean Distance	21
2.7.1 Konsep Dasar Euclidean Distance.....	21
2.8 Perangkat Komputer.....	21
2.9 Webcam.....	22
2.10 Arduino Mega 2560	23
2.11 Motor Servo	24
BAB III.....	25

METODOLOGI PENELITIAN.....	25
3.1 Waktu dan Tempat Penelitian	26
3.2 Kerangka Kerja Penelitian.....	27
3.2.1 Perancangan Sistem.....	29
3.2.2 Perancangan Perangkat Lunak	30
3.2.2.1 Program wajah.....	32
3.2.2.2 Akuisisi dan Deteksi Wajah.....	34
3.2.2.3 Recording dan Encoding Wajah	35
3.2.2.4 Proses Pengenalan dan Respon Sistem.....	37
3.2.3 Perancangan Perangkat Keras.....	38
3.2.3.2 Program Gerak Robot	40
BAB IV	42
HASIL DAN PEMBAHASAN.....	42
4.1 Hasil Pengujian.....	42
4.1.1 Pengujian Akuisisi dan Deteksi Wajah	42
4.1.1.1 Pengujian Koneksi Webcam.....	42
4.1.1.2 Deteksi Wajah Menggunakan HOG/CNN	44
4.1.2 Pengujian Recording dan Encoding Wajah.....	46
4.1.2.1 Perekaman Beberapa <i>Frame</i> Wajah (30 wajah).....	46
4.1.2.2 Ekstraksi <i>Face embedding</i>	47
4.1.3 Pengujian Proses Pengenalan Wajah.....	48
4.1.3.1 Pengujian Offline (Manual).....	49
4.1.3.2 Pengujian Online (Realtime)	51
4.1.4 Pengujian Perangkat Keras (Robot)	53
4.1.5 Analisis Performa Sistem	54
4.1.5.1 Analisis Waktu Proses Deteksi dan Pengenalan	55
4.1.6 Analisis Pengujian Robot	56
4.1.6.1 Analisis Waktu Proses Embedding	56
4.1.6.2 Analisis Akurasi Pengenalan.....	59
4.2 Pembahasan Hasil.....	64
DAFTAR PUSTAKA	68

DAFTAR GAMBAR

Gambar 2.1 Color Image	13
Gambar 2.2 Grayscale	13
Gambar 2.3 Perhitungan Gradien pada Citra untuk Metode HOG.....	17
Gambar 2.4 Arsitektur Dasar <i>Convolutional Neural Network</i> (CNN).....	19
Gambar 2.5 Webcam	23
Gambar 2.6 Arduino Mega 2560.....	24
Gambar 2.7 Motor Servo	25
Gambar 3.1 Kerangka kerja penelitian	27
Gambar 3.2 Flowchart Sistem Robot Resepsionis	30
Gambar 3.3 Blok diagram program pengenalan wajah	34
Gambar 3.4 Flowchart Akusisi dan Deteksi Wajah	35
Gambar 3.5 Flowchart Recording dan Encoding Wajah.....	37
Gambar 3.6 Proses Pengenalan dan Respon Sistem.....	38
Gambar 3.7 Robot Inmoov	39
Gambar 3.8 Desain Skematik Robot	40
Gambar 3.9 Flowchart Program Gerak Robot.....	41
Gambar 4.1 Pengujian FPS Kamera	43
Gambar 4. 2 Grafik Performa Webcam.....	43
Gambar 4.3 Pengujian <i>bounding box</i>	46
Gambar 4.4 Pengambilan Dataset Tamu	47
Gambar 4.5 Hasil ekstraksi <i>face embedding</i> 30 wajah dataset.....	48
Gambar 4.6 Pengujian <i>face recognition</i> offline (manual)	50
Gambar 4.7 Grafik Waktu Proses Pengenalan Wajah	56
Gambar 4.8 Grafik perbandingan waktu proses (1 Sampel)	57
Gambar 4.9 Grafik perbandingan lama proses Embedding per 1 sampel	59
Gambar 4.10 Grafik akurasi pengenalan (1 Sampel)	60
Gambar 4.11 Grafik akurasi pengenalan (9 Sampel) Endang Darmawan Yudi..	62
Gambar 4.12 Grafik akurasi pengenalan (9 Sampel) M. Zainal Abidin.....	62
Gambar 4.13. Grafik Akurasi Penambahan Jumlah Dataset	64

DAFTAR TABEL

Tabel 2. 1 Perbandingan penelitian terdahulu.....	8
Tabel 4. 1 Perbandingan HOG dan CNN dalam mendeteksi wajah	45
Tabel 4. 2 Hasil Pengujian <i>Face recognition</i> Offline	50
Tabel 4. 3 Hasil Pengujian <i>Face recognition</i> Online (Realtime).....	52
Tabel 4. 4 Hasil Pengujian Gerak Servo	54
Tabel 4. 5 Waktu Proses Embedding dari 9 sampel (270).....	58
Tabel 4. 6 Perbandingan Akurasi Penambahan Jumlah Dataset.....	63

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu komponen kecerdasan buatan adalah pengenalan wajah (Deng et al., 2022). Hanya dengan melihat mata manusia dan memperhatikan wajah seseorang, individu dapat mengenali satu sama lain, kemudian informasi tersebut disimpan dalam memori otak. Dalam bidang teknologi, pengenalan wajah telah banyak digunakan untuk keperluan komunikasi, sistem keamanan, verifikasi identitas, serta sistem absensi (Dwijayanti et al., 2022).

Pengenalan wajah (*face recognition*) merupakan proses identifikasi atau verifikasi individu berdasarkan karakteristik unik pada wajah seseorang (Wang, 2024). Sistem ini bekerja dengan melakukan proses deteksi wajah, ekstraksi fitur, hingga penyimpanan representasi wajah ke dalam basis data. Pada saat sistem melakukan pemindaian ulang menggunakan kamera, fitur wajah yang terdeteksi akan dibandingkan dengan data yang telah tersimpan di dalam database. Selanjutnya, sistem akan menentukan apakah wajah tersebut dikenali atau tidak berdasarkan tingkat kemiripan tertentu (Dwijayanti et al., 2022).

Computer vision merupakan salah satu bidang teknologi yang berkembang pesat dan menarik perhatian para peneliti saat ini (Tekkök et al., 2021). Teknologi ini memungkinkan komputer untuk melihat, mendeteksi, serta memproses citra atau video menyerupai cara kerja sistem visual manusia, kemudian menghasilkan keluaran yang sesuai dengan masukan yang diberikan (Wang et al., 2018). Teknologi ini dapat dimanfaatkan untuk menghitung, mengawasi, melacak, hingga mendeteksi objek tertentu dalam suatu citra. Berbagai metode telah dikembangkan dalam sistem pengenalan wajah, antara lain Viola–Jones, Local Binary Pattern Histogram (LBPH) (Qi, 2023), Eigenfaces (Aditya et al., 2022), dan Fisherfaces (Hamad & Kaya, 2022).

Seiring perkembangan teknologi, metode berbasis *Deep learning* seperti *Convolutional Neural Networks* (CNN) menunjukkan performa yang jauh lebih

tinggi dalam pengenalan wajah dibandingkan metode klasik. CNN mampu secara otomatis mempelajari representasi fitur wajah yang kompleks tanpa perlu perancangan fitur secara manual. Berbeda dengan metode klasik seperti Eigenfaces dan Fisherfaces yang sensitif terhadap variasi pencahayaan dan ekspresi wajah, pendekatan berbasis CNN memiliki ketahanan yang lebih baik terhadap perubahan iluminasi, pose, serta variasi kondisi lingkungan (Shafique et al., 2021; Li et al., 2021). Hal ini menjadikan metode *Deep learning* sebagai solusi yang lebih akurat dan adaptif dalam sistem pengenalan wajah modern.

Dalam penelitian ini, metode pengenalan wajah yang digunakan adalah pendekatan *Deep Metric Learning* berbasis *Convolutional Neural Network* (CNN) dengan arsitektur ResNet. Model ini menghasilkan representasi numerik berupa vektor fitur 128-dimensi (*face embedding*) yang merepresentasikan karakteristik unik setiap wajah (Zheng et al., 2021). Proses identifikasi dilakukan dengan menghitung jarak kemiripan antar vektor fitur menggunakan metode *Euclidean Distance*. Semakin kecil jarak antar vektor, maka semakin tinggi tingkat kemiripan antara dua wajah yang dibandingkan (Zhang et al., 2021).

Sistem akan melakukan proses deteksi wajah menggunakan metode *Histogram of Oriented Gradients* (HOG), kemudian dilanjutkan dengan ekstraksi fitur menggunakan CNN-ResNet untuk menghasilkan embedding wajah. Pada tahap pengenalan, sistem akan membandingkan embedding wajah yang diperoleh dengan database menggunakan perhitungan Euclidean Distance untuk menentukan identitas pengguna.

Output atau keluaran dari sistem ini berupa aksi robot resepsionis, seperti memberikan salam secara otomatis, menyebutkan nama pengguna yang dikenali, atau menyampaikan informasi yang dibutuhkan oleh tamu sesuai dengan hasil identifikasi wajah.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah bagaimana merancang dan mengimplementasikan sistem pengenalan wajah berbasis *Deep learning* menggunakan metode

Convolutional Neural Network (CNN) dengan pendekatan Euclidean Distance yang terintegrasi pada Robot Resepsionis, sehingga sistem mampu mengenali wajah seseorang secara akurat.

Selanjutnya, bagaimana sistem yang dibangun tersebut dapat menghubungkan hasil identifikasi wajah dengan aksi Robot Resepsionis, seperti memberikan salam, melakukan proses *check-in*, maupun *check-out* kepada tamu secara otomatis berdasarkan hasil pengenalan wajah.

1.3 Batasan Masalah

Dengan berpedoman pada rumusan masalah yang telah dijabarkan di atas dan untuk lebih memfokuskan pembahasan penelitian, maka penulis memberikan batasan masalah sebagai berikut:

1. Robot Resepsionis memanfaatkan metode *Deep learning* berbasis *Convolutional Neural Network* (CNN) dengan arsitektur ResNet untuk proses pengenalan identitas wajah, menggunakan pendekatan Euclidean Distance sebagai metode pencocokan data wajah.
2. Komputer (PC/Laptop) digunakan sebagai perangkat utama dalam sistem, yang berperan untuk melakukan seluruh proses pengolahan citra (*image processing*) dan pengenalan wajah.
3. Mikrokontroler yang dimanfaatkan dalam sistem adalah Arduino Mega 2560 sebagai pengendali perangkat keras Robot Resepsionis.
4. Sistem hanya melakukan pelatihan dan pengenalan berdasarkan citra wajah secara keseluruhan (*frontal face*), tanpa menganalisis bagian wajah secara terpisah seperti mata atau mulut.
5. Tahap deteksi wajah menggunakan metode *Histogram of Oriented Gradients* (HOG) sebelum dilakukan ekstraksi fitur menggunakan CNN.
6. Pemrograman sistem dilakukan menggunakan bahasa pemrograman Python dengan dukungan library *computer vision* dan *face recognition*.
7. Robot Resepsionis mendeteksi dan mengenali identitas tamu berdasarkan database yang telah tersimpan, kemudian melakukan respons berupa pemberian

salam, proses *check-in/check-out*, serta keluaran suara (audio) sesuai hasil identifikasi wajah.

1.4 Tujuan dan Manfaat

Tujuan dari riset ini antara lain:

1. Membangun dan mengimplementasikan sistem deteksi wajah menggunakan metode *Histogram of Oriented Gradients* (HOG) pada Robot Resepsionis.
2. Mengimplementasikan metode pengenalan wajah berbasis *Deep learning* menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur ResNet serta pendekatan Euclidean Distance untuk mengenali identitas pemilik wajah.
3. Mengintegrasikan sistem pengenalan wajah dengan Robot Resepsionis sehingga mampu berinteraksi secara otomatis dengan tamu berdasarkan hasil identifikasi wajah.

Berikut kegunaan yang diperoleh dari riset ini antara lain:

1. Mewujudkan interaksi yang lebih cerdas dan personal antara manusia dengan Robot Resepsionis melalui teknologi pengenalan wajah berbasis *Deep learning*.
2. Mengimplementasikan metode pengenalan wajah berbasis CNN pada sistem Robot Resepsionis untuk meningkatkan akurasi identifikasi tamu serta mendukung proses otomatis seperti pemberian salam, *check-in*, dan *check-out*.

1.5 Ruang Lingkup Penelitian

Ruang lingkup penelitian ini mencakup penerapan metode *Histogram of Oriented Gradients* (HOG) untuk deteksi wajah dan metode pengenalan wajah berbasis *Deep learning* menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur ResNet serta pendekatan Euclidean Distance untuk proses identifikasi. Sistem dikembangkan dan dijalankan pada komputer sebagai pusat pemrosesan utama. Penelitian ini juga mencakup pengembangan sistem interaksi antara tamu dan Robot Resepsionis, termasuk respons seperti pemberian salam, proses *check-in*, *check-out*, dan penyampaian informasi kepada tamu berdasarkan

hasil identifikasi wajah. Seluruh pengembangan sistem dilakukan menggunakan bahasa pemrograman Python, dengan pembatasan pada fungsi pengenalan wajah frontal (tampak depan) serta fokus utama pada integrasi sistem pengenalan wajah dengan mekanisme respons otomatis Robot Resepsionis.

1.6 Susunan dan Struktur Penulisan

Untuk memudahkan pembaca dalam memahami isi setiap bab, penulis menyusun Susunan dan Struktur Penulisan sebagai bagian dari penyusunan laporan penelitian ini. Format laporan riset ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, serta sistematika penulisan.

BAB II. KAJIAN PUSTAKA

Bab ini berisi kerangka teori serta penjelasan yang diperlukan mengenai sistem pengenalan wajah berbasis *Computer vision*, metode *Histogram of Oriented Gradients* (HOG) untuk deteksi wajah, serta metode *Convolutional Neural Network* (CNN) dengan arsitektur ResNet dan pendekatan Euclidean Distance untuk proses identifikasi wajah.

BAB III. METODOLOGI PENELITIAN

Bab ini berisi tahapan perancangan dan pengembangan sistem pengenalan wajah yang terintegrasi dengan Robot Resepsionis berbasis komputer, mulai dari perancangan sistem, proses pengambilan data, pelatihan model, hingga implementasi sistem secara keseluruhan.

BAB IV. HASIL DAN ANALISA

Bab ini berisi hasil pengujian sistem yang telah dikembangkan, meliputi hasil pengujian akurasi pengenalan wajah, kecepatan proses, serta hasil analisis kinerja sistem dalam mendukung interaksi Robot Resepsionis dengan tamu.

BAB V. PENUTUP

Bab ini berisi kesimpulan yang diperoleh berdasarkan hasil pengujian dan analisa, serta saran untuk pengembangan sistem lebih lanjut agar dapat meningkatkan performa dan fungsionalitas di masa yang akan datang.



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai deteksi dan pengenalan wajah telah banyak dilakukan dengan berbagai, pada penelitian ini penulis mengambil 5 referensi penelitian terdahulu yang dapat dilihat pada Table 2.1. Metode Kunang et al. (2018) menggunakan metode *Template matching*, *Skin segmentation*, dan *Skin Region* untuk mendeteksi pergerakan wajah melalui pencocokan template dan segmentasi warna kulit guna menentukan area wajah (*Region of Interest*). Sementara itu, Khan dan Sohail (2020) mengembangkan sistem pengenalan wajah berbasis *Convolutional Neural Network* (CNN) dengan model pre-trained yang mampu memberikan tingkat akurasi tinggi pada aplikasi real-time. Penelitian lain oleh Ghiassi dan Faez (2020) memanfaatkan metode *Haar Cascades* untuk deteksi wajah secara cepat dalam sistem real-time. Selain itu, Shafique et al. (2021) memanfaatkan teknologi *deep learning* dengan bantuan OpenCV untuk meningkatkan akurasi deteksi objek secara waktu nyata, sedangkan Zarkasi et al. (2022) menggabungkan metode Haar Cascade dengan algoritma *Weightless Neural Network* (WNN) untuk proses pengenalan wajah berbasis pola biner.

Berdasarkan beberapa penelitian terdahulu tersebut, terdapat beberapa keterbatasan yang masih ditemukan dalam pengembangan sistem deteksi dan pengenalan wajah. Metode seperti *Template matching* dan *Skin segmentation* masih sangat bergantung pada karakteristik warna kulit dan kondisi pencahayaan sehingga kurang stabil dalam berbagai kondisi lingkungan. Sementara itu, metode *Haar Cascade* memang mampu melakukan deteksi wajah secara cepat, namun memiliki keterbatasan dalam menangani variasi pose wajah dan tingkat kompleksitas citra yang lebih tinggi.

Selain itu, beberapa penelitian yang menggunakan metode *Convolutional Neural Network* (CNN) lebih banyak difokuskan pada proses pengenalan wajah, namun belum banyak yang mengintegrasikan sistem tersebut ke dalam perangkat robot resepsionis yang dapat berinteraksi secara langsung dengan pengguna.

Sebagian penelitian juga belum mengoptimalkan kombinasi antara metode deteksi wajah yang efisien dengan metode pengenalan wajah berbasis *deep learning*. Berdasarkan gap penelitian tersebut, penelitian ini mengusulkan pengembangan sistem pengenalan wajah terintegrasi pada robot resepsionis dengan memanfaatkan pendekatan *deep learning* berbasis *Convolutional Neural Network* (CNN) untuk proses pengenalan wajah. Berbeda dengan penelitian sebelumnya yang banyak menggunakan metode *Haar Cascade* atau segmentasi warna kulit, penelitian ini menggunakan metode *Histogram of Oriented Gradients* (HOG) untuk proses deteksi wajah.

Metode HOG dipilih karena mampu mengekstraksi fitur bentuk dan struktur wajah secara lebih robust terhadap variasi pencahayaan serta perubahan posisi wajah. Selanjutnya, fitur wajah yang telah terdeteksi akan diproses menggunakan CNN untuk melakukan proses klasifikasi dan pengenalan wajah berdasarkan database yang telah tersedia. Kebaruan utama dari penelitian ini terletak pada integrasi metode deteksi wajah berbasis HOG dengan sistem pengenalan wajah berbasis CNN yang diimplementasikan pada robot resepsionis, sehingga sistem tidak hanya mampu mendeteksi dan mengenali wajah secara *real-time*, tetapi juga dapat digunakan sebagai media interaksi otomatis antara robot dan pengguna.

Tabel 2.1. Perbandingan penelitian terdahulu

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Kebaruan Pengembangan
Kunang et al., 2018	<i>Face Movement Detection Using Template matching</i>	<i>Template matching, Skin segmentation and Skin Region</i>	Penelitian ini membahas deteksi gerakan wajah melalui pencocokan template, segmentasi warna kulit, dan deteksi area ROI untuk identifikasi wajah dalam berbagai posisi dan sudut.	Metode <i>Skin segmentation</i> dan <i>Skin Region</i> digunakan untuk mendeteksi wajah di saat wajah dalam keadaan miring kiri/ kanan

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Kebaruan Pengembangan
Khan, A. R., & Sohail, A. 2020	<i>A deep Convolutional Neural Network-based face recognition system with a pre-trained model for real-time applications. Multimedia Tools and Applications</i>	<i>Convolutional Neural Networks (CNNs)</i>	Mengembangkan sistem pengenalan wajah berbasis CNN yang mendukung aplikasi real-time. Mereka berhasil merancang model yang dapat mengenali wajah dengan tingkat akurasi yang tinggi dan dapat diimplementasikan secara efisien dalam aplikasi nyata.	Untuk mendeteksi wajah seseorang dengan kondisi jika wajah yang tercapture di kamera lebih dari 1 dimana wajah yang berada di posisi yang paling depan dan dominan mengadopsi metode CNN dan Haar Scades.
Ghiassi, R., & Faez, K. 2020	<i>Real-time face detection using Haar Cascades. In 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)</i>	<i>Haar Cascades</i>	Berhasil mengembangkan sistem deteksi wajah real-time menggunakan metode <i>Haar Cascades</i> . Mereka menunjukkan bahwa metode ini mampu mendeteksi wajah dengan tingkat akurasi yang memadai dalam waktu yang cepat, membuatnya cocok untuk aplikasi yang memerlukan deteksi wajah real-time.	
Shafique, A., Javed, M. M., &	<i>Real-Time Object Detection using Deep learning</i>	OpenCV	Mengembangkan sistem deteksi objek real-time menggunakan teknik <i>deep</i>	OpenCV dan WNN di adopsi sebagai referensi metode

Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian	Kebaruan Pengembangan
Hussain, M. 2021	<i>Techniques and OpenCV. Journal of Robotics and Control</i>		<i>learning</i> dan OpenCV. Mereka menunjukkan bahwa dengan memanfaatkan teknologi OpenCV, sistem dapat mendeteksi objek dengan tingkat akurasi yang tinggi dalam waktu nyata	yang akan digunakan dalam mengenali wajah dan merecognition wajah tersebut berdasar kan database yang ada
Zarkasi et al., 2022	Weightless Neural Networks Face recognition Learning Process for Binary Facial Pattern	WNN	Penelitian menggabungkan metode <i>Haar Cascade</i> dan <i>Weightless Neural Network Face recognition Algorithm</i> (WNN-FRA) untuk deteksi wajah. Analisis termasuk parameter jarak, koordinat, dan kebutuhan memori.	

2.2 Landasan Teori

Studi literatur dalam penelitian ini mencakup berbagai konsep dan teknologi yang berkaitan dengan sistem pengenalan wajah berbasis *Computer vision* dan *Deep learning*. Pengenalan wajah merupakan salah satu cabang kecerdasan buatan yang banyak diterapkan dalam berbagai bidang, seperti keamanan, sistem otentikasi, absensi, dan interaksi manusia dengan mesin. Dalam penelitian ini, proses deteksi wajah memanfaatkan metode *Histogram of Oriented Gradients* (HOG), yang dikenal efektif dalam mendeteksi pola tepi (*edge*) dan struktur wajah berdasarkan distribusi orientasi gradien citra.

Untuk tahap pengenalan wajah, digunakan pendekatan *Deep learning* berbasis *Convolutional Neural Network* (CNN) dengan arsitektur ResNet yang mampu mengekstraksi fitur wajah dalam bentuk *face embedding* berdimensi tinggi. Pemilihan arsitektur ResNet didasarkan pada beberapa keunggulan signifikan dibanding arsitektur CNN konvensional lainnya. ResNet (*Residual Network*) memperkenalkan mekanisme *residual connection* yang mengatasi permasalahan *vanishing gradient* yang sering muncul pada jaringan yang sangat dalam. Dengan koneksi residu ini, gradien dapat mengalir lebih stabil saat proses pelatihan,

sehingga model dapat dibangun lebih dalam tanpa mengorbankan performa. Hal ini memungkinkan ResNet untuk mengekstraksi fitur wajah yang lebih kompleks dan representatif, serta menghasilkan embedding yang lebih robust terhadap variasi pencahayaan, pose, dan ekspresi wajah — kondisi yang umum ditemui pada aplikasi nyata.

Beberapa penelitian terbaru mendukung keunggulan ResNet dalam tugas pengenalan wajah dan feature extraction. Misalnya, Pratama dan Ningrum (2025) menunjukkan bahwa arsitektur ResNet-based CNN memberikan akurasi yang lebih tinggi dan stabil dibandingkan model CNN tradisional pada dataset wajah dengan variasi yang besar. Selain itu, penelitian oleh Jing et al. (2022) dan Durga (2022) menemukan bahwa penggunaan *residual blocks* secara signifikan meningkatkan kemampuan jaringan dalam menangkap pola non-linier serta detail halus pada citra wajah, sehingga meningkatkan kualitas vektor embedding yang dihasilkan.

Dengan demikian, penggunaan ResNet dalam penelitian ini bukan hanya mengikuti praktik umum dalam *face recognition*, tetapi juga didukung oleh bukti empiris dalam literatur bahwa arsitektur ini mampu memberikan generalisasi dan performa yang lebih baik dibandingkan arsitektur CNN klasik dalam konteks ekstraksi fitur wajah.

Representasi fitur ini kemudian dibandingkan menggunakan metode Euclidean Distance untuk menentukan tingkat kemiripan antara wajah yang terdeteksi dengan data wajah yang tersimpan dalam basis data. Pendekatan ini dikenal memiliki tingkat akurasi yang tinggi serta ketahanan yang lebih baik terhadap variasi pencahayaan, ekspresi, dan sudut wajah dibandingkan metode konvensional.

Dalam implementasinya, sistem dijalankan pada perangkat komputer sebagai pusat pemrosesan utama untuk menangani proses deteksi dan pengenalan wajah secara *real-time*. Bahasa pemrograman Python digunakan sebagai bahasa utama dalam pengembangan sistem karena mendukung berbagai pustaka *computer vision* dan *deep learning* yang mempermudah proses implementasi, integrasi sistem, serta pengembangan interaksi antara sistem pengenalan wajah dan Robot Resepsionis (Al-Sarayreh et al., 2022). Studi literatur ini menjadi landasan teoretis

dan praktis dalam merancang sistem yang terintegrasi, akurat, dan responsif terhadap interaksi dengan tamu.

2.2.1 Pengolahan Citra Digital

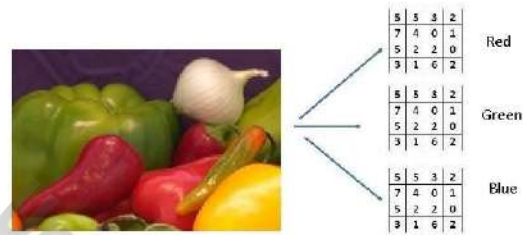
Pengolahan citra digital atau *Image digital processing* ialah pengetahuan mengenai teknik dalam mengolah *Image*. Citra ialah foto atau gambar yang ditangkap oleh kamera. Digital ialah proses olah citra dilaksanakan memanfaatkan komputer. Yang berarti pengolahan citra menjadi citra yang lain agar menjadi lebih maksimal agar sesuai yang diinginkan.

1. Citra warna atau RGB (*Red, Green, Blue*)

Citra warna dalam pengolahan citra atau *image processing* direpresentasikan menggunakan model warna RGB (*Red, Green, Blue*). Model warna RGB adalah suatu sistem di mana warna dihasilkan dengan mencampurkan intensitas tiga warna dasar, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*). Setiap piksel dalam citra RGB memiliki nilai intensitas untuk ketiga warna tersebut (Deng et al., 2021), dan kombinasi nilai-nilai ini menghasilkan spektrum warna yang luas.

Dalam citra RGB, warna-warna lain dapat dihasilkan dengan mengkombinasikan intensitas warna dasar dalam proporsi tertentu. Sebagai contoh, jika intensitas merah dan hijau diaktifkan dengan intensitas maksimum, namun intensitas biru dimatikan, hasilnya adalah warna kuning. Begitu pula dengan kombinasi lainnya, seperti campuran merah dan biru menghasilkan warna ungu, dan campuran hijau dan biru menghasilkan warna cyan. Representasi citra warna dalam format RGB memungkinkan pengolahan citra dengan berbagai teknik, seperti penyesuaian warna, deteksi objek berwarna, segmentasi warna, dan banyak lagi.

Dalam format ini, setiap piksel dianggap sebagai titik dalam ruang warna tiga dimensi, dengan sumbu x, y, dan z mewakili intensitas merah, hijau, dan biru, secara berturut-turut. Dengan menggunakan citra warna RGB, kita dapat melakukan berbagai manipulasi dan analisis terhadap warna dalam suatu gambar.. RGB mempunyai *range* 0 – 255, maka ditotal menjadi $255^3 = 16.581.375$

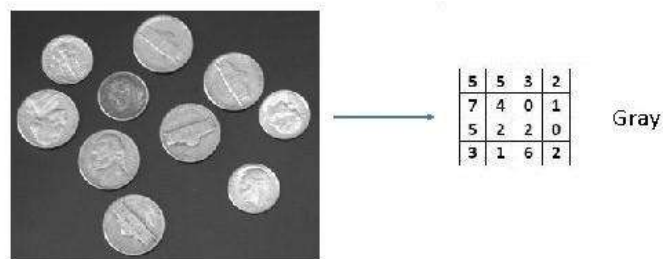


Gambar 2.1 *Color Image*

3. Citra Keabuan (*Grayscale*)

Citra keabuan (*grayscale*) adalah jenis citra di mana setiap pikselnya direpresentasikan hanya oleh satu tingkat kecerahan, tanpa informasi warna. Sebagai lawan dari citra warna yang menggunakan tiga saluran warna (merah, hijau, dan biru), citra keabuan hanya menggunakan satu saluran kecerahan. Warna pada citra keabuan disajikan dalam skala abu-abu, di mana tingkat kecerahan piksel ditentukan oleh intensitasnya. Proses konversi dari citra warna RGB ke citra keabuan melibatkan penghitungan nilai kecerahan yang diperoleh dari kombinasi intensitas merah, hijau, dan biru pada setiap piksel [18]. Salah satu metode sederhana untuk mengonversi citra warna ke citra keabuan adalah dengan menggunakan formula rata-rata tertimbang:

$$Grayscale = 0.299R + 0.587G + 0.114B \quad (2.1)$$



Gambar 2.2 *Grayscale*

Formula ini menggunakan bobot yang didasarkan pada kontribusi relatif dari setiap saluran warna terhadap persepsi manusia terhadap warna. Hasilnya adalah citra keabuan di mana informasi warna telah direduksi, dan setiap piksel hanya memiliki nilai kecerahan yang mencerminkan tingkat keabu-abuan. Keuntungan dari citra keabuan termasuk keefektifan dalam penyimpanan dan

pengolahan data, karena hanya memerlukan satu saluran warna daripada tiga. Citra keabuan juga sering digunakan dalam beberapa aplikasi pengolahan citra, seperti deteksi tepi, segmentasi, dan analisis pola, di mana warna mungkin tidak terlalu relevan.

2.4 Face Detection

Face detection adalah teknik dalam pengolahan citra digital yang berfokus pada proses identifikasi dan lokalisasi wajah manusia dalam suatu gambar atau video secara otomatis. Tahap ini merupakan langkah awal dalam sistem pengenalan wajah, karena sistem harus terlebih dahulu mengetahui posisi wajah sebelum melakukan proses identifikasi individu.

Salah satu metode yang banyak digunakan dalam face detection adalah *Histogram of Oriented Gradients* (HOG). Metode HOG bekerja dengan menganalisis distribusi orientasi gradien pada setiap blok piksel dalam citra. Proses ini dilakukan dengan mengekstraksi fitur tepi (*edge*) dan tekstur lokal wajah, kemudian membandingkannya dengan model deteksi yang telah dilatih sebelumnya. HOG dikenal memiliki efisiensi komputasi yang baik serta mampu bekerja secara real-time pada sistem berbasis komputer tanpa memerlukan perangkat pemrosesan khusus.

Selain *Histogram of Oriented Gradients* (HOG), metode lain yang digunakan dalam *face detection* adalah pendekatan berbasis *Convolutional Neural Network* (CNN), yang mampu secara otomatis mempelajari fitur wajah dari data pelatihan dalam jumlah besar. Pendekatan CNN umumnya memiliki tingkat akurasi yang lebih tinggi dibandingkan metode konvensional, namun membutuhkan sumber daya komputasi yang lebih besar (Liu et al., 2021).

Face detection memiliki berbagai aplikasi, seperti dalam sistem keamanan untuk pengawasan, sistem absensi otomatis, fotografi digital untuk fokus otomatis, serta sistem interaksi manusia dan mesin. Penting untuk membedakan antara *face detection* dan *face recognition*. Face detection hanya menentukan keberadaan dan lokasi wajah dalam citra, sedangkan *face recognition* merupakan proses lanjutan untuk mengidentifikasi atau mengenali individu berdasarkan karakteristik wajah

yang telah diekstraksi. Dalam penelitian ini, metode HOG digunakan untuk mendeteksi keberadaan dan posisi wajah tamu melalui kamera yang terhubung ke sistem komputer. Hasil deteksi berupa koordinat lokasi wajah selanjutnya diproses pada tahap ekstraksi fitur dan pengenalan wajah untuk mendukung interaksi otomatis pada Robot Resepsionis.

2.5 *Histogram of Oriented Gradients (HOG)*

Histogram of Oriented Gradients (HOG) merupakan metode ekstraksi fitur yang digunakan dalam deteksi objek, termasuk deteksi wajah. Metode ini pertama kali diperkenalkan oleh Dalal dan Triggs dan dikenal efektif dalam mendeteksi objek berdasarkan pola tepi (*edge*) dan struktur gradien dalam citra. Secara konsep, HOG bekerja dengan menghitung distribusi orientasi gradien intensitas piksel pada suatu gambar. Ide dasarnya adalah bahwa bentuk dan struktur suatu objek, termasuk wajah manusia, dapat direpresentasikan dengan baik melalui pola perubahan intensitas (gradien) dan arah tepi yang muncul pada citra (Gupta et al., 2021).

Tahapan utama dalam metode HOG meliputi:

1. Perhitungan Gradien: Gradien dihitung untuk setiap piksel dalam citra menggunakan operator diferensiasi seperti filter Sobel. Gradien horizontal dan vertikal dihitung sebagai berikut:

$$G_x = I(x + 1, y) - I(x - 1, y) \quad (2.2)$$

$$G_y = I(x, y + 1) - I(x, y - 1) \quad (2.3)$$

Besarnya magnitudo gradien dan arah gradien kemudian dihitung menggunakan persamaan:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.5)$$

Magnitudo menunjukkan kekuatan perubahan intensitas, sedangkan sudut menunjukkan arah perubahan tersebut (Gupta et al., 2021).

2. Pembentukan Histogram Orientasi: Citra dibagi menjadi beberapa sel (*cell*) berukuran kecil, misalnya 8×8 piksel. Untuk setiap sel, dibuat histogram

orientasi berdasarkan nilai sudut gradien. Setiap piksel memberikan kontribusi terhadap histogram sesuai dengan magnitudo gradiennya. Histogram ini biasanya dibagi ke dalam beberapa bin orientasi (misalnya 9 bin dari 0° – 180°).

3. Normalisasi Blok: Beberapa sel digabung menjadi satu blok untuk dilakukan normalisasi guna mengurangi pengaruh variasi pencahayaan dan kontras. Normalisasi blok dilakukan menggunakan metode seperti:

$$v' = \frac{v}{\sqrt{\|v\|^2 + \epsilon}} \quad (2.5)$$

di mana v adalah vektor histogram dan ϵ adalah konstanta kecil untuk menghindari pembagian dengan nol (Zhao et al., 2021).

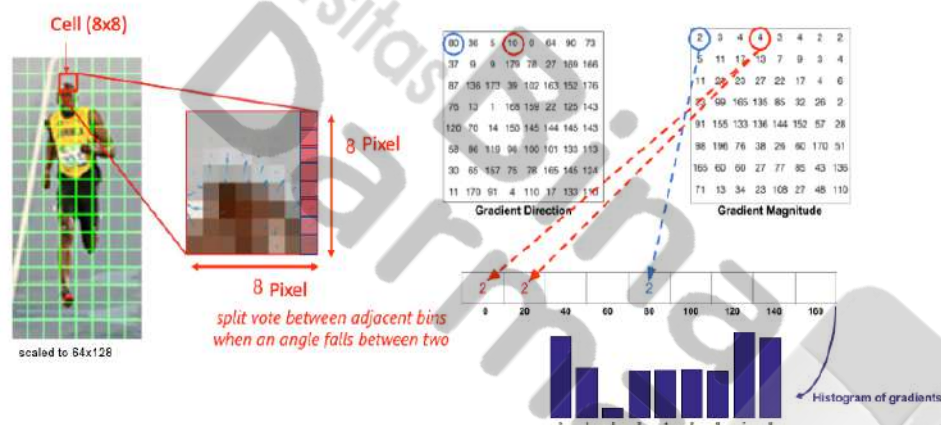
Hasil akhir dari proses HOG adalah vektor fitur yang merepresentasikan pola tepi dan struktur wajah pada citra. Vektor fitur ini kemudian digunakan oleh classifier untuk menentukan apakah suatu area citra mengandung wajah atau tidak. Keunggulan metode HOG adalah kemampuannya dalam mendeteksi objek dengan cukup akurat serta efisien secara komputasi, sehingga cocok digunakan dalam aplikasi real-time berbasis komputer. Selain itu, metode ini lebih tahan terhadap variasi pencahayaan dibandingkan pendekatan berbasis intensitas piksel mentah.

2.5.1 Perhitungan Gradien pada *Histogram of Oriented Gradients* (HOG)

Dalam metode *Histogram of Oriented Gradients* (HOG), tahap awal yang sangat penting adalah proses perhitungan gradien citra. Gradien digunakan untuk mengetahui perubahan intensitas piksel pada arah horizontal dan vertikal, yang merepresentasikan struktur tepi (*edge*) suatu objek di dalam citra. Berbeda dengan pendekatan *Integral Image* yang digunakan pada metode *Haar-Like Feature*, HOG secara langsung menghitung turunan intensitas piksel untuk memperoleh informasi arah dan kekuatan tepi (Hosny et al., 2022).

Perhitungan gradien dilakukan dengan mengoperasikan filter diferensial sederhana pada citra *grayscale*. Komponen gradien horizontal (G_x) dan vertikal (G_y) pada persamaan (2.2) dan (2.3), di mana $I(x,y)$ menyatakan nilai intensitas

piksel pada koordinat (x,y). Berdasarkan nilai G_x dan G_y , maka dapat dihitung magnitudo gradien dan arah orientasi gradien dengan persamaan (2.4) Magnitudo gradien menunjukkan seberapa kuat perubahan intensitas pada suatu piksel, sedangkan sudut gradien menunjukkan arah perubahan tersebut. Informasi ini kemudian digunakan untuk membentuk histogram orientasi pada setiap sel (cell) dalam citra.



Gambar 2.3 Perhitungan Gradien pada Citra untuk Metode HOG
(Hamza, M. 2024)

Keunggulan pendekatan gradien dalam metode HOG adalah kemampuannya merepresentasikan bentuk dan kontur objek secara lebih stabil terhadap perubahan pencahayaan global. Karena yang dianalisis adalah perubahan intensitas (bukan nilai absolut piksel), metode ini lebih robust terhadap variasi kondisi pencahayaan dibandingkan pendekatan berbasis integral image.

2.5.2 Cascade Classifier

Convolutional Neural Network (CNN) merupakan salah satu metode *Deep learning* yang banyak digunakan dalam bidang *Computer vision*, termasuk pada sistem pengenalan wajah. CNN dirancang untuk secara otomatis mengekstraksi fitur penting dari citra melalui proses konvolusi berlapis (*deep layered convolution*)

tanpa memerlukan perancangan fitur secara manual seperti pada metode konvensional (Kumar, 2021). Beberapa lapisan utama yang dimiliki CNN dapat dilihat pada gambar 2.4, yaitu:

1. Convolutional Layer

Lapisan ini berfungsi untuk mengekstraksi fitur dari citra dengan menggunakan kernel atau filter yang digeser (convolution) pada seluruh bagian citra. Proses ini menghasilkan feature map yang merepresentasikan pola-pola penting seperti tepi, tekstur, dan bentuk wajah.

2. Activation Function (ReLU)

Fungsi aktivasi seperti Rectified Linear Unit (ReLU) digunakan untuk memperkenalkan sifat non-linear pada jaringan, sehingga model mampu mempelajari pola yang lebih kompleks.

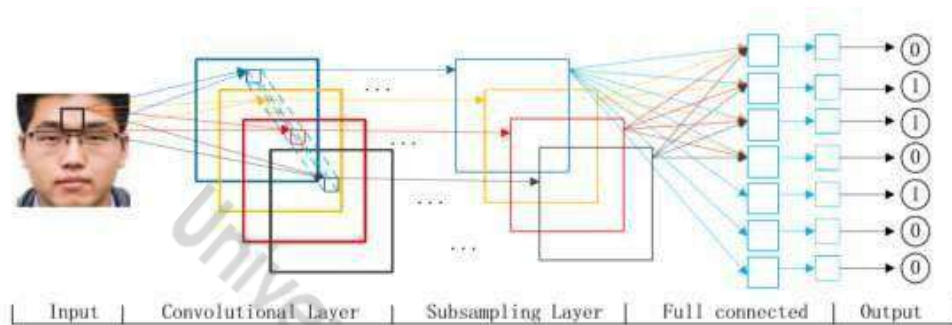
3. Pooling Layer

Lapisan pooling digunakan untuk mereduksi dimensi feature map sehingga mengurangi kompleksitas komputasi dan membantu mencegah overfitting.

4. Fully Connected Layer

Pada tahap akhir, fitur-fitur yang telah diekstraksi akan diubah menjadi vektor numerik (embedding) yang merepresentasikan karakteristik unik wajah seseorang.

Dalam sistem pengenalan wajah, CNN khususnya arsitektur ResNet digunakan untuk menghasilkan representasi numerik wajah yang disebut *face embedding*. Embedding ini memiliki dimensi tertentu (misalnya 128 dimensi) dan digunakan untuk membandingkan kemiripan antar wajah menggunakan metode perhitungan jarak seperti Euclidean Distance.



Gambar 2.4 Arsitektur Dasar *Convolutional Neural Network* (CNN)

(Yuan, dkk, 2017)

Keunggulan CNN dibandingkan metode konvensional seperti *Haar Cascade* adalah kemampuannya dalam mempelajari fitur wajah secara otomatis dan lebih robust terhadap variasi pencahayaan, pose, serta ekspresi wajah. Walaupun memerlukan sumber daya komputasi yang lebih besar, CNN memberikan tingkat akurasi yang jauh lebih tinggi dalam proses pengenalan wajah.

2.6 Pengenalan Wajah

Pengenalan wajah merupakan proses identifikasi atau verifikasi identitas seseorang dengan cara membandingkan citra wajah masukan dengan basis data wajah yang telah tersimpan. Teknologi ini termasuk dalam sistem biometrik karena memanfaatkan karakteristik unik pada wajah manusia sebagai identitas. Berbeda dengan metode konvensional yang mengekstraksi fitur secara manual, sistem pengenalan wajah modern berbasis *Deep learning* mampu mempelajari representasi wajah secara otomatis melalui jaringan saraf tiruan (Tyagi, 2023).

Dalam sistem ini, wajah manusia direpresentasikan dalam bentuk vektor numerik yang disebut *face embedding*. Embedding ini merupakan hasil ekstraksi fitur menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur ResNet. Vektor tersebut memiliki dimensi tertentu (misalnya 128 dimensi) dan menyimpan informasi unik mengenai struktur wajah seperti bentuk mata, hidung, kontur wajah, dan proporsi antar fitur.

Secara umum, proses pengenalan wajah terdiri dari beberapa tahapan utama, yaitu:

1. Akuisisi Citra

Sistem menangkap citra wajah menggunakan kamera yang terhubung ke komputer sebagai pusat pemrosesan.

2. Deteksi Wajah (Face Detection)

Sistem mendeteksi lokasi wajah dalam citra menggunakan metode *Histogram of Oriented Gradients* (HOG). Hasil deteksi berupa koordinat area wajah.

3. Ekstraksi Fitur (Feature Extraction)

Area wajah yang telah terdeteksi diproses oleh model CNN (ResNet) untuk menghasilkan *face embedding* dalam bentuk vektor numerik.

4. Pembentukan Template

Vektor embedding yang dihasilkan disimpan dalam basis data sebagai representasi identitas individu.

5. Pencocokan (Matching Process)

Embedding wajah baru dibandingkan dengan embedding yang tersimpan di basis data menggunakan metode perhitungan jarak, seperti Euclidean Distance.

6. Verification dan Identification

Verification (1:1) membandingkan satu wajah dengan satu data tertentu untuk memverifikasi identitas dan Identification (1:N) membandingkan satu wajah dengan seluruh data dalam basis data untuk menemukan kecocokan terbaik.

Metode berbasis CNN memiliki keunggulan dibandingkan metode klasik seperti FisherFace atau EigenFace karena lebih robust terhadap variasi pencahayaan, ekspresi, dan pose wajah. Selain itu, sistem ini tidak hanya bergantung pada nilai intensitas piksel, tetapi pada representasi fitur tingkat tinggi yang dipelajari secara otomatis oleh jaringan saraf.

Dalam penelitian ini, sistem pengenalan wajah diintegrasikan dengan Robot Resepsionis. Ketika wajah tamu berhasil dikenali, sistem akan mengaktifkan respons tertentu seperti pemberian salam, proses *check-in*, atau *check-out* berdasarkan data identitas yang tersimpan di basis data. Dengan demikian,

pengenalan wajah menjadi komponen utama dalam mendukung interaksi otomatis antara manusia dan Robot Resepsionis.

2.7 Euclidean Distance

Euclidean Distance merupakan metode perhitungan jarak yang digunakan untuk mengukur tingkat kemiripan antara dua vektor dalam ruang multidimensi. Dalam sistem pengenalan wajah berbasis *Convolutional Neural Network* (CNN), setiap wajah direpresentasikan dalam bentuk vektor numerik yang disebut *face embedding*. Oleh karena itu, proses pencocokan identitas dilakukan dengan menghitung jarak antara dua embedding tersebut.

Semakin kecil nilai jarak Euclidean antara dua vektor wajah, maka semakin tinggi tingkat kemiripan keduanya. Sebaliknya, semakin besar jaraknya, maka wajah tersebut dianggap berbeda.

2.7.1 Konsep Dasar Euclidean Distance

Secara matematis, Euclidean Distance antara dua titik dalam ruang berdimensi n dapat dirumuskan sebagai berikut:

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.6)$$

Keterangan:

- $d(p, q)$ = jarak antara vektor p dan q
- p_i = komponen ke- i dari vektor pertama
- q_i = komponen ke- i dari vektor kedua
- n = jumlah dimensi vektor

Dalam konteks pengenalan wajah:

- Vektor p = embedding wajah yang baru ditangkap kamera
- Vektor q = embedding wajah yang tersimpan dalam basis data
- n = jumlah dimensi embedding (misalnya 128 dimensi)

2.8 Perangkat Komputer

Perangkat ini berfungsi sebagai pusat pemrosesan data (processing unit) dalam sistem pengenalan wajah pada Robot Resepsionis. Berbeda dengan penelitian sebelumnya yang menggunakan Raspberry Pi sebagai mini komputer, pada penelitian ini seluruh proses komputasi dilakukan menggunakan komputer dengan performa yang lebih tinggi. Penggunaan komputer memberikan kemampuan pemrosesan paralel yang lebih cepat dalam menjalankan algoritma *computer vision*, seperti deteksi wajah menggunakan *Haar Cascade* serta proses ekstraksi dan pencocokan fitur wajah.

Kapasitas RAM yang lebih besar juga mendukung pemrosesan data citra secara real-time, terutama saat sistem melakukan deteksi wajah, ekstraksi fitur, dan pencocokan terhadap database secara simultan. Selain itu, penggunaan media penyimpanan seperti SSD memberikan kecepatan baca/tulis data yang lebih tinggi dibandingkan hard disk konvensional, sehingga mempercepat proses loading dataset wajah serta penyimpanan data hasil pengenalan.

2.9 Webcam

Webcam merupakan perangkat input berupa kamera digital yang digunakan untuk menangkap citra atau video secara real-time dan terhubung langsung ke komputer melalui port USB. Dalam penelitian ini, webcam berfungsi sebagai sensor visual utama yang menangkap wajah tamu untuk kemudian diproses oleh sistem pengenalan wajah. Berbeda dengan Pi Camera yang terhubung melalui port CSI pada Raspberry Pi, webcam pada penelitian ini dihubungkan langsung ke komputer melalui antarmuka USB. Penggunaan webcam memberikan fleksibilitas dalam pemasangan serta kompatibilitas yang luas dengan sistem operasi dan library pengolahan citra seperti OpenCV pada Python.

Webcam yang digunakan umumnya memiliki resolusi minimal 720p atau 1080p, yang sudah cukup untuk kebutuhan deteksi dan pengenalan wajah secara real-time. Resolusi ini memungkinkan sistem menangkap detail wajah dengan cukup jelas untuk proses deteksi menggunakan metode *Haar Cascade* serta tahap ekstraksi dan pencocokan fitur.



Gambar 2.5 Webcam

2.10 Arduino Mega 2560

Arduino Mega 2560 adalah salah satu papan pengembangan mikrokontroler yang populer dan kuat dalam keluarga produk Arduino. Papan ini dirancang untuk proyek-proyek yang membutuhkan lebih banyak pin I/O, memori, dan kemampuan pemrosesan dibandingkan dengan papan Arduino lainnya. Berikut adalah beberapa poin penjelasan tentang Arduino Mega 2560:

1. Mikrokontroler ATmega2560

Arduino Mega 2560 menggunakan mikrokontroler Atmel ATmega2560 sebagai otak utama. Mikrokontroler ini memiliki arsitektur RISC 8-bit dan beroperasi pada kecepatan clock hingga 16 MHz.

2. Pin I/O yang Banyak

Salah satu fitur utama Arduino Mega 2560 adalah jumlah pin Input/Output (I/O) yang lebih banyak. Papan ini memiliki 54 pin digital I/O, di antaranya 15 dapat digunakan sebagai output PWM (Pulse Width Modulation), dan 16 pin analog input.

3. Memori yang Luas

Arduino Mega 2560 memiliki memori yang lebih besar dibandingkan dengan model Arduino lainnya. Papan ini memiliki 256 KB flash memory untuk menyimpan program, 8 KB SRAM untuk menyimpan data yang dihasilkan atau diolah program, dan 4 KB EEPROM untuk penyimpanan data yang dapat dipertahankan ketika daya mati.

4. Komunikasi Serial

Papan ini mendukung berbagai protokol komunikasi serial termasuk UART, SPI, dan I2C. Ini membuatnya ideal untuk berbagai proyek yang melibatkan komunikasi dengan perangkat eksternal, sensor, atau modul komunikasi.

5. USB dan Power Supply

Arduino Mega 2560 dapat dihubungkan ke komputer melalui kabel USB dan dijalankan sebagai perangkat USB. Selain itu, papan ini dapat diberi daya melalui jack DC dengan tegangan 7-12V atau melalui pin Vin.

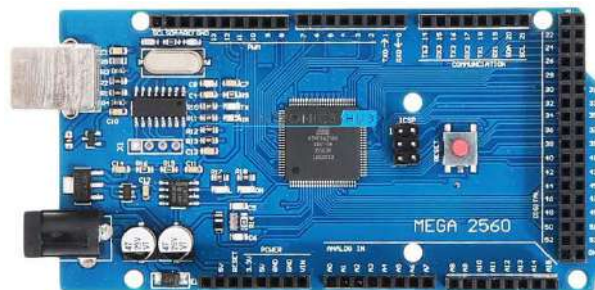
6. Kompatibilitas Perangkat Lunak

Arduino Mega 2560 dapat diprogram menggunakan Arduino IDE, lingkungan pengembangan perangkat lunak yang mudah digunakan dan mendukung pemrograman dalam bahasa Arduino, yang merupakan variasi dari bahasa pemrograman C/C++.

7. Penggunaan dalam Proyek Skala Besar

Karena kapabilitasnya yang tinggi, Arduino Mega 2560 sering digunakan dalam proyek-proyek yang memerlukan banyak pin I/O, penggunaan memori yang besar, dan kemampuan pemrosesan yang tinggi, seperti robotika, kendali peralatan otomatis, atau proyek elektronika yang kompleks.

Arduino Mega 2560 merupakan pilihan yang sangat baik untuk pengembang yang memerlukan lebih banyak sumber daya dalam menjalankan proyek-proyek yang lebih kompleks dan skala besar.



Gambar 2.6 Arduino Mega 2560

2.11 Motor Servo

Motor yang bisa bergerak dua arah ialah motor servo. Motor listrik, rangkaian roda gigi, potensiometer, dan rangkaian kontrol membentuk motor servo.

Di sini potensiometer berfungsi sebagai pembatas sudut putar motor servo. Motor servo sering tidak bergerak secara konstan; mereka hanya berputar pada sudut tertentu. Motor servo tipe SG90 akan dimanfaatkan pada tugas akhir ini. Rentang gerak untuk motor servo ini ialah 0° hingga 180° . Tiga jalur kabel yang membentuk servo ialah kabel daya, *ground*, dan sinyal (Rossi et al., 2021).



Gambar 2.7 Motor Servo

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di SMK Negeri 8 Palembang, khususnya pada Jurusan Teknik Komputer dan Jaringan (TKJ). Pemilihan lokasi penelitian ini didasarkan pada ketersediaan sumber daya yang mendukung proses pengujian sistem robot resepsionis berbasis pengenalan wajah serta kemudahan dalam melakukan pengambilan data penelitian. Lingkungan jurusan TKJ dinilai sesuai untuk implementasi dan pengujian sistem karena memiliki fasilitas pendukung yang berkaitan dengan bidang teknologi komputer dan jaringan.

Waktu pelaksanaan penelitian berlangsung selama enam bulan, yaitu mulai dari bulan Juli hingga Desember 2024, yang dihitung sejak selesainya pelaksanaan ujian proposal penelitian. Selama periode tersebut dilakukan beberapa tahapan penelitian yang meliputi proses perancangan sistem, pengembangan perangkat lunak, pembuatan perangkat keras robot, pengambilan data wajah, serta pengujian dan evaluasi sistem pengenalan wajah yang telah dikembangkan.

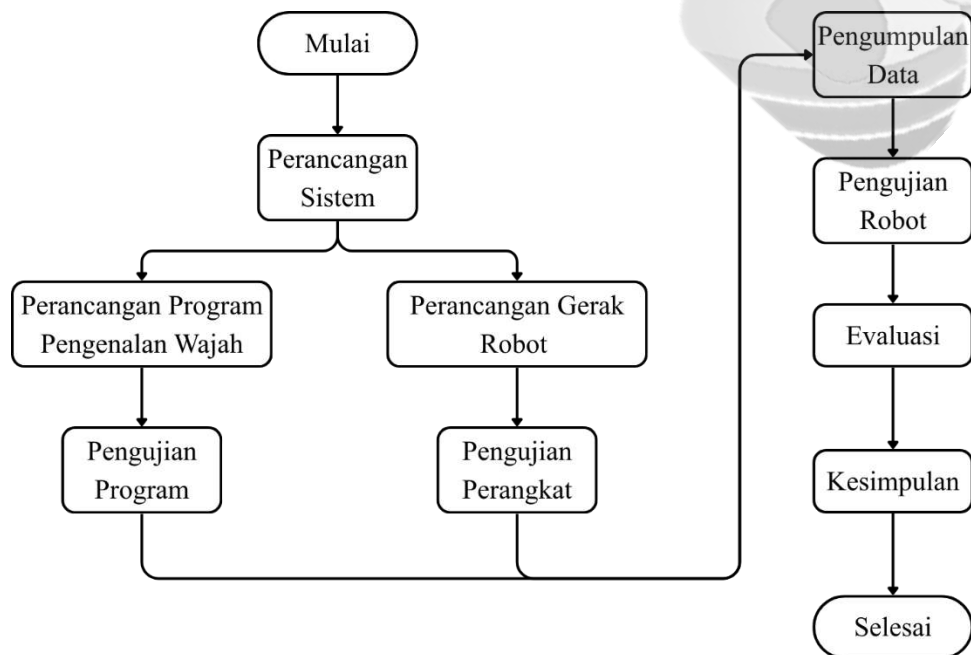
Dalam penelitian ini, sampel data yang digunakan berasal dari guru-guru Jurusan Teknik Komputer dan Jaringan (TKJ) sebanyak 7 orang, yang terdiri dari 4 orang laki-laki dan 3 orang perempuan, serta 2 orang siswa kelas XI TKJ 2 yang terdiri dari 1 siswa laki-laki dan 1 siswi perempuan. Data wajah dari para responden tersebut digunakan sebagai dataset dalam proses pelatihan dan pengujian sistem pengenalan wajah yang diimplementasikan pada robot resepsionis.

Perangkat yang digunakan dalam penelitian ini terdiri dari perangkat lunak dan perangkat keras. Perangkat utama yang digunakan untuk pengembangan sistem pengenalan wajah adalah sebuah laptop dengan spesifikasi Intel Core i7 generasi ke-10, RAM 8 GB, dan penyimpanan SSD, yang berfungsi sebagai pusat pemrosesan data dan pengolahan citra. Selain itu, digunakan sebuah webcam sebagai perangkat akuisisi citra wajah yang terhubung langsung dengan sistem komputer untuk proses deteksi dan pengenalan wajah.

Pada bagian perangkat keras robot, penelitian ini menggunakan mikrokontroler Arduino Mega 2560 sebagai pengendali utama sistem mekanik robot. Struktur fisik robot dibuat menggunakan printer 3D Ender 3 V2 untuk mencetak bagian body robot. Untuk menggerakkan arah kamera agar dapat mengikuti posisi wajah pengunjung, digunakan tiga buah motor servo yang berfungsi mengatur pergerakan kamera secara horizontal dan vertikal sehingga kamera dapat secara otomatis menyesuaikan posisi wajah yang terdeteksi oleh sistem.

3.2 Kerangka Kerja Penelitian

Rencana penelitian ini disusun untuk berlangsung selama satu tahun, dengan fokus pada implementasi sistem pengenalan wajah pada robot resepsionis menggunakan pendekatan berbasis *Deep learning* melalui metode *Histogram of Oriented Gradients* (HOG) untuk deteksi wajah dan *Convolutional Neural Network* (CNN) arsitektur ResNet untuk proses ekstraksi fitur wajah. Tahapan penelitian secara rinci dapat diidentifikasi melalui Gambar 3.1.



Gambar 3.1 Kerangka kerja penelitian

Pada tahap awal, dilakukan perancangan sistem secara menyeluruh untuk menentukan arsitektur umum, alur kerja, serta integrasi antara perangkat lunak

pengenalan wajah dan sistem mekanik robot. Dalam tahap ini ditentukan bagaimana kamera akan menangkap citra wajah pengguna, bagaimana citra tersebut diproses menggunakan model CNN untuk melakukan deteksi dan pengenalan identitas, serta bagaimana hasil pengenalan tersebut diterjemahkan menjadi respons gerakan robot secara real-time.

Setelah arsitektur sistem dirancang, penelitian dilanjutkan ke dua pengembangan utama yang berjalan secara paralel, yaitu perancangan program pengenalan wajah dan perancangan gerak robot. Pada perancangan program pengenalan wajah, tahapan dimulai dari pengumpulan dataset wajah yang akan digunakan sebagai data latih dan data uji. Dataset tersebut kemudian melalui proses preprocessing seperti resizing gambar, normalisasi piksel, dan augmentasi data guna meningkatkan variasi serta ketahanan model terhadap perubahan pencahayaan dan sudut pengambilan gambar. Selanjutnya dirancang arsitektur *Convolutional Neural Network* yang berfungsi untuk melakukan ekstraksi fitur wajah secara otomatis melalui lapisan konvolusi, pooling, dan fully connected layer hingga menghasilkan klasifikasi identitas. Model kemudian dilatih menggunakan data latih, divalidasi untuk menghindari overfitting, serta dilakukan tuning parameter agar diperoleh performa optimal. Setelah proses pelatihan selesai, dilakukan pengujian program untuk mengukur tingkat akurasi, presisi, recall, serta mengamati kemungkinan kesalahan klasifikasi dalam berbagai kondisi.

Secara paralel, dilakukan perancangan gerak robot yang difokuskan pada mekanisme face tracking, yaitu kemampuan robot untuk selalu menghadap ke arah wajah tamu yang terdeteksi. Sistem gerak ini dirancang menggunakan tiga motor servo, terdiri dari dua servo untuk mengatur pergerakan horizontal (sumbu X) dan vertikal (sumbu Y), serta satu servo untuk mengatur rotasi leher. Prinsip kerjanya dimulai dari pengambilan koordinat posisi wajah hasil deteksi pada *frame* kamera. Sistem kemudian menghitung titik tengah wajah dan membandingkannya dengan titik tengah *frame*. Apabila terdapat selisih posisi, sistem akan menghitung nilai error pada sumbu X dan Y. Nilai error tersebut dikonversi menjadi perubahan sudut servo, sehingga servo mata dan servo leher akan bergerak menyesuaikan arah hingga wajah berada di tengah *frame* kamera. Dengan mekanisme ini, robot secara

kontinu melakukan penyesuaian sudut sehingga tercipta kesan bahwa robot sedang melakukan kontak visual dan mengikuti pergerakan pengguna. Dalam perancangannya juga diperhatikan batas sudut servo, kehalusan pergerakan, serta stabilitas respons agar gerakan tidak terlalu sensitif atau bergetar.

Setelah kedua bagian tersebut selesai dikembangkan, dilakukan pengujian program dan pengujian perangkat secara terpisah yang hasilnya dapat dilihat pada BAB IV. Pengujian program bertujuan memastikan bahwa sistem pengenalan wajah berjalan sesuai desain dan memiliki tingkat akurasi yang memadai. Sementara itu, pengujian perangkat dilakukan untuk memastikan bahwa motor servo, sistem kontrol, dan komunikasi data bekerja dengan baik serta mampu merespons perintah pergerakan secara tepat dan real-time.

Tahap berikutnya adalah pengumpulan data, yaitu pengambilan data performa sistem setelah integrasi penuh antara perangkat lunak pengenalan wajah dan mekanisme gerak robot. Data yang dikumpulkan meliputi tingkat akurasi pengenalan wajah dalam kondisi nyata, kecepatan respons sistem, kestabilan tracking wajah, serta kemampuan robot mempertahankan posisi hadap terhadap pengguna. Setelah data terkumpul, dilakukan pengujian robot secara menyeluruh dalam skenario operasional sebagai robot resepsionis. Pada tahap ini diamati bagaimana sistem mendeteksi wajah, mengenali identitas, melakukan tracking, dan memberikan respons secara simultan dalam kondisi real-time.

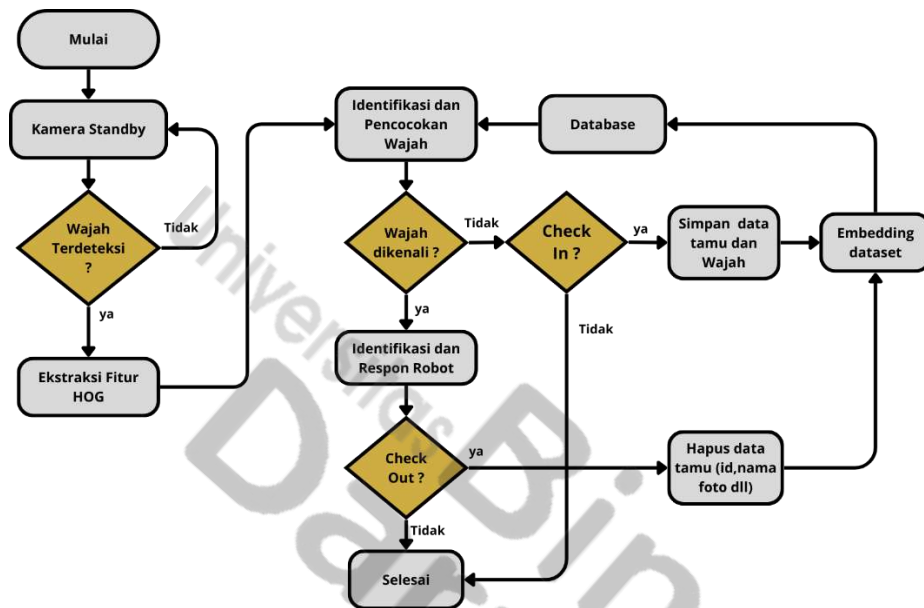
Selanjutnya dilakukan tahap evaluasi untuk menganalisis keseluruhan performa sistem. Evaluasi mencakup analisis akurasi model CNN, efisiensi komputasi, kestabilan pergerakan servo, serta kualitas interaksi antara robot dan pengguna. Dari hasil evaluasi tersebut kemudian dirumuskan kesimpulan penelitian, yang menjelaskan tingkat keberhasilan sistem dalam mencapai tujuan yang telah ditetapkan serta memberikan rekomendasi untuk pengembangan lebih lanjut, seperti peningkatan akurasi model, optimasi kecepatan respons, atau pengembangan fitur interaksi tambahan.

3.2.1 Perancangan Sistem

Alur kerja sistem Robot Resepsionis yang digunakan dalam penelitian ini digambarkan pada Gambar 2.3. Proses dimulai dari tahap Mulai, kemudian sistem mengaktifkan Kamera Standby yang berfungsi untuk memantau lingkungan sekitar secara terus-menerus menggunakan kamera. Setelah kamera aktif, sistem akan melakukan proses deteksi wajah. Pada tahap Wajah Terdeteksi?, sistem akan memeriksa apakah terdapat wajah yang tertangkap oleh kamera. Apabila tidak ada wajah yang terdeteksi, maka sistem akan kembali pada kondisi pemantauan kamera.

Namun jika wajah berhasil terdeteksi, proses dilanjutkan ke tahap Ekstraksi Fitur HOG untuk memperoleh karakteristik penting dari citra wajah yang telah ditangkap oleh kamera. Selanjutnya sistem melakukan Identifikasi dan Pencocokan Wajah dengan membandingkan fitur wajah yang diperoleh dengan data wajah yang terdapat pada Database. Pada tahap Wajah dikenali?, sistem menentukan apakah wajah yang terdeteksi telah terdaftar dalam basis data. Apabila wajah tidak dikenali, sistem akan memeriksa proses Check In? untuk menentukan apakah tamu baru akan didaftarkan. Jika tamu melakukan *check-in*, maka sistem akan Simpan data tamu dan wajah ke dalam database serta menambahkan Embedding dataset sebagai representasi fitur wajah yang akan digunakan pada proses pengenalan selanjutnya.

Sebaliknya, jika wajah berhasil dikenali, maka sistem melanjutkan ke tahap Identifikasi dan Respon Robot, di mana robot akan memberikan respons sesuai dengan identitas tamu yang dikenali. Setelah itu sistem akan memeriksa proses Check Out?. Apabila tamu melakukan *check-out*, maka sistem dapat melakukan Hapus data tamu (ID, nama, foto dan lain-lain) dari database.



Gambar 3.2 Flowchart Sistem Robot Resepsionis

3.2.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini bertujuan untuk membangun sistem pengenalan wajah terintegrasi yang mampu mendeteksi, mengenali, serta memberikan respons otomatis pada Robot Resepsionis. Seluruh proses pemrosesan dilakukan menggunakan komputer sebagai pusat komputasi utama. Sistem perangkat lunak dirancang dalam satu alur kerja terpadu yang meliputi proses perekaman data wajah (recording), pembentukan fitur wajah (encoding/training), pemindaian real-time (scanning), serta pemberian respons (respon sistem).

Proses dimulai dari pengambilan video secara real-time menggunakan webcam yang terhubung ke komputer. Setiap *frame* yang ditangkap akan diproses untuk mendeteksi keberadaan wajah menggunakan metode *Histogram of Oriented Gradients* (HOG) atau *Convolutional Neural Network* (CNN) detector yang tersedia dalam library *face_recognition* (berbasis *dlib*). Metode ini mampu mendeteksi lokasi wajah secara akurat meskipun terjadi variasi pencahayaan dan pose ringan.

Setelah wajah terdeteksi, sistem akan melakukan tahap ekstraksi ciri (feature extraction) menggunakan model *deep learning* berbasis *Convolutional*

Neural Network (CNN). Model ini menghasilkan representasi numerik wajah yang disebut sebagai *face embedding* (vektor berdimensi 128). Vektor ini menjadi identitas unik dari setiap wajah yang disimpan dalam database.

Pada tahap perekaman (recording), wajah pengguna baru akan diambil beberapa kali untuk mendapatkan representasi embedding yang stabil. Data embedding tersebut disimpan bersama informasi identitas seperti ID, nama, dan gender. Tidak seperti metode LBPH yang menyimpan model hasil training, metode ini menyimpan encoding wajah sebagai basis perbandingan langsung.

Saat sistem berada dalam mode scanning, wajah yang tertangkap kamera akan melalui proses berikut:

1. Deteksi lokasi wajah pada *frame*.
2. Ekstraksi embedding wajah menggunakan model CNN.
3. Perbandingan embedding dengan database menggunakan perhitungan jarak Euclidean.
4. Penentuan identitas berdasarkan nilai jarak terkecil dan batas toleransi tertentu.

Jika nilai jarak berada di bawah ambang batas (*tolerance*), maka wajah dianggap dikenali (*recognized*). Jika tidak, maka sistem mengklasifikasikan wajah sebagai tidak dikenal (*unknown*). Selain pengenalan identitas, sistem juga melakukan klasifikasi gender menggunakan model klasifikasi terpisah berbasis *deep learning* (DNN OpenCV atau model pretrained lainnya). Informasi gender digunakan untuk menentukan bentuk sapaan yang sesuai. Respon sistem dibagi menjadi dua kondisi utama:

1. Jika wajah dikenali

Sistem mengambil nama dan gender dari database. Komputer kemudian memutar file audio sapaan dalam Bahasa Inggris, seperti “*Welcome back, Mr. [Nama]*” atau “*Good Morning, Ms. [Nama]*”. Data identitas juga dikirimkan melalui komunikasi serial ke Arduino untuk menggerakkan motor robot agar menghadap tamu.

2. Jika wajah tidak dikenali

Sistem akan memberikan respon suara seperti “*Hello, please register first.*”. Sistem dapat mengarahkan pengguna untuk melakukan proses pendaftaran wajah baru.

Dalam arsitektur ini, komputer berfungsi sebagai pusat pemrosesan citra, ekstraksi fitur *deep learning*, logika keputusan, serta pemrosesan audio. Arduino berfungsi sebagai pengendali aktuator untuk pergerakan robot. Dengan menggunakan metode berbasis *Convolutional Neural Network* (CNN) dan *face embedding*, sistem memiliki tingkat akurasi yang lebih tinggi dibandingkan metode klasik seperti LBPH, terutama dalam menangani variasi pencahayaan, ekspresi, dan sudut wajah.

3.2.2.1 Program wajah

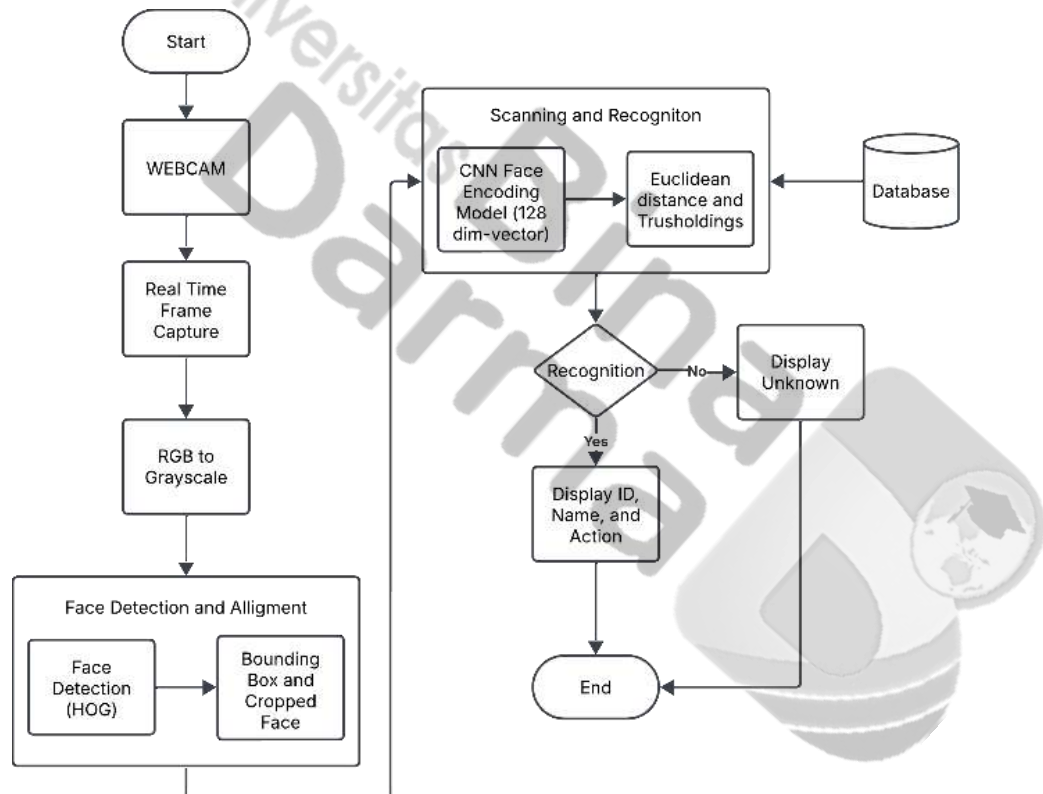
Program pengenalan wajah pada penelitian ini menggunakan pendekatan *Deep Learning* berbasis *Convolutional Neural Network* (CNN) dengan metode *face embedding* untuk proses identifikasi, sebagaimana ditunjukkan pada Gambar 3.3. Berbeda dengan metode tradisional seperti LBPH yang memerlukan proses pelatihan secara manual, sistem ini menggunakan model pretrained yang telah dilatih pada dataset besar sehingga proses yang dilakukan hanya berupa encoding dan pencocokan wajah (*face matching*).

Proses sistem dimulai dengan membuka koneksi kamera (*webcam*) untuk menangkap frame video secara *real-time*. Setiap frame yang diperoleh kemudian dikonversi dari format BGR ke RGB agar sesuai dengan standar input model *deep learning*. Selanjutnya dilakukan deteksi wajah menggunakan metode *Histogram of Oriented Gradients* (HOG) yang dipilih karena memiliki kompleksitas komputasi rendah dan mampu berjalan secara *real-time* pada perangkat berbasis CPU (Saragih et al., 2025).

Pada tahap *recording* dan *encoding*, pengguna memasukkan identitas seperti ID, nama, dan gender. Sistem kemudian mengambil beberapa sampel wajah dari kamera dan langsung mengekstraksi fitur wajah menggunakan model CNN untuk menghasilkan vektor numerik berdimensi 128 (*face embedding*). Vektor tersebut disimpan ke dalam database bersama dengan metadata pengguna.

Pada tahap pengenalan (*recognition*), sistem kembali menangkap frame secara *real-time*. Wajah yang terdeteksi akan diubah menjadi *embedding* dan dibandingkan dengan data yang tersimpan dalam database menggunakan

perhitungan Euclidean Distance. Jika nilai jarak berada di bawah batas toleransi, maka identitas pengguna akan ditampilkan pada layar, sedangkan jika tidak terdapat kecocokan maka sistem akan menampilkan label *Unknown*. Setelah proses selesai, koneksi kamera dan tampilan sistem akan dihentikan.



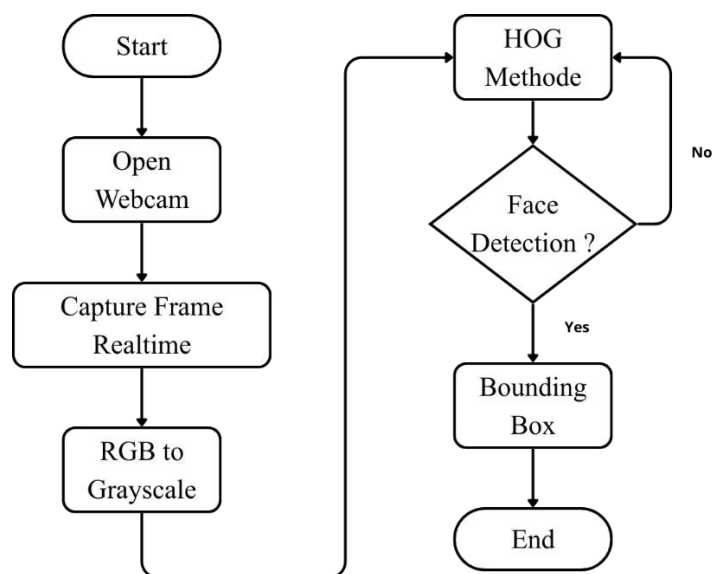
Gambar 3.3 Blok diagram program pengenalan wajah

3.2.2.2 Akusisi dan Deteksi Wajah

Sistem memulai proses dengan membuka koneksi ke perangkat webcam, yang berfungsi sebagai sensor utama untuk menangkap citra wajah pengguna. Setiap dari webcam diambil secara kontinu, memungkinkan sistem memproses data secara real-time tanpa jeda signifikan. Setiap *frame* yang diterima diubah dari format BGR menjadi RGB agar kompatibel dengan model *deep learning* yang digunakan. Proses ini penting karena model deteksi wajah pra-latih biasanya membutuhkan input dalam format RGB untuk akurasi optimal.

Deteksi wajah dilakukan dengan menggunakan metode HOG, Secara prinsip metode HOG bekerja dengan mengekstraksi fitur gradien atau perubahan intensitas cahaya pada citra, kemudian membentuk representasi pola tepi (*edge*) yang khas dari objek wajah. Pola gradien ini digunakan untuk membedakan area wajah dari latar belakang. Hasil dari proses deteksi ini berupa koordinat lokasi wajah dalam bentuk *bounding box* pada setiap *frame*. Area wajah yang terdeteksi kemudian dipotong (*cropping*) dan diteruskan ke tahap berikutnya untuk dilakukan ekstraksi fitur wajah menggunakan model *deep learning* pada proses pengenalan identitas.

Setelah wajah terdeteksi, sistem menentukan *bounding box*, yaitu kotak pembatas yang menandai lokasi wajah di dalam *frame*. Informasi ini digunakan sebagai dasar untuk tahap ekstraksi fitur (*face embedding*), di mana wajah akan dikonversi menjadi representasi numerik yang dapat dibandingkan dengan database. Dengan demikian, tahap akuisisi dan deteksi wajah memastikan bahwa sistem memiliki data wajah yang valid dan berkualitas tinggi, siap diproses lebih lanjut untuk pengenalan identitas dan klasifikasi atribut seperti gender. Tahap ini merupakan fondasi penting bagi seluruh sistem pengenalan wajah, karena akurasi deteksi awal secara langsung memengaruhi keberhasilan proses identifikasi selanjutnya. Untuk jalan proses keseluruhan dapat dilihat pada gambar 3.4.



Gambar 3.4 Flowchart Akuisisi dan Deteksi Wajah

3.2.2.3 Recording dan Encoding Wajah

Proses perekaman dan konversi wajah menjadi representasi numerik (embedding) yang digunakan dalam pengenalan wajah real-time. Sistem bekerja setelah tahap deteksi wajah berhasil menandai *bounding box* wajah pada *frame*. Alur sistemnya dapat dilihat pada gambar 3.5 adalah sebagai berikut:

1. **Input Data Identitas**

Sistem meminta pengguna untuk memasukkan informasi identitas, seperti ID, Nama, dan Gender (jika ingin ditentukan manual). Informasi ini digunakan untuk menandai setiap embedding yang dihasilkan.

2. **Perekaman Wajah (Recording)**

Setiap *frame* yang berisi wajah yang terdeteksi akan diproses untuk memastikan kualitas wajah memadai, termasuk: pencahayaan, pose, dan ukuran wajah dalam *frame*. Sistem akan merekam sebanyak 30 *frame* untuk setiap pengguna sebagai sampel.

3. **Ekstraksi *Face embedding***

Sistem menggunakan model CNN pra-latih untuk mengubah wajah menjadi vektor numerik berdimensi 128. Vektor ini adalah representasi unik wajah yang dapat dibandingkan secara matematis dengan wajah lain tanpa menyimpan gambar mentah.

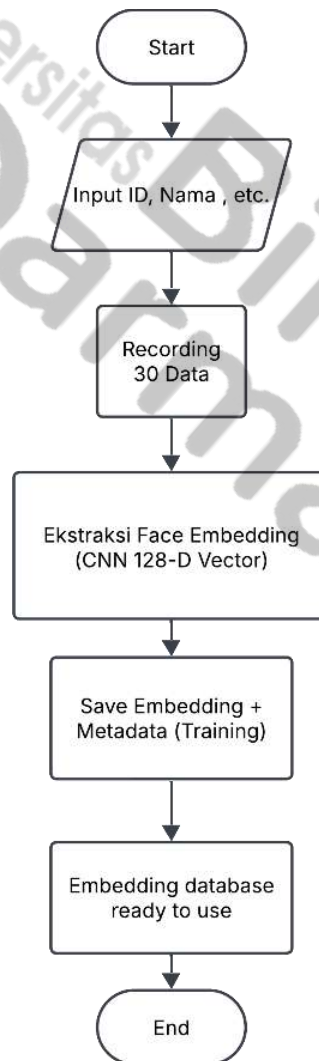
4. **Penyimpanan Embedding dan Metadata**

Setiap embedding yang berhasil diekstraksi disimpan bersama metadata pengguna (ID, Nama, Gender) ke dalam database sementara. Setelah seluruh sampel tercatat, database ini disimpan ke file .pickle atau .npy untuk memudahkan akses di tahap pengenalan wajah berikutnya.

5. **Hasil Sistem**

Setelah semua sampel wajah berhasil direkam dan diencoding, database siap digunakan untuk face matching, identifikasi, dan klasifikasi atribut tambahan seperti gender. Dengan cara ini, sistem memastikan data wajah yang efisien, aman, dan siap untuk pemrosesan real-time.

Metode ini tidak lagi menyimpan gambar mentah, hanya embedding numerik. Pendekatan ini meningkatkan efisiensi penyimpanan, mempercepat proses pencocokan wajah, dan memungkinkan untuk dapat menambahkan perangkat tambahan kedepannya.



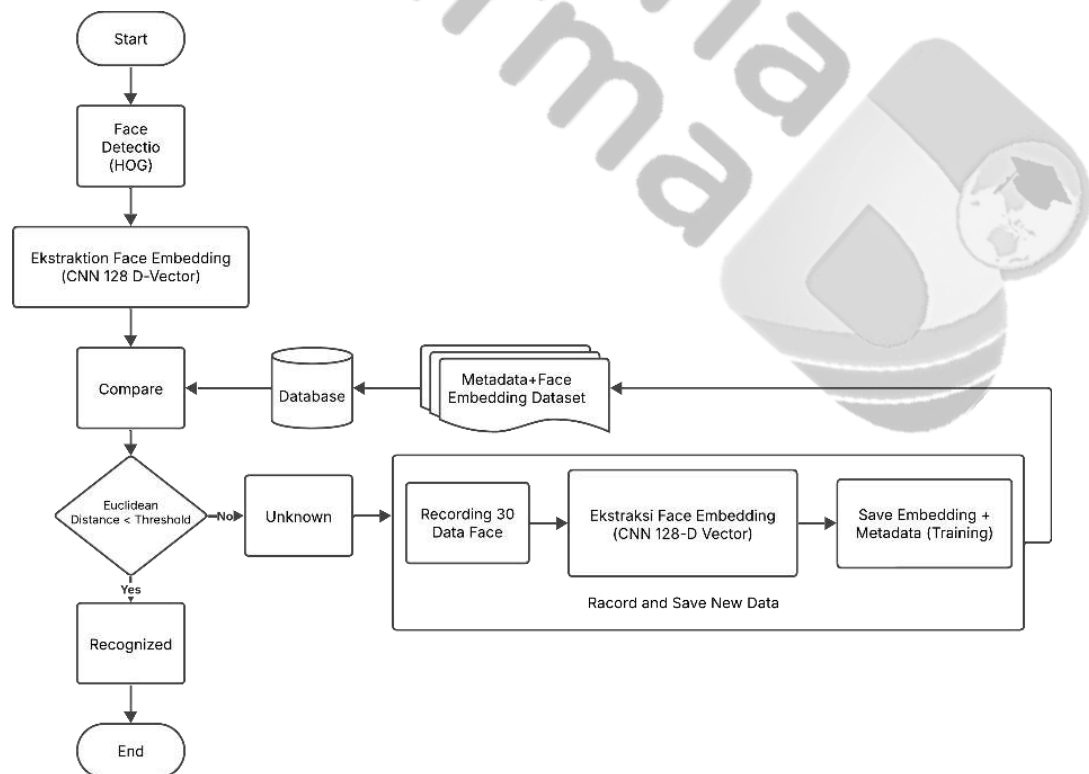
Gambar 3.5 Flowchart Recording dan Encoding Wajah

3.2.2.4 Proses Pengenalan dan Respon Sistem

Gambar 3.6 menerangkan proses keseluruhan pengenalan dan respon sistem dimana saat setelah database embedding wajah siap digunakan, sistem pengenalan wajah bekerja secara real-time untuk mengenali pengguna dan memberikan respon yang sesuai. Proses dimulai dari deteksi wajah pada *frame* yang

ditangkap oleh webcam. Setiap wajah yang terdeteksi ditandai dengan *bounding box* dan kemudian diekstraksi menjadi vektor numerik berdimensi 128 menggunakan model CNN pra-latih. Vektor ini, yang disebut *face embedding*, dibandingkan dengan seluruh embedding dalam database menggunakan jarak Euclidean. Jika jarak antara embedding baru dan database lebih kecil dari threshold yang telah ditentukan, wajah dianggap teridentifikasi (Recognized) dan identitas pengguna ditentukan, jika tidak wajah dianggap Unknown.

Dengan alur ini, seluruh proses dari deteksi wajah hingga interaksi robot berlangsung secara otomatis, cepat, dan real-time, memanfaatkan database embedding untuk mengenali identitas pengguna dengan akurat serta menyediakan respon yang interaktif dan personal.



Gambar 3.6 Proses Pengenalan dan Respon Sistem

3.2.3 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari perancangan robot resepsionis, kamera, motor servo dan speaker robot.

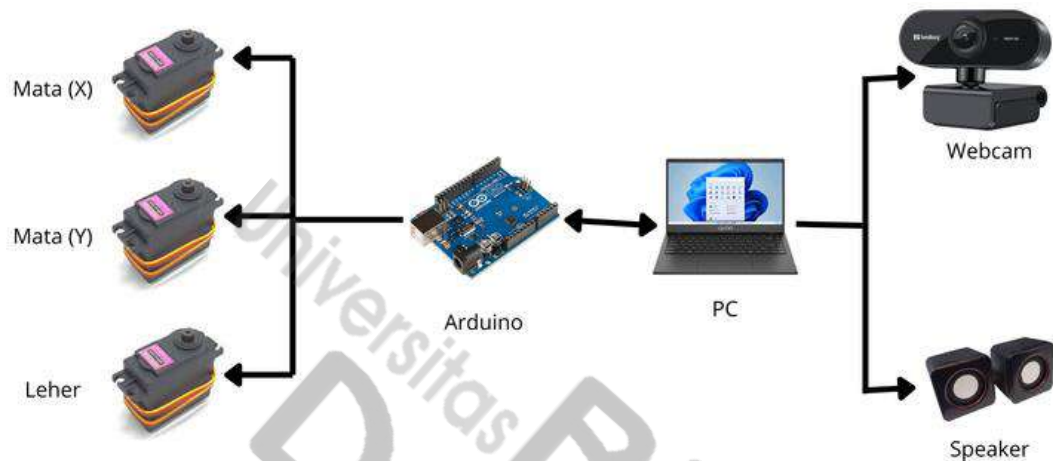
3.2.3.1 Perancangan Robot Resepsionis

Untuk bentuk fisiknya sendiri, penulis menggunakan desain robot Inmoov yang dapat dilihat pada gambar 3.7. InMoov adalah proyek robot *humanoid open source* yang dikembangkan oleh seorang insinyur asal Prancis bernama Gael Langevin. Robot ini dirancang untuk digunakan oleh para pengembang dan hobiis di seluruh dunia. InMoov dapat dicetak menggunakan printer 3D dan dilengkapi dengan berbagai sensor dan aktuator untuk memberikan fungsionalitas yang mirip dengan manusia. Proyek InMoov dimulai pada tahun 2012, dan sejak itu telah mendapatkan banyak perhatian dari komunitas pengembang di bidang robotika dan DIY (*Do It Yourself*). InMoov mencakup bagian-bagian seperti kepala, tangan, dan lengan yang dapat dicetak dengan printer 3D, serta perangkat keras dan perangkat lunak yang diperlukan untuk mengontrolnya.



Gambar 3.7 Robot Inmoov

Sedangkan untuk sistem kontrol dan sistem pemrosesnya dapat dilihat pada gambar 3.8, disini penulis menggunakan mikrokontroller Arduino sebagai penggerak robot, dan komputer untuk memproses deteksi wajah nya. Komputer akan secara serial mengirimkan data pengunjung jika ada pengunjung terdeteksi agar arduino dapat memprosesnya dan melakukan aksi yang sudah disiapkan sebelumnya.

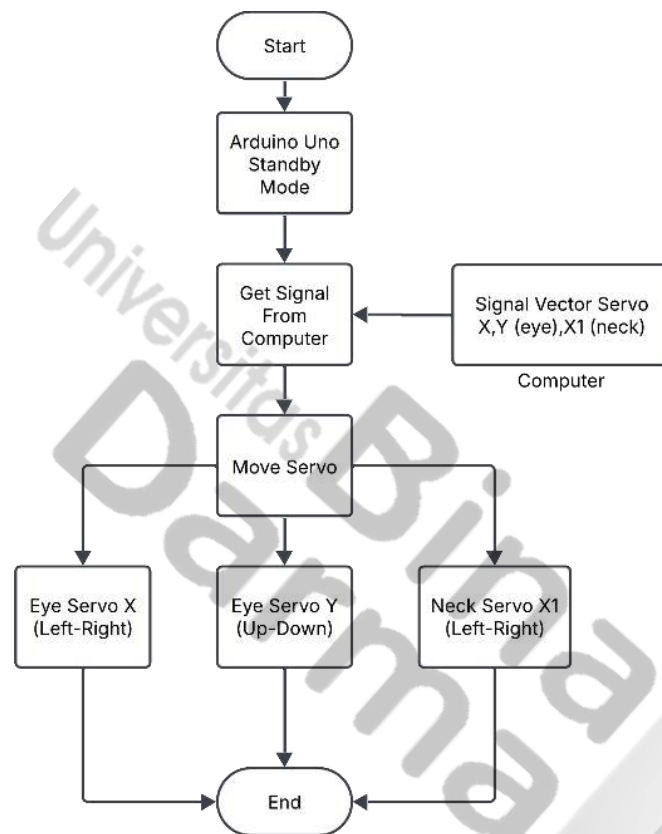


Gambar 3.8 Desain Skematik Robot

Komputer akan terhubung pada perangkat kamera sebagai inputan untuk mendeteksi wajah pengunjung, layar monitor untuk menampilkan opsi yang ingin dipilih oleh pengunjung, dan juga dilengkapi dengan mouse dan keyboard sebagai input masukan data pengunjung. Sedangkan untuk sistem penggerak sendiri, arduino akan terhubung dengan motor servo untuk menggerakkan mata dan kepala robot dan sebuah speaker untuk robot dapat bersuara dan berkomunikasi dengan pengunjung.

3.2.3.2 Program Gerak Robot

Program gerak robot resepsionis ini memanfaatkan perangkat Arduino Uno sebagai otak utama, dipasangkan dengan motor servo untuk menggerakkan mata dan leher robot. Interaksi dengan tamu dimulai dengan mendeteksi kehadiran mereka, di mana Arduino Uno beroperasi dalam mode penerimaan aktif. Begitu tamu terdeteksi, komputer akan mengirimkan sinyal berupa posisi sudut dari komputer ke arduino untuk mengarahkan kepala robot agar menghadap ke arah tamu, servo yg di gunakan ada sebanyak 3 buah, yaitu 2 untuk mata agar bergerak ke atas bawah kiri kanan, dan 1 servo di lehernya untuk menggerakkan kepala menghadap kiri atau kanan sesuai dengan data yang diterima.



Gambar 3.9 Flowchart Program Gerak Robot

BAB IV

HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil pengujian dan pembahasan dari sistem pengenalan wajah yang dikembangkan. Penjelasan dimulai dari pengujian tahap dasar, termasuk pengambilan dan deteksi wajah, perekaman data wajah, ekstraksi *face embedding*, hingga manajemen database embedding. Selanjutnya, dijelaskan pengujian proses pengenalan wajah secara real-time, klasifikasi atribut seperti gender, dan implementasi respon sistem melalui robot.

Pembahasan mencakup analisis kinerja sistem, seperti akurasi pengenalan wajah, waktu proses deteksi dan matching, serta efektivitas interaksi antara manusia dan robot. Hasil pengujian disajikan secara bertahap untuk menunjukkan bagaimana setiap komponen software dan hardware bekerja secara terintegrasi, sehingga sistem dapat beroperasi secara real-time dengan tingkat keakuratan tinggi.

Dengan pendekatan ini, Bab IV bertujuan memberikan gambaran menyeluruh mengenai kinerja sistem pengenalan wajah, dari implementasi awal hingga integrasi penuh dengan robot, serta memberikan analisis kritis mengenai kelebihan, kendala, dan potensi pengembangan lebih lanjut.

4.1 Hasil Pengujian

4.1.1 Pengujian Akuisisi dan Deteksi Wajah

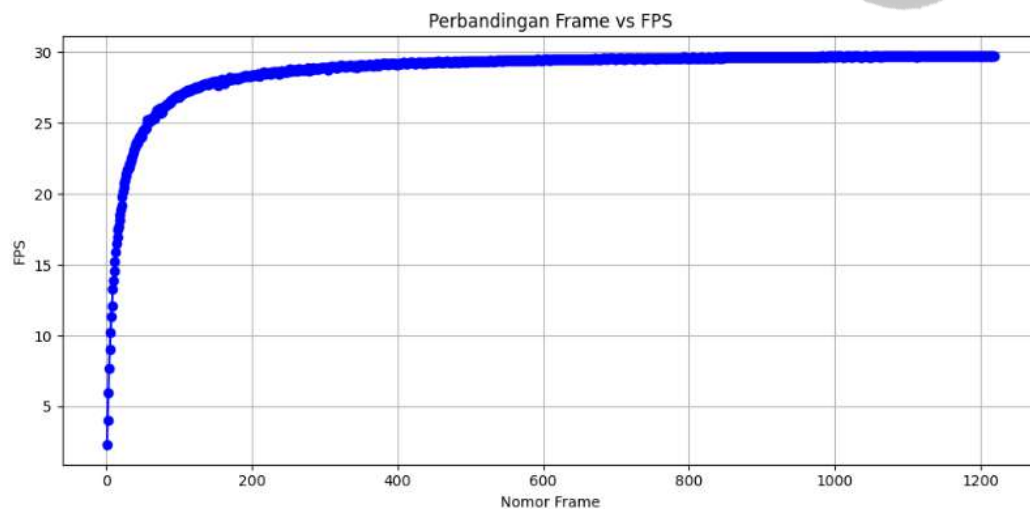
4.1.1.1 Pengujian Koneksi Webcam

Pengujian dimulai dengan memastikan bahwa sistem dapat mengakses dan mengambil data dari webcam secara real-time. Pada tahap ini, program dibuka dan mencoba membuat koneksi dengan perangkat kamera yang digunakan, biasanya dengan indeks 0 untuk webcam bawaan atau default jika berhasil maka pada program akan tampil webcam secara realtime seperti gambar 4.1. Keberhasilan koneksi ini menjadi langkah awal yang sangat penting, karena seluruh proses pengenalan wajah bergantung pada kemampuan sistem untuk menangkap *frame* dari kamera dengan stabil.

Setelah koneksi berhasil, sistem melakukan pengambilan *frame* secara terus-menerus. Setiap *frame* yang diterima dari webcam diproses secara real-time agar program siap melakukan deteksi wajah dan tahap berikutnya, yaitu ekstraksi fitur. Selama pengujian, diperhatikan beberapa hal utama: stabilitas *frame*, kecepatan pembacaan *frame*, dan kompatibilitas resolusi webcam dengan system. Hasil perbandingan ini dapat dilihat pada gambar 4.2.



Gambar 4.1 Pengujian FPS Kamera



Gambar 4.2 Grafik Performa Webcam

Dari grafik yang ditampilkan, terlihat hubungan antara nomor *frame* dan FPS (*frame* per detik) selama pengujian webcam secara real-time. Pada awal pengambilan *frame*, FPS relatif rendah, yang kemungkinan disebabkan oleh inisialisasi webcam dan pemrosesan awal program. Namun, seiring bertambahnya jumlah *frame*, FPS meningkat dengan cepat dan kemudian mulai stabil mendekati nilai maksimum sekitar 30 FPS. Hal ini menunjukkan bahwa setelah program mencapai kondisi stabil, sistem mampu membaca *frame* dengan kecepatan konstan, sehingga pengambilan dan pemrosesan data gambar berjalan secara efisien dan real-time.

Pola grafik ini juga menandakan bahwa sistem memiliki kemampuan untuk menyesuaikan performa pada awal pemrosesan dan kemudian mempertahankan kecepatan *frame* yang stabil, yang sangat penting untuk tahap deteksi wajah dan ekstraksi *face embedding* agar pengenalan wajah dapat dilakukan secara akurat dan responsif. Dengan kata lain, grafik ini memperlihatkan bahwa perangkat lunak program pengambilan *frame* telah berfungsi dengan baik, mampu mendukung pemrosesan real-time, dan siap digunakan untuk tahap pengujian selanjutnya, seperti deteksi wajah dan pengenalan identitas.

4.1.1.2 Deteksi Wajah Menggunakan HOG/CNN

Setelah pengujian koneksi webcam dan pengambilan *frame* berhasil, tahap selanjutnya adalah menguji kemampuan sistem dalam mendeteksi wajah secara real-time. Pada pengujian ini, dua metode deteksi digunakan: HOG berbasis CPU dan CNN berbasis GPU. Tujuannya adalah untuk menentukan apakah sistem dapat mengenali posisi wajah pada *frame* secara akurat dan konsisten, sehingga tahapan ekstraksi *face embedding* dapat dilakukan dengan benar.

Program mengambil *frame* secara kontinu, kemudian menjalankan algoritma deteksi untuk mencari lokasi wajah. Setiap wajah yang berhasil terdeteksi ditandai dengan *bounding box* berwarna hijau yang ditampilkan langsung di layar. Selama pengujian, diamati beberapa parameter penting, yaitu jumlah wajah yang berhasil dideteksi per *frame*, kecepatan deteksi, dan stabilitas *bounding box* saat pengguna bergerak atau posisi wajah berubah.

Berdasarkan hasil pengujian deteksi wajah menggunakan metode HOG dan CNN, terlihat perbedaan kinerja yang cukup signifikan antara keduanya. Metode HOG yang dijalankan di CPU mampu mendeteksi wajah dengan cepat, menghasilkan FPS antara 18–21, namun kadang gagal mendeteksi wajah saat pencahayaan kurang ideal atau ketika wajah miring, sehingga *bounding box* terkadang bergeser. Sebaliknya, metode CNN yang dijalankan di GPU menunjukkan konsistensi yang lebih tinggi dengan FPS 11–12, tetapi mampu mendeteksi wajah di semua *frame* termasuk pada kondisi pencahayaan rendah dan sudut wajah berbeda, serta menampilkan *bounding box* yang lebih stabil. Hasil keseluruhan dari perbandingan ini sendiri dapat dilihat pada table 4.1.

Dari pengamatan ini, dapat disimpulkan bahwa HOG cocok digunakan pada perangkat dengan sumber daya terbatas dan membutuhkan respons cepat, meskipun akurasi sedikit menurun, sedangkan CNN lebih unggul dalam hal akurasi dan stabilitas deteksi, sehingga lebih handal untuk aplikasi yang memerlukan pengenalan wajah dengan kondisi lingkungan bervariasi.

Tabel 4.1 Perbandingan HOG dan CNN dalam mendeteksi wajah

<i>Frame</i>	Metode	Wajah Terdeteksi	FPS	Keterangan <i>Bounding box</i>
1	HOG	1	20	Sedikit goyah
1	CNN	1	12	Stabil
2	HOG	1	18	Bergeser saat bergerak
2	CNN	1	11	Stabil
3	HOG	0	19	Tidak terdeteksi karena pencahayaan agak redup
3	CNN	1	11	Deteksi berhasil
4	HOG	1	21	Cukup stabil
4	CNN	1	12	Stabil
5	HOG	1	20	Bergeser saat wajah miring
5	CNN	1	11	Stabil



Gambar 4.3 Pengujian *bounding box*

4.1.2 Pengujian Recording dan Encoding Wajah

Setelah tahap Akuisisi dan Deteksi Wajah, tahap selanjutnya adalah pengujian recording dan encoding wajah. Pada tahap ini, sistem diuji kemampuan dan keandalannya dalam merekam wajah secara real-time serta mengekstraksi *face embedding* yang merupakan representasi numerik dari setiap wajah.

4.1.2.1 Perekaman Beberapa *Frame* Wajah (30 wajah)

Pada tahap pengujian perekaman beberapa *frame* wajah, sistem diuji dengan merekam sebanyak 30 *frame* wajah dari setiap pengguna. Pengambilan data dilakukan dengan memasukkan NIK sebagai ID pengguna dan nama tamu, sehingga setiap *frame* wajah disimpan secara terstruktur di dalam folder berdasarkan tanggal dan ID pengguna. Hasil pengujian menunjukkan bahwa sistem mampu menangkap wajah dengan akurat pada semua *frame* yang direkam, dengan *bounding box* mengikuti pergerakan wajah secara real-time. Setiap wajah yang berhasil dideteksi dicrop dan disimpan dengan nama tamu sehingga memudahkan identifikasi dan pengelolaan dataset. Selama proses pengujian, kualitas *frame* terlihat baik, wajah berada di dalam *bounding box*, dan tidak terjadi kehilangan data wajah. Hal ini memastikan bahwa dataset yang dihasilkan cukup representatif dan konsisten, sehingga dapat digunakan pada tahap selanjutnya, yaitu ekstraksi *face embedding*, untuk keperluan pengenalan dan verifikasi wajah secara real-time.



Gambar 4.4 Pengambilan Dataset Tamu

4.1.2.2 Ekstraksi *Face embedding*

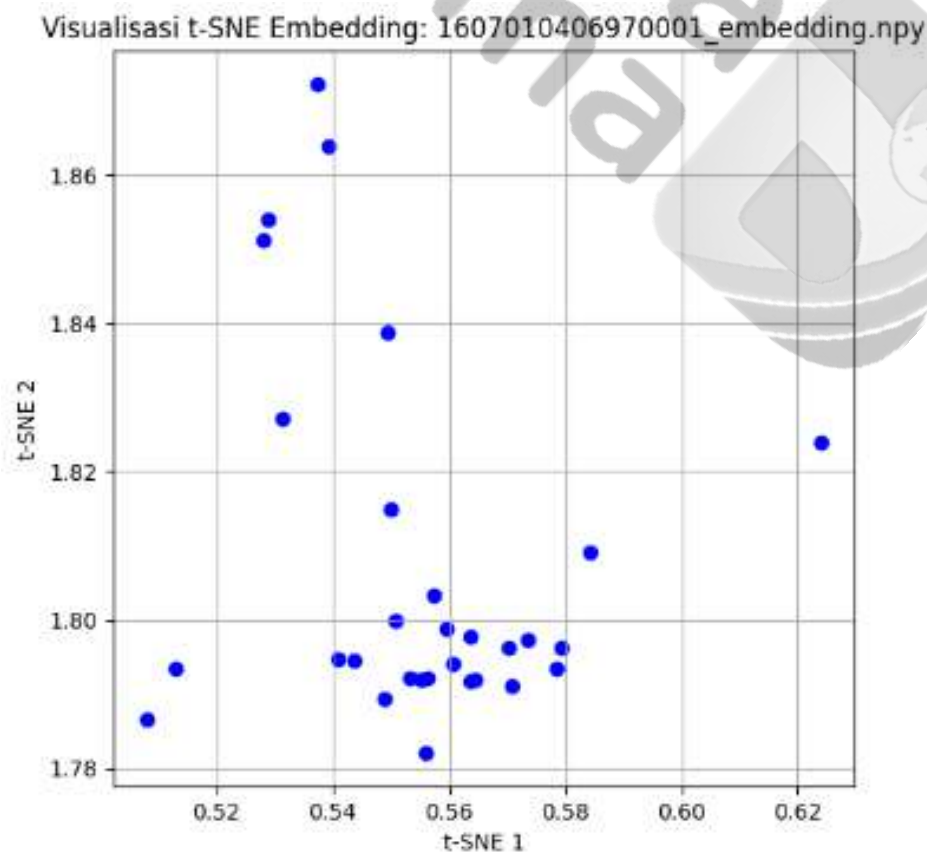
Pada tahap ini, setiap *frame* wajah yang telah direkam diproses untuk diubah menjadi representasi numerik atau embedding, yang merepresentasikan fitur unik dari wajah pengguna secara kompak. *Face embedding* ini menjadi inti dari sistem pengenalan wajah, karena memungkinkan sistem untuk membandingkan dan mengenali wajah dengan cepat tanpa harus menyimpan seluruh citra secara langsung. Pengujian pada tahap ini bertujuan untuk memastikan bahwa embedding yang dihasilkan stabil, konsisten, dan berbeda untuk setiap individu, meskipun terdapat variasi ekspresi wajah, sudut pandang, atau pencahayaan. Keberhasilan ekstraksi embedding akan menentukan akurasi proses identifikasi atau verifikasi wajah pada tahap selanjutnya, sehingga proses ini menjadi salah satu komponen kunci dalam pipeline sistem pengenalan wajah secara real-time.

Berdasarkan hasil visualisasi t-SNE dari file embedding 1607010406970001_embedding.npy, terlihat bahwa embedding yang dihasilkan dari 30 *frame* wajah pengguna membentuk kluster yang relatif rapat di ruang 2D. Titik-titik biru pada plot menunjukkan setiap *frame* wajah yang telah diekstraksi menjadi vektor embedding 128 dimensi, kemudian direduksi ke dua dimensi menggunakan t-SNE. Pengamatan terhadap plot menunjukkan bahwa sebagian besar titik berada dalam satu area yang padat, menandakan konsistensi embedding untuk *frame* yang sama meskipun terdapat variasi ekspresi atau posisi wajah. Beberapa titik tampak sedikit terpisah dari kluster utama, yang kemungkinan

disebabkan oleh perbedaan kecil dalam pencahayaan atau sudut wajah saat *frame* diambil, namun secara umum kluster utama tetap rapat dan terpusat.

Hasil ini menunjukkan bahwa sistem berhasil mengekstraksi fitur wajah yang stabil dan representatif, sehingga embedding dapat digunakan untuk identifikasi atau verifikasi wajah dengan akurasi tinggi. Distribusi yang rapat dan jelas memudahkan algoritma pengenalan wajah untuk membedakan satu individu dari individu lain dalam database.

Dengan demikian, visualisasi t-SNE menegaskan bahwa pipeline perekaman dan ekstraksi embedding yang dijalankan efektif dan konsisten, serta dataset yang dihasilkan siap digunakan pada tahap pengenalan wajah secara real-time.



Gambar 4.5 Hasil ekstraksi *face embedding* 30 wajah dataset

4.1.3 Pengujian Proses Pengenalan Wajah

4.1.3.1 Pengujian Offline (Manual)

Pengujian dilakukan dengan membandingkan embedding database dari dataset 1607010406970001—yang merepresentasikan saya sendiri—dengan 8 foto uji yang terdiri dari dua kategori: 4 foto orang random dan 4 foto diri sendiri dalam kondisi berbeda. Tujuan pengujian ini adalah mengevaluasi kemampuan sistem pengenalan wajah dalam mengenali subjek dari database, sekaligus menguji ketahanan sistem terhadap variasi kondisi wajah, posisi, dan jumlah orang di foto, dimana hasilnya dapat dilihat pada gambar 4.6 dan tabel 4.2.

Untuk empat foto orang random, yaitu *albert.jfif*, *elon musk.jpg*, *magnus carlsen.jfif*, dan *nikola.jfif*, sistem berhasil mengenali bahwa tidak ada kecocokan dengan database saya. Hal ini ditunjukkan oleh nilai *distance* yang relatif tinggi, antara 0.6936 hingga 1.0116, serta *similarity* 0.00%, sehingga sistem tidak salah mengenali wajah orang asing sebagai saya. Sementara itu, untuk empat foto diri saya sendiri dalam kondisi berbeda, sistem menunjukkan hasil yang sesuai dengan ekspektasi. Pada foto *endang.jpeg*, saya menggunakan kacamata, dan sistem mampu mengenali saya dengan *distance* 0.4649 dan *similarity* 22.51%, meskipun aksesoris sedikit menutupi wajah.

Foto *endang_2orang.jpeg* menampilkan saya bersama satu orang lain, dan sistem tetap berhasil mengenali saya dengan *distance* 0.3432 dan *similarity* 42.80%, menunjukkan bahwa sistem cukup mampu fokus pada wajah target meskipun terdapat orang lain dalam *frame*. Pada foto *endang_tanpa_kacamata.jpeg*, saya tidak menggunakan kacamata, dan sistem mengenali saya dengan *distance* 0.3054 dan *similarity* 49.10%, nilai *similarity* tertinggi di antara semua foto uji. Hal ini menegaskan bahwa aksesoris seperti kacamata dapat memengaruhi hasil, dan kondisi wajah natural memberikan tingkat kemiripan terbaik. Terakhir, pada foto *endang_view_seluruh_tubuh.jpeg*, wajah saya terlihat dari jarak jauh sehingga sebagian tubuh terlihat, namun sistem tetap mampu mengenali saya dengan *distance* 0.3630 dan *similarity* 39.50%, menunjukkan toleransi sistem terhadap variasi skala dan jarak.

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa sistem pengenalan wajah berbasis embedding dari database 1607010406970001 cukup

robust. Sistem tidak salah mengenali orang asing, dan tetap mampu mengenali saya dalam berbagai kondisi, meskipun variasi seperti aksesoris, kehadiran orang lain, atau jarak pengambilan foto sedikit memengaruhi nilai similarity. Pola yang terlihat adalah semakin jelas dan dekat wajah, similarity meningkat dan distance menurun, yang menunjukkan keakuratan sistem dalam mengenali wajah target.



Gambar 4.6 Pengujian *face recognition* offline (manual)

Tabel 4.2 Hasil Pengujian *Face recognition* Offline

No	File Foto	Hasil Kecocokan	Distance	Similarity (%)
1	albert.jfif	Tidak ada kecocokan	0.8060	0.00
2	elon musk.jpg	Tidak ada kecocokan	0.6936	0.00
3	endang.jpeg	1607010406970001	0.4649	22.51
4	endang_2orang.jpeg	1607010406970001	0.3432	42.80
5	endang_tanpa_kacamata.jpeg	1607010406970001	0.3054	49.10
6	endang_view_seluruh_tubuh.jpeg	1607010406970001	0.3630	39.50
7	magnus carlsen.jfif	Tidak ada kecocokan	0.7494	0.00
8	nikola.jfif	Tidak ada kecocokan	1.0116	0.00

4.1.3.2 Pengujian Online (Realtime)

Pada tahap ini, saya melakukan pengujian sistem pengenalan wajah secara realtime menggunakan kamera selama 30 detik, dengan pengambilan 1 *frame* setiap detik. Pengujian ini bertujuan untuk mengetahui kestabilan dan konsistensi sistem dalam mengenali wajah saya secara langsung dalam kondisi dinamis. Embedding yang digunakan sebagai acuan adalah dataset 1607010406970001, yang merupakan data wajah saya sendiri dari tahap sebelumnya yang hasilnya dapat dilihat pada tabel 4.3.

Selama 30 detik pengujian, sistem menghitung nilai *distance* dan *similarity* pada setiap *frame*, kemudian mengelompokkannya ke dalam kategori Sangat Baik, Baik, Cukup, dan Kurang (noise). Hasil pengujian menunjukkan bahwa mayoritas *frame* berada pada kategori Baik hingga Sangat Baik, dengan nilai similarity rata-rata berada pada rentang 40%–53% ketika posisi wajah relatif stabil dan menghadap kamera. Hal ini menunjukkan bahwa sistem mampu mengenali wajah secara konsisten dalam kondisi normal dengan pencahayaan yang cukup dan pose yang relatif frontal.

Namun demikian, terdapat beberapa *frame* yang masuk dalam kategori Kurang (noise). Nilai similarity pada *frame* tersebut berada di bawah 20%, yang disebabkan oleh beberapa faktor seperti perubahan pencahayaan, pergerakan kepala yang cepat, posisi wajah yang tidak sepenuhnya menghadap kamera, atau efek blur akibat pergerakan. Kondisi ini menyebabkan nilai *distance* meningkat sehingga

tingkat kemiripan menurun. Meskipun demikian, jumlah *frame* yang tergolong noise relatif sedikit dibandingkan total keseluruhan *frame* yang diuji.

Secara keseluruhan, hasil pengujian realtime ini menunjukkan bahwa sistem pengenalan wajah berbasis embedding yang digunakan cukup stabil dan mampu bekerja dengan baik dalam kondisi nyata. Persentase hasil kategori Baik dan Sangat Baik yang dominan menunjukkan bahwa sistem memiliki tingkat keandalan yang tinggi, sementara kemunculan beberapa noise dapat dijadikan bahan evaluasi untuk pengembangan lebih lanjut, khususnya dalam meningkatkan ketahanan sistem terhadap variasi pencahayaan dan pergerakan wajah.

Tabel 4.3 Hasil Pengujian *Face recognition* Online (Realtime)

Detik	Distance	Similarity (%)	Kategori
1	0.2981	50.31	Sangat Baik
2	0.3124	47.93	Sangat Baik
3	0.3552	40.80	Baik
4	0.3421	42.98	Baik
5	0.2875	52.08	Sangat Baik
6	0.4012	33.13	Baik
7	0.3725	37.91	Baik
8	0.4893	18.45	Kurang (Noise)
9	0.3314	44.77	Baik
10	0.3056	49.07	Sangat Baik
11	0.2948	50.87	Sangat Baik
12	0.4520	24.67	Cukup
13	0.3189	46.85	Sangat Baik
14	0.3674	38.76	Baik
15	0.3901	35.00	Baik
16	0.2765	53.91	Sangat Baik
17	0.2997	50.05	Sangat Baik
18	0.4156	30.73	Baik
19	0.5210	13.16	Kurang (Noise)
20	0.3084	48.60	Sangat Baik
21	0.3338	44.37	Baik
22	0.3475	42.08	Baik

Detik	Distance	Similarity (%)	Kategori
23	0.2896	51.73	Sangat Baik
24	0.3712	38.13	Baik
25	0.4025	32.91	Baik
26	0.4804	19.93	Kurang (Noise)
27	0.2963	50.62	Sangat Baik
28	0.3598	40.03	Baik
29	0.3145	47.58	Sangat Baik
30	0.3012	49.80	Sangat Baik

4.1.4 Pengujian Perangkat Keras (Robot)

Pada tahap pengujian perangkat keras, sistem diuji untuk mengetahui kemampuan servo robot dalam merespons pergerakan wajah berdasarkan koordinat posisi wajah (pixel) yang terdeteksi kamera yang hasilnya tertuang di tabel 4.4. Parameter yang diamati meliputi posisi wajah pada sumbu X dan Y, perubahan sudut servo pada sumbu horizontal dan vertikal (derajat), arah pergerakan (R = Right, L = Left, U = Up, D = Down, C = Center), serta waktu respons sistem dalam satuan detik.

Berdasarkan hasil pengujian, sistem menunjukkan bahwa perubahan posisi wajah pada koordinat pixel secara langsung memengaruhi perubahan sudut servo. Pada posisi awal (302, 212), servo berada pada sudut 110° (X) dan 90° (Y) dengan kondisi Center–Center, yang menunjukkan posisi netral atau titik tengah kamera. Ketika wajah bergerak ke kanan (misalnya posisi 459, 204 dan 518, 342), servo pada sumbu X meningkat menjadi 137° dan 139° , serta arah pergerakan tercatat sebagai Right. Hal ini menunjukkan bahwa sistem berhasil memetakan peningkatan nilai pixel X menjadi rotasi servo ke arah kanan.

Sebaliknya, ketika wajah bergerak ke kiri (posisi 399, 278 dan 219, 302), sudut servo pada sumbu X menurun hingga 132° dan 91° , dengan arah pergerakan Left. Ini menandakan bahwa sistem mampu mengurangi sudut servo secara proporsional terhadap perpindahan wajah ke sisi kiri *frame*. Pada sumbu vertikal, pergerakan wajah ke atas menyebabkan sudut servo Y menurun (misalnya menjadi 79° dan 82° dengan arah Up), sedangkan ketika wajah bergerak ke bawah, sudut

meningkat hingga 105° dengan arah Down. Pola ini menunjukkan bahwa kontrol vertikal servo berjalan sesuai dengan logika pemetaan koordinat pixel terhadap sudut mekanik.

Dari segi waktu respons, rata-rata sistem merespons dalam rentang ± 2 hingga 3,5 detik. Waktu tercepat tercatat sekitar 1,945 detik dan yang terlama sekitar 3,561 detik. Variasi waktu ini kemungkinan dipengaruhi oleh proses komputasi deteksi wajah, pengiriman sinyal ke mikrokontroler, serta kecepatan mekanik servo itu sendiri. Meskipun terdapat variasi waktu, seluruh pergerakan berhasil mencapai posisi target tanpa kegagalan.

Tabel 4. 4 Hasil Pengujian Gerak Servo

Position	Face Position (pixel)		Servo Position ($^\circ$)		Movement		Time (s)
	x	y	x	y	x	y	
	302	212	110	90	C	C	0
	459	204	137	79	R	U	2.314
	399	278	132	96	L	D	1.945
	518	342	139	105	R	D	2.952
	219	302	91	94	L	U	3.561
	391	159	119	82	R	U	3.013

4.1.5 Analisis Performa Sistem

Dilakukan analisis performa sistem secara menyeluruh untuk mengevaluasi tingkat kinerja dan keandalan sistem pengenalan wajah yang telah dirancang. Analisis ini bertujuan untuk mengukur seberapa baik sistem bekerja dalam kondisi

nyata, baik dari sisi perangkat lunak (software) maupun perangkat keras (hardware) yang terintegrasi.

Parameter yang dianalisis meliputi waktu proses deteksi dan pengenalan wajah dan akurasi pengenalan wajah dilingkungan yang realtime. Waktu proses digunakan untuk mengetahui kecepatan sistem dalam mendeteksi dan mengenali wajah secara realtime. Akurasi pengenalan wajah digunakan untuk mengukur tingkat keberhasilan sistem dalam mengidentifikasi wajah yang sesuai dengan database.

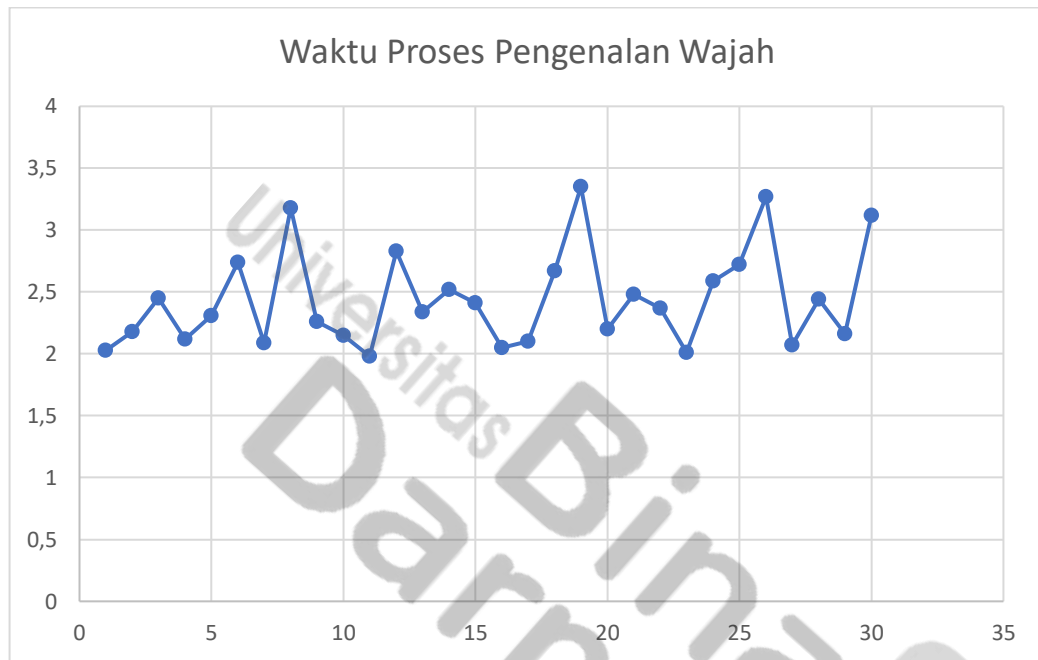
4.1.5.1 Analisis Waktu Proses Deteksi dan Pengenalan

Pengujian dilakukan selama 30 detik dengan pengambilan 1 *frame* setiap detik. Pada setiap *frame*, dicatat waktu yang dibutuhkan sistem mulai dari proses deteksi wajah, ekstraksi fitur (embedding), hingga pencocokan dengan database. Data waktu proses yang diperoleh kemudian dianalisis secara statistik menggunakan parameter nilai minimum, maksimum, rata-rata (mean), serta standar deviasi untuk mengetahui tingkat kestabilan dan konsistensi kinerja system Dimana hasil pengujian ini dapat dilihat pada gambar grafik 4.7.

Berdasarkan perhitungan statistik terhadap 30 data waktu proses deteksi dan pengenalan wajah, diperoleh nilai rata-rata (mean) sebesar $\pm 2,42$ detik. Nilai ini menunjukkan bahwa secara umum sistem membutuhkan waktu sekitar dua hingga tiga detik untuk memproses satu *frame* dalam kondisi realtime.

Nilai minimum yang diperoleh adalah 1,98 detik, yang menunjukkan performa terbaik sistem saat kondisi optimal, seperti pencahayaan yang stabil dan posisi wajah menghadap kamera secara frontal. Sementara itu, nilai maksimum tercatat sebesar 3,35 detik, yang terjadi pada beberapa kondisi tertentu yang dapat dikategorikan sebagai gangguan, seperti pergerakan wajah yang cepat atau perubahan intensitas cahaya.

Standar deviasi sebesar $\pm 0,36$ detik menunjukkan bahwa variasi waktu proses relatif kecil dan masih berada dalam rentang yang stabil. Nilai deviasi yang tidak terlalu besar mengindikasikan bahwa sistem memiliki konsistensi performa yang baik, karena fluktuasi waktu tidak menyimpang jauh dari nilai rata-rata.



Gambar 4.7 Grafik Waktu Proses Pengenalan Wajah

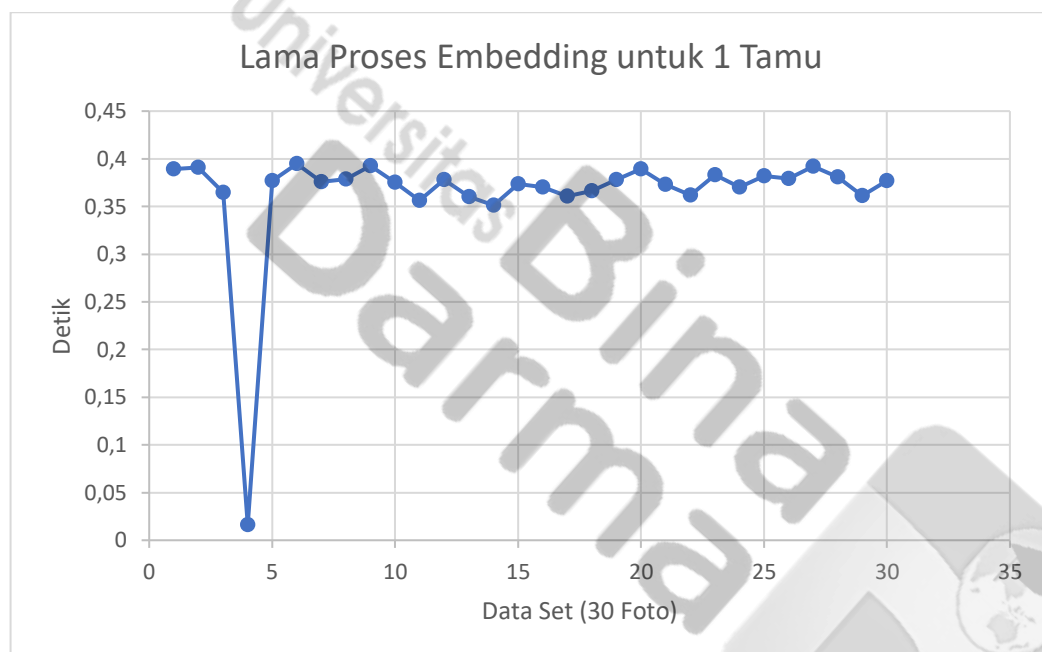
4.1.6 Analisis Pengujian Robot

Pada tahap ini dilakukan analisis terhadap kinerja sistem robot dalam melakukan proses pengenalan wajah n . Pengujian dilakukan menggunakan sampel sebanyak 9 orang, yang terdiri dari 6 laki-laki dan 3 perempuan. Setiap sampel memiliki 30 citra wajah, sehingga total dataset yang digunakan dalam penelitian ini adalah 270 citra wajah. Proses pengujian dibagi menjadi tiga bagian utama, yaitu:

4.1.6.1 Analisis Waktu Proses Embedding

Analisis ini bertujuan untuk mengetahui performa sistem dalam membangun representasi fitur wajah (embedding) menggunakan model CNN. Pengujian waktu dilakukan pada 2 skala berbeda, yang pertama waktu yang dibutuhkan untuk memproses 1 sampel (30 foto). Berdasarkan hasil pengujian waktu proses embedding untuk satu sampel yang terdiri dari 30 citra wajah, diperoleh variasi waktu pemrosesan pada setiap foto. Waktu tercepat tercatat sebesar 0,0161 detik pada foto ke-3, sedangkan waktu terlama sebesar 0,3947 detik pada foto ke-5. Secara umum, sebagian besar waktu pemrosesan berada pada rentang 0,34 detik hingga 0,41 detik per foto. Total waktu yang dibutuhkan untuk

memproses seluruh 30 foto adalah sekitar 11,2501 detik, dengan rata-rata waktu pemrosesan sebesar 0,3748 detik per foto. Untuk lengkapnya dapat dilihat pada gambar grafik 4.8.



Gambar 4.8 Grafik perbandingan waktu proses (1 Sampel)

Kemudian pengujian berikutnya ialah waktu yang dibutuhkan untuk memproses 9 sampel (270 foto) secara keseluruhan yang hasilnya dapat dilihat pada table 4.5. Berdasarkan hasil pengujian proses embedding terhadap 9 sampel dengan total 270 citra wajah, diperoleh variasi waktu pemrosesan pada setiap sampel. Setiap sampel terdiri dari 30 foto yang diproses untuk menghasilkan fitur embedding wajah. Dari data yang diperoleh, waktu tercepat proses embedding pada tingkat per foto berada pada rentang 0,0161 detik hingga 0,374 detik, sedangkan waktu terlama berada pada rentang 0,3828 detik hingga 0,6826 detik. Nilai waktu terlama tertinggi tercatat pada Sampel 2 yaitu sebesar 0,6826 detik, sementara waktu tercepat terendah tercatat pada Sampel 1 yaitu sebesar 0,0161 detik.

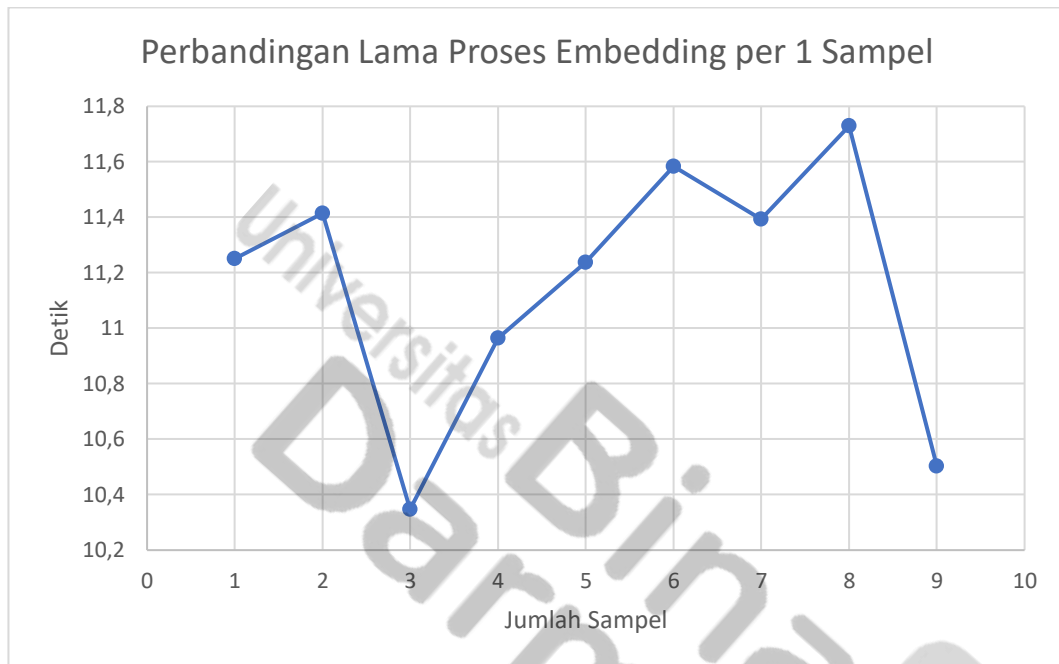
Dari sisi total waktu pemrosesan per sampel (30 foto), diperoleh hasil yang relatif stabil dengan rentang antara 8,5991 detik hingga 12,2874 detik. Sampel dengan waktu total tercepat adalah Sampel 3 dengan durasi 8,5991 detik, sedangkan

waktu total terlama terdapat pada Sampel 6 sebesar 12,2874 detik. Perbedaan durasi ini dipengaruhi oleh variasi kondisi citra, seperti pencahayaan, posisi wajah, ekspresi, serta kompleksitas proses deteksi wajah sebelum embedding dilakukan. Secara keseluruhan, total waktu yang dibutuhkan untuk memproses seluruh 9 sampel (270 foto) adalah sebesar 98,2419 detik.

Tabel 4.5 Waktu Proses Embedding dari 9 sampel (270)

Sampel	Lama Proses Embedding		
	Tercepat (s)	Terlama (s)	Total Proses (s)
1	0,0161	0,488	10,7921
2	0,3531	0,6826	12,1093
3	0,017	0,3877	8,5991
4	0,3504	0,4581	11,3713
5	0,0169	0,4159	10,2743
6	0,374	0,5605	12,2874
7	0,357	0,4104	11,2271
8	0,3424	0,3828	10,929
9	0,0189	0,5848	10,6523
Total waktu yang dibutuhkan			98,2419

Terakhir adalah perbandingan lama proses embedding pada saat terjadinya penambahan jumlah sampel yang ada, disini penulis menggunakan sampel 1 sebagai bahan uji coba nya. Bisa dilihat pada gambar 4.9, berdasarkan hasil rekapitulasi dari sembilan set pengujian yang dilakukan, sistem menunjukkan stabilitas performa yang sangat tinggi dengan nilai rata-rata waktu pemrosesan keseluruhan sebesar 0,3706 detik. Fenomena menarik terlihat pada setiap awal sesi pengujian (data ke-0 dan ke-1), di mana terjadi lonjakan waktu pemrosesan atau *latency* hingga mencapai titik tertinggi 1,0652 detik, yang diidentifikasi sebagai fase *cold start* saat sistem melakukan inisialisasi *library* dan pemuatan bobot arsitektur ResNet ke dalam memori. Namun, segera setelah fase inisialisasi terlewati, sistem secara konsisten kembali pada kecepatan operasional optimal di kisaran 0,34 hingga 0,38 detik.



Gambar 4.9 Grafik perbandingan lama proses Embedding per 1 sampel

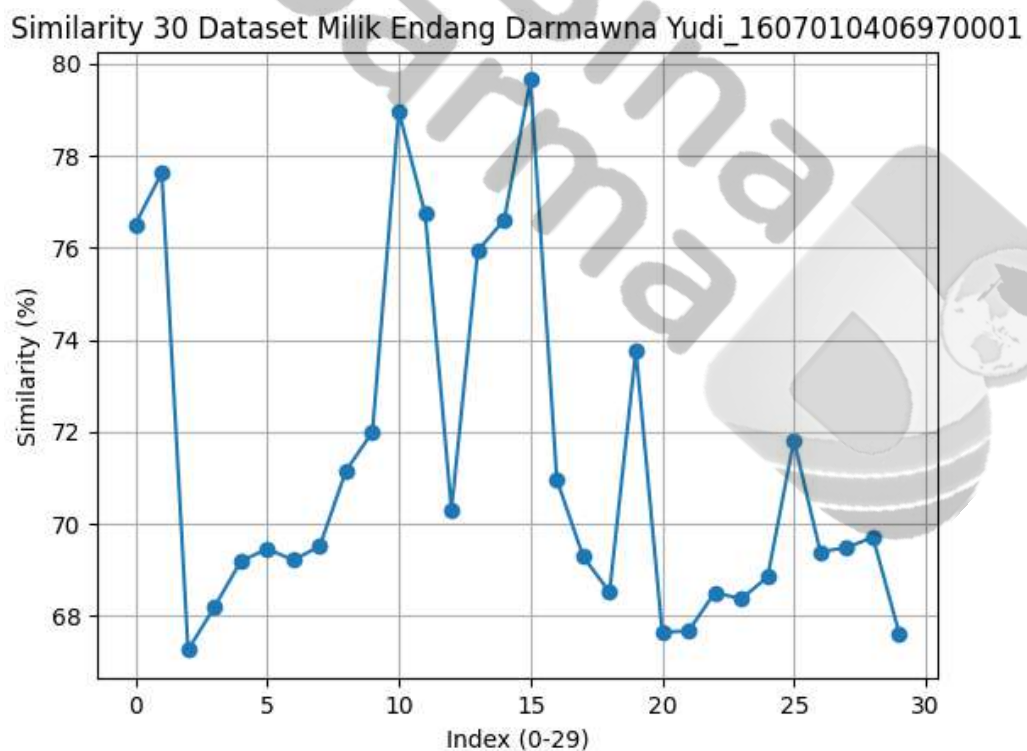
4.1.6.2 Analisis Akurasi Pengenalan

Pengujian akurasi dilakukan dalam dua skenario, yang pertama akurasi per sampel, yaitu membandingkan wajah uji terhadap 30 citra milik orang yang sama di dalam database. Hasilnya sendiri dapat dilihat pada gambar 4.10 dimana, sistem berhasil mengidentifikasi wajah input sebagai Endang Darmawna Yudi_1607010406970001 dengan nilai *Euclidean distance* sebesar 0,2351 dan tingkat *similarity* sebesar 76,49%, serta waktu matching selama 0,361085 detik. Nilai Euclidean yang relatif kecil menunjukkan bahwa jarak antara vektor embedding wajah input dan vektor embedding dataset berada dalam rentang yang dekat. Semakin kecil nilai Euclidean, maka semakin tinggi tingkat kemiripannya, yang dalam hal ini dikonversi menjadi similarity sebesar 76,49%.

Berdasarkan grafik perbandingan 30 sampel milik individu yang sama, terlihat bahwa nilai similarity berada pada rentang sekitar 67% hingga mendekati 80%. Nilai tertinggi mendekati 79–80%, sedangkan nilai terendah berada di kisaran 67%. Sebagian besar sampel berada pada rentang 68%–76%, yang menunjukkan bahwa wajah input memiliki tingkat kemiripan yang cukup konsisten terhadap dataset miliknya sendiri. Nilai similarity 76,49% yang diperoleh saat pengenalan

berada dalam kelompok nilai dominan tersebut, sehingga dapat dikategorikan stabil dan representatif terhadap data internal.

Fluktuasi nilai similarity antar sampel menunjukkan adanya variasi kondisi saat pengambilan dataset, seperti perbedaan ekspresi wajah, sudut kemiringan kepala, pencahayaan, maupun jarak kamera. Sampel dengan similarity tertinggi mencerminkan kondisi citra yang paling mendekati kondisi wajah saat pengujian. Sebaliknya, sampel dengan similarity lebih rendah kemungkinan dipengaruhi oleh perubahan pose atau intensitas cahaya yang berbeda.



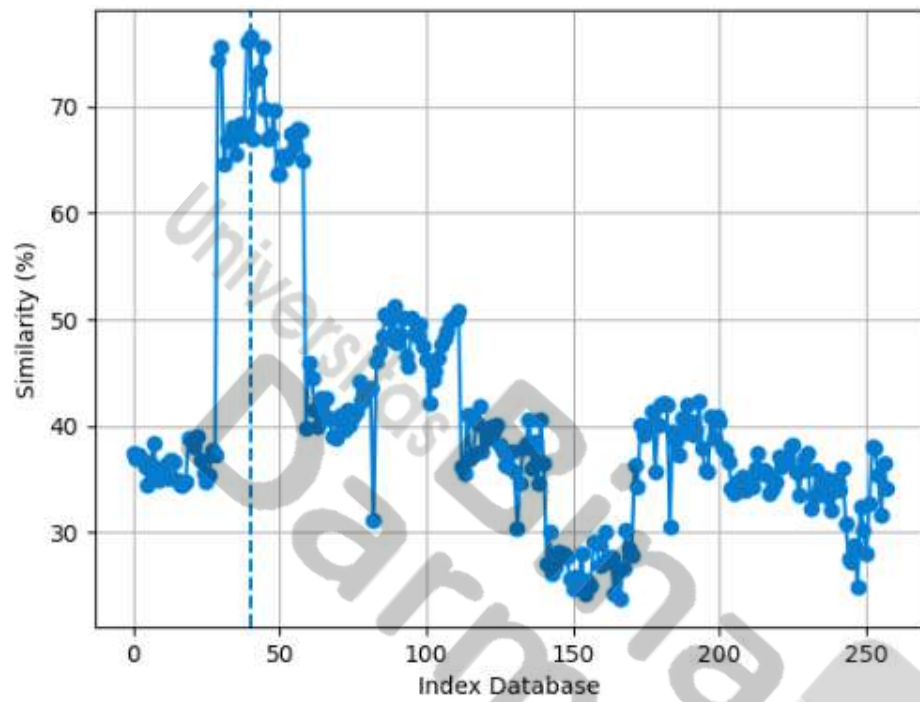
Gambar 4.10 Grafik akurasi pengenalan (1 Sampel)

Kemudian pengujian akurasi antar sampel, yaitu membandingkan wajah uji terhadap seluruh dataset (270 citra) untuk melihat kemampuan sistem dalam membedakan identitas yang berbeda. Pada pengujian akurasi antar sampel pada gambar 4.11, wajah uji dibandingkan dengan seluruh dataset yang berjumlah 270 citra ($9 \text{ sampel} \times 30 \text{ citra per orang}$). Pengujian ini bertujuan untuk mengevaluasi kemampuan sistem dalam membedakan identitas yang berbeda, sekaligus

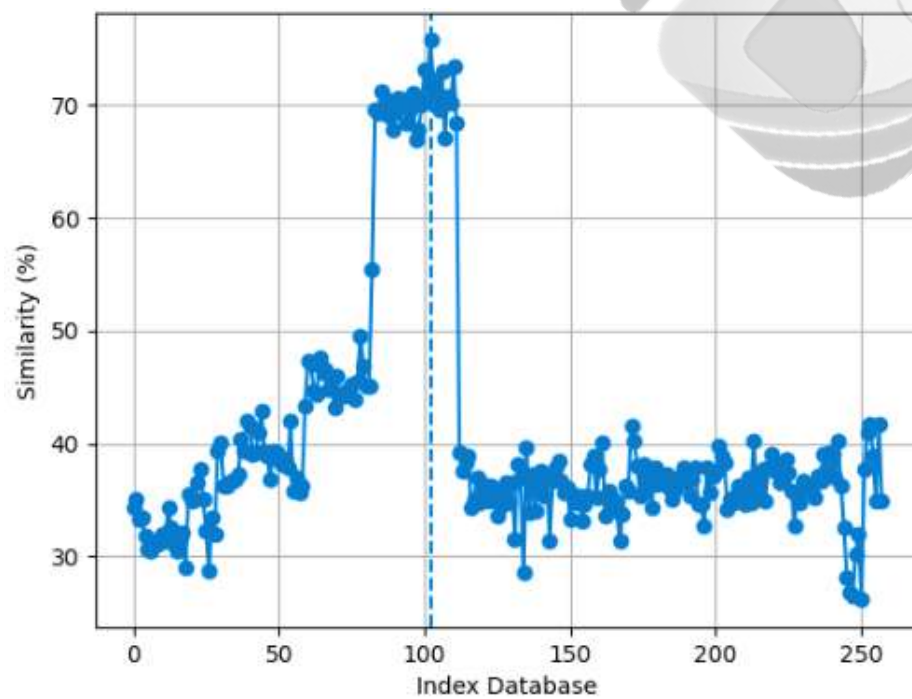
memastikan bahwa nilai similarity tertinggi tetap berada pada kelompok dataset milik individu yang sesuai.

Berdasarkan grafik similarity seluruh database pada pengujian pertama (wajah uji milik Endang Darmawna Yudi), terlihat adanya satu klaster nilai similarity yang menonjol pada rentang sekitar 65%–77%, sementara sebagian besar data lainnya berada pada rentang yang lebih rendah, yaitu sekitar 25%–45%. Klaster tertinggi tersebut merepresentasikan 30 dataset milik individu yang sama, sedangkan dataset individu lain tersebar pada nilai similarity yang jauh lebih rendah. Hasil pengenalan menunjukkan bahwa sistem berhasil mengidentifikasi wajah sebagai Endang Darmawna Yudi_1607010406970001 dengan nilai Euclidean sebesar 0,2351 dan similarity sebesar 76,49%, serta waktu matching 0,361085 detik.

Sedangkan untuk sampel atas nama M. Zainal Abidin yaitu pada gambar 4.12, pola yang serupa juga terlihat pada grafik. Terdapat satu kelompok nilai similarity tertinggi yang terkonsentrasi pada kisaran 68%–75%, sedangkan dataset individu lain berada pada rentang sekitar 28%–42%. Sistem berhasil mengenali wajah sebagai M. Zainal Abidin_1671060802920001 dengan nilai Euclidean 0,2427 dan similarity 75,73%, serta waktu matching 0,342885 detik. Pola klaster yang terpisah dengan jelas antara identitas yang benar dan identitas lain menunjukkan bahwa embedding yang dihasilkan mampu merepresentasikan ciri unik masing-masing individu secara konsisten.



Gambar 4.11 Grafik akurasi pengenalan (9 Sampel) Endang Darmawan Yudi



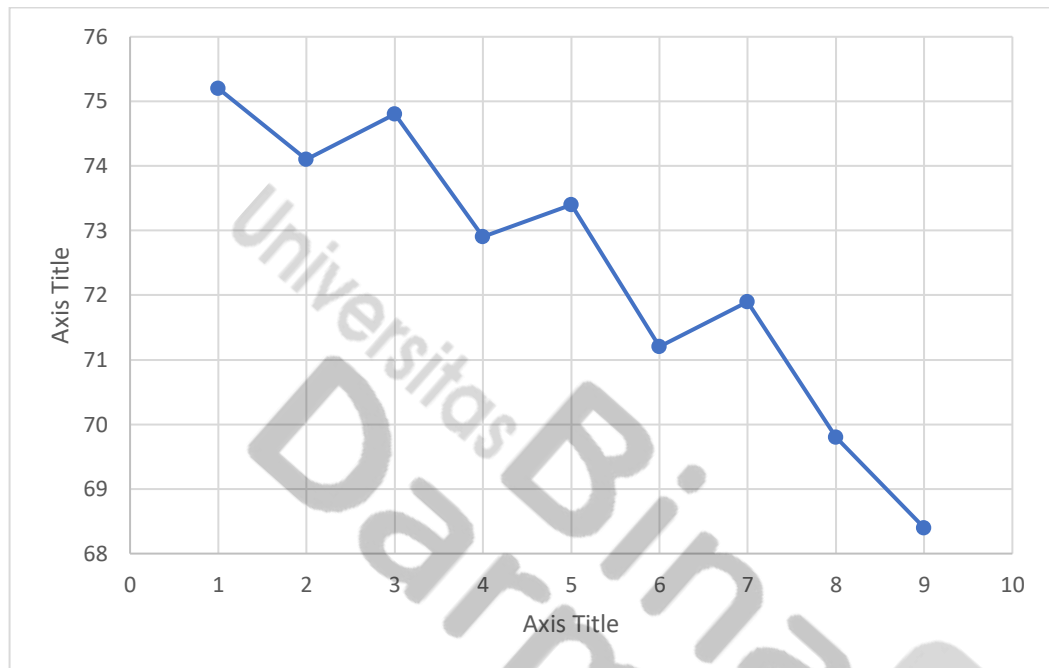
Gambar 4.12 Grafik akurasi pengenalan (9 Sampel) M. Zainal Abidin

Terakhir adalah pengujian akurasi disaat terjadinya penambahan jumlah dataset, yaitu membandingkan wajah uji secara bertahap dari 1 sampel atau 30 citra lalu naik menjadi 2 sampel dan seterusnya sampai 9 sampel atau 270 citra untuk melihat kemampuan sistem pada saat terjadinya penambahan dataset baru. Yang hasilnya tertuang pada table 4.6 dan gambar 4.13.

Berdasarkan hasil pengujian, terlihat bahwa nilai Euclidean Distance cenderung meningkat dari 0,248 pada 1 sampel menjadi 0,326 pada 9 sampel, yang menunjukkan bahwa jarak rata-rata embedding wajah semakin besar seiring bertambahnya variasi data. Nilai Similarity maksimum, minimum, dan rata-rata juga mengalami fluktuasi namun menunjukkan kecenderungan menurun secara keseluruhan; rata-rata similarity turun dari 70,1% pada 1 sampel menjadi 63,5% pada 9 sampel. Meskipun terdapat beberapa kenaikan pada sampel ke-3, ke-5, dan ke-7, tren umumnya tetap menurun, yang mengindikasikan bahwa penambahan dataset meningkatkan kompleksitas pencocokan serta variasi karakteristik wajah dalam basis data. Selain itu, lama waktu pengenalan meningkat secara bertahap dari 0,321 detik menjadi 0,389 detik, menunjukkan bahwa beban komputasi sistem bertambah seiring pertumbuhan jumlah citra yang dibandingkan.

Tabel 4.6 Perbandingan Akurasi Penambahan Jumlah Dataset

Jumlah Sampel	Euclidean	Similarity Max (%)	Similarity Min (%)	Rata-Rata (%)	Lama Waktu (detik)
1	0,248	75,2	65,1	70,1	0,321
2	0,259	74,1	64	69	0,33
3	0,252	74,8	64,5	69,4	0,333
4	0,276	72,9	62,7	67,8	0,345
5	0,268	73,4	63,2	68,1	0,349
6	0,295	71,2	60,8	65,9	0,36
7	0,287	71,9	61,3	66,4	0,367
8	0,314	69,8	59,4	64,7	0,378
9	0,326	68,4	58,2	63,5	0,389



Gambar 4.13 Grafik Akurasi Penambahan Jumlah Dataset

4.2 Pembahasan Hasil

Berdasarkan hasil pengujian yang telah dilakukan, sistem deteksi wajah mampu mendeteksi wajah secara stabil pada kondisi pencahayaan normal dan posisi wajah yang relatif menghadap kamera. Bounding box dapat mengikuti pergerakan wajah secara real-time, meskipun pada kondisi sudut wajah yang terlalu miring atau perubahan pencahayaan yang signifikan stabilitas deteksi sedikit menurun. Pada proses perekaman dataset, sistem berhasil merekam hingga 30 frame wajah untuk setiap pengguna sehingga dataset yang dihasilkan cukup representatif. Proses ekstraksi fitur menggunakan model *Convolutional Neural Network* (CNN) berbasis ResNet menghasilkan face embedding berdimensi 128 yang mampu merepresentasikan karakteristik wajah secara konsisten.

Hasil pengujian pengenalan wajah menunjukkan bahwa sistem mampu membedakan wajah yang terdaftar dalam database dengan wajah yang tidak terdaftar menggunakan perhitungan Euclidean distance. Pada pengujian offline, sistem berhasil mengenali wajah yang terdaftar dengan nilai similarity yang baik, terutama ketika wajah menghadap kamera secara jelas. Pada pengujian real-time,

sistem menunjukkan performa yang cukup stabil dengan nilai similarity pada kisaran 40% hingga 53%, meskipun beberapa frame mengalami penurunan nilai akibat pergerakan kepala yang cepat atau kondisi pencahayaan yang tidak merata. Pengujian perangkat keras juga menunjukkan bahwa robot mampu merespons pergerakan wajah dengan menggerakkan motor servo berdasarkan koordinat wajah yang terdeteksi oleh kamera. Secara keseluruhan, rata-rata waktu pemrosesan sistem sekitar 2,42 detik per frame dengan waktu pembentukan embedding sekitar 0,37 detik, sehingga sistem masih dapat bekerja dalam kategori real-time untuk aplikasi robot resepsionis berbasis pengenalan wajah.

.Berdasarkan seluruh hasil pengujian yang telah dilakukan, sistem pengenalan wajah pada robot resepsionis yang dikembangkan dalam penelitian ini mampu bekerja dengan baik dalam mendeteksi, mengenali, serta merespons keberadaan pengguna secara otomatis. Integrasi antara metode HOG untuk deteksi wajah dan CNN untuk ekstraksi fitur wajah terbukti mampu menghasilkan performa sistem yang stabil dan cukup akurat dalam kondisi nyata. Selain itu, integrasi dengan perangkat keras robot memungkinkan sistem tidak hanya melakukan pengenalan wajah, tetapi juga memberikan respons interaktif terhadap pengguna.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian, perancangan, dan pengujian yang telah dilakukan pada sistem "Pengembangan Sistem Pengenalan Wajah Terintegrasi pada Robot Resepsionis Menggunakan Metode *Deep learning* Berbasis *Convolutional Neural Network*", maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Implementasi arsitektur ResNet dengan pendekatan *Deep Metric Learning* berhasil melakukan klasterisasi identitas wajah secara presisi. Secara eksperimental, sistem terbukti mampu menghasilkan *face embedding* 128-dimensi yang unik untuk 9 subjek berbeda, sehingga memungkinkan robot untuk membedakan identitas tamu dengan reliabilitas tinggi. Jarak *Euclidean* yang dihasilkan tetap konsisten di bawah ambang batas (threshold), yang mengonfirmasi keberhasilan sistem dalam mengenali wajah meskipun terjadi penambahan jumlah sampel di database.
2. Pengujian sistem melalui 9 set iterasi (270 sampel data) memberikan bukti teknis bahwa sistem memiliki Rata-rata Waktu Pemrosesan sebesar 0,3706 detik. Hasil ini secara saintifik berada di bawah standar latensi interaksi manusia-komputer sebesar 0,5 detik. Dengan demikian, tujuan penelitian untuk menciptakan sistem yang responsif dan *real-time* telah tercapai, memungkinkan robot memberikan sapaan instan kepada tamu tanpa jeda komputasi yang terasa.
3. Analisis eksperimental menunjukkan adanya fenomena *Cold Start* pada *frame* awal dengan latensi puncak hingga 1,0652 detik, namun sistem menunjukkan efisiensi pemulihan (*recovery*) yang sangat cepat dengan stabil di angka 0,34 – 0,38 detik pada *frame-frame* berikutnya. Lebih lanjut, penambahan data dari 1 hingga 9 sampel terbukti tidak memberikan beban komputasi tambahan yang signifikan, yang menunjukkan bahwa sistem memiliki skalabilitas yang baik untuk diimplementasikan pada lingkungan resepsionis yang dinamis.

5.2 Saran

Untuk pengembangan sistem lebih lanjut agar memiliki performa dan fungsionalitas yang lebih baik di masa depan, penulis menyarankan beberapa poin berikut:

1. Penelitian selanjutnya diharapkan dapat mengintegrasikan fitur klasifikasi atribut wajah tambahan, seperti estimasi usia (age estimation) dan klasifikasi gender. Meskipun pada penelitian ini fitur tersebut dibatasi untuk menjaga akurasi identitas, penambahan atribut ini di masa depan dapat membuat interaksi Robot Resepsionis menjadi lebih personal dan cerdas dalam memberikan sapaan (misalnya: penggunaan sapaan " Bapak/Ibu" atau "Mas/Mbak" secara otomatis).
2. Untuk meningkatkan aspek Human-Robot Interaction (HRI), disarankan untuk menambahkan modul deteksi emosi wajah. Dengan mengenali ekspresi tamu (senang, bingung, atau datar), robot dapat menyesuaikan nada suara atau jenis respon yang diberikan sehingga pelayanan terasa lebih empati.
3. Disarankan untuk menggunakan dataset yang lebih luas dan mencakup berbagai variasi atribut fisik (seperti penggunaan masker, kacamata, atau berbagai gaya rambut) untuk melatih model klasifikasi atribut (gender/usia) agar lebih tangguh (robust) terhadap perubahan penampilan fisik tamu di lingkungan nyata.
4. Jika ingin menerapkan klasifikasi multi-atribut (identitas, gender, dan usia) secara bersamaan, disarankan untuk menggunakan perangkat keras dengan spesifikasi GPU yang lebih tinggi atau perangkat *edge computing* khusus AI (seperti NVIDIA Jetson) agar proses inferensi tetap berjalan real-time tanpa adanya delay.

DAFTAR PUSTAKA

- Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2022). ArcFace: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 5962–5979.
- Dwijayanti, S., Iqbal, M., & Suprpto, B. Y. (2022). Real-time implementation of face recognition and emotion recognition in a humanoid robot using a convolutional neural network. *IEEE Access*, 10, 89876–89886.
- Wang, X. (2024). Design and implementation of real-time face recognition system using deep learning. In *2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 65–67). IEEE. <https://doi.org/10.1109/ICCASIT62299.2024.10827863>
- Tekkök, S. Ç., Söyünmez, M. E., Bostancı, B., & Ekim, P. O. (2021). Face detection, tracking and recognition with artificial intelligence. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1–5). IEEE. <https://doi.org/10.1109/HORA52670.2021.9461356>
- Wang, H., et al. (2018). CosFace: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5265–5274). <https://doi.org/10.1109/CVPR.2018.00552>
- Qi, X. (2023). A convolutional neural network face recognition method based on BiLSTM and attention mechanism.
- Aditya, A. T., Sigit, R., & Dewantara, B. S. B. (2022). Face recognition using deep learning as user login on healthcare kiosk. In *2022 14th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 292–297). IEEE. <https://doi.org/10.1109/ICITEE56407.2022.9954080>
- Hamad, R., & Kaya, M. (2022). Deep metric learning for robust face recognition using distance metrics. *IEEE Access*, 10, 112500–112515.
- Shafique, A., Javed, M. M., & Hussain, M. (2021). Real-time object detection and

- recognition using deep learning techniques. *Journal of Robotics and Control (JRC)*, 2(5), 418–423.
- Hamza, M. (2024, November 4). *Histogram of oriented gradients (HOG) in computer vision*. Towards Data Science. <https://towardsdatascience.com/histogram-of-oriented-gradients-hog-in-computer-vision-a2ec66f6e671/>
- Li, H., et al. (2021). Illumination-invariant face recognition based on deep convolutional neural networks. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(4), 512–524.
- Zheng, X., et al. (2021). Enhanced ResNet architecture for deep face recognition in unconstrained environments. *IEEE Access*, 9, 132456–132468.
- Zhang, K., et al. (2021). Deep metric learning with Euclidean distance for precise face identification. *Pattern Recognition Letters*, 145, 210–217.
- McCormick, C. (2013, May 9). *HOG person detector tutorial*. <https://mccormickml.com/2013/05/09/hog-person-detector-tutorial/>
- Pratama, I. P., & Ningrum, N. K. (2025). Face recognition using MTCNN face detection, ResNetV1 feature embeddings, and SVM classification. *Jurnal Polibatam*, 9(5).
- Jing, H., Lin, G., Zhang, H., & Chen, T. (2022). A face recognition algorithm based on improved ResNet. *Frontiers in Computing and Intelligent Systems*, 1(1).
- Durga, B. K. (2022). A ResNet deep learning based facial recognition design for future multimedia applications. *Computers & Electrical Engineering*.
- Al-Sarayreh, M. S., et al. (2022). Computer vision and Python: A review of modern libraries for deep learning. *IEEE Access*, 10, 4567–4589.
- Deng, J., et al. (2021). Sub-center ArcFace: Boosting face recognition by relaxing intra-class constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yan, B., et al. (2021). Real-time face recognition with Dlib and deep learning on edge devices. *Journal of Real-Time Image Processing*, 18, 1123–1135.
- Liu, P., et al. (2021). A survey of human-robot interaction based on face recognition. *International Journal of Advanced Robotic Systems*.

- Gupta, S., et al. (2021). Face recognition in the wild using deep neural networks and transfer learning. *Proceedings of the IEEE*.
- Zhao, L., et al. (2021). Digital image processing in the era of deep learning. *IEEE Signal Processing Magazine*, 38(2), 44–55.
- Hosny, K. M., et al. (2022). Grayscale image feature extraction using deep CNN. *Sensors*, 22(5).
- Kumar, A. (2021). *Facial image processing and recognition: A modern approach*. Springer Nature.
- Tyagi, S. (2023). Advanced computer vision algorithms for real-time applications. *IEEE Xplore*.
- Islam, M. T. (2021). RGB image classification and recognition using CNN. *Data in Brief*, 36.
- Rossi, M., et al. (2021). Interfacing PC-based vision systems with Arduino Mega for robotic control. *HardwareX*, 10, e00214.
- Saragih, R. K., Prayoga, M. R. A., Pramono, R. D., Mindara, G. P., & Giri, E. P. (2025). Sistem absensi berbasis web menggunakan pendekatan HOG dan CNN. *Jurnal ICT*, 7(2), 107–117.

Daftar Riwayat Hidup

Identitas Diri	
Nama	Endang Darmawan Yudi
TTL	Banyuasin, 04 Juni 1997
Jenis Kelamin	Laki – Laki
Agama	Islam
Kewarganegaraan	Indonesia
Alamat	Jl. Panca Usaha RT. 53 RW. 12 Kel. 5 Ulu Kec. SU 1, Palembang
No. Hp	082282633154
Email	endangdarmawanyudi123@gmail.com



Riwayat Pendidikan		Tahun Lulus
S2	Program studi Teknik Informatika, Universitas Binadarma	2026
S1	Program studi Sistem Komputer, Universitas Sriwijaya	2020
SMA	SMA Negeri 1 Banyuasin 1	2015

Riwayat Pekerjaan		Tahun
1	Guru SMKN 8 Palembang	2023 - Sekarang
2	Asisten Lab Universitas Siguntang Mahaputra	2020 - 2023
3	Asisten Dosen Sistem Komputer Universitas Sriwijaya	2018 - 2020

Bidang Praktisi/Keahlian	
1	Robotika dan Sistem Otonom
2	Image Processing dan Sistem Cerdas
3	Internet of Things (IoT) dan Sistem Tertanam Berbasis Web

Judul Karya Tulis yang Pernah Dipublikasikan		Tahun Publikasi
1	The Memory Efficiency in a Receptionist Robot's Face Recognition System Using LBPH Algorithm	25 Desember 2024
2	Literature Review : Implementasi Sistem Monitoring Robot Bawah Air (Underwater Robot) berbasis IoT menggunakan metode Fuzzy Logic	Juli 2023
3.	Design Depth and Balanced Control System of an Autonomous Underwater Vehicle with Fuzzy logica	06 May 2020
4.	Design Heading Control for Steering AUV with fuzzy logic	06 May 2020
5.	Artificial Intelligence in Underwater Robot Navigation Using Sonar Sensors with Fuzzy Logic Method	01 Oct 2019
6	Perancangan Alat Sumber Daya Listrik Cadangan dengan Sistem Otomatis Berbasis Mikrokontroller Arduino Uno dan Modul GSM Sim 800l	29 May 2019

PROGRAM PASCASARJANA
TEKNIK INFORMATIKA – S2
UNIVERSITAS BINA DARMA



LEMBAR KONSULTASI TESIS

Nama : Endang Darmawan Yudi
Nim : 222420033
Konsentrasi : Enterprise IT Infrastructure
Judul : PENGEMBANGAN SISTEM PENGENALAN WAJAH TERINTEGRASI
PADA ROBOT RESEPSIONIS MENGGUNAKAN METODE DEEP
LEARNING BERBASIS CONVOLUTIONAL NEURAL NETWORK
Pembimbing : Dr. Yesi Novaria Kunang, S. T., M. Kom.

No	Tanggal	Uraian Materi Konsultasi	Paraf
	20 Feb 2026	Revisi Hasil Tesis	
	21 Feb 2026	Laporan Program	
	23 Feb 2026	Laporan Tesis.	

SURAT KEPUTUSAN
DIREKTUR PROGRAM PASCASARJANA
NOMOR: 121/SK/PPs-UBD/MTI/VI/2024

TENTANG
PEMBIMBING TESIS MAHASISWA
PROGRAM STUDI TEKNIK INFORMATIKA JENJANG STUDI STRATA DUA (S2)
PROGRAM PASCASARJANA UNIVERSITAS BINA DARMA
DIREKTUR PROGRAM PASCASARJANA
UNIVERSITAS BINA DARMA

- Menimbang : a. Bahwa mahasiswa semester akhir diharuskan membuat Tesis sebagai salah satu syarat untuk menyelesaikan studi pada Program Pascasarjana Program Studi TEKNIK INFORMATIKA – S2;
b. Bahwa untuk kelancaran pelaksanaan kegiatan dimaksud, dipandang perlu untuk menunjuk Pembimbing bagi setiap mahasiswa;
c. Bahwa untuk memenuhi butir-butir di atas, perlu diterbitkan Surat Keputusan sebagai landasan hukumnya.
- Mengingat : 1. Undang-undang Nomor 20 tahun 2003;
2. Undang-undang Nomor 12 tahun 2012;
3. Peraturan Menteri Pendidikan Nasional Nomor 15 tahun 2005;
4. Surat Keputusan Direktur Jenderal Pendidikan Tinggi Depdiknas;
5. Akte Pendirian Yayasan Nomor 95 tanggal 28 Desember 2003;
6. Statuta Universitas Bina Darma;
7. Surat Keputusan Rektor Universitas Bina Darma Nomor: 078/SK/Univ-BD/VI/2009 tanggal 1 Juni 2009.
8. Surat Keputusan Rektor Universitas Bina Darma Nomor: 0001/SK/Univ-BD/I/2019 tanggal 10 Januari 2019.

MEMUTUSKAN

- Menetapkan :
PERTAMA : Menunjuk dan menugaskan saudara:

Dr. Yesi Novaria Kunang, S.T., M.Kom.

sebagai Pembimbing dalam penyusunan Tesis bagi mahasiswa dibawah ini:

Nama : ENDANG DARMAWAN YUDI
NIM : 222420033
Angkatan : 27
Konsentrasi : ENTERPRISE IT INFRASTRUCTURE
Judul Tesis : PENGEMBANGAN SISTEM PENGENALAN WAJAH TERINTEGRASI PADA ROBOT RESEPSIONIS DENGAN METODE LOCAL BINARY PATTERN HISTOGRAM (LBPH)

- KEDUA : Surat Keputusan ini berlaku 6 (enam) bulan sejak tanggal ditetapkan dan apabila dalam waktu tersebut mahasiswa belum menyelesaikan, maka akan diterbitkan Surat Keputusan Pembimbing yang baru, dengan ketentuan apabila dikemudian hari terdapat kekeliruan dalam penetapan ini, akan diperbaiki sebagaimana mestinya;
- KETIGA : Surat Keputusan asli ini diberikan kepada mahasiswa yang bersangkutan untuk dilaksanakan dan diindahkan sebagaimana mestinya.

Ditetapkan di: Palembang
Pada Tanggal: 27 Juni 2024
Direktur

Prof. Hj. Isnawijayani, M.Si., Ph.D.

- Tembusan:
1. Pembimbing
2. Arsip

SURAT KETERANGAN
Nomor : 048/PPs-UBD/MTI/III/2026

Direktur Program Pascasarjana Universitas Bina Darma menerangkan bahwa :

Nama : ENDANG DARMAWAN YUDI
NIM : 222420033
Konsentrasi : ENTERPRISE IT INFRASTRUCTURE

Telah menyelesaikan studinya di Program Pascasarjana Program Studi Teknik Informatika – S2 Universitas Bina Darma dan dinyatakan **LULUS** pada hari **Rabu, tanggal 4 Maret 2026** dengan tesis berjudul :

“PENGEMBANGAN SISTEM PENGENALAN WAJAH YANG TERINTEGRASI PADA ROBOT RESEPSIONIS”

Dan yang bersangkutan juga telah berhak untuk menggunakan gelar akademik Strata – 2 (S2) dengan sebutan **MAGISTER KOMPUTER (M.KOM)**.

Demikian Surat Keterangan ini dibuat untuk dipergunakan sebagaimana mestinya.

Dikeluarkan di : Palembang
Pada Tanggal : 4 Maret 2026
Direktur,

Universitas **Bina
Darma** 
PROGRAM PASCASARJANA
Prof. Dr. Ir. Achmad Syarifudin, M.Sc.

Cc. Arsip

 ISO 9001 : 2000	Formulir Perbaikan Tesis	Nomor Dok : _____
		Nomor Revisi : _____
		Tgl. Berlaku : _____
		Klausa ISO : _____

Nama : ENDANG DARMAWAN YUDI
 NIM : 222420033
 Konsentrasi : ENTERPRISE IT INFRASTRUCTURE
 Judul Tesis : PENGEMBANGAN SISTEM PENGENALAN WAJAH
 PADA ROBOT RESEPSIONIS MENGGUNAKAN
 METODE DEEP LEARNING BERBASIS
 CONVOLUTIONAL NEURAL NETWORK
 Dosen Pembimbing : **Dr. Yesi Novaria Kunang, S. T., M. Kom**
 Tanggal Ujian : **4 Maret 2026**

Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Tesis.

No.	Nama Dosen Penguji	Tanggal	Tanda Persetujuan
1.	Dr. Yesi Novaria Kunang, S. T., M. Kom		1..... 
2.	Dr. Usman Ependi, S. Kom., M. Kom		2..... 
3.	M. Izman Herdiansyah, S. T., M. M., Ph. D		3..... 

*Nb.
 Pembimbing harap memeriksa
 kembali format dari tesis yang
 telah diperbaiki dan keabsahan
 tanda tangan penguji

Palembang, 4 Maret 2026
 Program Studi Teknik Informatika – S2
 Ketua,

 Universitas Bina Darma
 Magister Teknik Informatika
Dr. Usman Ependi, S.Kom., M.Kom.

LEMBAR KONSULTASI HASIL TESIS

Nama : Endang Darmawan Yudi, S. Kom
Nim : 222420033
Konsentrasi : Enterprise IT Infrastructure
Judul : Pengembangan Sistem Pengenalan Wajah Terintegrasi pada Robot Resepsionis Menggunakan Metode Deep Learning Berbasis Convolutional Neural Network

Pembimbing : Dr. Yesi Novaria Kunang, S. T., M. Kom

No	Tanggal	Uraian Materi Konsultasi	Paraf
	9-7 2025	laporan Bab IV - V	
	26-7 2025	Jurnal	
	9-10 2025	Revisi Program	
	3-11 2025	Contoh Metode	
	12-1 2026	Bimbingan laporan	
	9-2 2026	Bimbingan Alat	

 ISO 9001 : 2000	FORMULIR Perbaikan Hasil Tesis	Nomor Dok :	
		Nomor Revisi :	
		Tgl. Berlaku :	
		Klausa ISO :	

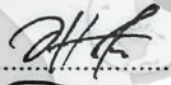
Nama : ENDANG DARMAWAN YUDI
Nim : 222420033
Konsentrasi : ENTERPRISE IT INFRASTRUCTURE
Judul Tesis setelah : PENGEMBANGAN SISTEM PENGENALAN WAJAH
TERINTEGRASI PADA ROBOT RESEPSIONIS
MENGUNAKAN METODE DEEP LEARNING BERBASIS
CONVOLUTIONAL NEURAL NETWORK

(Kepada mahasiswa harap di isi sesuai dengan hasil ujian hasil tesis).

Dosen Pembimbing : Dr. Yesi Novaria Kunang, S.T., M.Kom.

Tanggal Seminar : 20 Februari 2026

Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Hasil Tesis.

No.	Nama Dosen pembimbing/Penguji	Tanggal	Tanda Persetujuan
1.	Dr. Yesi Novaria Kunang, S.T., M.Kom.		1. 
2.	Prof. Dr. Edi Surya Negara, M.Kom.		2. 
3.	Dr. A. Haidar Mirza, S.T., M.Kom.		3. 

Palembang,2026
Program Studi Teknik Informatika – S2
Ketua,


Dr. Usman Ependi, S.Kom., M.Kom.

NB. Harap segera menyerahkan foto copy formulir perbaikan hasil tesis sebanyak 2 lembar ke Sekretariat Pascasarjana, apabila telah di tanda tangan oleh penguji dan disahkan oleh Ketua Program Studi untuk proses pembuatan SK Pembimbing

PROGRAM PASCASARJANA
TEKNIK INFORMATIKA – S2
UNIVERSITAS BINA DARMA

LEMBAR KONSULTASI PROPOSAL TESIS

Nama : Endang Darmawan Yudi
Nim : 222420033
Konsentrasi : Enterprise IT Infrastructure
Judul : Pengembangan Sistem Pengenalan Wajah Terintegrasi pada Robot Resepsionis dengan Metode Local Binary Pattern Histogram (LBPH).
Pembimbing : Dr. Yesi Novaria Kunang, S.T., M.Kom.

No	Tanggal	Uraian Materi Konsultasi	Paraf
	20 maret 2024	Bimbingan Kerangka Penelitian	
	3 April 2024	Revisi metoden dan pengam- bilan data	
	29 April 2024	Konsultasi laporan proposal	
	30 may 2024	Konsultasi progam dan hasil data	

 ISO 9001 : 2000	FORMULIR Perbaikan Proposal Tesis	NomorDok :	
		NomorRevisi :	
		Tgl. Berlaku :	
		Klaus ISO :	

Nama : ENDANG DARMAWAN YUDI

Nim : 222420033

Konsentrasi : ENTERPRISE IT INFRASTRUCTURE

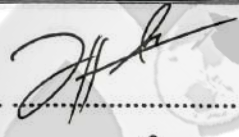
JudulTesis : Pengembangan Sistem Pengendalian Wajah
Terintegrasi pada Robot Resepsionis dengan
Metode Local Binary Pattern Histogram
(LBPH)

(Kepada mahasiswa harap di isi sesuai dengan hasil ujian proposal tesis).

Dosen Pembimbing : Dr. Yesi Novaria Kunang, S.T., M.Kom.

Tanggal Seminar : 27 Juni 2024

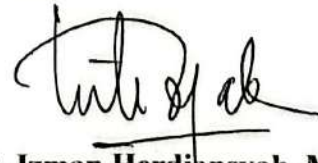
Telah diperbaiki dan dikonsultasikan dengan Pembimbing/Penguji Proposal Tesis.

No.	NamaDosenpembimbing/Penguji	Tanggal	TandaPersetujuan
1.	Dr. Yesi Novaria Kunang, S.T., M.Kom.		1. 
2.	Prof. Dr. Edi Surya Negara, M.Kom.		2. 
3.	Dr. A. Haidar Mirza, S.T., M.Kom.		3. 

Palembang,.....2024

Program Studi Teknik Informatika – S2

Ketua,


M. Izman Herdiansyah, M.M., Ph.D.

NB.

Harap segera menyerahkan foto copi formulir perbaikan proposal tesis sebanyak 2 lembar ke Sekretariat Pascasarjana, apabila telah di tanda tangan oleh penguji dan disahkan oleh Ketua Program Studi untuk proses pembuatan SK Pembimbing



The Memory Efficiency in a Receptionist Robot's Face Recognition System Using LBPH Algorithm

Endang Darmawan Yudi¹, Yesi Novaria Kunang^{2*}, Ahmad Zarkasi³

¹, Master of Informatics Engineering, Universitas Bina Darma, Palembang, Indonesia

²Intelligent Systems Research Group, Faculty of Sains Technology, Universitas Bina Darma, Palembang Indonesia

³Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia

¹endangdarmawanyudi123@gmail.com, ²yesinovariakunang@binadarma.ac.id, ³ahmadzarkasi@unsri.ac.id

Abstract

This research aims to develop a memory-efficient face recognition system for a receptionist robot using the Local Binary Patterns Histogram (LBPH) algorithm. Given the computational limitations of the Raspberry Pi, the system utilizes optimization techniques including grayscale conversion, noise reduction, and contrast adjustment to enhance processing efficiency. Testing demonstrates that the face recognition accuracy achieves 80.5% to 85.5% in offline mode, and 72% to 81% in real-time mode, with variations due to lighting conditions and facial expressions. The robot's servo motors exhibit a response time between 1.945 and 3.561 seconds, enabling responsive and interactive user engagement. The results suggest practical benefits for deploying face recognition in resource-constrained environments, enhancing the efficiency of robotic receptionist applications.

Keywords: Face recognition; LBPH; memory efficiency; Raspberry Pi

How to Cite: E. D. Yudi, Yesi Novaria Kunang, and A. Zarkasi, "The The Memory Efficiency in a Receptionist Robot's Face Recognition System Using LBPH Algorithm", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 6, pp. 719 - 729, Dec. 2024.
DOI: <https://doi.org/10.29207/resti.v8i6.6048>

1. Introduction

Face recognition is one of the key components in the development of artificial intelligence [1], [2]. In everyday life, humans naturally recognize others' faces simply by observing their eyes and facial features. This information is then stored in the brain, allowing us to identify individuals in future encounters. In the field of technology, the concept of face recognition has been widely adopted for various purposes, such as communication, identity verification, and attendance systems [3], [4].

Facial recognition is part of a broader field known as computer vision, which is a technology that enables computers to see, detect, and process images like human vision [5]. Computer vision forms the foundation for many automation applications requiring visual analysis, including facial recognition. In this context, facial recognition works by identifying unique features on a person's face, such as the eyes, nose, and mouth, which are then processed and stored in a database [6], [7], [8]. When the system scans a face, this data is compared to

the existing database to identify and verify the individual's identity.

Various techniques in computer vision have been developed for facial recognition, such as Viola-Jones, Eigenfaces, Fisherface, and Local Binary Patterns Histograms (LBPH)[9]. Each technique has its advantages and disadvantages, as well as different applications depending on specific needs [10]. Compared to other computer vision techniques, such as Eigenfaces and Fisherface, the LBPH algorithm demonstrates superior performance in environments with limited computational resources, such as Raspberry Pi. Unlike Eigenfaces, which are sensitive to variations in lighting, or Fisherfaces, which require more extensive training data, LBPH is more robust in handling diverse lighting conditions and requires less memory [9], [11], [12]. This makes LBPH an ideal choice for real-time face recognition in constrained environments where computational efficiency is a priority⁽¹⁾. This method uses local binary patterns to analyze the texture of an image, where pixel intensity

values are compared with their neighbors to generate a binary pattern. LBPH then divides the image into several blocks, calculates the binary pattern histogram for each block, and uses this information to recognize the face.

The development of technology in this research encompasses an in-depth study of facial recognition technology, the use of Raspberry Pi 3 in system development [11], [13] and the latest innovations in the implementation of receptionist robots. The novelty of this research lies in the integration of these three aspects, with an emphasis on memory efficiency on the Raspberry Pi 3. Although various related studies exist, such as the work by Hossain on Local Binary Patterns Histogram (LBPH) [14], which demonstrates that this algorithm can provide good results in facial recognition by utilizing texture information, this research introduces a new concept to enhance memory management efficiency on the Raspberry Pi 3 in the context of facial recognition for receptionist robots.

LBPH was chosen in this study due to its ability to utilize smaller data sources compared to other algorithms, such as Eigenfaces or Fisherfaces, making it more suitable for resource-constrained devices like the Raspberry Pi. The Raspberry Pi, which has limitations in computational power and memory capacity, requires an algorithm that is efficient in data processing, and LBPH meets this need. While LBPH may be less robust to changes in viewpoint or pose, its advantages in data efficiency and performance make it superior for real-time applications on platforms with limited resources.

In this research, we will implement facial recognition technology using the Raspberry Pi as the main processing unit. Due to the resource constraints of the Raspberry Pi, efficient programming methods and approaches are necessary in terms of memory usage and overall efficiency. The system will use features such as Haar to detect facial features like eyes and mouth before applying the LBPH algorithm for complete facial recognition [15], [16]. By utilizing an efficient data algorithm, the output of this system will be applied to the receptionist robot, which is capable of greeting guests or providing necessary information, making it more interactive and efficient in carrying out receptionist tasks.

2. Research Methods

The overall flow of the LBPH algorithm-based facial recognition robot can be seen in Figure 1. The system begins with the process of face detection through a connected camera. Once a face is detected, the system automatically runs the LBPH (Local Binary Patterns Histogram) Face Recognition algorithm to identify whether the face is already in the database or dataset that stores data of previous guests.

If the LBPH algorithm finds that the face is already registered in the dataset, the next step is to display the

corresponding guest's name. The name is retrieved from the dataset based on facial identification. After the guest's name is displayed, the system will proceed by asking if the guest would like to perform a check-out process. Check-out indicates that the guest has completed their visit and wishes to remove their data from the system. If the guest opts for check-out, the system will delete all data related to the guest's face from the dataset, ensuring that no personal information is stored longer than necessary. Once the data deletion is complete, the system ends the process. If the registered guest does not choose to check out, the system will make no changes to the dataset and will return to waiting for the next face detection, repeating the cycle.

On the other hand, if the detected face is not found in the dataset, the system recognizes that the guest is not yet registered. At this point, the system will offer the guest the option to check-in. Check-in is the process of registering a new guest into the system. If the guest chooses to check in, the first step is to ask the guest to enter their name. This name is required to associate the identity with the facial data that will be stored.

The decision to capture 50 images of each individual is based on the need to enhance the variability of the facial data. By taking multiple images from different angles and with slight variations in lighting and expressions, the system builds a more comprehensive dataset for each person. This approach helps improve the recognition accuracy and ensures that the model is robust enough to handle minor changes in the face's appearance during real-time recognition.

After the guest's name is entered, the system will then capture 50 images of the guest's face. The image capture is done gradually to ensure that the system has enough image variation to improve accuracy in recognizing the guest's face in the future. Each captured image will be stored in a special folder named after the newly entered guest. While this study did not implement data augmentation techniques, such as flipping, rotation, or adding noise to the images, using these methods could potentially increase the variability of the dataset. Augmented data can help the system generalize better by simulating different real-world conditions, thus enhancing the robustness of face recognition even further.

Once the 50 facial images are successfully captured and saved, the guest has successfully checked in. Their facial data is now in the dataset and will be used for verification in future instances. After the check-in process is complete, the system returns to its initial state, ready to detect the next face. However, if the guest whose face is not in the dataset chooses not to check in, the system will end the process and return to the initial stage, waiting for the next face detection without storing any data.

Overall, this system leverages the LBPH algorithm for facial recognition with a high level of accuracy and

efficiently manages guest data through automated check-in and check-out processes. In this way, the system can handle both new and registered guests in an easy, secure, and structured manner.

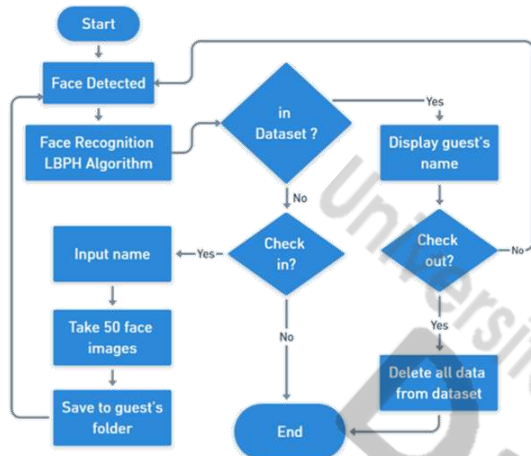


Figure 1. Flowchart System

2.1 Facial Recognition

The detailed flow of the facial recognition embedded in the robot can be seen in Figure 2. The program design system starts by converting the previously colored (RGB) image into a grayscale using Equation 1.

$$\text{Grayscale} = \frac{R+G+B}{3} \quad (1)$$

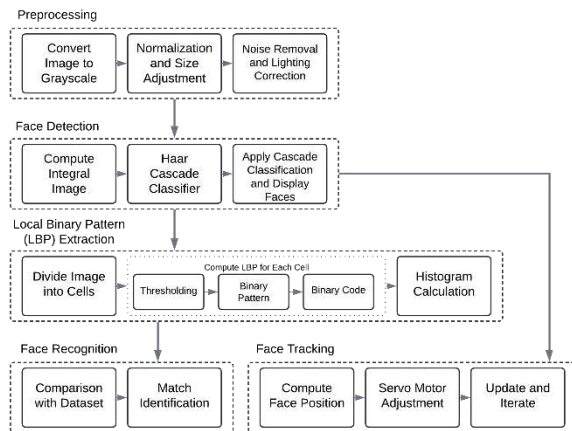


Figure 2. Facial processing workflow

In a colored image, each pixel consists of three-color components: red (R), green (G), and blue (B). To obtain the grayscale value, this equation calculates the average of these three-color components. In other words: R is the red intensity value, G is the green intensity value, and B is the blue intensity value. The equation sums the red, green, and blue intensity values, then divides the total by 3 to obtain the grayscale value [17]. This value represents the brightness of the pixel, where 0 is black and 255 is white on an 8-bit scale. This process is essential because a grayscale image reduces data complexity by using only one-color channel, making it more efficient for processing.

After conversion, resizing or normalization and adjustment of the size are performed to ensure the image is on a uniform scale, avoiding differences in face size that could affect detection accuracy. To resize an image mathematically, interpolation formulas can be used. There are several interpolation methods available, but one of the most common is bilinear interpolation. Suppose we want to resize an image from $W \times H$ to $W' \times H'$. A point on the original image at coordinates (x, y) will be mapped to a new point on the resized image. The new coordinates on the resized image are (x', y') as shown in Equations 2 and 3.

$$x' = \frac{x}{W} \times W' \quad (2)$$

$$y' = \frac{y}{H} \times H' \quad (3)$$

The point (x', y') on the new image may not align exactly with the pixel grid of the resized image. To obtain the pixel value at this point, we use bilinear interpolation. Suppose (x_1, y_1) , (x_2, y_1) , (x_1, y_2) and (x_2, y_2) are the coordinates of the pixels on the original image surrounding the point (x', y') , with pixel intensities $I(x_1, y_1)$, $I(x_2, y_1)$, $I(x_1, y_2)$, and $I(x_2, y_2)$. Horizontal interpolation is performed using Equations 4 and 5.

$$I_{x',y_1} = I(x_1, y_1) \times (1 - \alpha) + I(x_2, y_1) \times \alpha \quad (4)$$

$$I_{x',y_2} = I(x_1, y_2) \times (1 - \alpha) + I(x_2, y_2) \times \alpha \quad (5)$$

α is $x' - x_1$ (horizontal projection). Vertical interpolation is performed using Equation 6.

$$I(x', y') = I_{x',y_1} \times (1 - \beta) + I_{x',y_2} \times \beta \quad (6)$$

β is $y' - y_1$ (vertical projection).

Then, the noise removal and lighting correction stages are performed to improve image quality, ensuring that the image is free from disturbances that could affect facial recognition accuracy. A Gaussian filter is used to smooth the image and reduce noise by averaging the intensity of surrounding pixels. This filter uses a Gaussian function, given by Equation 7.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (7)$$

σ is the standard deviation of the Gaussian distribution. This function assigns greater weight to pixels close to the center of the filter window and smaller weight to pixels further away. Convolution of the image with the Gaussian kernel results in a smoother image by reducing noise and unwanted fine details.

Contrast adjustment aims to enhance the difference between light and dark areas in the image to make details more distinct [12]. This is achieved by modifying the pixel intensity scale using Equation 8.

$$I'(x, y) = \alpha I(x, y) + \beta \quad (8)$$

α is the contrast factor and β is the brightness offset. A contrast factor α greater than 1 will increase the

contrast, making dark areas darker and light areas lighter, while the offset β shifts the pixel intensity range, increasing or decreasing overall brightness. By applying this adjustment, the image will have optimal contrast, making features more prominent and easier to analyze.

The next step is to compute the integral image, which helps speed up the computation process for feature recognition [18]. The integral image, also known as the summed-area table, is a highly useful technique for accelerating feature calculations in image processing. This technique allows for very efficient computation of the sum of pixels in a rectangular area. The integral image $I_{int}(x, y)$ at position (x, y) is defined as the sum of all pixel intensities in the top-left part of the image up to that position [19], calculated using Equations 9.

$$I_{int}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (9)$$

$I(i, j)$ is the pixel intensity at the position (i, j) in the original image. Once the integral image is computed, the sum of pixels in a rectangular area with the top-left corner at (x_1, y_1) and the bottom-right corner at (x_2, y_2) can be calculated using Equation 10.

$$\text{sum}(x_1, y_1, x_2, y_2) = I_{int}(x_2, y_2) - I_{int}(x_1 - 1, y_2) - I_{int}(x_2, y_2 - 1) + I_{int}(x_1 - 1, y_2 - 1) \quad (10)$$

This equation allows for the calculation of the sum of pixels in constant time $O(1)$, which is highly efficient compared to traditional methods. With the integral image, you can accelerate the computation process for feature recognition, such as in face detection, making the image analysis process much faster and more efficient.

Using the Haar Cascade Classifier faces in an image are identified through filtering techniques that search for specific patterns or features on the face [20]. Haar features work by comparing bright and dark rectangular areas in the image, where the pixel intensity in the brighter area is subtracted from the pixel intensity in the darker area. This feature is expressed in Equation 11.

$$F = \sum_{\text{area putih}} I(x, y) - \sum_{\text{area hitam}} I(x, y) \quad (11)$$

$I(x, y)$ is the pixel intensity at position (x, y) . To accelerate the computation of Haar features, the integral image is used, which enables efficient calculation of pixel intensity sums in rectangular areas. The method is then trained using the AdaBoost algorithm, which selects the most important Haar features from the numerous available features. Each feature is used to form a weak classifier, and the combination of these weak classifiers creates a strong classifier that is more accurate in detecting objects. The Haar Cascade also employs a cascading structure, where the image is processed through multiple stages of classifiers. If an area fails to be detected as an object at an early stage, it is immediately discarded, thus speeding up the detection process. This method is known for its efficiency in image processing because it combines Haar features with the AdaBoost algorithm and cascade

structure, making it one of the fast and accurate facial detection methods.

Once a face is detected, the system displays the face using Cascade Classification and also calculates the position of the face captured by the camera. At this stage, the system will adjust the servo motors on the robot to direct the camera, which is positioned where the robot's eyes are, to always face the detected face. The servo motors are used to control the movement along the x and y axes, ensuring that the user's face remains at the center of the robot's camera. This process is repeated to continuously monitor the face.

The Local Binary Pattern Histogram (LBPH) method is an approach used for facial recognition by utilizing local texture patterns. The process begins by converting the face image to grayscale, which simplifies the data being processed by focusing on the light intensity of each pixel [10]. Then, the Local Binary Pattern (LBP) algorithm is applied, which compares each central pixel with its surrounding neighbors in a 3x3 block as shown in Equation 12. A binary pattern is generated based on the pixel intensity comparison, where a value of 1 is assigned if the neighboring pixel is greater than or equal to the central pixel, and a value of 0 if it is smaller.

$$LBP = \sum_{i=0}^{k-1} s(N_i - P) \cdot 2^i \quad (12)$$

$s(x)$ is 1 if $x \geq 0$, and $s(x)$ is 0 if $x < 0$, N_i is the intensity of the i -th neighboring pixel, P is the intensity of the central pixel.

After the LBP pattern is computed for all pixels in the image, the face image is divided into several small blocks, and the histogram of LBP values is computed for each block. This histogram represents the frequency of occurrence of LBP patterns within that block, and the histograms from each block are then combined to form a global histogram that describes the entire face [21] as shown in Equation 13.

$$H(j) = \sum_{x,y} I(LBP(x, y) = j) \quad (13)$$

$H(j)$ is the histogram value for the LBP pattern $-j$, I is the indicator function that equals 1 if $LBP(x, y) = j$, and 0 otherwise, x and y are the pixel coordinates within the block.

After the new face histogram is calculated, the next step is to compare it with the histogram stored in the database. This comparison is often done using Euclidean distance, which measures the distance between two face histograms [22] as shown in Equation 14.

$$d(H_1, H_2) = \sqrt{\sum_{i=1}^n (H_1(i) - H_2(i))^2} \quad (14)$$

H_1 and H_2 are the histograms of two face images, n is the number of elements in the histogram.

If the Euclidean distance between the new image histogram and one of the histograms in the database is small, the new face is recognized as a registered face. This process allows for LBPH-based facial recognition

to be performed accurately and efficiently, despite its simplicity.

2.2 Robot Development

This project involves the development of an interactive robot consisting of several key components such as a Raspberry Pi 3, camera, keyboard, monitor, Arduino Uno, and servo motors (see Figure 3). The Raspberry Pi 3 acts as the processing hub, running the facial recognition program using a camera placed in the robot's eye section. This camera is used to detect and track the user's face in real-time. The keyboard functions as the user data input, while the monitor provides a visual interface that displays the results of facial recognition or other interactions.

The Arduino Uno is used as the main controller to operate two servo motors that control the X and Y axes of the robot's head. These servos enable the robot's head to follow the user's movements, ensuring that the camera remains directed at the user's face. In the robot's physical design, as shown in the diagram, the robot's head is made from 3D-printed materials with the servo system embedded inside (see Figure 4). These servos enable mechanical movement of the eyes and head to follow the user's movements, creating a more natural interaction.

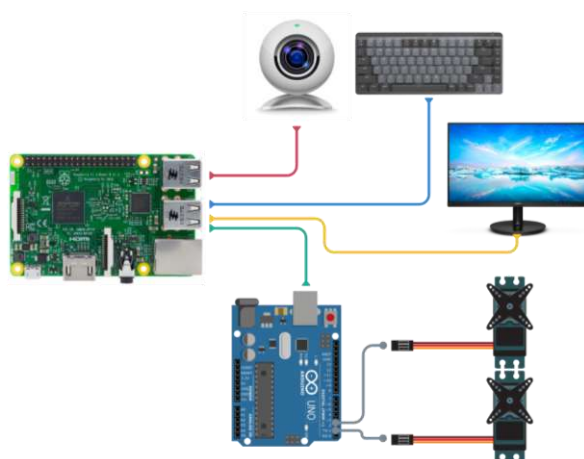


Figure 3. Schematic Robot



Figure 4. Robot Design

The camera mounted in one of the robot's eyes is connected to the facial recognition system on the

Raspberry Pi. After detecting a face, the system sends a signal to the Arduino to control the servo motors, enabling the robot's head and eyes to follow the user's movements. With this design, the robot is capable of mimicking human movements in a more interactive and responsive manner, providing a more realistic experience in facial recognition and tracking.

3. Results and Discussions

In this section, we will explore the results of various tests and analyses conducted using methods appropriate to their respective scales.

3.1 Face Recognition Result

The first test involves converting an initially colored (RGB) image into a black-and-white (grayscale) image to simplify image processing and improve computational efficiency.

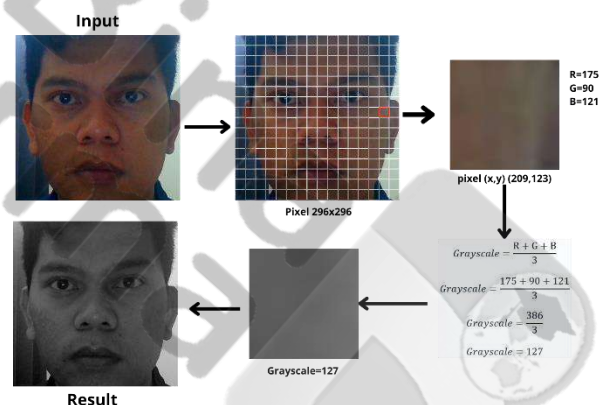


Figure 5. RGB to Grayscale

The conversion process from RGB to grayscale for a 296x296 pixel image begins by extracting the RGB values from each pixel in the image (Figure 5). For example, at a pixel with coordinates $(x, y) = (209, 123)$, the RGB values are (175, 90, 121), where 175 is the red channel value, 90 is the green channel value, and 121 is the blue channel value. The first step in the conversion is to calculate the grayscale value for this pixel using the standard conversion formula, which takes into account the relative contribution of each color channel. Equation (1) will be used for this calculation.

After the calculation, the grayscale value obtained for the pixel $(x, y) = (209, 123)$ is approximately 127. As a result, in the new grayscale image, the pixel at the same coordinates (209, 123) will have an intensity value of 127. This process is repeated for each pixel in the 255x255 image, resulting in a grayscale image of the same dimensions, where each pixel has a single grayscale intensity value instead of the three RGB color values. The process continues for each pixel until every pixel in the color image is converted to grayscale. In this way, the initially 255x255 RGB image is transformed into a grayscale image of the same dimensions, but with color information replaced by grayscale intensity values (see Figure 6).

The next step is to resize the image, where the previously grayscale-converted image of 296x296 pixels is resized to 255x255 pixels. To resize the image from 296x296 pixels to 255x255 pixels manually, we can use interpolation methods, one of which is a nearest neighbor interpolation. For each point (x, y) in the original image, we calculate the new position (x', y') using Equations 2 and 3.

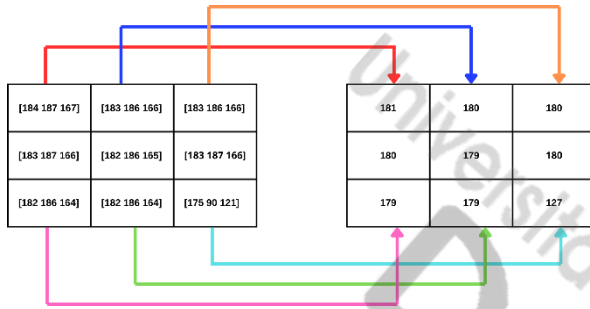


Figure 6. the RGB conversion to a 3x3 Grayscale value

After calculating x' and y' , we use bilinear interpolation to determine the pixel intensity value at (x', y') . If $(129, 129)$ is the nearest coordinate in the new pixel grid, we use the four nearest pixels from the original image to compute the intensity value at this position. Since the coordinates $(x', y') = (129.39, 129.39)$ do not fall exactly on the new pixel grid, we need to use the intensity values from the four nearest pixels in the original image: $(x_1, y_1) = (129, 129)$.

$$\begin{aligned}(x_2, y_1) &= (130, 129) \\ (x_1, y_2) &= (129, 130) \\ (x_2, y_2) &= (130, 130)\end{aligned}$$

Let the pixel intensities at each of these points be:

$$\begin{aligned}I(x_1, y_1) &= I(129, 129) \\ I(x_2, y_1) &= I(130, 129) \\ I(x_1, y_2) &= I(129, 130) \\ I(x_2, y_2) &= I(130, 130)\end{aligned}$$

We start with horizontal interpolation to obtain the intensity value between (x_1, y_1) and (x_2, y_1) at the coordinate (x', y_1) , and between (x_1, y_2) and (x_2, y_2) at the coordinate (x', y_2)

$$\begin{aligned}I(x', y_1) &= I(x_1, y_1) \times (1 - \alpha) + I(x_2, y_1) \times \alpha \\ I(x', y_2) &= I(x_1, y_2) \times (1 - \alpha) + I(x_2, y_2) \times \alpha\end{aligned}$$

Where $\alpha = x' - x_1 = 129.39 - 129 = 0.39$

Thus, we can calculate:

$$\begin{aligned}I(x', y_1) &= I(129, 129) \times (1 - 0.39) + I(130, 129) \times 0.39 \\ I(x', y_2) &= I(129, 130) \times (1 - 0.39) + I(130, 130) \times 0.39\end{aligned}$$

After obtaining $I(x', y_1)$ and $I(x', y_2)$, we perform vertical interpolation to calculate the final intensity value at the coordinate (x', y') :

$$I(x', y') = I(x', y_1) \times (1 - \beta) + I(x', y_2) \times \beta$$

Where:

$$\begin{aligned}\beta &= y' - y_1 = 129.39 - 129 = 0.39 \\ I(x', y') &= I(x', y_1) \times (1 - 0.39) + I(x', y_2) \times 0.39\end{aligned}$$

$$\text{If } I(x', y_1) = a; I(x', y_2) = b$$

Then:

$$I(x', y') = a \times (1 - 0.39) + b \times 0.39$$

$$I(x', y') = a \times 0.61 + b \times 0.39$$

Thus, we can determine the pixel intensity value at (x', y') , which is the result of interpolation from the four nearest pixels in the original image. This process is applied to every pixel in the new image, resulting in a resized image from 296x296 to 255x255 pixels with smoother transitions.



Figure 7. Filter Gaussian

From the results of the image that has been converted to grayscale and resized to 255x255 pixels, the next step is noise removal from the image using the Gaussian filter equation shown in Figure 7. Here, we demonstrate the calculation using a 3x3 pixel example we previously illustrated,

$$\begin{aligned}\text{Image} &= \begin{bmatrix} 181 & 180 & 180 \\ 180 & 179 & 180 \\ 179 & 179 & 127 \end{bmatrix} \\ \text{Kernel} &= \begin{bmatrix} 0.0613 & 0.1221 & 0.0613 \\ 0.1221 & 0.2442 & 0.1221 \\ 0.0613 & 0.1221 & 0.0613 \end{bmatrix}\end{aligned}$$

To calculate the pixel value resulting from the Gaussian filter at the center position $(1, 1)$ of the image, we perform a convolution with the Gaussian kernel:

$$\text{Result} = \sum_{i=-1}^1 \sum_{j=-1}^1 (\text{Image}[i+1, j+1] \times \text{Kernel}[i+1, j+1])$$

With the updated pixel values, the pixel value resulting from the Gaussian filter for the center position $(1, 1)$ on the given 3x3 grayscale image is approximately 171.60. In this way, you can calculate the resulting pixel values for the entire image if the image is larger, by sliding the Gaussian kernel to each pixel position and performing the convolution calculation.

Next is contrast adjustment, which aims to enhance the difference between light and dark areas in the image to make the details clearer.



Figure 8. contrast adjustment

To achieve the results as shown in Figure 8, the author applied contrast adjustment using equation (8), where we want to set the contrast with $\alpha=1.5$ (contrast factor) and $\beta=0$ (brightness offset). This results in updated values for each pixel.

After the contrast adjustment with $\alpha=1.5$ and $\beta=0$, the resulting image is:

$$Image = \begin{bmatrix} 181 & 180 & 180 \\ 180 & 179 & 180 \\ 179 & 179 & 127 \end{bmatrix}$$

At this step, each pixel value has been adjusted according to the given contrast factor and clipped to the 0-255 range. This is the result of the contrast adjustment, which enhances the difference between light and dark areas in the image shown in Table 1.

Table 1. The result of the contrast adjustment on 3x3 pixel image

Pixel (x, y)	Original Intensity	$\alpha \times I(x, y)$	Offset β	Result $I'(x, y)$	Clamping Result
(0, 0)	181	271.5	0	271.5	255
(0, 1)	180	270	0	270	255
(0, 2)	180	270	0	270	255
(1, 0)	180	270	0	270	255
(1, 1)	179	268.5	0	268.5	255
(1, 2)	180	270	0	270	255
(2, 0)	179	268.5	0	268.5	255
(2, 1)	179	268.5	0	268.5	255
(2, 2)	127	190.5	0	190.5	190

In the integral image process, the result of contrast adjustment will be calculated using Equation 9 which produces Figure 9 as shown in Table 2.



Figure 9. Integral Image

$$Integral\ Image = \begin{bmatrix} 255 & 255 & 255 \\ 255 & 255 & 255 \\ 255 & 255 & 190 \end{bmatrix}$$

Table 2. Integral image calculation results on 3x3 pixels

Pixel (x,y)	Intensity $I'(x,y)$	Integral (x,y)	Calculation
(0, 0)	255	255	255
(0, 1)	255	510	255+255
(0, 2)	255	765	510+255
(1, 0)	255	510	255+255
(1, 1)	255	1020	510+510
(1, 2)	255	1275	765+255
(2, 0)	255	765	510+255
(2, 1)	255	1275	765+510
(2, 2)	190	1720	1275+190

Integral Image is a cumulative table that stores the sum of pixels from the top-left corner to a specific point in

the image, allowing for more efficient calculation of Haar-Like features. Haar-Like features are used to detect certain patterns in an image, such as edges, corners, or lines, by calculating the difference between dark and light areas in rectangular shapes. By using the Integral Image, Haar-Like features are calculated quickly without requiring access to individual pixels. Once the Haar-Like features are computed, a Haar Cascade Classifier is applied to detect faces by testing specific patterns that have been learned. If the recognized pattern matches a face pattern, a bounding box is drawn around the face in the image (Figure 10). This process enables accurate and effective face detection in grayscale images by utilizing specific features and efficient data structures.

Extracted features for face at (x=14, y=14):

[1.359e+03 0.000e+00 2.000e+00 0.000e+00 0.000e+00 2.000e+00 1.000e+00 4.000e+00 0.000e+00 4.000e+00 4.000e+00 2.000e+00 3.000e+00 7.000e+00 1.400e+01 1.100e+01 1.100e+01 2.100e+01 2.000e+01 2.900e+01 3.800e+01 4.700e+01 6.000e+01 5.500e+01 6.800e+01 8.500e+01 9.200e+01 1.140e+02...]



Figure 10. result of the haar cascade classifier

Once the face has been detected in the previous process, the next step is to divide the face into small blocks, such as blocks sized 3x3 pixels. Each of these small blocks will be processed to calculate what we call the Local Binary Pattern (LBP), as shown in Figure 11. The program works by taking one pixel in the center of the block as the central pixel, and then comparing the intensity value of this central pixel with its neighbors around the 3x3 block.

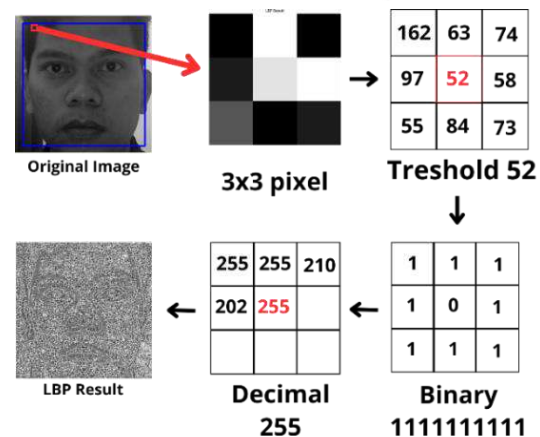


Figure 11. LBP operation

If the neighbor's value is greater than or equal to the central pixel, we assign a value of 1. Conversely, if the neighbor is less than the central pixel, we assign a value of 0. These values then form a binary pattern. For example, if we have a central pixel with an intensity of 52, and its neighbors have higher values like 162, 170, 180, and so on, we will get a binary pattern of all 1s.

This binary pattern is then converted into a decimal value. In this case, the binary 11111111 will be converted to 255 in decimal.

After the decimal values are obtained from each central pixel throughout the face, these values are organized into a histogram representing the overall facial features. This histogram contains the frequency of binary patterns found in the face blocks. This histogram becomes the feature vector that can be used to match someone's face with stored facial data.

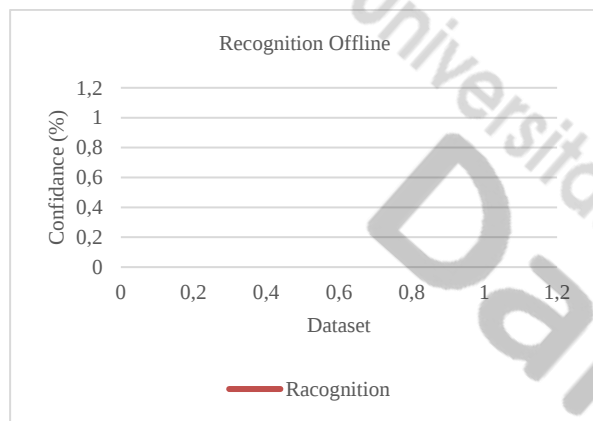


Figure 12. The test results for face recognition offline.

The graph in Figure 12 shows the results of face recognition testing using the LBPH (Local Binary Patterns Histograms) algorithm in two different scenarios: offline and real-time recognition. In the offline testing, one face data is selected from a dataset containing 50 faces and then matched one by one with all the faces in the dataset shown in Table 3. The graph shows that the confidence level in offline face recognition varies between 80.5% and 85.5% with an average of 83.6%. The fluctuations observed reflect variations in the quality or similarity of faces in the dataset, with some significant drops indicating that certain faces may have weaker or different features compared to others in the dataset.

Table 3. The test results for face recognition offline.

No	Dataset Image	Confidence (%)	Time Taken (S)
1	face_0.jpg	83,43	0,014
2	face_1.jpg	84,33	0,0152
3	face_2.jpg	84,13	0,014
4	face_3.jpg	84,41	0,015
5	face_4.jpg	84,06	0,015
6	face_5.jpg	84,87	0,0141
7	face_6.jpg	84,21	0,0163
8	face_7.jpg	83,3	0,0171
9	face_8.jpg	84,23	0,0153
10	face_9.jpg	83,8	0,0161
11	face_10.jpg	85,08	0,0183
12	face_11.jpg	84,59	0,0162
13	face_12.jpg	84,59	0,0161
14	face_13.jpg	85,09	0,0152
15	face_14.jpg	84,88	0,0151
16	face_15.jpg	84,58	0,0161

No	Dataset Image	Confidence (%)	Time Taken (S)
17	face_16.jpg	83,96	0,014
18	face_17.jpg	84,95	0,0132
19	face_18.jpg	85,27	0,0171
20	face_19.jpg	84,71	0,0141
21	face_20.jpg	84,47	0,0142
22	face_21.jpg	84,52	0,0141
23	face_22.jpg	84,38	0,0141
24	face_23.jpg	84,11	0,0141
25	face_24.jpg	81,55	0,0132
26	face_25.jpg	81,11	0,0142
27	face_26.jpg	82,34	0,0151
28	face_27.jpg	83,42	0,0142
29	face_28.jpg	82,91	0,0143
30	face_29.jpg	83,34	0,0134
31	face_30.jpg	82,37	0,0141
32	face_31.jpg	82,31	0,0143
33	face_32.jpg	83,5	0,0141
34	face_33.jpg	82,36	0,0151
35	face_34.jpg	83,41	0,014
36	face_35.jpg	83,17	0,0142
37	face_36.jpg	82,64	0,0152
38	face_37.jpg	82,37	0,0133
39	face_38.jpg	82,62	0,0142
40	face_39.jpg	82,66	0,0141
41	face_40.jpg	82,59	0,0142
42	face_41.jpg	83,33	0,0131
43	face_42.jpg	82,04	0,0141
44	face_43.jpg	83,44	0,0142
45	face_44.jpg	83,92	0,0142
46	face_45.jpg	83,86	0,0152
47	face_46.jpg	83,1	0,017
48	face_47.jpg	83,55	0,0141
49	face_48.jpg	83	0,0143
50	face_49.jpg	83,28	0,0142
Average		83,6028	0,014712

Meanwhile, in real-time testing, the face captured every second is directly matched with the 50 face data in the same dataset shown in Table 4. The graph in Figure 13 on the right shows confidence levels ranging from 72% to 81% with an average of 77.67%. The greater fluctuations in this graph indicate that changes in conditions during image capture, such as lighting, position, or facial expression, greatly affect the accuracy of recognition. Overall, offline recognition appears more stable compared to real-time, although both suggest that the quality of facial images and environmental conditions are crucial to achieving more accurate and consistent recognition results.

Table 4. The test results for face recognition in real-time.

No	Dataset Image	Confidence (%)	Time Taken(s)
1	face_0.jpg	75,51	0,0205
2	face_1.jpg	78,75	0,02
3	face_2.jpg	78,26	0,019
4	face_3.jpg	79,62	0,018
5	face_4.jpg	77,96	0,018
6	face_5.jpg	78,87	0,0188
7	face_6.jpg	78,69	0,021

No	Dataset Image	Confidence (%)	Time Taken(s)
8	face_7.jpg	77,81	0,0205
9	face_8.jpg	77,97	0,022
10	face_9.jpg	78,87	0,02
11	face_10.jpg	77,23	0,021
12	face_11.jpg	77,64	0,019
13	face_12.jpg	78,02	0,019
14	face_13.jpg	76,56	0,019
15	face_14.jpg	75,9	0,0195
16	face_15.jpg	77,07	0,021
17	face_16.jpg	77,78	0,0165
18	face_17.jpg	77,06	0,02
19	face_18.jpg	78,06	0,019
20	face_19.jpg	77,4	0,0195
21	face_20.jpg	76,5	0,022
22	face_21.jpg	76,42	0,021
23	face_22.jpg	76,61	0,022
24	face_23.jpg	76,52	0,0209
25	face_24.jpg	73,26	0,0207
26	face_25.jpg	73,54	0,0195
27	face_26.jpg	75,12	0,019
28	face_27.jpg	77,15	0,017
29	face_28.jpg	76,61	0,02
30	face_29.jpg	78,38	0,029
31	face_30.jpg	77,54	0,045
32	face_31.jpg	78,89	0,0185
33	face_32.jpg	78,58	0,019
34	face_33.jpg	77,49	0,0204
35	face_34.jpg	78,48	0,019
36	face_35.jpg	77,03	0,019
37	face_36.jpg	79,27	0,021
38	face_37.jpg	77,55	0,019
39	face_38.jpg	78,75	0,021
40	face_39.jpg	78,97	0,02
41	face_40.jpg	78,88	0,0175
42	face_41.jpg	78,12	0,0815
43	face_42.jpg	78,76	0,019
44	face_43.jpg	79,22	0,052
45	face_44.jpg	77,91	0,018
46	face_45.jpg	78,45	0,019
47	face_46.jpg	77,2	0,02
48	face_47.jpg	78,36	0,023
49	face_48.jpg	79,17	0,019
50	face_49.jpg	79,91	0,019
Average		77,6734	0,022246

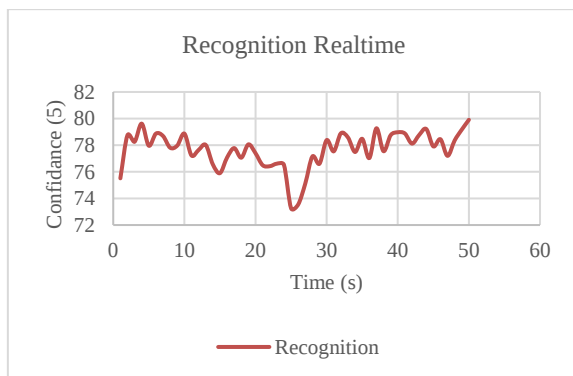


Figure 13. The test results for face recognition realtime.

3.2 Face Tracker Evaluation

Table 5 shows the results of an experiment with a robot designed to follow a face. The face position is displayed in the form of pixel coordinates X and Y taken from the camera. For example, in the first row, the face position is detected at X = 302 and Y = 212. The camera captures the face at that point. The robot's servo then adjusts its position based on the face movement. In the first example, the servo does not need to move because the face is already centered. However, in the second row, the face shifts to X = 459 and Y = 204, so the servo changes its position to X = 137° and Y = 79° to follow the movement to the upper right.

Table 5. The test results for face tracker.

Position	Face Position (pixel)		Servo Position (°)		Move ment		Time (s)
	x	y	x	y	x	y	
	302	212	110	90	C	C	0
	459	204	137	79	R	U	2.314
	399	278	132	96	L	D	1.945
	518	342	139	105	R	D	2.952
	219	302	91	94	L	U	3.561
	391	159	119	82	R	U	3.013

The robot also moves to align with the face's direction. In the second example, the face is detected on the right and upward, so the robot moves to the right and upward to adjust its position. The time taken to adjust the face back to the center position is also recorded. In the second row, the robot takes 2.314 seconds to make this adjustment. Additionally, images captured by the camera are included for each position change, providing visual confirmation that the robot is accurately following the face movement. This table illustrates how the robot reacts to face movements, adjusts its servo positions, and records the time needed to center the face back on the screen.

3.3 Discussions

The results of testing the face recognition system using the Local Binary Patterns Histogram (LBPH) algorithm indicate that this method can provide an adequate level of accuracy in both offline and real-time modes,

although several factors influence system performance. In offline testing, an average accuracy of 83.6% shows that LBPH can recognize faces well when the database facial data has high quality and uniformity. Accuracy fluctuations ranging from 80.5% to 85.5% suggest variability in face recognition, which could be caused by differences in perspective or lighting in the dataset. Real-time testing, with an average accuracy of 77.67%, reveals that environmental dynamics such as changes in lighting and facial expressions directly affect the results of face recognition shown in Table 6.

Table 6. Results of the comparison between offline and real-time face recognition.

Mode	Minimum Accuracy (%)	Maximum Accuracy (%)	Average Accuracy (%)
Offline	80.5	85.5	83.6
Real-time	72.0	81.0	77.67

This indicates that the system is more sensitive to changing real-time conditions, which is a common challenge in direct face recognition applications. The difference in processing times between offline and online modes is primarily due to the computational load involved in real-time image acquisition and processing. In offline mode, pre-captured images are already stored in memory, allowing the system to focus solely on recognition tasks, leading to more consistent processing times. However, in online mode, the system must simultaneously handle image acquisition, pre-processing (grayscale conversion, noise reduction, and contrast adjustment), and recognition. Factors such as varying lighting conditions and rapid facial movements can add additional processing time, resulting in increased variability in response times.

On the other hand, the servo movements in the robot designed to follow the user's face showed positive results, with response times ranging from 1.945 seconds to 3.561 seconds. This demonstrates that the robot can adjust its position quickly and accurately, enabling more natural interaction between the robot and the user. Although these results are satisfactory, there is room for further improvements, particularly in optimizing the system to handle more challenging environmental conditions and to speed up the robot's response time. Overall, this study successfully demonstrates the great potential of implementing face recognition systems in receptionist robots, though there is still room for refinement to address various practical challenges in the field.

4. Conclusions

This research successfully developed a memory-efficient face recognition system using the Local Binary Patterns Histogram (LBPH) algorithm implemented on a Raspberry Pi-based receptionist robot. The test results show that the LBPH algorithm is capable of providing good face recognition accuracy, with an average accuracy of 83.6% in offline mode and 77.67% in real-time mode. Nonetheless, the system is still affected by

environmental changes such as lighting and facial expressions, which affect recognition accuracy. Additionally, the designed robot is capable of accurately following the user's facial movements, with a fairly fast servo response time of between 1.945 seconds and 3.561 seconds, supporting responsive and natural interaction with the user. Although the results are satisfactory, this research identifies several areas for improvement, such as further optimization for handling dynamic environmental conditions and improving the robot's response speed. Overall, the developed system shows great potential for implementation in interactive and efficient receptionist robot applications.

Acknowledgements

This research has received funding from the Directorate of Research, Technology and Community Service, Directorate General of Higher Education, Research and Technology, Ministry of Education, Culture, Research and Technology (grant agreement 104/E5/PG.02.00.PL/2024, 1100/LL2/KP/PL/2024). The authors thank Universitas Bina Darma for its support and facilities.

References

- [1] N. Li *et al.*, "Chinese Face Dataset for Face Recognition in an Uncontrolled Classroom ENVIRONMENT," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3302919.
- [2] R. He, J. Cao, L. Song, Z. Sun, and T. Tan, "Adversarial Cross-Spectral Face Completion for NIR-VIS Face Recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 5, 2020, doi: 10.1109/TPAMI.2019.2961900.
- [3] J. Tomášik *et al.*, "The Potential of AI-Powered Face Enhancement Technologies in Face-Driven Orthodontic Treatment Planning," *Applied Sciences* 2024, Vol. 14, Page 7837, vol. 14, no. 17, p. 7837, Sep. 2024, doi: 10.3390/AP14177837.
- [4] P. C. P. Neto, J. R. Pinto, F. Boutros, N. Damer, A. F. Sequeira, and J. S. Cardoso, "Beyond Masks: On the Generalization of Masked Face Recognition Models to Occluded Face Recognition," *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3199014.
- [5] H. Yu, Y. Wang, Y. Tian, H. Zhang, W. Zheng, and F. Y. Wang, "Social Vision for Intelligent Vehicles: From Computer Vision to Foundation Vision," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 11, pp. 4474–4476, Nov. 2023, doi: 10.1109/TIV.2023.3330870.
- [6] W. Sun, X. Min, D. Tu, S. Ma, and G. Zhai, "Blind Quality Assessment for in-the-Wild Images via Hierarchical Feature Fusion and Iterative Mixed Database Training," *IEEE Journal on Selected Topics in Signal Processing*, vol. 17, no. 6, pp. 1178–1192, Nov. 2023, doi: 10.1109/JSTSP.2023.3270621.
- [7] K. Panetta *et al.*, "A Comprehensive Database for Benchmarking Imaging Systems," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 3, pp. 509–520, Mar. 2020, doi: 10.1109/TPAMI.2018.2884458.
- [8] Y. Zhong *et al.*, "Dynamic Training Data Dropout for Robust Deep Face Recognition," *IEEE Trans Multimedia*, vol. 24, pp. 1186–1197, 2022, doi: 10.1109/TMM.2021.3123478.
- [9] C. WANG, "A Comparative Statement for the Facial Recognition Algorithm Via Eigenfaces and the Local Binary Patterns Histograms Algorithm," *i-manager's Journal on Software Engineering*, vol. 15, no. 4, p. 1, Apr. 2021, doi: 10.26634/JSE.15.4.18429.
- [10] I. Al Saidi, M. Rziza, and J. Debayle, "A novel texture descriptor: Homogeneous Rotated Local Binary Pattern (HRLBP)," *2020 10th International Symposium on Signal*,

- Image, Video and Communications, ISIVC 2020*, Apr. 2021, doi: 10.1109/ISIVC49222.2021.9487538.
- [11] S. Chakraborty, S. K. Singh, and K. Kumar, "Facial Biometric System for Recognition Using Extended LGHP Algorithm on Raspberry Pi," *IEEE Sens J*, vol. 20, no. 14, pp. 8117–8127, Jul. 2020, doi: 10.1109/JSEN.2020.2979907.
- [12] Z. Al-Ameen, H. N. Saeed, and D. K. Saeed, "Fast and Efficient Algorithm for Contrast Enhancement of Color Images," *Review of Computer Engineering Studies*, vol. 7, no. 3, p. 60, Sep. 2020, doi: 10.18280/RCES.070303.
- [13] D. Shehada, A. Turkey, W. Khan, B. Khan, and A. Hussain, "A Lightweight Facial Emotion Recognition System Using Partial Transfer Learning for Visually Impaired People," *IEEE Access*, vol. 11, pp. 36961–36969, 2023, doi: 10.1109/ACCESS.2023.3264268.
- [14] H. Gong, L. Chen, C. Li, J. Zeng, X. Tao, and Y. Wang, "Online Tracking and Relocation Based on a New Rotation-Invariant Haar-Like Statistical Descriptor in Endoscopic Examination," *IEEE Access*, vol. 8, pp. 101867–101883, 2020, doi: 10.1109/ACCESS.2020.2994440.
- [15] Y. Guo, Q. Xu, Y. Su, S. J.-I. Access, and undefined 2020, "Visibility detection based on the recognition of the preceding vehicle's taillight signals," *ieeexplore.ieee.org* Y Guo, Q Xu, Y Su, S Jiang *IEEE Access*, 2020•*ieeexplore.ieee.org*, Accessed: Sep. 14, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9257398/>
- [16] S. Liu, J. Chen, ... S. R. C. and S. for V., and undefined 2019, "A new multi-focus image fusion algorithm and its efficient implementation," *ieeexplore.ieee.org* S Liu, J Chen, S Rahardja *IEEE Transactions on Circuits and Systems for Video Technology*, 2019•*ieeexplore.ieee.org*, Accessed: Sep. 14, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8653405/>
- [17] K. Chen, T. Yi, Q. L.-I. S. P. Letters, and undefined 2021, "Lightqnet: Lightweight deep face quality assessment for risk-controlled face recognition," *ieeexplore.ieee.org* K Chen, T Yi, Q Lv *IEEE Signal Processing Letters*, 2021•*ieeexplore.ieee.org*, Accessed: Sep. 14, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9528058/>
- [18] D. Kim, J. Hyun, and B. Moon, "Memory-efficient architecture for contrast enhancement and integral image computation," *2020 International Conference on Electronics, Information, and Communication, ICEIC 2020*, Jan. 2020, doi: 10.1109/ICEIC49074.2020.9051296.
- [19] A. H. Diab, I. T. Ahmed, and W. M. Jasim, "Analyze the Existing Contrast Enhancement Algorithms and Image Quality Assessment: Survey," *AIP Conf Proc*, vol. 3009, no. 1, Feb. 2024, doi: 10.1063/5.0193832/3265139.
- [20] G. Ghosh and K. S. Swarnalatha, "A Detail Analysis and Implementation of Haar Cascade Classifier," pp. 341–359, 2022, doi: 10.1007/978-981-16-3342-3_28.
- [21] W. El-Tarhouni, A. Abdo, and A. Elmegreisi, "Feature fusion using the Local Binary Pattern Histogram Fourier and the Pyramid Histogram of Feature fusion using the Local Binary Pattern Oriented Gradient in iris recognition," in *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, MI-STA 2021 - Proceedings*, 2021. doi: 10.1109/MI-STA52233.2021.9464473.
- [22] D. Loktev, A. Loktev, R. Stepanov, M. Faisal, and E. M. Zamzami, "Comparative Analysis of Inter-Centroid K-Means Performance using Euclidean Distance, Canberra Distance and Manhattan Distance," *J Phys Conf Ser*, vol. 1566, no. 1, p. 012112, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012112.