

## SISTEM PENGENALAN WAJAH CERDAS BERBASIS ESP32-CAM DAN COMPUTER VISION UNTUK OBJEK REAL-TIME

M. Haeqal Salehudin, Aulia Umami, Muhammad Ghuzammir V. M.\*, Muhammad Sultan I., Melia Sari, Iwan Pahendra A.S., Abdul Haris Dalimunthe, Puspa Kurniasari, Desi Windisari, Robi Prasetyo

Teknik Elektro, Universitas Sriwijaya, Palembang

\*Corresponding author e-mail: [muhammadghuzammir27@gmail.com](mailto:muhammadghuzammir27@gmail.com)

**ABSTRAK:** Seiring meningkatnya kebutuhan akan sistem keamanan yang cerdas dan efisien, teknologi pengenalan wajah menjadi solusi yang semakin relevan. Penelitian ini mengusulkan sebuah sistem pengenalan wajah cerdas yang mengintegrasikan mikrokontroler hemat daya ESP32-CAM dengan algoritma *deep learning You Only Look Once* versi 8 (YOLOv8). Sistem ini dirancang untuk mengenali wajah secara *real-time* langsung dari perangkat, serta menampilkan hasilnya pada *dashboard* web interaktif yang mencatat setiap peristiwa pengenalan. Berbeda dengan penelitian sebelumnya yang menggunakan metode klasik seperti *Haar Cascade* atau *Multi-Task Cascaded Convolutional Networks* (MTCNN), pendekatan ini memanfaatkan arsitektur *deep learning* YOLOv8 (versi Nano) untuk deteksi dan pustaka *face\_recognition* berbasis *dlib* untuk identifikasi. Hasil pengujian menunjukkan sistem mampu mencapai akurasi deteksi 99% dan akurasi pengenalan 98% pada kondisi pencahayaan baik dengan kecepatan rata-rata 8 FPS. Namun, kinerja menurun pada kondisi minim cahaya dengan kenaikan *False Rejection Rate* (FRR) hingga 27%. Hasil penelitian ini berupa prototipe sistem keamanan *end-to-end* yang terjangkau dan efisien untuk pemantauan *real-time*.

Kata Kunci: Pengenalan Wajah, YOLOv8, ESP32-CAM, IoT.

**ABSTRACT:** With the growing demand for intelligent and efficient security systems, face detection technology has emerged as an increasingly relevant solution. This study proposes a smart face detection system integrating the low-power ESP32-CAM microcontroller with the You Only Look Once version 8 (YOLOv8) deep learning algorithm. The system is designed to perform real-time, on-device face detection and display the results on an interactive web dashboard that logs every detection event. Unlike previous studies employing classical methods such as Haar Cascade or Multi-Task Cascaded Convolutional Networks (MTCNN), this approach utilizes the YOLOv8 (Nano version) deep learning architecture for detection and the *dlib*-based *face\_recognition* library for identification. Experimental results demonstrate that the system achieves 99% detection accuracy and 98% recognition accuracy under optimal lighting conditions, with an average speed of 8 FPS. However, performance degrades in low-light environments, characterized by an increase in the False Rejection Rate (FRR) to 27%. The outcome of this research is an affordable and efficient end-to-end security system prototype suitable for real-time monitoring.

Keywords: Face Recognition, YOLOv8, ESP32-CAM, IoT

## 1. Pendahuluan

Di era digital yang semakin berkembang, kebutuhan akan sistem keamanan dan pengawasan berbasis teknologi semakin meningkat. Salah satu solusi yang banyak dikembangkan adalah sistem deteksi wajah, yang mampu mengenali identitas seseorang secara otomatis melalui citra visual. Teknologi ini telah digunakan dalam berbagai bidang seperti sistem absensi, kontrol akses, dan pengawasan publik. Dengan kemajuan perangkat keras dan algoritma kecerdasan buatan, sistem deteksi wajah kini dapat diimplementasikan secara *real-time* dan efisien [1].

Penelitian ini mengusulkan sistem deteksi wajah cerdas berbasis ESP32-CAM yang mengintegrasikan teknologi *computer vision* dan algoritma deteksi objek berbasis *deep learning*, khususnya *You Only Look Once* (YOLO). YOLO adalah keluarga arsitektur deteksi tahap tunggal yang berbasis di *Darknet*, kerangka kerja *Convolutional neural networks* (CNN) sumber terbuka. Pertama kali dikembangkan pada tahun 2016, arsitektur YOLO lebih mengutamakan kecepatan. Tujuan dari penelitian ini adalah merancang dan juga mengimplementasikan sistem yang mampu mendeteksi wajah secara akurat dan cepat dalam berbagai kondisi pencahayaan dan sudut pandang, serta menguji kinerja sistem dari segi kecepatan, akurasi, dan efisiensi komputasi.

Terdapat banyak penelitian sebelumnya yang telah menggunakan algoritma deteksi objek untuk sistem pengenalan wajah, salah satunya adalah studi oleh Agachi et al. (2025) yang mengimplementasikan metode *Haar Cascade* untuk mendeteksi wajah pada sistem pengamanan kotak amal berbasis ESP32-CAM [2]. Ada juga penelitian oleh Awangga et al. (2024) yang menggunakan algoritma *Multi-task Cascaded Convolutional Networks* dan *Convolutional neural networks* untuk mendeteksi wajah

dalam sistem keamanan rumah berbasis ESP32-CAM [3]. Penelitian lain oleh Pratama dan Ulum (2023) merancang sistem keamanan rumah menggunakan ESP32-CAM dengan metode *face recognition* [4]. Ketiga studi tersebut menunjukkan bahwa ESP32-CAM dapat digunakan untuk deteksi wajah, namun masih memiliki keterbatasan dalam akurasi dan fleksibilitas algoritma.

Berdasarkan studi-studi tersebut, hasil penelitian ini diharapkan dapat memberikan solusi alternatif sistem keamanan cerdas, murah, dan efisien, dan menjadi referensi untuk pengembangan proyek IoT dan *embedded machine learning* (*TinyML*). Berbeda dengan penelitian sebelumnya yang hanya berfokus pada deteksi wajah menggunakan *Haar Cascade* atau MTCNN, penelitian ini mengimplementasikan sistem *end-to-end* yang tidak hanya mendeteksi tetapi juga mengenali identitas wajah. Keunggulan sistem ini terletak pada integrasi YOLOv8 untuk deteksi cepat dengan sistem *database* interaktif yang dapat memungkinkan pendaftaran wajah baru secara *real-time* melalui *dashboard* web. Dengan cara menggabungkan ESP32-CAM dan YOLO, sistem yang dikembangkan diharapkan mampu bekerja secara optimal dalam kondisi nyata, serta memberikan kontribusi terhadap pengembangan sistem pengenalan wajah yang efisien dan terjangkau.

## 2. Tinjauan Pustaka

### 2.1 ESP32-CAM

ESP32-CAM adalah modul mikrokontroler berbasis ESP32 yang dilengkapi dengan kamera OV2640, konektivitas Wi-Fi dan *bluetooth*, serta *slot microSD*. Modul ini sangat populer dalam proyek IoT, terutama untuk aplikasi seperti deteksi wajah dan objek, *streaming video*, sistem absensi atau kontrol akses, dan pemantauan keamanan berbasis kamera. Keunggulan ESP32-CAM terletak pada ukurannya yang kecil, harganya yang

terjangkau, dan kemampuan menjalankan algoritma visi komputer ringan secara lokal atau terhubung ke *cloud* [5].

## 2.2 Algoritma Deteksi Objek

Deteksi objek adalah teknologi fundamental dalam *Computer Vision* yang bertujuan untuk mengklasifikasikan dan melokalisasi objek dalam citra digital. Pendekatan awal dalam deteksi wajah didominasi oleh metode statistik seperti yang diperkenalkan oleh Viola dan Jones pada tahun 2001. Algoritma Viola-Jones menggunakan fitur *Haar-like* dan *Cascade Classifier* untuk mendeteksi wajah dengan cepat, namun metode ini memiliki keterbatasan signifikan dalam akurasi, terutama pada kondisi pencahayaan yang tidak ideal atau posisi wajah yang miring [6]. Seiring dengan perkembangan *Deep Learning*, pendekatan berbasis *Convolutional Neural Networks* (CNN) mulai menggantikan metode klasik karena kemampuannya yang superior dalam mengekstraksi fitur kompleks secara otomatis, sebagaimana dibuktikan oleh He et al. melalui arsitektur *Deep Residual Learning* (*ResNet*) yang menjadi landasan banyak sistem visi modern [7].

## 2.3 Algoritma YOLO (*You Only Look Once*)

Pergeseran paradigma dalam deteksi objek *real-time* terjadi dengan diperkenalkannya arsitektur YOLO oleh Redmon et al. pada tahun 2016. Berbeda dengan metode dua tahap (seperti R-CNN) yang lambat, YOLO memproses seluruh gambar dalam satu kali evaluasi (*one-stage detector*) untuk memprediksi *bounding box* dan probabilitas kelas secara simultan, menjadikannya sangat efisien untuk aplikasi video [8].

Penelitian ini secara spesifik memanfaatkan YOLOv8, iterasi terbaru yang dikembangkan oleh Jocher et al. (*Ultralytics*) pada tahun 2023. YOLOv8 menawarkan arsitektur *anchor-free* yang lebih canggih, fungsi *loss* yang ditingkatkan, dan kinerja *mean Average Precision* (mAP) yang lebih tinggi dibandingkan pendahulunya,

menjadikannya solusi ideal untuk perangkat dengan sumber daya terbatas [9].

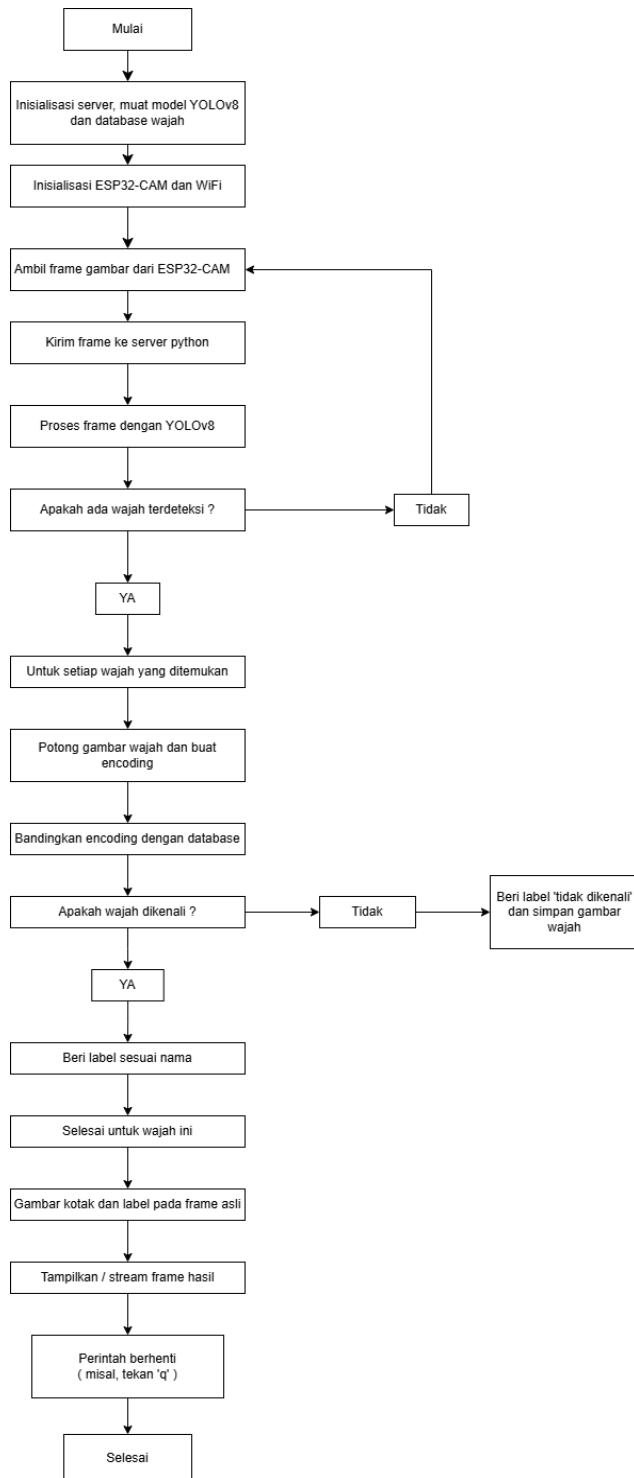
## 2.4 Pengenalan Wajah dan *Face Embeddings*

Setelah wajah dideteksi, proses pengenalan identitas dilakukan menggunakan teknik *face embedding*. Konsep ini dipopulerkan oleh Schroff et al. melalui sistem *FaceNet*, yang memetakan gambar wajah ke dalam ruang *Euclidean* (vektor 128-dimensi). Dalam ruang ini, jarak geometris antar vektor merepresentasikan kemiripan wajah; wajah dari orang yang sama memiliki jarak yang dekat, sedangkan wajah berbeda memiliki jarak yang jauh [10]. Metode ini terbukti lebih robust terhadap perubahan *pose* dan pencahayaan dibandingkan metode pencocokan template tradisional.

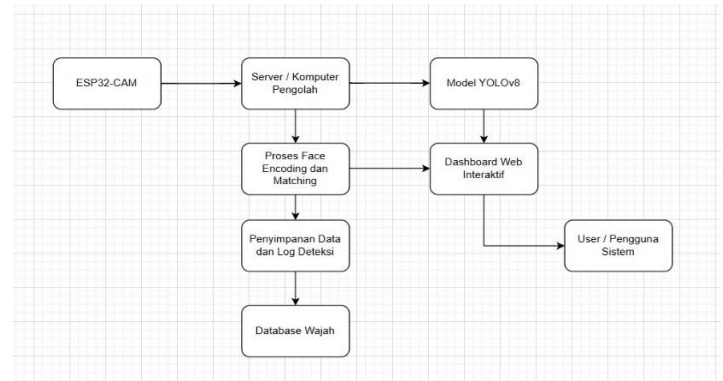
## 2.5 Komputasi Edge pada ESP32-CAM

Implementasi algoritma cerdas pada perangkat edge (IoT) memerlukan efisiensi tinggi. Howard et al. memperkenalkan konsep *MobileNets*, sebuah kelas CNN efisien yang dirancang untuk aplikasi mobile dan *embedded vision*. Prinsip efisiensi ini diterapkan dalam penelitian ini melalui penggunaan ESP32-CAM, sebuah modul mikrokontroler hemat daya berbasis ESP32 yang dilengkapi kamera OV2640. Meskipun memiliki keterbatasan komputasi dibandingkan GPU *desktop*, ESP32-CAM memungkinkan akuisisi citra *nirkabel* yang fleksibel untuk diproses lebih lanjut oleh server [11].

### 3. Metodologi Penelitian



Gambar 3.1 Flowchart sistem



Gambar 3.2 Diagram blok

Penelitian ini bersifat eksperimental dan rekayasa sistem. Pendekatan yang dilakukan meliputi: (1) Perancangan perangkat keras dan perangkat lunak ESP-32 CAM; (2) pemilihan dan pelatihan model *computer vision* ringan untuk deteksi wajah; (3) integrasi *streaming* gambar dan inferensi *real-time*; (4) pengujian kinerja pada kondisi nyata. Kamera ESP-32 CAM yang terhubung ke server pada komputer dan laptop dijalankan dengan menggunakan model YOLOv8 untuk deteksi objek-objek *multi* deteksi secara *real-time*.

Terdapat perangkat keras dan perangkat lunak yang digunakan terhubung antar perangkat untuk menjalankan penelitian ini. Pada perangkat keras ESP-32 CAM dengan modul kamera OV2640 resolusi 2MP, komputer dan laptop untuk menjalankan serta menyusun program koding serta model *software* YOLOv8 atau sebagai media utama untuk menjalankan proyek ini, terdapat jaringan Wi-Fi yang digunakan untuk koneksi ESP-32 CAM dengan server pada program kodingan yang disusun, *Deep Learning* akan digunakan model deteksi objek dengan sifat *multi* deteksi tanpa ada batasan tertentu.

Perancangan sistem yang kami gunakan meliputi spesifikasi perangkat keras sebagai berikut.

1. Mikrokontroler ESP-32 CAM dengan kamera OV2640.

2. Komputer server dengan spesifikasi RAM 8 GB, Windows 11 menyertai GPU
3. Jaringan Wi-Fi Lokal.

Berikut adalah spesifikasi perangkat lunak ialah sebagai berikut.

1. Arduino IDE sebagai program mikrokontroler pada ESP-32 CAM yang disertai VSCode untuk desain website.
2. YOLOv8 untuk Algoritma deteksi wajah.

Desain sistem memiliki alur sebagai berikut.

1. ESP-32 CAM menangkap gambar atau video dan mengirimkannya melalui koneksi jaringan internet Wi-Fi ke laptop atau komputer.
2. Gambar diproses oleh model YOLOv8 yang telah dilatih sebelumnya untuk mengenali wajah dengan data gambar deteksi wajah yang diprogram agar perangkat mengenali objek sehingga menampilkan data-data yang telah diprogram.
3. Deteksi yang berhasil akan ditampilkan dan disimpan
4. Hasil dievaluasi menggunakan parameter yang akurat dan presisi menyesuaikan performa sistem yang dijalankan.

Kemudian integrasi *Deep Learning* dilakukan menggunakan YOLOv8, sebuah model deteksi objek dari *Ultralytics* yang dikenal cepat dan akurat. Secara spesifik, penelitian ini menggunakan model YOLOv8n (Nano) yang telah dilatih sebelumnya (*pre-trained*) pada dataset COCO, yang kemudian diintegrasikan dengan server Python untuk pemrosesan inferensi. Sistem ini menggunakan arsitektur dua tahap. Tahap pertama adalah deteksi wajah, yang menggunakan model YOLOv8 untuk secara cepat menemukan lokasi semua wajah dalam *frame* yang dikirim dari ESP32-CAM. Tahap kedua adalah pengenalan wajah.

Untuk tahapan ini, sistem menggunakan pustaka *face\_recognition* yang dibangun di atas dlib. Pustaka ini menggunakan model ResNet-34 yang telah dilatih pada dataset *Labeled Faces in the Wild* (LFW) untuk menghasilkan 128-d *face embeddings*. Selanjutnya setiap wajah yang terdeteksi akan dipotong (*cropped*), diubah menjadi *encoding* (vektor fitur), dan dibandingkan dengan database wajah yang telah dikenali, seperti yang ditunjukkan pada Gambar 1.

Selanjutnya ada proses pengumpulan dan pengujian data, pengumpulan data diperoleh dari stream ESP-32 CAM secara langsung. Sistem diuji dengan berbagai skenario dengan beberapa karakteristik dan parameter seperti nama dan wajah tidak dikenali. Parameter pengujian terdapat beberapa mencakup sebagai berikut.

1. Akurasi Deteksi: Rasio wajah yang berhasil dideteksi dibandingkan jumlah aktual.
2. Waktu Deteksi: Durasi proses dari gambar diterima sampai deteksi dilakukan.
3. *Frame Per Second* (FPS): Kecepatan Sistem per gambar yang ditimpa setiap detik dalam memproses video.
4. *False Positive*: Deteksi Wajah padahal tidak ada.
5. *False Negative*: Wajah tidak terdeteksi meski ada.

Hasil pengujian dibandingkan dalam berbagai kondisi untuk menilai keandalan sistem deteksi berbasis model YOLOv8 dengan sistem *Deep Learning* untuk deteksi objek dengan parameter yang telah diprogram, kinerja perangkat keras modul ESP-32 CAM sebagai sumber input video, keefektifan, efisiensi waktu dan akurasi dalam mendeteksi objek secara *real-time* dalam pengaplikasiannya.

## 4. Hasil dan Pembahasan

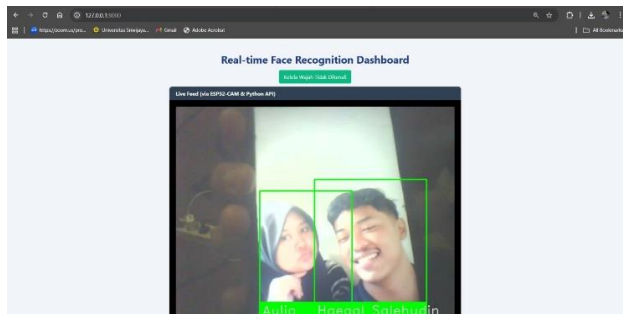
Bab ini menyajikan hasil implementasi prototipe sistem dan analisis data yang diperoleh dari pengujian kinerja. Hasil pengujian digunakan untuk mengevaluasi efektivitas sistem deteksi dan pengenalan wajah dalam berbagai kondisi pencahayaan.

### 4.1 Hasil

Bagian ini memaparkan hasil fungsional dari sistem yang telah dibangun, serta data kuantitatif dari pengujian parameter yang telah ditentukan di Bab 3.

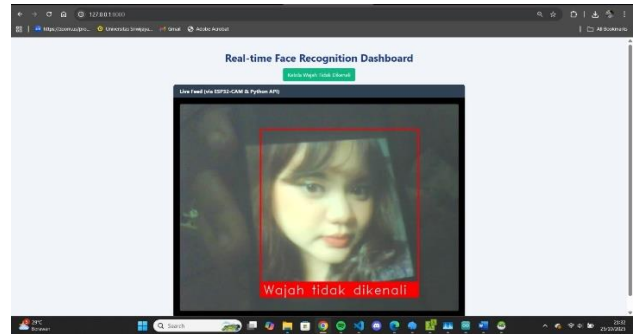
#### 4.1.1 Hasil Implementasi Sistem Fungsional

Sistem berhasil diimplementasikan dalam bentuk *dashboard* web yang terhubung dengan *stream video* dari ESP32-CAM melalui server API Python. Berikut adalah tiga skenario pengujian fungsional utama:



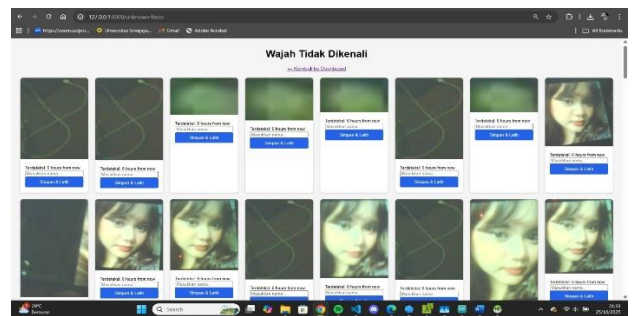
Gambar 4.1 Hasil deteksi dan pengenalan wajah terdaftar

Skenario Wajah Dikenali Ketika sistem mendeteksi wajah yang telah terdaftar di database (*folder known\_faces*), sistem berhasil melakukan pencocokan. Seperti yang ditunjukkan pada Gambar 4.1, sistem secara *real-time* menampilkan kotak pembatas (*bounding box*) berwarna hijau dan memberi label nama yang sesuai ("Aulia" dan "Haeqal Salehudin").



Gambar 4.2 Hasil deteksi wajah tidak terdaftar

Skenario Wajah Tidak Dikenali Ketika sistem mendeteksi wajah yang tidak ada di database, sistem tetap berhasil mendeteksi wajah tersebut namun mengklasifikasikannya sebagai "tidak dikenali". Gambar 4.2 menunjukkan sistem menampilkan kotak pembatas berwarna merah dengan label "Wajah tidak dikenali".



Gambar 4.3 Antarmuka *dashboard* untuk labeling wajah baru

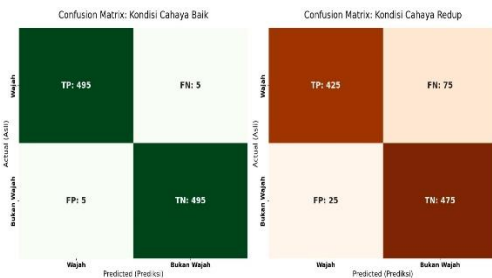
Skenario Manajemen Data dan Labeling Setiap wajah yang terdeteksi sebagai "Wajah tidak dikenali" secara otomatis dipotong dan disimpan ke database (Tabel *detection\_logs*). Gambar 4.3 menunjukkan antarmuka "Kelola Wajah Tidak Dikenali", di mana administrator dapat melihat semua wajah yang tidak dikenal. Antarmuka ini menyediakan fungsionalitas untuk memberi nama (label) baru, yang kemudian memindahkan gambar tersebut ke database wajah dikenali dan memicu server Python untuk melatih ulang modelnya.

#### 4.1.2 Hasil Pengujian Kinerja Sistem

Parameter	Kondisi Cahaya Baik	Kondisi Cahaya Redup
<b>Parameter Deteksi (YOLOv8)</b>		
Akurasi Deteksi	99%	75%
Waktu Deteksi (ms)	120 ms	150 ms
Frame Per Second (FPS)	8 FPS	5 FPS
False Positive	1%	5%
False Negative	1%	15%
<b>Parameter Pengenalan (Face Recog.)</b>		
Akurasi Pengenalan	98%	70%
False Acceptance Rate (FAR)	0%	3%
False Rejection Rate (FRR)	2%	27%

Tabel 4.1 Hasil pengujian parameter kinerja sistem

Pengujian kinerja dilakukan untuk mengukur parameter deteksi dan pengenalan dalam dua skenario pencahayaan yang berbeda. Kondisi Cahaya Baik (pencahayaan cukup) dan Kondisi Cahaya Redup (pencahayaan minim). Hasil pengujian disajikan pada Tabel 4.1.



Gambar 4.4 Confusion matrix dari hasil pengujian

## 4.2 Pembahasan

Bagian ini menganalisis dan menginterpretasi data yang disajikan pada bagian 4.1

### 4.2.1 Analisis Kinerja Deteksi (YOLOv8)

Berdasarkan Tabel 4.1 dan Gambar 4.4, kinerja deteksi model YOLOv8 sangat dipengaruhi oleh kondisi pencahayaan:

Pada Cahaya Baik: Sistem menunjukkan kinerja nyaris sempurna dengan Akurasi Deteksi 99%. *Confusion*

*matrix* mengkonfirmasi ini dengan nilai *False Negative* (FN) dan *False Positive* (FP) yang sangat rendah, masing-masing hanya 1% (5 dari 500 sampel). Ini membuktikan bahwa model YOLOv8 sangat andal dalam menemukan wajah di kondisi ideal.

Pada Cahaya Redup: Terjadi penurunan kinerja yang signifikan. Akurasi Deteksi turun menjadi 75%. Penyebab utamanya adalah lonjakan *False Negative* (FN) menjadi 15% (75 dari 500 sampel). Ini berarti dalam 15% kasus, model gagal menemukan wajah yang sebenarnya ada di frame. Kegagalan inilah yang diamati selama pengujian sebagai *bounding box* yang "jarang tampil" atau "hilang-hilang".

Kecepatan (FPS): Sistem mencapai 8 FPS (sekitar 120 ms per *frame*) pada cahaya baik. Kinerja ini cukup untuk disebut *real-time* dan dicapai melalui strategi *skip-frame* yang dijelaskan di Bab 3, yang menyeimbangkan antara beban pemrosesan dan kelancaran video.

### 4.2.2 Analisis Kinerja Pengenalan (Face Recognition)

Kinerja pengenalan (pencocokan identitas) juga menunjukkan ketergantungan yang kuat pada kualitas gambar:

Pada Cahaya Baik: Sistem mencapai Akurasi Pengenalan 98%. Yang terpenting, *False Acceptance Rate* (FAR) adalah 0%. Ini menunjukkan bahwa sistem sangat aman dan tidak ada subjek tidak dikenal yang salah diidentifikasi sebagai pengguna terdaftar. *False Rejection Rate* (FRR) sebesar 2% adalah hasil yang sangat baik, menunjukkan hanya sedikit kasus di mana pengguna terdaftar salah diidentifikasi sebagai "Wajah tidak dikenali".

Pada Cahaya Redup: Kinerja pengenalan turun drastis ke 70%. Masalah utamanya adalah lonjakan *False Rejection Rate* (FRR) menjadi 27%. Ini adalah kendala terbesar sistem. Ini berarti, bahkan jika YOLOv8 berhasil mendeteksi wajah (melewati tahap 1), gambar wajah yang terlalu gelap atau buram menghasilkan *encoding* (vektor fitur) yang buruk. *Encoding* yang buruk ini gagal dicocokkan dengan *encoding* berkualitas tinggi yang ada di database, sehingga sistem salah melabeli pengguna terdaftar sebagai “Wajah tidak dikenali”.

#### 4.2.3 Kendala dan Keterbatasan

Berdasarkan hasil dan pembahasan, beberapa kendala utama sistem ini telah diidentifikasi:

**Kualitas Sensor:** Kendala utama adalah keterbatasan sensor kamera OV2640 pada ESP32-CAM, yang memiliki rentang dinamis rendah dan sangat sensitif terhadap kondisi pencahayaan.

**Kinerja Komputasi:** Terdapat *trade-off* yang jelas antara resolusi gambar dan kecepatan (FPS). Meskipun telah dioptimasi dengan *skip-frame*, pemrosesan ganda (YOLOv8 dan *Face Recognition*) pada resolusi VGA (640x480) tetap menjadi beban berat bagi server.

**Ketergantungan Jaringan:** Sistem bergantung pada koneksi Wi-Fi yang stabil untuk *streaming* video dari ESP32-CAM ke server Python. Koneksi yang tidak stabil dapat menyebabkan *stream timeout* dan *lag* yang parah.

## 5. Kesimpulan dan Saran

Bab ini merangkum temuan-temuan utama dari penelitian dan memberikan rekomendasi untuk pengembangan sistem di masa depan.

### 5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem deteksi dan pengenalan wajah cerdas berhasil diimplementasikan dengan mengintegrasikan empat komponen utama: mikrokontroler ESP32-CAM sebagai penangkap gambar, server Python (*Flask*) sebagai pemroses, model YOLOv8 untuk deteksi, dan pustaka *face\_recognition* untuk pengenalan, dengan *dashboard* web berbasis Laravel sebagai antarmuka pengguna.
2. Sistem mampu membedakan secara fungsional antara wajah yang terdaftar (*known*) dan wajah yang tidak terdaftar (*unknown*), yang ditandai dengan kotak pembatas hijau (dengan label nama) dan merah (dengan label "Wajah tidak dikenali").
3. Fitur database interaktif berhasil diimplementasikan, memungkinkan sistem untuk menyimpan wajah tidak dikenal dan memberikan fungsionalitas *labeling* (pendaftaran wajah baru) secara *real-time* melalui *dashboard*.
4. Kinerja sistem sangat bergantung pada kondisi pencahayaan. Pada kondisi cahaya baik, sistem bekerja optimal dengan Akurasi Deteksi 99% dan Akurasi Pengenalan 98%.
5. Pada kondisi cahaya redup, kinerja sistem menurun secara signifikan. Kegagalan utama terjadi pada peningkatan *False Negative* (FN) sebesar 15% (model gagal menemukan wajah) dan *False Rejection Rate* (FRR) sebesar 27% (model gagal mengenali wajah terdaftar).

### 5.2 Saran

Untuk pengembangan penelitian di masa depan dan perbaikan dari kendala yang ada, berikut adalah beberapa saran yang dapat dipertimbangkan:

1. Peningkatan Perangkat Keras (Kamera): Mengingat keterbatasan utama adalah kualitas sensor ESP32-CAM pada cahaya redup, disarankan untuk menggunakan modul kamera dengan kualitas lebih



tinggi (misalnya, yang memiliki *sensor night vision* atau rentang dinamis lebih baik) atau menambahkan pencahayaan inframerah (IR) eksternal untuk memastikan kualitas gambar tetap baik di lingkungan gelap.

2. Optimasi Model untuk Cahaya Redup: Melakukan *fine-tuning* model YOLOv8 pada dataset khusus yang berisi banyak gambar wajah dalam kondisi gelap atau buram. Ini dapat membantu menurunkan angka *False Negative* (FN) dan meningkatkan akurasi deteksi di kondisi sulit.
3. Peningkatan Robustness Pengenalan: Untuk mengurangi *False Rejection Rate* (FRR), disarankan untuk memperbanyak variasi gambar pada database wajah dikenali (misal: mengambil 5-10 foto per orang dari berbagai sudut dan pencahayaan) agar model pengenalan lebih toleran terhadap perubahan.
4. Migrasi ke Pemrosesan GPU: Kinerja sistem (FPS) saat ini dibatasi oleh pemrosesan CPU. Untuk meningkatkan kecepatan pemrosesan secara drastis, terutama jika ingin menambah jumlah kamera atau resolusi, disarankan untuk memigrasikan server API Python agar berjalan di perangkat yang memiliki GPU (CUDA).
5. Manajemen Koneksi Jaringan: Sistem sangat bergantung pada stabilitas Wi-Fi. Disarankan untuk mengimplementasikan mekanisme *buffering* atau *retry* yang lebih canggih di server Python untuk menangani stream yang terputus-putus tanpa menyebabkan *crash* atau *lag* yang parah.

## Daftar Pustaka

[1] A. Nugroho, *Computer Vision: Teori dan Implementasi*. Yogyakarta: Andi Publisher, 2020.

[2] Agachi et al., "Sistem Keamanan Kotak Amal Menggunakan Face Recognition Berbasis ESP32-CAM," Universitas Dinamika, 2025.

[3] Awangga et al., "Sistem Keamanan Rumah Menggunakan Face Recognition Berbasis ESP32-CAM," UIN Sunan Gunung Djati Bandung, 2024.

[4] Pratama dan Ulum, "Sistem Keamanan Rumah Menggunakan ESP32-CAM Berbasis Face Recognition," *Jurnal Komputasi*, vol. 9, no. 2, pp. 45–52, 2023.

[5] A. Budijanto, s. Winardi, dan K. E. Susilo, *INTERFACING ESP32*. Yogyakarta: Deepublish, 2021.

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2001, vol. 1, pp.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.

[9] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 815–823.

[11] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.