

2213020110_Muhammad Choirul Anwar.docx.pdf

by Turnitin Arise

Submission date: 01-Jul-2026 09:44PM (UTC-0700)

Submission ID: 2992454861

File name: 2213020110_Muhammad_Choirul_Anwar.docx.pdf (2.85M)

Word count: 10447

Character count: 65245

BAB I PENDAHULUAN

A. Latar Belakang

Energi surya merupakan salah satu sumber energi terbarukan yang banyak dimanfaatkan karena ramah lingkungan dan memiliki potensi besar untuk memenuhi kebutuhan energi listrik. Panel surya sebagai perangkat utama dalam pemanfaatan energi ini harus berada dalam kondisi optimal agar mampu menghasilkan daya secara maksimal (Effendi et al., 2024). Perawatan dan pemantauan kondisi panel surya menjadi aspek penting untuk menjamin performa sistem, terutama pada instalasi yang digunakan dalam jangka panjang.

Permasalahan yang muncul adalah kerusakan fisik seperti retak, goresan, *shading*, hingga penumpukan debu yang dapat menyebabkan penurunan efisiensi panel surya (Wulan et al., 2025). Banyak sistem deteksi berbasis *computer vision* membutuhkan perangkat berkinerja tinggi seperti GPU ataupun komputer spesifikasi besar untuk menjalankan model deteksi *modern* (Prasetya et al., 2024). Kondisi ini menyebabkan implementasi *monitoring* visual menjadi kurang terjangkau dan sulit diterapkan pada lingkungan dengan keterbatasan sumber daya komputasi.

Beberapa penelitian terdahulu telah memanfaatkan *deep learning* untuk mendeteksi kerusakan panel surya. Studi oleh (Wiliani et al., 2025) menggunakan metode peningkatan kontras, yaitu *Contrast Limited Adaptive Histogram Equalization* (CLAHE) yang berfokus pada tahap *preprocessing* untuk memperjelas ciri kerusakan sebelum diklasifikasi. Penelitian lain oleh (Prasetya et al., 2024) berhasil mencapai *precision* di atas 85%. Namun, kedua penelitian tersebut dijalankan pada *platform* komputasi berbasis GPU yang membutuhkan daya dan biaya tinggi.

Di sisi lain, penggunaan perangkat *IoT* seperti *ESP32-CAM* telah banyak diteliti sebagai media deteksi objek ringan. Penelitian oleh (Pratama & Laksmiati, 2024) menunjukkan bahwa *ESP32-CAM* dapat menjalankan

model deteksi sederhana seperti *YOLOv8* untuk aplikasi *real-time*, meskipun dengan keterbatasan kapasitas komputasi. Hal yang sama ditemukan dalam penelitian (Auliya et al., 2023), di mana *ESP32-CAM* dinyatakan layak untuk sistem *monitoring* berbasis *computer vision* skala kecil, namun belum diterapkan secara spesifik pada deteksi panel surya.

Berdasarkan permasalahan tersebut, diperlukan sistem deteksi kondisi panel surya yang bersifat ringan, *real-time*, dan dapat berjalan pada perangkat rendah daya seperti *ESP32-CAM*. Penelitian ini mengusulkan implementasi sistem deteksi berbasis arsitektur *hybrid* dua tahap, di mana *ESP32-CAM* berfungsi sebagai modul akuisisi citra sekaligus melakukan deteksi ringan awal (*lightweight pre-detection*) seperti deteksi gerakan atau keberadaan objek abnormal di permukaan panel. Jika terdeteksi adanya perubahan atau objek, *ESP32-CAM* akan mengirimkan *video stream* ke komputer *edge* (laptop) untuk diproses lebih lanjut dengan model *YOLOv8* yang telah dilatih untuk melakukan klasifikasi detail ke dalam 7 kelas kondisi panel. Dengan pendekatan ini, sistem *monitoring* panel surya diharapkan menjadi lebih efisien dalam penggunaan bandwidth dan daya, serta lebih responsif dalam mendeteksi anomali secara *real-time*.

B. Identifikasi Masalah

Belum tersedianya sistem deteksi kondisi panel surya yang menerapkan pendekatan deteksi *hybrid* dua tahap, di mana *ESP32-CAM* berperan sebagai *pre-detector* ringan dan *YOLOv8* sebagai *classifier* akurat. Integrasi ini diharapkan dapat mengurangi beban komputasi dan bandwidth dengan hanya mengirimkan *frame* yang relevan ke laptop.

68

C. Rumusan Masalah

Adapun rumusan masalah yang didapatkan yaitu:

1. Bagaimana merancang dan mengintegrasikan sistem deteksi kondisi panel surya yang terdiri dari modul *ESP32-CAM* sebagai *pre-detector* ringan dan aplikasi komputer berbasis *YOLOv8* sebagai penerima dan pemroses utama *YOLOv8*?

2. Bagaimana kinerja model *YOLOv8* dalam mendeteksi kondisi panel surya pada perangkat komputasi ringan secara *real-time*?

8

D. Batasan Masalah

Agar penelitian ini lebih terarah dan terfokus, maka ditetapkan batasan masalah sebagai berikut

1. Penelitian hanya difokuskan pada deteksi kondisi panel surya berupa kotoran burung, cacat, debu, daun, tidak cacat, kerusakan fisik, dan salju.
2. Metode deteksi menggunakan model *YOLOv8* yang dioptimalkan untuk komputasi ringan.
3. *Hardware* utama menggunakan *ESP32-CAM* sebagai perangkat pemrosesan visual
4. Lokasi pengujian dilakukan pada instalasi panel surya skala kecil.
5. Pengujian dilakukan dalam kondisi pencahayaan siang hari yang cukup.
6. Sistem hanya memberikan *output* berupa hasil deteksi visual, tanpa perhitungan efisiensi daya listrik.
7. Hasil akhir penelitian berupa *prototype* sistem deteksi kondisi panel surya yang dapat beroperasi secara *real-time* dengan notifikasi jika ditemukan anomali.

8

E. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Mengimplementasikan sistem *monitoring* panel surya berbasis arsitektur *hybrid* dengan pemrosesan tingkat awal pada *ESP32-CAM* untuk efisiensi pengiriman data dan inferensi utama menggunakan *YOLOv8* di laptop
2. Menyediakan evaluasi kinerja *YOLOv8* pada perangkat komputasi rendah daya sebagai referensi bagi penelitian *edge-computing* dan *computer vision*.

60

F. Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat dalam pengembangan teknologi *monitoring* panel surya yang lebih efisien dan terjangkau melalui

pemanfaatan *ESP32-CAM* sebagai perangkat pendeteksi awal visual. Sistem yang dirancang dapat menjadi solusi alternatif bagi instalasi panel surya skala kecil maupun menengah yang membutuhkan deteksi kondisi secara *real-time* tanpa ketergantungan pada perangkat komputasi tinggi. Selain itu, hasil penelitian ini dapat menjadi referensi bagi pengembangan metode *computer vision* pada perangkat komputasi ringan, khususnya dalam implementasi model *YOLOv8* pada *edge device*, sehingga dapat mendorong penelitian lanjutan dalam bidang energi terbarukan, *IoT*, dan *edge-AI*.

LANDASAN TEORI

A. Teori dan Penelitian Terdahulu

1. Landasan teori

20

a. Panel Surya (*Photovoltaic*)

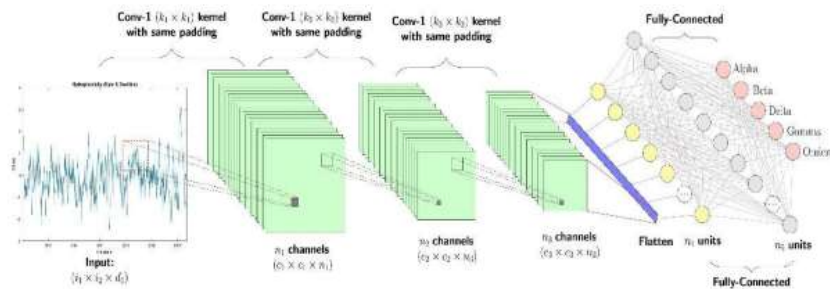
Panel surya atau *photovoltaic* (PV) adalah perangkat semikonduktor yang berfungsi mengubah energi foton dari sinar matahari menjadi energi listrik arus searah (DC). Menurut (Abdillah et al., 2025) efisiensi panel surya sangat bergantung pada kondisi fisik permukaannya, dimana gangguan seperti debu, kotoran burung, atau keretakan (*micro-crack*) dapat menghambat penyerapan cahaya dan menyebabkan fenomena *hotspot*. Sementara itu, (Widia et al., 2023) mendefinisikan panel surya sebagai komponen utama dalam PLTS yang kinerjanya bersifat fluktuatif tergantung pada radiasi matahari dan kebersihan modul. Pendapat lain dikemukakan oleh (Assalam & Muhammad, 2024) yang menyatakan bahwa degradasi performa panel surya seringkali disebabkan oleh faktor eksternal yang tidak terdeteksi secara visual dalam waktu lama, sehingga menurunkan *lifetime* perangkat.

b. *Deep Learning* dan *Convolutional Neural Network* (CNN)

38

Computer Vision adalah cabang dari kecerdasan buatan yang melatih komputer untuk menafsirkan dan memahami dunia visual. Menurut (Tresya et al., 2021), *computer vision* bekerja dengan meniru cara kerja sistem visual manusia menggunakan algoritma digital untuk memproses, menganalisis, dan memahami citra digital. Dalam penerapannya, *computer vision* sering dikombinasikan dengan *Deep Learning*. (Berliana & Totok, 2024) menjelaskan bahwa *Deep Learning* adalah sub-bidang *Machine Learning* yang menggunakan arsitektur *Neural Networks* berlapis banyak (seperti CNN) untuk mengekstrak fitur objek secara otomatis tanpa rekayasa fitur manual.

40



Gambar 2.1 *Arsitektur Deep Convolutional Neural Network*

Sumber : shutterstock

Operasi matematis fundamental dalam *Convolutional Neural Network* (CNN) meliputi konvolusi dan fungsi aktivasi.

- 1). Operasi Konvolusi Berfungsi untuk mengekstrak fitur dari citra *input*. Secara matematis, operasi konvolusi diskrit dua dimensi didefinisikan sebagai berikut:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \dots (2.1)$$

Keterangan:

- a). $S(i, j)$: Hasil peta fitur (*feature map*) pada posisi (i, j)
 - b). I : Citra *input*
 - c). K : *Kernel (filter)*
 - d). m, n : Indeks dimensi *kernel*
- 2). Fungsi Aktivasi (*SiLU*) *YOLOv8* dan arsitektur modern lainnya menggunakan fungsi aktivasi *non-linear Sigmoid Linear Unit (SiLU)* atau sering disebut *Swish* untuk memungkinkan jaringan mempelajari pola yang kompleks. Rumus *SiLU* didefinisikan sebagai:

$$f(x) = x \cdot \sigma(x) = \frac{x}{1 + e^{-x}} \dots (2.2)$$

Dimana $\sigma(x)$ adalah fungsi *sigmoid* logistik.

Lebih lanjut, (Yuliany & Rachman, 2022) menekankan bahwa keunggulan *Deep Learning* dalam pengolahan citra terletak pada kemampuannya menangani variasi data yang besar dan kompleks dengan akurasi yang terus meningkat seiring bertambahnya jumlah data latih

c. Deteksi Perubahan (*Frame Differencing*)

Metode ini sangat cocok untuk *ESP32-CAM* karena tidak membutuhkan memori besar (*SRAM* 520 KB) seperti arsitektur *Deep Learning*. Konsepnya adalah membandingkan intensitas piksel antara dua *frame* video yang diambil secara berurutan untuk mendeteksi adanya objek atau anomali yang masuk ke area panel surya.

Rumus Matematis Deteksi Perubahan:

Operasi ini dilakukan dengan mengurangkan citra pada waktu t dengan citra pada waktu $t - 1$. Selisih absolut dari kedua citra tersebut didefinisikan sebagai:

$$D(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \dots (2.3)$$

Keterangan:

- 1) $D(x, y)$: Citra selisih (*difference image*).
- 2) $I_t(x, y)$: Intensitas piksel pada *frame* saat ini.
- 3) $I_{t-1}(x, y)$: Intensitas piksel pada *frame* sebelumnya.

Selanjutnya, untuk menentukan apakah ada objek yang cukup signifikan untuk memicu pengiriman data ke laptop, digunakan fungsi ambang batas (*thresholding*):

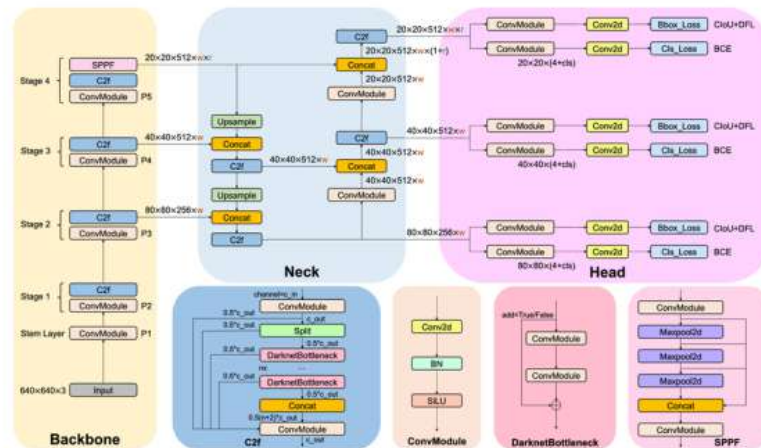
$$B(x, y) \{1, \text{ jika } D(x, y) > T, \text{ lainya } 0 \dots (2.4)$$

Di mana T adalah nilai *threshold* yang ditentukan. Jika jumlah piksel bernilai 1 melebihi batas tertentu, maka *ESP32-CAM* akan mengaktifkan fungsi *sendFrame()* ke laptop untuk dianalisis oleh *YOLOv8*

29

d. Algoritma *YOLO (You Only Look Once)* Versi 8

YOLO (You Only Look Once) adalah algoritma deteksi objek *real-time* yang memproses keseluruhan citra dalam satu kali evaluasi jaringan syaraf tiruan. Spesifik mengenai versi terbaru, (Jocher et al., 2024) menjelaskan bahwa *YOLOv8* merupakan iterasi mutakhir yang dikembangkan oleh Ultralytics dengan fitur *anchor-free detection*.



Gambar 2.2 Arsitektur *YOLOv8*

Sumber : ResearchGate

- 1). Arsitektur *YOLOv8* Secara struktural, arsitektur *YOLOv8* terdiri dari tiga bagian utama:
 - a). *Backbone* (CSPDarknet): Menggunakan struktur C2f (*Cross Stage Partial with 2 fusion*) yang menggantikan modul C3 pada versi sebelumnya. Modul ini dirancang untuk memperkaya aliran gradien informasi dan mengurangi beban komputasi, namun tetap mempertahankan bobot yang ringan.
 - b). *Neck* (PANet): Menggunakan *Path Aggregation Network* (PANet) untuk menggabungkan fitur dari berbagai skala resolusi gambar, memastikan objek kecil (seperti retakan halus) dan objek besar dapat terdeteksi dengan baik.
 - c). *Head* (*Decoupled*): *YOLOv8* menggunakan *Decoupled Head* yang memisahkan proses klasifikasi objek dan regresi

bounding box. Hal ini berbeda dengan versi lama yang menyatukannya, sehingga meningkatkan akurasi lokalisasi.

- 2). Fungsi Kerugian (*Loss Function*) Pelatihan model *YOLOv8* bertujuan meminimalkan *Loss Function* total yang merupakan gabungan dari *Box Loss* dan *Class Loss*. Untuk regresi *bounding box*, *YOLOv8* menggunakan *CIOU* (*Complete Intersection over Union*) dan *DFL* (*Distribution Focal Loss*).

Rumus *CIOU Loss* memperhitungkan area irisan, jarak pusat, dan rasio aspek:

$$L_{CIOU} = 1 - IoU + \frac{\rho^{2(b, b^{gt})}}{c^2} + \alpha v \dots (2.5)$$

Keterangan:

- a). *IoU*: *Intersection over Union* (Area Irisan / Area Gabungan).
- b). $\rho^{2(b, b^{gt})}$: Jarak *Euclidean* antara titik pusat kotak prediksi *b* dan kotak *ground truth* b^{gt} .
- c). *c*: Panjang diagonal dari kotak pembatas terkecil yang mencakup kedua kotak.
- d). α : Parameter pembobot *trade-off* positif.
- e). *v*: Parameter yang mengukur konsistensi rasio aspek.

Pendapat serupa disampaikan oleh (Cahyanti, 2023) yang menyebutkan bahwa *YOLOv8* menawarkan keseimbangan (*trade-off*) terbaik antara kecepatan (FPS) dan akurasi (*mAP*) untuk perangkat *edge computing* dibandingkan pendahulunya.

e. Mikrokontroler *ESP32-CAM*

ESP32-CAM adalah modul pengembangan berbasis *ESP32* yang dilengkapi dengan antarmuka kamera. Menurut (Wibowo et al., 2025) *ESP32-CAM* merupakan modul *Low-Cost IoT* yang mengintegrasikan *Wi-Fi*, *Bluetooth*, dan kamera *OV2640*, sehingga memungkinkan

pengambilan dan pengiriman citra nirkabel. (Nurul et al., 2022) menambahkan bahwa meskipun memiliki keterbatasan memori (SRAM 520 KB), *ESP32-CAM* mampu difungsikan sebagai *IP Camera* yang melakukan *streaming* data video menggunakan protokol *HTTP* atau *RTSP*.

Dalam arsitektur sistem ini, *ESP32-CAM* tidak melakukan pemrosesan *YOLO* secara internal (*on-device inference* yang berat), melainkan bertindak sebagai *node* akuisisi citra yang mengirimkan *frame* gambar dalam format *MJPEG* (*Motion JPEG*) ke server lokal untuk diproses. Hal ini didukung oleh (Ardiansyah et al., 2022) yang menyatakan bahwa *ESP32-CAM* adalah solusi paling efisien untuk purwarupa sistem visi komputer berbasis *IoT* karena kompatibilitasnya yang tinggi dengan *Arduino IDE* dan konsumsi daya yang rendah.

f. Metrik Evaluasi

Untuk mengukur keberhasilan model deteksi, digunakan alat ukur statistik berbasis *Confusion Matrix*. (Awaludin et al., 2022) menjelaskan *Confusion Matrix* sebagai tabel yang memvisualisasikan performa algoritma klasifikasi.

Berdasarkan parameter tersebut, kinerja sistem dievaluasi menggunakan rumus-rumus berikut:

- 1). Presisi (*Precision*) Menurut (Yuliany & Rachman, 2022), presisi mengukur seberapa akurat prediksi positif model.

$$Precision(P) = \frac{\{TP\}}{\{TP + FP\}}(2.6)$$

- 2). Recall (Sensitivitas) Mengukur kemampuan model dalam menemukan kembali seluruh objek yang ada.

$$Recall(R) = \frac{\{TP\}}{\{TP + FN\}}(2.7)$$

- 3). *Mean Average Precision (mAP)* Ini adalah metrik paling krusial dalam deteksi objek. *mAP* dihitung dengan mencari rata-rata dari *Average Precision (AP)* untuk setiap kelas. *AP* sendiri adalah area di bawah kurva *Precision-Recall (PR Curve)*. Secara integral didefinisikan sebagai:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i = \frac{1}{N} \sum_{i=1}^N \int_0^1 P(R) dR \dots (2.8)$$

Dimana N adalah jumlah kelas (dalam penelitian ini: kotoran burung, cacat, debu, daun, tidak cacat, kerusakan fisik, dan salju) dan $P(R)$ adalah fungsi presisi terhadap recall.

2. Kajian Pustaka

Penelitian mengenai deteksi panel surya dan implementasi *computer vision* pada perangkat *IoT* telah banyak dilakukan sebelumnya. Berikut adalah kajian terhadap lima penelitian terdahulu yang relevan:

- a. Penelitian pertama dilakukan oleh (Andi et al., 2025) dengan judul "*Optimalisasi Deteksi Kerusakan Elektrikal Panel Surya dengan Transfer Learning dan Augmentasi Terkontrol berbasis YOLOv8*". Permasalahan yang diangkat adalah sulitnya mendeteksi kerusakan spesifik pada panel surya secara otomatis. Peneliti menggunakan algoritma *YOLOv8* dengan teknik *transfer learning* pada dataset citra panel. Hasil penelitian menunjukkan bahwa model mampu mencapai akurasi deteksi di atas 90% untuk klasifikasi kerusakan. Perbedaan dengan penelitian ini adalah Andi et al. berfokus pada analisis citra dataset statis di komputer, sedangkan penelitian penulis mengimplementasikan sistem akuisisi data *real-time* menggunakan *ESP32-CAM*.
- b. Penelitian kedua oleh (Pratama & Laksmiati, 2024) berjudul "*Prototipe Sistem Deteksi Burung Menggunakan ESP32-Cam dan Algoritma YOLO*". Masalah yang dihadapi adalah hama burung

yang mengganggu area pertanian/panel. Solusi yang ditawarkan adalah integrasi kamera *ESP32-CAM* sebagai *input* visual untuk algoritma *YOLO*. Hasilnya membuktikan bahwa *ESP32-CAM* mampu mengirimkan *stream* video yang cukup baik untuk diproses oleh *YOLO* dengan latensi yang dapat diterima. Letak perbedaannya adalah pada objek deteksi (burung vs kondisi fisik panel surya) dan fokus penelitian penulis lebih pada klasifikasi jenis kondisi panel (bersih/retak/debu).

- c. Penelitian yang dilakukan oleh (Nugroho et al., 2025) berfokus pada klasifikasi kondisi permukaan panel surya menggunakan metode *deep learning*. Penelitian ini membandingkan empat arsitektur jaringan saraf konvolusional (CNN), yaitu MobileNetV2, DenseNet121, InceptionV3, dan *Xception*, dalam membedakan citra panel surya yang bersih dan berdebu. *Dataset* yang digunakan berupa citra visual RGB permukaan panel surya yang mengalami tingkat debu berbeda, kemudian diolah melalui proses pelatihan pada masing-masing arsitektur. Hasil penelitian menunjukkan bahwa model-model CNN tersebut mampu melakukan klasifikasi dengan akurasi yang cukup tinggi, serta ditemukan bahwa arsitektur tertentu lebih unggul dalam mengidentifikasi perbedaan tekstur dan pola debu pada panel surya.
- d. Penelitian keempat oleh (Auliya et al., 2023) berjudul "*Sistem Pemantauan Slot Parkir Menggunakan Sensor Ultrasonik JSN-SR04T dan Pengenalan Plat Nomor Kendaraan dengan ESP32-CAM*". Studi ini membahas pemanfaatan *ESP32-CAM* untuk *monitoring* visual di area parkir. Solusi yang diterapkan adalah pengambilan citra plat nomor untuk diolah sistem. Hasilnya menunjukkan *ESP32-CAM* valid digunakan sebagai perangkat *monitoring* visual berbiaya rendah. Perbedaannya adalah penelitian ini menggabungkan sensor ultrasonik dan fokus pada teks (OCR).

- e. Penelitian kelima dilakukan oleh (Apridiansyah et al., 2024) berjudul "Pengolahan Citra Berbasis Video *Processing* dengan Metode *Frame Difference* untuk Deteksi Gerak". Penelitian ini berangkat dari kebutuhan sistem pemantauan otomatis yang efisien untuk mendeteksi pergerakan objek melalui media CCTV. Peneliti menggunakan metode *Frame Difference* yang bekerja dengan cara membandingkan selisih nilai intensitas piksel antara dua *frame* video berurutan. Hasil penelitian menunjukkan bahwa metode ini memiliki keunggulan pada beban komputasi yang sangat ringan dan efisiensi tinggi dalam mengenali perubahan visual secara *real-time*.
- f. Penelitian keenam dilakukan oleh (Wiliani et al., 2025) yang berjudul "Solar Panel *Defect Detection Using CLAHE and Deep Learning Classification*" menggunakan metode *Contrast Limited Adaptive Histogram Equalization* (CLAHE) untuk *preprocessing* gambar panel surya sebelum diklasifikasi menggunakan CNN. Permasalahan yang dihadapi adalah citra panel surya dengan kontras rendah yang menyulitkan deteksi kerusakan. Solusi yang ditawarkan adalah penerapan CLAHE untuk meningkatkan kontras lokal pada gambar, kemudian menggunakan arsitektur CNN *custom* untuk klasifikasi. Hasil penelitian menunjukkan akurasi klasifikasi mencapai 92.3% pada *dataset thermal imaging* panel surya.

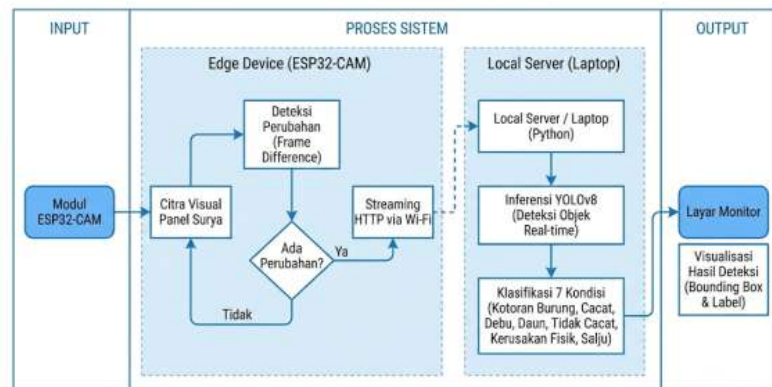
Berdasarkan tinjauan terhadap enam penelitian di atas, bahwa penggunaan metode *Deep Learning* (CNN dan YOLOv8) terbukti sangat efektif untuk klasifikasi dan deteksi kerusakan panel surya dengan akurasi yang tinggi. Namun, implementasi tersebut umumnya masih bergantung pada perangkat komputasi tinggi (GPU) atau *dataset* statis. Di sisi lain, penggunaan ESP32-CAM sebagai perangkat IoT telah terbukti valid untuk pemantauan visual berbiaya rendah, meskipun memiliki keterbatasan memori untuk menjalankan model AI yang kompleks secara mandiri.

Penelitian ini mengusulkan arsitektur *hybrid* yang nantinya dapat mengintegrasikan efisiensi metode *Frame Difference* pada sisi *edge* (*ESP32-CAM*) sebagai *filter* pemicu deteksi ringan, dan akurasi tinggi dari model *YOLOv8* pada sisi laptop untuk proses inferensi utama. Dengan pendekatan ini, sistem tidak hanya mampu mendeteksi berbagai kondisi panel surya secara akurat dan *real-time*, tetapi juga lebih efisien dalam penggunaan *bandwidth* dan sumber daya komputasi perangkat *IoT*.

83

B. Kerangka Berpikir

Kerangka berpikir dalam penelitian ini menggambarkan alur kerja sistem deteksi kondisi panel surya mulai dari pengambilan data hingga penyajian informasi kepada pengguna. Sistem dirancang untuk mengatasi masalah inspeksi manual dengan mengotomatisasi proses visualisasi kondisi panel.



Gambar 2.3 Kerangka Berpikir

67

Berdasarkan alur di atas, kerangka berpikir penelitian ini dapat dijelaskan sebagai berikut:

1. *Input* (Masukan): Data masukan berupa citra visual permukaan panel surya yang ditangkap secara terus-menerus (*real-time*) oleh modul

kamera pada perangkat *ESP32-CAM* yang terpasang menghadap panel surya.

2. Proses:

- a. *ESP32-CAM* bertindak sebagai perangkat *edge* yang melakukan analisis citra awal menggunakan metode *Frame Difference* untuk mendeteksi perubahan intensitas piksel secara lokal. Modul hanya akan mengirimkan *stream* video apabila terdeteksi adanya anomali atau perubahan visual pada permukaan panel
 - b. Pada komputer laptop, program Python menerima *stream* video tersebut dan memproses setiap *frame* gambar menggunakan model *Deep Learning YOLOv8* yang telah dilatih. Arsitektur *YOLOv8* mengekstrak fitur melalui *Backbone C2f*, menggabungkan fitur di *Neck PANet*, dan melakukan prediksi koordinat serta kelas pada *Decoupled Head*.
 - c. Model melakukan deteksi dan klasifikasi terhadap tujuh kelas objek gangguan pada panel, yaitu: *bird drop* (kotoran burung), *defective* (cacat), *dust* (debu), *leaf* (daun), *non defective* (tidak cacat), *physical damage* (kerusakan fisik), dan *snow* (salju).
3. *Output* (Keluaran): Hasil keluaran sistem ditampilkan pada antarmuka aplikasi *desktop* berupa tampilan visual yang memberikan kotak penanda (*bounding box*) pada area kerusakan, label kondisi, dan nilai kepercayaan (*confidence score*). Informasi ini menjadi dasar bagi teknisi untuk memantau kondisi fisik panel secara *real-time* dan mengambil keputusan pemeliharaan yang diperlukan.

BAB III

METODE PENELITIAN

A. Desain Penelitian

1. Jenis Penelitian

Penelitian ini merupakan penelitian pengembangan (*development research*) yang bertujuan untuk merancang dan mengimplementasikan sistem deteksi kondisi panel surya berbasis *computer vision* menggunakan *ESP32-CAM* sebagai perangkat akuisisi citra yang dilengkapi dengan logika deteksi perubahan (*Frame Difference*) sebagai *filter* data, serta model *YOLOv8* sebagai algoritma deteksi objek utama yang dijalankan pada aplikasi *desktop* berbasis Python. Sistem ini dikembangkan untuk mengotomatisasi inspeksi visual panel surya melalui analisis citra digital secara *real-time* tanpa memerlukan perangkat komputasi berkinerja tinggi. Model *YOLOv8* digunakan untuk melakukan ekstraksi fitur, deteksi, dan klasifikasi kondisi panel surya seperti bersih, berdebu, dan rusak dalam satu tahap pemrosesan. Proses penelitian meliputi perancangan sistem, pengolahan *dataset* citra panel surya, pelatihan dan pengujian model *YOLOv8*, serta integrasi *ESP32-CAM* dengan aplikasi *desktop*. Hasil penelitian ini berupa *prototype* sistem deteksi kondisi panel surya yang ringan, efisien, dan layak diterapkan pada lingkungan dengan keterbatasan sumber daya komputasi.

2. Variabel Penelitian

1. Variabel Independen

1. Citra panel surya

Citra panel surya yang digunakan merupakan gambar permukaan panel yang diperoleh melalui *ESP32-CAM* secara *real-time*. Citra tersebut mencakup berbagai kondisi panel surya, seperti panel bersih, panel berdebu, dan panel dengan kerusakan permukaan. Variasi kondisi citra ini menjadi variabel utama yang dianalisis oleh sistem deteksi berbasis *computer vision*.

2. Ambang batas (*Threshold*) *Frame Difference*

Ambang batas yang menentukan kapan sistem *ESP32-CAM* harus memicu (*trigger*) pengiriman *stream video* ke laptop untuk diproses lebih lanjut oleh model *YOLOv8*. *Threshold* ini diterapkan pada hasil perhitungan selisih intensitas piksel antara dua *frame* berurutan yang ditangkap oleh kamera *ESP32-CAM*..

3. Model *YOLOv8*

Model *YOLOv8* digunakan sebagai algoritma deteksi objek untuk melakukan proses ekstraksi fitur, pendeteksian, dan klasifikasi kondisi panel surya. Penggunaan *YOLOv8* sebagai model *deep learning* menjadi variabel bebas karena model ini menentukan cara sistem mengenali dan mengklasifikasikan kondisi panel berdasarkan citra masukan.

2. Variabel Dependen

1. Hasil Klasifikasi Kondisi Panel Surya

Hasil klasifikasi berupa label kondisi panel surya, yaitu kotoran burung, cacat, debu, daun, tidak cacat, kerusakan fisik, dan salju, yang dihasilkan berdasarkan analisis citra panel surya oleh model *YOLOv8*.

2. Akurasi Deteksi Sistem

Akurasi sistem merupakan ukuran ketepatan model *YOLOv8* dalam mendeteksi dan mengklasifikasikan kondisi panel surya, yang dihitung berdasarkan perbandingan antara hasil prediksi sistem dan data sebenarnya (*ground truth*).

3. Efisiensi Pengiriman Data

Efisiensi pengiriman data mengacu pada kemampuan sistem deteksi *hybrid* untuk mengurangi volume data yang dikirimkan dari *ESP32-CAM* ke laptop dengan memanfaatkan algoritma *Frame Difference* sebagai *filter* awal. Sistem ini

dirancang untuk hanya mengirimkan *stream video* ketika terdeteksi perubahan signifikan pada permukaan panel surya, sehingga menghindari pengiriman data kosong atau redundan yang tidak mengandung informasi anomali.

3. Variabel kontrol

1. *Dataset* citra panel surya

Dataset yang digunakan bersumber dari platform Roboflow dengan jumlah 3.962 gambar yang telah dianotasi. *Dataset* dikontrol agar tetap konsisten dalam jumlah data, format anotasi, dan kelas objek selama proses pelatihan dan pengujian model.

2. Perangkat Keras dan Lingkungan Pemrosesan

Perangkat keras yang digunakan berupa komputer atau laptop dengan spesifikasi yang sama selama proses pelatihan dan pengujian model YOLOv8. Pengendalian ini dilakukan untuk memastikan bahwa hasil pengujian tidak dipengaruhi oleh perbedaan spesifikasi *hardware*.

3. Kondisi pengujian sistem

Pengujian sistem dilakukan pada kondisi pencahayaan siang hari dan jaringan *Wi-Fi* yang stabil. Kondisi ini dikontrol agar tidak terjadi perbedaan signifikan yang dapat mempengaruhi hasil deteksi sistem.

3. Metode Pengumpulan data

Data untuk pengembangan model diperoleh melalui studi dokumenter dengan mengunduh *dataset* publik dari platform Roboflow. Roboflow merupakan platform terkemuka yang menyediakan berbagai *dataset computer vision* yang telah teranotasi untuk keperluan penelitian dan pengembangan teknologi kecerdasan buatan. *Dataset* ini telah dianotasi secara manual dengan tujuh kelas kerusakan dan kondisi utama yang sering ditemui pada panel surya, yaitu: *Bird Drop* (kotoran burung), *Defective* (cacat), *Dust* (debu), *Leaf* (daun), *Non Defective* (tidak cacat), *Physical Damage* (kerusakan fisik), dan *Snow* (salju). Anotasi ini dibuat

dalam format *bounding box* yang kompatibel dengan arsitektur *YOLOv8*. Pengelompokan dan pelabelan yang komprehensif ini bertujuan membantu algoritma *deep learning* untuk mempelajari ciri visual yang spesifik dari setiap kondisi, sehingga dapat mengidentifikasi dan mengklasifikasikan kerusakan panel surya secara akurat pada sistem yang dikembangkan.

B. Instrumen Penelitian

1. Perangkat Keras (*Hardware*)

Perangkat keras yang diperlukan dalam proses sistem yaitu sebagai berikut :

Tabel 3. 1 Perangkat Keras (*Hardware*)

Kapasitas Penyimpanan	512GB SSD
CPU	AMD Ryzen 3 PRO 2300U APU
RAM	8GB
GPU	AMD Radeon Vega 6
Modul Akuisisi Citra	ESP32-CAM (kamera OV2640, WiFi 2.4GHz)
Objek Penelitian	Panel Surya 10 wp <i>polycrystalline</i>

2. Perangkat Lunak (*Software*)

Perangkat lunak yang diperlukan dalam proses sistem yaitu sebagai berikut :

Tabel 3. 2 Perangkat Lunak (*Software*)

Sistem Operasi	Windows 11 Home 64-bit
Pengembangan <i>IoT</i>	Arduino IDE 2.3.2, Library esp32cam

Pengembangan AI & CV	Python 3.9.18, PyTorch 2.0.1, Ultralytics YOLOv8 8.0.196, OpenCV 4.8.1, Google Colab
IDE & Tools	Visual Studio Code, Git, Roboflow CLI

3. Dataset

Dataset utama adalah *dataset* publik dari Roboflow. Detail lengkapnya pada Tabel 3.3.

Tabel 3. 3 Dataset

Nama <i>Dataset</i>	Solar panel <i>Dataset</i>
Sumber	https://universe.roboflow.com/solar-panel-q3vlt/solar-panel-gydkp
Total Gambar	3.962 gambar
Kelas Objek	bird_drop, defective, dust, leaf, non_defective, physical_damage, snow
<i>Split Data</i>	Train: 2.774 img (70%) Validation: 792 img (20%) Test: 396 img (10%)

4. Analisis hasil

Analisis hasil pada penelitian ini bertujuan untuk menjelaskan metode evaluasi yang digunakan dalam menilai kinerja sistem deteksi dan klasifikasi kondisi panel surya berbasis arsitektur *hybrid* yang melibatkan *ESP32-CAM* dan algoritma *YOLOv8*. Pada tahap proposal ini, analisis difokuskan secara mendalam pada perencanaan teknik evaluasi dan pemodelan perhitungan matematis yang akan diterapkan, tanpa membahas atau menginterpretasikan nilai hasil pengujian aktual karena proses implementasi masih dalam tahap rancangan.

Evaluasi kinerja sistem dilakukan secara bertahap, dimulai dari efektivitas pemicu (*trigger*) pada perangkat *edge* hingga akurasi klasifikasi pada *server*. Untuk mengukur keberhasilan deteksi objek oleh model *YOLOv8*, digunakan alat ukur statistik berbasis *Confusion Matrix*. Metode ini membandingkan hasil prediksi model dengan kondisi sebenarnya

(*ground truth*) pada data uji yang telah ditentukan. *Confusion Matrix* secara visual menggambarkan hubungan antara kelas hasil prediksi dan kelas aktual, sehingga dapat memetakan tingkat ketepatan maupun pola kesalahan sistem dalam mengidentifikasi kondisi panel surya. Matriks evaluasi ini terdiri dari empat komponen utama, *True Positive* (TP) yang menunjukkan jumlah kerusakan yang berhasil terdeteksi dengan benar, *False Positive* (FP) yang menunjukkan objek yang salah terdeteksi. *False Negative* (FN) yang menunjukkan adanya kerusakan namun luput dari deteksi sistem, serta *True Negative* (TN) yang menunjukkan kondisi bersih yang divalidasi dengan benar oleh sistem⁶.

Berdasarkan komponen dalam *Confusion Matrix* tersebut, tingkat Akurasi sistem dihitung untuk mengetahui proporsi prediksi yang benar terhadap seluruh total data pengujian. Akurasi menjadi metrik evaluasi mendasar yang menggambarkan performa umum model dalam melakukan klasifikasi citra panel surya. Secara matematis, perhitungan akurasi dilakukan dengan menjumlahkan nilai *True Positive* dan *True Negative*, kemudian membaginya dengan total keseluruhan data (TP, TN, FP, FN), lalu dikalikan seratus persen.

Selain aspek akurasi, analisis hasil juga mencakup evaluasi pada sisi *ESP32-CAM* untuk menilai efisiensi transmisi data. Peneliti akan menganalisis keberhasilan algoritma *Frame Difference* dalam memicu pengiriman *stream* video hanya saat terdeteksi anomali, yang bertujuan untuk membuktikan efisiensi penggunaan *bandwidth* pada arsitektur yang dirancang. Nilai parameter TP, TN, FP, dan FN nantinya diperoleh dari hasil pengujian sistem menggunakan *dataset* uji dari Roboflow yang berjumlah 3.962 gambar. Hasil perhitungan ini akan menjadi landasan objektif untuk menilai apakah model *YOLOv8* dan perangkat *ESP32-CAM* layak digunakan dalam sistem *monitoring real-time*. Pembahasan mendalam mengenai nilai numerik akurasi dan performa sistem secara kuantitatif akan dipaparkan sepenuhnya pada laporan akhir penelitian setelah tahap implementasi dan pengujian lapangan selesai dilaksanakan.

C. Tempat Dan Jadwal Penelitian

Tabel 3.4 Jadwal Penelitian

No	Kegiatan	Bulan Ke -																							
		1				2				3				4				5				6			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur	■	■	■	■	■	■																		
2	Pengumpulan Data			■	■	■	■	■	■	■	■	■	■												
3	Desain Sistem								■	■	■	■	■	■	■										
4	Coding atau Implementasi										■	■	■	■	■	■	■								
5	Uji Coba															■	■	■	■	■	■	■	■	■	
6	Evaluasi dan Analisis Hasil																	■	■	■	■	■	■	■	■
7	Penulisan Laporan			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Jadwal penelitian yang dilakukan selama 6 bulan. Kegiatan penelitian secara rinci dijelaskan pada tabel sebagai berikut :

D. Objek Penelitian / Subjek Penelitian

1. Analisis Kebutuhan Sistem

a. Kebutuhan Fungsional

- 1). Sistem harus mampu mendeteksi perubahan visual ringan menggunakan metode *Frame Difference* pada sisi mikrokontroler sebagai pemicu (*trigger*) pengiriman data.
- 2). Sistem harus mampu menerima dan menampilkan *streaming* citra panel surya secara *real-time* yang dikirimkan oleh *ESP32-CAM*.

- 3). Sistem harus mampu menjalankan proses deteksi dan klasifikasi kondisi panel surya menggunakan algoritma *YOLOv8* pada aplikasi *desktop* berbasis Python.
- 4). Sistem harus mampu menampilkan hasil deteksi berupa citra panel surya yang telah diberi *bounding box* dan label kondisi secara jelas kepada pengguna.

b. Kebutuhan non-fungsional

Sistem harus memiliki kinerja yang stabil dengan waktu respons yang cepat, akurasi deteksi objek yang baik, serta mampu berjalan pada perangkat komputasi dengan spesifikasi menengah tanpa memerlukan GPU khusus. Selain itu, aplikasi harus berjalan dengan lancar dan memberikan tampilan visual yang mudah dipahami oleh pengguna.

c. Kebutuhan pengguna

Sistem harus mudah digunakan oleh pengguna atau teknisi, menampilkan hasil deteksi kondisi panel surya secara jelas dan informatif. Pengguna diharapkan dapat dengan mudah memahami kondisi panel surya yang terdeteksi melalui tampilan visual tanpa memerlukan keahlian khusus di bidang *computer vision*.

d. Kebutuhan data

Data yang digunakan berupa citra digital panel surya yang diperoleh dari *dataset* Roboflow dan citra hasil *streaming ESP32-CAM*. Data harus memiliki anotasi yang jelas dan konsisten agar dapat diproses dengan baik oleh model *YOLOv8* serta menghasilkan informasi deteksi yang mudah dipahami oleh pengguna.

2. Objek Penelitian

Objek penelitian ini adalah prototipe sistem deteksi kondisi panel surya berbasis arsitektur *hybrid* menggunakan *ESP32-CAM* dan *YOLOv8*. Perangkat keras (*ESP32-CAM*) bertindak sebagai *edge device* untuk akuisisi dan penyaringan citra, sementara perangkat lunak (aplikasi Python) bertindak sebagai pusat inferensi cerdas. Sistem ini

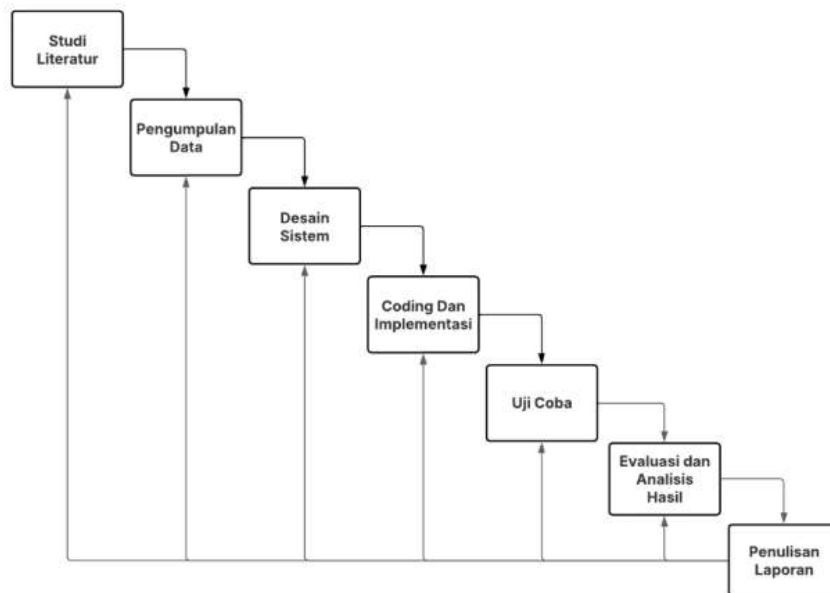
memungkinkan pengguna untuk memantau kondisi fisik panel surya secara visual melalui antarmuka aplikasi *desktop* yang menampilkan hasil deteksi berupa *bounding box* dan klasifikasi kondisi.

23 3. Subjek Penelitian

Subjek dalam penelitian ini adalah pengguna sistem yang melakukan pemantauan kondisi panel surya menggunakan aplikasi *desktop* yang dikembangkan. Interaksi antara pengguna dan sistem menjadi aspek penting untuk memastikan bahwa sistem mampu menyajikan informasi deteksi secara jelas, akurat, dan mudah dipahami. Umpan balik dari subjek penelitian digunakan sebagai bahan evaluasi terhadap kinerja dan kemudahan penggunaan sistem deteksi kondisi panel surya.

E. Prosedur Penelitian

Sistem deteksi kondisi panel surya berbasis *ESP32-CAM* dan *YOLOv8* ini dikembangkan dengan mengikuti metode *Waterfall* dan metode kuantitatif. 23 Metode *Waterfall* merupakan metode pengembangan perangkat lunak yang memiliki tahapan yang jelas, terstruktur, dan berurutan (linier), sehingga setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Pendekatan ini bertujuan untuk meminimalkan kesalahan dalam proses pengembangan sistem dan memastikan sistem dibangun secara sistematis (Utomo et al., 2022). 53 Metode kuantitatif digunakan dalam penelitian ini untuk mengukur dan mengevaluasi kinerja sistem deteksi kondisi panel surya yang dikembangkan. Evaluasi dilakukan dengan menganalisis hasil pengujian model *YOLOv8* 59 menggunakan metrik pengukuran seperti akurasi, *precision*, *recall*, dan *mean average precision (mAP)* berdasarkan data pengujian dan validasi yang diperoleh dari *dataset* citra panel surya (Wahyudi, 2022). Pendekatan ini memungkinkan peneliti untuk menilai tingkat keberhasilan sistem secara objektif dan terukur.



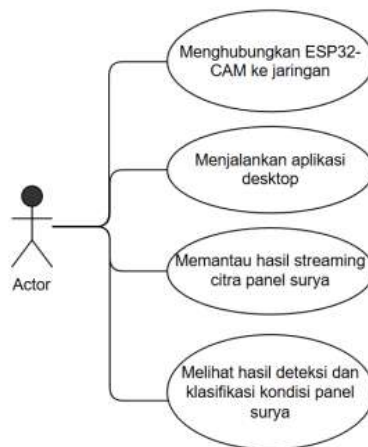
Gambar 3.1 Flowchart Waterfall

F. Teknik Analisis Data

1. Desain Sistem

a. Arsitektur Sistem((System Architecture))

1). Diagram Use case

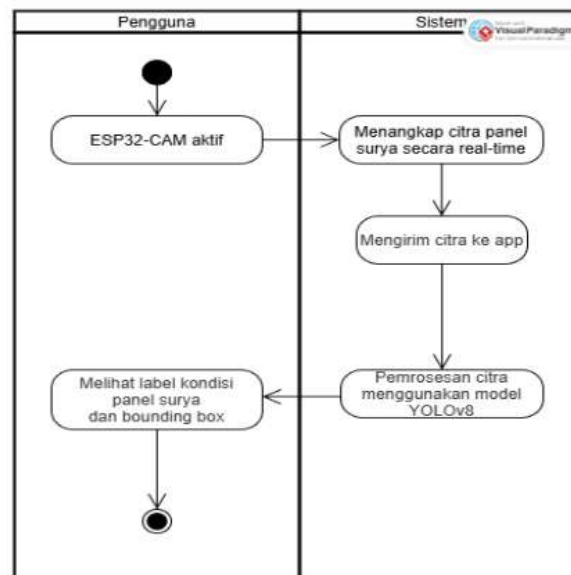


Gambar 3.2 Diagram Use Case

Berdasarkan Gambar 3.2 Diagram Use Case, sistem memiliki satu aktor utama yaitu pengguna (operator/teknisi). Pengguna dapat melakukan beberapa aktivitas, yaitu

menghubungkan *ESP32-CAM* ke jaringan, menjalankan aplikasi *desktop*, memantau hasil *streaming* citra panel surya, serta melihat hasil deteksi dan klasifikasi kondisi panel surya. Sistem akan menampilkan hasil deteksi berupa *bounding box*, label kondisi panel surya, dan hasil prediksi model *YOLOv8*

2). Activity Diagram Sistem



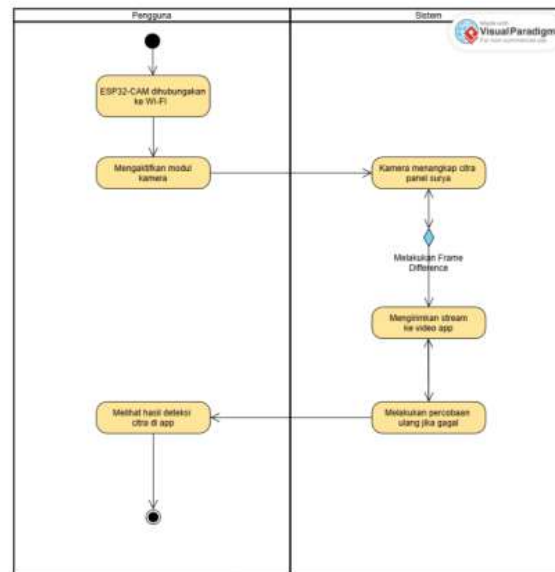
Gambar 3.3 Activity Diagram Sistem

Berdasarkan Gambar 3.3, *activity diagram* menjelaskan alur kerja sistem deteksi kondisi panel surya secara keseluruhan. Proses dimulai ketika sistem *ESP32-CAM* aktif maka kamera menangkap citra panel surya serta melakukan pemrosesan lokal menggunakan metode *Frame Difference* untuk menganalisis selisih intensitas piksel antar *frame* guna mendeteksi adanya pergerakan atau anomali visual. Apabila perubahan piksel tersebut melampaui ambang batas (*threshold*) yang ditentukan, maka Citra tersebut kemudian dikirimkan melalui jaringan *Wi-Fi* ke aplikasi *desktop* berbasis Python.

Selanjutnya, sistem melakukan pemrosesan citra menggunakan model *YOLOv8* untuk mendeteksi dan mengklasifikasikan kondisi

panel surya. Setelah proses deteksi selesai, hasil berupa label kondisi panel surya dan *bounding box* ditampilkan kepada pengguna melalui antarmuka aplikasi *desktop*.

3). Activity Diagram Streaming ESP32-CAM



Gambar 3.4 Activity Diagram ESP32-CAM

Activity diagram 3.4 menggambarkan proses kerja *ESP32-CAM* sebagai perangkat *input* citra sekaligus pemroses tingkat awal (*edge processing*). Proses dimulai ketika *ESP32-CAM* dihubungkan ke jaringan *Wi-Fi* dan mengaktifkan modul kamera. Kamera akan menangkap citra panel surya secara kontinu.

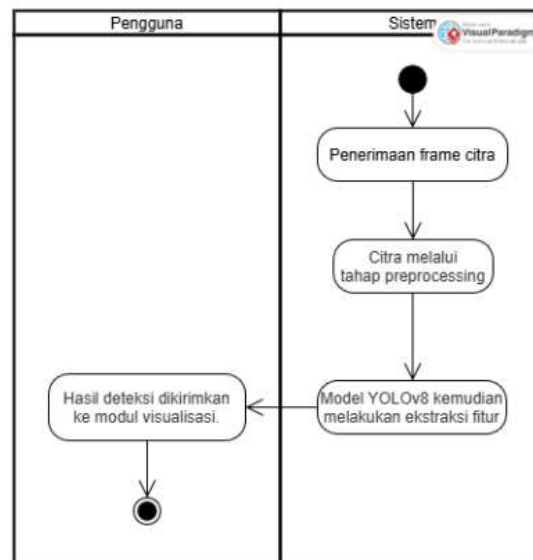
Berbeda dengan sistem *streaming* konvensional, modul ini menerapkan logika deteksi perubahan (*Frame Difference*). Sistem akan membandingkan citra saat ini dengan citra sebelumnya. Apabila terdeteksi perubahan visual yang signifikan, modul baru akan mengirimkan *stream* video ke server atau komputer lokal menggunakan protokol *HTTP*. Jika tidak ada perubahan, sistem kembali ke mode penangkapan citra (*standby*) untuk menghemat *bandwidth*. Apabila koneksi jaringan terputus saat pengiriman,

sistem akan melakukan percobaan koneksi ulang hingga *streaming* kembali berjalan dengan normal.

1). Activity Diagram YOLOv8

Activity diagram YOLOv8 menjelaskan proses deteksi objek yang dilakukan pada aplikasi *desktop*. Proses dimulai dengan penerimaan *frame* citra dari *ESP32-CAM*. Selanjutnya, citra akan melalui tahap *preprocessing* seperti *resize* dan normalisasi.

Model *YOLOv8* kemudian melakukan ekstraksi fitur menggunakan *backbone* C2f, penggabungan fitur pada *neck* PANet, serta proses prediksi pada *decoupled head* untuk menghasilkan *bounding box*, label kelas, dan nilai *confidence*. Hasil deteksi selanjutnya dikirimkan ke modul visualisasi.

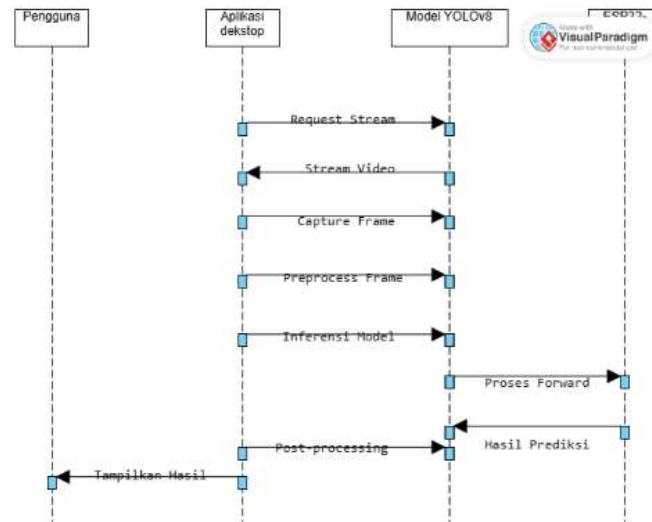


Gambar 3.5 Activity Diagram YOLOv8

2). Sequence Diagram (Diagram Sekuensial)

Sequence diagram menggambarkan interaksi antara komponen sistem dalam satu siklus pemrosesan *frame*. *Sequence diagram* menunjukkan alur komunikasi antara komponen sistem. Aplikasi *desktop* meminta *stream* dari *ESP32-CAM*, menangkap *frame*, memprosesnya, dan melakukan inferensi menggunakan

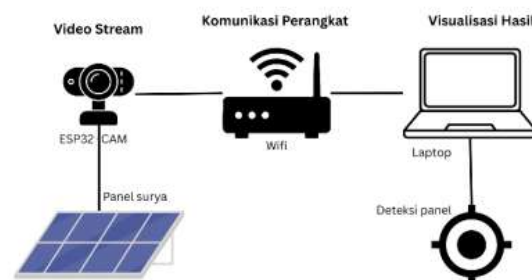
model *YOLOv8*. Hasilnya diproses dan ditampilkan kembali kepada pengguna.



Gambar 3.6 Sequence Diagram (Diagram Sekuensial)

3). Alur Sistem

Alur sistem ini dirancang untuk mendukung deteksi dan *monitoring* kondisi panel surya secara *real-time* dengan pendekatan arsitektur bertingkat (*tiered architecture*). Sistem terdiri dari dua komponen utama: *ESP32-CAM* sebagai perangkat *edge* yang melakukan deteksi awal, serta aplikasi *desktop* berbasis Python sebagai *unit* pemroses klasifikasi utama.



Gambar 3.7 Alur sistem

ESP32-CAM berfungsi sebagai *node* akuisisi yang mengambil gambar panel surya secara berkelanjutan. Modul ini melakukan pemrosesan lokal tingkat awal menggunakan algoritma

Frame Difference untuk mendeteksi perubahan visual. Berbeda dengan sistem *streaming* biasa, modul hanya akan mengirimkan *video stream* via *Wi-Fi (HTTP)* apabila terdeteksi adanya anomali atau objek pada permukaan panel surya. Mekanisme ini bertindak sebagai *filter* cerdas untuk mengurangi pengiriman data yang tidak perlu.

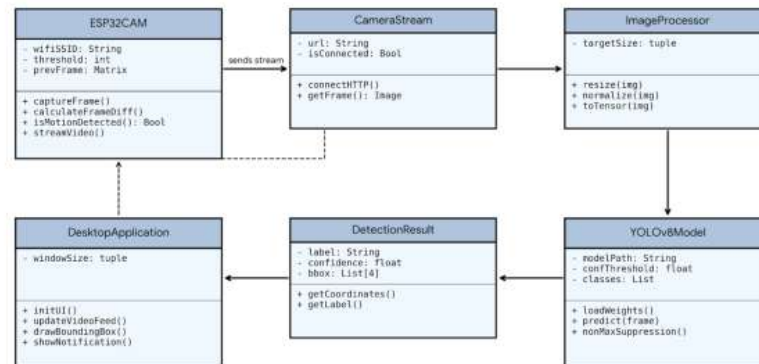
Aplikasi *desktop* menerima *stream* yang dipicu tersebut, melakukan pra-pemrosesan citra (*resize*, normalisasi, konversi format), dan memprosesnya menggunakan model *YOLOv8* yang telah dilatih. Model ini melakukan deteksi dan klasifikasi kondisi panel surya secara mendalam dalam satu tahap inferensi, menghasilkan *bounding box*, kelas kondisi (7 kelas), dan tingkat kepercayaan (*confidence score*).

Hasil deteksi langsung divisualisasikan pada *frame* video untuk pemantauan *real-time* oleh pengguna, sekaligus disimpan untuk evaluasi kinerja sistem melalui analisis *Confusion Matrix*. Arsitektur *hybrid* ini menyeimbangkan beban kerja dengan memisahkan deteksi gerak ringan di sisi *edge* dan komputasi berat (inferensi AI) di sisi *desktop*, sehingga menjaga efisiensi *bandwidth* dan stabilitas sistem secara keseluruhan.

b. Desain Database (Database Design)

1. Class Diagram (Diagram Kelas)

Class diagram pada penelitian ini menggambarkan struktur kelas dan hubungan antar komponen dalam sistem deteksi kondisi panel surya berbasis arsitektur *hybrid*. *Diagram* ini menunjukkan alur pemrosesan data citra yang terbagi menjadi dua tingkat: tingkat *edge* (*ESP32-CAM*) dan tingkat *server* (Laptop/PC).



Gambar 3.8 Class Diagram (Diagram Kelas)

Kelas ESP32CAM berperan sebagai perangkat cerdas pada sisi *edge*. Berbeda dengan sistem konvensional, kelas ini dilengkapi dengan atribut *motionThreshold()* dan metode *calculateFrameDiff()* untuk menjalankan algoritma *Frame Difference*. Kelas ini menangkap citra panel surya, membandingkannya dengan *frame* sebelumnya, dan hanya mengaktifkan metode *streamVideo()* untuk mengirim data via *Wi-Fi* jika terdeteksi perubahan visual yang signifikan.

Data citra yang dikirimkan kemudian diterima oleh kelas *CameraStream()* pada sisi server, yang bertugas mengelola protokol komunikasi *HTTP* dan memastikan penerimaan *frame* secara stabil. *Frame* citra tersebut selanjutnya diteruskan ke kelas *ImageProcessor* untuk melalui tahap *preprocessing*, meliputi penyesuaian ukuran (*resize*) dan normalisasi matriks piksel agar sesuai dengan *tensor input model*.

Citra yang telah siap diproses masuk ke kelas *YOLOv8Model*. Kelas ini memuat bobot model (*weights*) yang telah dilatih dan menjalankan metode *predict()* untuk melakukan deteksi objek. Hasil inferensi disimpan dalam kelas *DetectionResult*, yang memuat struktur data berupa label kondisi (7 kelas), nilai kepercayaan (*confidence*), dan koordinat *bounding box*. Terakhir, informasi ini dikelola oleh kelas *DesktopApplication()* untuk ditampilkan kepada

pengguna dalam antarmuka grafis (*GUI*) yang menyajikan visualisasi *real-time* dan status kondisi panel.

2. Desain Dataset Citra

Dataset citra yang digunakan dalam penelitian ini berjumlah 3.962 gambar yang diperoleh dari platform Roboflow. *Dataset* terdiri dari beberapa kelas kondisi panel surya, seperti kotoran burung, cacat, debu, daun, tidak cacat, kerusakan fisik, dan salju. *Dataset* disimpan pada penyimpanan lokal komputer untuk mempercepat proses *training* dan inferensi. Adapun deskripsi masing-masing kelas adalah sebagai berikut.

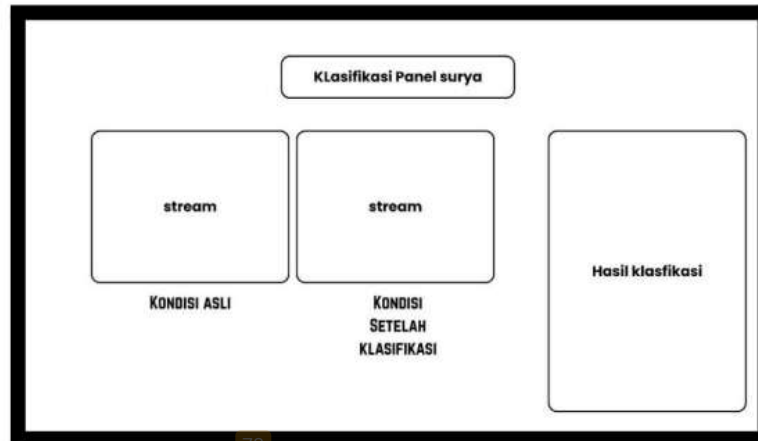
Tabel 3.5 Dataset Panel Surya

No	Citra Panel	Deskripsi
1		Retak atau pecah pada panel surya yang mengganggu fungsi.
2		Penumpukan partikel debu pada permukaan panel surya.

No	Citra Panel	Deskripsi
3		Kondisi Panel surya yang tidak sempurna
4		Panel surya tertutup daun yang menyebabkan <i>shading</i> .
5		Panel surya dalam kondisi normal dan bersih.
6		Panel surya tertutup kotoran burung yang menghambat penyerapan cahaya.
7		Panel surya tertutup salju sehingga cahaya terhalang.

No	Citra Panel	Deskripsi

c. Desain Antarmuka Pengguna (*UI/UX Design*)



Gambar 3.9 Desain Antarmuka Pengguna

Desain antarmuka pengguna pada sistem ini menampilkan halaman utama untuk klasifikasi kondisi panel surya secara *real-time*. Pada bagian kiri ditampilkan kondisi asli panel surya berupa *streaming* video langsung dari *ESP32-CAM*, sehingga pengguna dapat melihat keadaan panel sebelum diproses. Bagian tengah menampilkan kondisi panel surya setelah dilakukan klasifikasi oleh model *YOLOv8*, yang dilengkapi dengan penanda objek dan label hasil deteksi. Sementara itu, bagian kanan menampilkan hasil klasifikasi yang berisi informasi kondisi panel surya berdasarkan hasil pendeteksian. Antarmuka dirancang sederhana dan terstruktur untuk memudahkan pengguna dalam memantau kondisi panel surya.

2. Simulasi Proses Penyelesaian Masalah

Pada bagian teknik analisis data dilakukan simulasi perhitungan manual algoritma yang digunakan meliputi :

a. Data input

Pengguna melakukan *input* citra panel surya yang telah ditentukan yaitu citra panel surya yang memiliki kondisi berbeda-beda yang diberi label setiap kondisi panel surya yang ada di *dataset* pada tabel sebagai berikut:

Tabel 3.6 Citra Panel Surya Label

No	Citra Panel	Label
1		<i>Physical Damage</i> (Kerusakan Fisik)
2	 shutterstock.com - 2664815875	<i>Dust</i> (Debu)
3		<i>Defective</i> (Cacat)
4		

No	Citra Panel	Label
		<i>Leaf</i> (Daun)
5		<i>Non Defective</i> (Tidak Cacat)
6		<i>Bird Drop</i> (Kotoran Burung)
7		<i>Snow</i> (Salju)

b. Proses Pengolahan Data

Proses pengolahan data pada sistem ini dilakukan secara bertingkat. Tahap pertama adalah perhitungan deteksi gerak (*Frame Difference*) pada *ESP32-CAM*, dan tahap kedua adalah perhitungan algoritma *YOLOv8* pada laptop jika deteksi gerak terpenuhi. Berikut adalah simulasi perhitungannya:

1). Perhitungan Deteksi Perubahan (*Frame Difference* di *ESP32-CAM*)

Sebelum data diproses oleh *YOLOv8*, sistem membandingkan intensitas piksel antara *frame* saat ini (I_t) dengan *frame* sebelumnya ($I_{t=1}$) untuk menentukan apakah ada objek baru.

Sampel Data:

- Frame* Sebelumnya ($I_{t=1}$): Kondisi panel normal (sedikit lebih gelap).
- Frame* Saat Ini (t): Kondisi panel dengan objek baru (lebih terang/berbeda).

Matriks *Frame* Sebelumnya ($I_{t=1}$):

[170 160 145 170 155 140 165 150 135]

Matriks *Frame* Saat Ini (I_t) / Data Input *YOLO*:

[180 165 150 175 160 145 170 155 140]

Rumus Selisih Mutlak:

$$D(x, y) = |I_t(x, y) - I_{t-1}(x, y)| \dots (3.1)$$

Perhitungan Selisih per Piksel:

Baris 1: $180-170=10$; $165-160=5$; $150-145=5$

Baris 2: $175-170=5$; $160-155=5$; $145-140=5$

Baris 3: $170-165=5$; $155-150=5$; $140-135=5$

Total Perubahan (Sum of Differences):

$$Total = 10 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 = 50$$

Keputusan (*Decision*): Misalkan nilai ambang batas (*Threshold*) ditentukan sebesar 30. Karena Total (50) > *Threshold* (30), maka status = "Ada Perubahan". Sistem *ESP32-CAM* mengirimkan citra ke Laptop untuk diproses *YOLOv8*.

2). Perhitungan *Preprocessing* (Normalisasi)

Pada penelitian ini menggunakan sampel citra panel surya berukuran 3×3 *pixel*. Berikut adalah nilai intensitas *pixel*:

	Col 1	Col 2	Col 3
--	-------	-------	-------

Row 1	180	165	150
Row 2	175	160	145
Row 3	170	155	140

Normalisasi dilakukan dengan formula:

$$Inorm = \frac{I_{original}}{255} \dots (3.2)$$

Contoh perhitungan :

$$Inorm = \frac{180}{255} = 0.706 \quad Inorm = \frac{165}{255} = 0.647 \quad Inorm = \frac{150}{255} = 0.588$$

Hasil normalisasi :

	Col 1	Col 2	Col 3
Row 1	0.706	0.647	0.588
Row 2	0.686	0.627	0.569
Row 3	0.667	0.608	0.549

3). Perhitungan Konvolusi (*Feature Extraction*)

Kernel 3×3 untuk deteksi *edge*:

	Col 1	Col 2	Col 3
Row 1	1	0	-1
Row 2	1	0	-1
Row 3	1	0	-1

Formula konvolusi :

$$S(i, j) = \sum_{\{m\}} \cdot \sum_{\{n\}} \cdot I(m, n) \cdot K(i - m, j - n) \dots (3.3)$$

Perhitungan :

$$\begin{aligned} S &= (0.706 \times 1) + (0.647 \times 0) + (0.588 \times -1) \\ &+ (0.686 \times 1) + (0.627 \times 0) + (0.569 \times -1) + (0.667 \times 1) \\ &+ (0.608 \times 0) + (0.549 \times -1) \quad S = 0.706 - 0.588 \\ &+ 0.686 - 0.569 + 0.667 - 0.549 = 0.353 \end{aligned}$$

Hasil *Feature Map*: 0.353

4). Perhitungan Aktivasi *SiLU*

Formula *SiLU* (*Sigmoid Linear Unit*):

$$f(x) = \frac{x}{1 + e^{(-x)}} \dots (3.4)$$

Perhitungan untuk $x = 0.353$:

$$f(0.353) = \frac{0.353}{1 + e^{(-0.353)}} = \frac{0.353}{1.703} = 0.207$$

Hasil setelah aktivasi : 0.207

5). Perhitungan Prediksi *Bounding Box* dan Klasifikasi

Output dari *Grid Cell* (20, 20) dengan *stride* 16:

Bounding Box: $tx = 0.3$, $ty = 0.4$, $tw = 0.15$, $th = 0.12$

Class Logits: [0.5, 0.8, 2.3, 0.3, 0.2, 0.6, 0.1]

a) Konversi *Bounding Box*:

$$x_{\{center\}} = (20 + \sigma(0.3)) \times 16 = (20 + 0.574) \times 16 = 329.2$$

$$y_{center} = (20 + 0.599) \times 16 = 329.6$$

$$width = e^{0.15} \times 16 = 18.6$$

$$height = e^{0.12} \times 16 = 18.0$$

Hasil: BBox [329.2, 329.6, 18.6, 18.0]

b) Klasifikasi dengan *Softmax*:

$$P(class_i) = \frac{e^{z_i}}{\sum_{j=1}^7 e^{z_j}} \dots (3.5)$$

Perhitungan:

$$\sum e^{z_j} = 1.649 + 2.226 + 9.974 + 1.350 + 1.221 + 1.822 + 1.105 = 19.347$$

Probabilitas kelas *Dust* (index 2):

$$p^{Dust} = \frac{e^{2.3}}{19.347} = \frac{9.974}{19.347} = 0.515$$

Hasil Prediksi: *Dust* dengan *confidence* 51.5%

c. Data Output

Proses pengolahan data diawali dengan penerapan algoritma

Frame Difference pada *ESP32-CAM* untuk menghitung selisih piksel

antar-frame. Berdasarkan simulasi, diperoleh nilai total perubahan sebesar 50 yang melampaui ambang batas (*threshold*), sehingga sistem memicu pengiriman citra ke server. Pada tahap selanjutnya di sisi server, citra melalui proses normalisasi piksel untuk menstandarisasi nilai intensitas (menjadi 0.706). Data tersebut kemudian diproses menggunakan operasi konvolusi dengan *kernel* 3×3 untuk ekstraksi fitur tepi (menghasilkan nilai 0.353), dilanjutkan dengan fungsi aktivasi *SiLU* untuk pengaktifan neuron (0.207). Tahap akhir melibatkan prediksi *bounding box* dan klasifikasi objek menggunakan fungsi *Softmax*, yang menghasilkan deteksi akhir berupa kelas *Dust* dengan tingkat kepercayaan (*confidence*) sebesar 51.5% pada koordinat [329.2, 329.6, 18.6, 18.0]

15
BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian

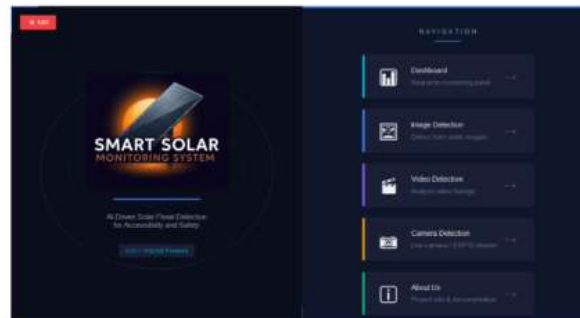
1. Implementasi Desain Sistem

a. Implementasi Lembar Kerja

Pada bagian ini akan dipaparkan lembar kerja untuk keperluan penelitian, sebagai berikut:

1) Lembar Kerja *Navigation*

Lembar kerja utama adalah halaman awal ketika pengguna membuka sistem. Pada lembar kerja terdapat nama sistem yang dibuat, memiliki lima menu navigasi utama yaitu: *Dashboard*, *Image Detection*, *Video Detection*, *Camera Detection* dan *About Us*. Pada Gambar 4.1 Lembar Kerja *Navigation*.



Gambar 4.1 Lembar Kerja *Navigation*.

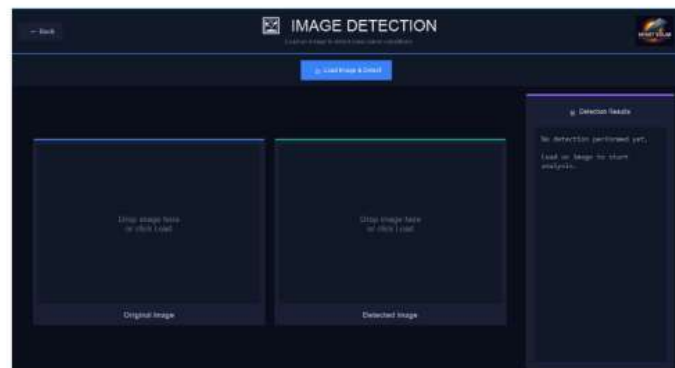
2) Lembar Kerja *Dashboard*

Lembar kerja ini berfungsi untuk menampilkan data sensor secara *real-time*. Dengan menampilkan empat parameter utama yaitu Tegangan, Arus, Daya, dan Kapasitas Baterai (*Battery Level*). Data ini ditampilkan untuk memudahkan pengguna dalam memantau kesehatan elektrik panel surya. menunjukkan bahwa koneksi antara sensor dan aplikasi berjalan dengan stabil.

Gambar 4.2 Lembar Kerja *Dashboard*

3) Lembar Kerja *Image Detection*

Halaman *image detection* digunakan untuk melakukan deteksi kondisi panel surya menggunakan gambar *input* dari pengguna. Pengguna dapat mengunggah gambar panel surya kemudian sistem akan melakukan proses deteksi menggunakan model *YOLOv8*.

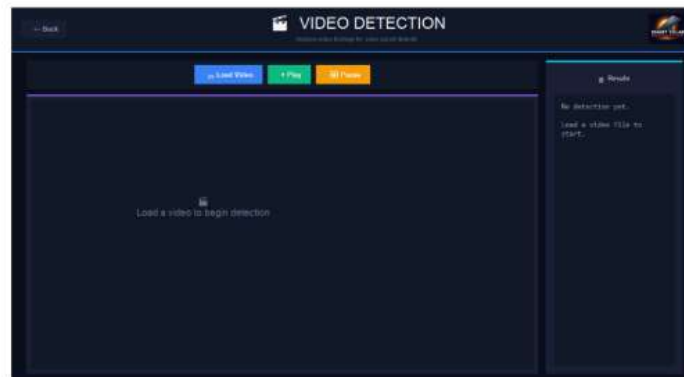


79

Gambar 4.3 Lembar Kerja *Image Detection*

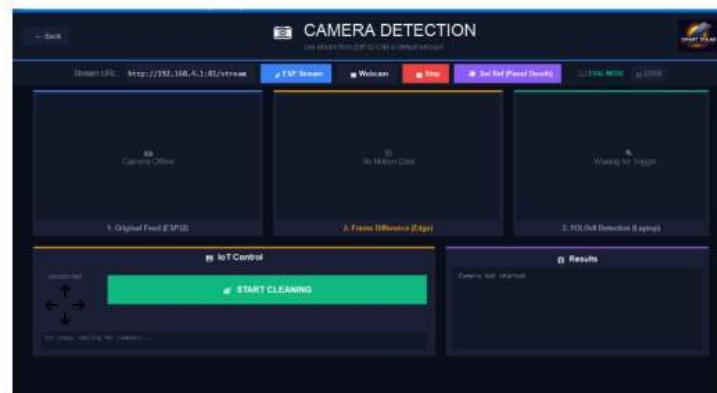
4) Lembar Kerja *Video Detection*

Lembar kerja *video detection* digunakan untuk melakukan deteksi kondisi panel surya menggunakan *file* video. Sistem akan memproses setiap *frame* video menggunakan model *YOLOv8* secara otomatis.

Gambar 4.4 Lembar Kerja *Video Detection*

5) Lembar Kerja *Camera Detection*

Halaman *camera detection* digunakan untuk melakukan deteksi kondisi panel surya secara langsung menggunakan *streaming* kamera *ESP32-CAM*. Sistem akan menerima video *stream* kemudian melakukan deteksi secara *real-time*.

Gambar 4.5 Lembar Kerja *Camera Detection*

b. Keterkaitan Antar Halaman Sistem

1) Halaman *Navigation*

Halaman *navigation* digunakan sebagai penghubung antar halaman pada aplikasi. Pengguna dapat berpindah menuju halaman *dashboard*, *image detection*, *video detection*, dan *camera detection* melalui menu navigasi yang tersedia.

Alur kerja:

- a) Pengguna membuka aplikasi.
- b) Sistem menampilkan menu navigasi utama.
- c) Pengguna memilih fitur yang ingin dijalankan.

2) Halaman *Dashboard*

Halaman *dashboard* digunakan sebagai pusat *monitoring* data sensor panel surya secara *real-time*. Sistem menampilkan empat parameter yaitu tegangan, arus, daya, dan kapasitas baterai. Informasi tersebut digunakan untuk memudahkan pengguna dalam memantau kondisi elektrik panel surya secara langsung.

Alur kerja:

- a) Sensor membaca data elektrik panel surya.
- b) Data dikirim menuju sistem melalui *ESP32*.
- c) Sistem menerima data sensor secara *realtime*.
- d) *Dashboard* menampilkan nilai tegangan, arus, daya, dan *battery level*.
- e) Pengguna dapat melakukan *monitoring* kondisi panel surya secara langsung melalui aplikasi.

3) Halaman *Image Detection*

Halaman *image detection* digunakan untuk melakukan deteksi kondisi panel surya menggunakan *input* gambar.

Alur kerja:

- a) Pengguna mengunggah gambar panel surya.
- b) Sistem melakukan *preprocessing* citra.
- c) Model *YOLOv8* melakukan deteksi objek.
- d) Hasil klasifikasi ditampilkan berupa *bounding box*, label, dan *confidence score*.

4) Halaman *Video Detection*

Halaman *video detection* digunakan untuk melakukan deteksi kondisi panel surya menggunakan *file* video.

Alur kerja:

- a) Pengguna memilih *file* video.
 - b) Sistem membaca *frame* video secara berurutan.
 - c) *YOLOv8* melakukan inferensi pada setiap *frame*.
 - d) Hasil deteksi ditampilkan secara *realtime* pada video.
- 5) Halaman *Camera Detection*

Halaman *camera detection* merupakan halaman utama *monitoring realtime* berbasis *ESP32-CAM*. Pada halaman ini proses *streaming*, *frame difference*, dan deteksi *YOLOv8* dilakukan dalam satu halaman terintegrasi.

Alur kerja:

- a) *ESP32-CAM* menangkap citra panel surya secara *realtime*.
- b) Sistem melakukan proses *Frame Difference* untuk mendeteksi perubahan visual.
- c) Jika terdeteksi perubahan, *frame* dikirim ke laptop melalui *streaming HTTP*.
- d) Laptop menerima *frame* dan melakukan *preprocessing* citra.
- e) Model *YOLOv8* melakukan deteksi kondisi panel surya.
- f) Hasil deteksi ditampilkan dalam bentuk *bounding box*, label kelas, dan *confidence score*.
- g) Sistem menampilkan hasil *monitoring* pada panel hasil deteksi dan *results* secara *realtime*.

57

2. Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan seluruh fitur pada sistem deteksi kondisi panel surya berbasis *ESP32-CAM* dan *YOLOv8* dapat berjalan sesuai dengan fungsi yang telah dirancang. Pengujian dilakukan pada setiap halaman sistem meliputi *navigation*, *dashboard*, *image detection*, *video detection*, dan *camera detection*

69

62

a. Pengujian *White Box*

Pengujian *white box* dilakukan dengan menguji fungsi-fungsi utama sistem berdasarkan alur logika program dan proses yang berjalan pada aplikasi.

1) Pengujian Fungsi *Streaming ESP32-CAM*

Fungsi *streaming* digunakan untuk menerima video *stream* dari *ESP32-CAM* maupun *webcam* lokal. Sistem menggunakan library *OpenCV* melalui fungsi *cv2.VideoCapture()* untuk membuka koneksi *streaming*.

Tabel 4.1 Pengujian Fungsi *Streaming*

Aspek	Keterangan
Nama Fungsi	<i>open_camera()</i> dan <i>cv2.VideoCapture()</i>
Tujuan	Membuka <i>streaming ESP32-CAM</i>
Input	URL <i>stream ESP32</i>
Proses	Membaca <i>stream</i> menggunakan <i>OpenCV</i>
Output	Video <i>realtime</i> tampil pada <i>canvas</i>
Hasil yang Diharapkan	<i>Streaming</i> berjalan stabil

Berdasarkan hasil pengujian, sistem berhasil menerima video *streaming* dari *ESP32-CAM* secara *realtime* dan menampilkannya pada halaman *camera detection*.

2) Pengujian Fungsi *Frame Difference*

Fungsi *Frame Difference* digunakan sebagai simulasi *edge processing* untuk mendeteksi perubahan visual sebelum dilakukan inferensi *YOLOv8*.

Tabel 4.2 Pengujian Fungsi *Frame Difference*

Aspek	Keterangan
Nama Fungsi	<code>cv2.absdiff()</code> dan <code>cv2.threshold()</code>
Tujuan	Mendeteksi perubahan visual antar <i>frame</i>
<i>Input</i>	Dua <i>frame</i> citra <i>grayscale</i>
Proses	Menghitung selisih piksel dan melakukan <i>thresholding</i>
<i>Output</i>	Area perubahan visual pada <i>frame</i>
Hasil yang Diharapkan	Perubahan objek berhasil terdeteksi

Berdasarkan hasil pengujian, metode *Frame Difference* mampu mendeteksi perubahan visual pada panel surya sebelum dilakukan proses deteksi menggunakan *YOLOv8*.

3) Pengujian Fungsi *YOLOv8 Detection*

Fungsi *YOLOv8* digunakan untuk mendeteksi kondisi panel surya secara *realtime* ketika sistem mendeteksi adanya perubahan visual.

Tabel 4.3 Pengujian Fungsi YOLOv8

Aspek	Keterangan
Nama Fungsi	<i>self.model.predict()</i>
Tujuan	Melakukan deteksi kondisi panel surya menggunakan YOLOv8
Input	Frame hasil streaming kamera
Proses	Inferensi model YOLOv8 klasifikasi objek
Output	<i>Bounding box</i> , label kelas, <i>confidence score</i>
Hasil yang Diharapkan	Objek panel surya berhasil dideteksi sesuai kondisi sebenarnya

Hasil pengujian menunjukkan bahwa model YOLOv8 mampu mendeteksi beberapa kondisi

panel surya seperti *Bird Drop*, *Defective*, *Dust*, *Non Defective*, *Physical Damage*, *Snow*, dan *Leaf* secara *realtime*.

4) Pengujian Fungsi *Monitoring Realtime*

Fungsi *monitoring realtime* digunakan untuk menampilkan hasil deteksi secara langsung pada antarmuka sistem.

Tabel 4.4 Pengujian Fungsi *Monitoring*

Aspek	Keterangan
Nama Fungsi	<code>update_camera_frame()</code>
Tujuan	Memperbarui tampilan hasil deteksi secara <i>realtime</i>
Input	<i>Frame</i> hasil inferensi <i>YOLOv8</i>
Proses	Menampilkan <i>frame</i> pada <i>canvas</i> GUI Tkinter
Output	Tampilan <i>realtime</i> hasil <i>monitoring</i>
Hasil yang Diharapkan	Tampilan sistem berjalan <i>realtime</i> tanpa gangguan

Berdasarkan hasil pengujian, sistem mampu menampilkan hasil deteksi secara *realtime* pada halaman *camera detection* dengan *delay* yang relatif kecil.

6
b. Pengujian *Black Box*

1) Tabel Pengujian *Black Box* – *Dashboard Monitoring*

Pada Tabel 4.4 *Black Box* – *Dashboard Monitoring* dilakukan pengujian untuk mengetahui apakah sistem dapat menampilkan data sensor panel surya secara *realtime*. Pengujian dilakukan pada parameter tegangan, arus, daya, dan kapasitas baterai. Sebagai berikut:

Tabel 4.4 *Black Box* – *Dashboard Monitoring*

No	Skenario Uji	Input	Ekspektasi Output	Output Sistem	Status
1	Menampilkan tegangan	Data sensor <i>voltage</i>	Nilai tegangan tampil <i>realtime</i>	Tegangan berhasil tampil	Sesuai

No	Skenario Uji	Input	Ekspektasi Output	Output Sistem	Status
2	Menampilkan arus	Data sensor <i>current</i>	Nilai arus tampil <i>realtime</i>	Arus berhasil tampil	Sesuai
3	Menampilkan daya	Data sensor power	Nilai daya tampil <i>realtime</i>	Daya berhasil tampil	Sesuai
4	Menampilkan <i>battery level</i>	Data sensor baterai	Nilai kapasitas baterai tampil <i>realtime</i>	<i>Battery level</i> berhasil tampil	Sesuai

2) Tabel Pengujian *Black Box* – Image Detection

Pada Tabel 4.5 *Black Box* – Image Detection dilakukan pengujian untuk mengetahui apakah sistem mampu melakukan deteksi kondisi panel surya menggunakan gambar *input*. Pengujian dilakukan pada proses pembacaan gambar, deteksi objek, dan hasil inferensi *YOLOv8*. Sebagai berikut:

Tabel 4.5 *Black Box* – Image Detection

No	Skenario Uji	Input	Ekspektasi Output	Output Sistem	Status
1	Mengunggah gambar panel surya	<i>File</i> gambar panel surya	Sistem membaca gambar	Gambar berhasil tampil	Sesuai
2	Deteksi objek pada gambar	Gambar panel surya	<i>Bounding box</i> dan label tampil	Objek berhasil terdeteksi	Sesuai
3	Menampilkan <i>confidence score</i>	Hasil inferensi <i>YOLOv8</i>	<i>Confidence score</i> tampil	<i>Confidence score</i> berhasil tampil	Sesuai
4	Mengunggah gambar tidak jelas	Gambar blur atau <i>noise</i>	Sistem tetap memproses gambar	Deteksi tetap berjalan	Sesuai

3) Tabel Pengujian *Black Box* – Video Detection

Pada Tabel 4.6 *Black Box – Video Detection* dilakukan pengujian untuk mengetahui apakah sistem dapat melakukan deteksi kondisi panel surya menggunakan *file video*. Pengujian dilakukan pada proses pembacaan video, inferensi *frame*, dan tampilan hasil deteksi *realtime*. Sebagai berikut:

Tabel 4.6 *Black Box – Video Detection*

No	Skenario Uji	Input	Ekspektasi Output	Output Sistem	Status
1	Mengunggah <i>file video</i>	<i>File video</i> panel surya	Video berhasil diproses	Video berhasil tampil	Sesuai
2	Deteksi objek pada video	<i>Frame video</i>	<i>Bounding box</i> dan label tampil	Objek berhasil terdeteksi	Sesuai
3	Menampilkan hasil <i>realtime</i>	Hasil inferensi video	Hasil deteksi tampil <i>realtime</i>	Deteksi berjalan <i>realtime</i>	Sesuai
4	Mengunggah video resolusi rendah	Video kualitas rendah	Sistem tetap memproses video	Video tetap diproses	Sesuai

4) Tabel Pengujian *Black Box – Camera Detection*

Pada Tabel 4.7 *Black Box – Camera Detection* dilakukan pengujian untuk mengetahui apakah sistem *monitoring realtime* berbasis *ESP32-CAM* dan *YOLOv8* dapat berjalan dengan baik pada berbagai kondisi pengujian. Pengujian dilakukan pada proses *streaming*, *frame difference*, dan deteksi objek panel surya. Sebagai berikut:

Tabel 4.7 *Black Box – Camera Detection*

No	Skenario Uji	Input	Ekspektasi Output	Output Sistem	Status
1	Membuka <i>streaming ESP32-CAM</i>	URL <i>stream ESP32</i>	Video <i>streaming</i> tampil <i>realtime</i>	<i>Streaming</i> berhasil tampil	Sesuai
2	Mendeteksi perubahan <i>frame</i>	Gerakan pada <i>frame</i>	Area perubahan terdeteksi	<i>Frame difference</i> berhasil tampil	Sesuai
3	Mendeteksi objek panel surya	<i>Frame</i> panel surya	<i>Bounding box</i> dan label objek tampil	Objek berhasil terdeteksi	Sesuai
4	Menjalankan <i>webcam</i> lokal	<i>Webcam</i> aktif	Tampilan <i>webcam</i> muncul	<i>Webcam</i> berhasil tampil	Sesuai
5	Menjalankan tombol stop	Klik tombol stop	<i>Streaming</i> berhenti	Kamera berhasil berhenti	Sesuai
6	Menampilkan hasil <i>realtime</i>	Hasil inferensi <i>YOLOv8</i>	Hasil <i>monitoring</i> tampil <i>realtime</i>	<i>Monitoring</i> berjalan <i>realtime</i>	Sesuai

11 3. Pengujian Non fungsional

Pengujian non-fungsional dilakukan untuk mengetahui performa sistem dalam menjalankan proses deteksi objek pada berbagai mode operasi, meliputi deteksi gambar statis, deteksi video, dan deteksi *real-time* menggunakan *ESP32-CAM*. Aspek yang diuji meliputi waktu respons sistem, stabilitas proses deteksi, serta kemampuan sistem dalam menampilkan hasil deteksi secara berkelanjutan.

Tabel 4.8 Pengujian Waktu Respon Sistem

No	Fitur	Parameter Pengujian	Hasil
1	Image Detection	Waktu pemrosesan gambar hingga hasil tampil	0.48 detik

2	<i>Video Detection</i>	Waktu awal pemrosesan video	0,78 detik
3	<i>Camera Detection</i>	Waktu penerimaan <i>stream ESP32-CAM</i>	0,26 detik
4	<i>YOLOv8 Detection</i>	Waktu inferensi rata-rata per <i>frame</i>	85,4 ms

Tabel 4.9 Pengujian Stabilitas Sistem

No	Skenario	Jumlah Pengujian	Berhasil	Keterangan
1	Deteksi gambar statis	20 kali	20	Sangat stabil untuk pemrosesan <i>file</i> tunggal.
2	Deteksi video	20 kali		
3	<i>Streaming ESP32-CAM</i>	20 kali	17	3 kali koneksi terputus (<i>timeout</i>) akibat panas pada modul <i>hardware ESP32</i> .
4	Inferensi <i>YOLOv8 real-time</i>	20 kali	15	2 kali terjadi kegagalan <i>render</i> GUI Tkinter akibat beban komputasi CPU 100% dan juga terputusnya koneksi ke <i>ESP32- CAM</i> .

Tabel 4.10 Pengujian Waktu Respon Sistem

No	Kelas	Jumlah Pengujian	Hasil Konsisten	Keterangan
1	<i>Physical Damage</i>	10	Ya (7/10)	3 kali salah klasifikasi sebagai <i>Defective</i> karena pola keretakan menyerupai cacat sel dalam.
2	<i>Dust</i>	10	Ya (5/10)	Debu tipis sangat sulit dideteksi dan sering dianggap sebagai panel bersih (<i>Non Defective</i>).
3	<i>Defective</i>	10	Ya (6/10)	4 kali gagal terdeteksi jika ukuran cacat sel terlalu kecil atau tertutup bayangan samar.
4	<i>Leaf</i>	10	Ya (8/10)	Bentuk daun sangat unik, namun 2 kali terlewat jika tertimbun kotoran lain.
5	<i>Non Defective</i>	10	Ya (8/10)	2kali dideteksi sebagai <i>Dust</i> karena pantulan cahaya matahari yang membuat panel terlihat buram.
6	<i>Bird Drop</i>	10	Ya (7/10)	3 kali gagal dideteksi jika kotoran burung berwarna putih abu-abu menyerupai debu tebal.
7	<i>Snow</i>	10	Ya (9/10)	Karakter warna putih tebal sangat khas, hanya 1 kali terdeteksi sebagai <i>Bird Drop</i> karena pola salju yang membulat kecil.

Berdasarkan hasil pengujian non-fungsional, sistem mampu menjalankan fitur deteksi gambar, video, dan *real-time* dengan waktu respons yang cepat serta performa yang cukup stabil. Sebagian besar pengujian berhasil dijalankan sesuai dengan fungsi yang dirancang, meskipun masih terdapat beberapa kendala pada proses *streaming ESP32-CAM* dan deteksi *real-time* akibat keterbatasan perangkat keras dan koneksi jaringan.

Selain itu, sistem menunjukkan tingkat konsistensi yang baik dalam mengklasifikasikan berbagai kondisi panel surya. Meskipun terdapat beberapa kesalahan klasifikasi pada kelas yang memiliki karakteristik visual serupa, secara keseluruhan sistem telah mampu mendukung proses *monitoring* dan deteksi kondisi panel surya secara otomatis dengan baik.

4. Alat Ukur

Alat ukur evaluasi yang digunakan pada penelitian ini yaitu *Confusion Matrix*. *Confusion Matrix* merupakan sebuah tabel yang digunakan untuk mengukur kinerja model deteksi objek dengan membandingkan hasil prediksi model terhadap label sebenarnya pada *dataset* pengujian.

Melalui *Confusion Matrix* dapat diketahui jumlah data yang berhasil dideteksi dengan benar (*True Positive*), objek yang salah terdeteksi (*False Positive*), serta objek yang gagal dideteksi (*False Negative*). Selain itu, *Confusion Matrix* juga digunakan sebagai dasar perhitungan nilai *Precision*, *Recall*, *F1-Score*, dan *Mean Average Precision (mAP)* untuk mengevaluasi performa model secara menyeluruh.

Pada penelitian ini evaluasi dilakukan terhadap tujuh kelas kondisi panel surya yaitu *Bird Drop*, *Defective*, *Dust*, *Non Defective*, *Physical Damage*, *Snow*, dan *Leaf*.



46

Gambar 4.6 Confusion Matrix

Pada Gambar 4.6 ditampilkan hasil *Confusion Matrix* model YOLOv8 terhadap *dataset* pengujian. Sebagian besar nilai berada pada diagonal utama yang menunjukkan bahwa model mampu mengenali objek sesuai kelasnya. Namun masih terdapat beberapa kesalahan klasifikasi antar kelas yang memiliki karakteristik visual serupa.

Berdasarkan hasil evaluasi menggunakan *Confusion Matrix* diperoleh nilai *True Positive* (TP) sebesar 5107. Nilai *False Positive* (FP) sebesar 1908, sedangkan nilai *False Negative* (FN) sebesar 2331. Nilai-nilai tersebut selanjutnya digunakan untuk menghitung metrik evaluasi berupa *Precision*, *Recall*, *F1-Score*, dan *Mean Average Precision* (mAP).

a. Precision

Precision merupakan perbandingan antara jumlah objek yang berhasil dideteksi dengan benar terhadap seluruh objek yang diprediksi oleh model.

$$Precision = \frac{TP}{TP+FP} Precision = \frac{5107}{5107+1908} Precision = \frac{5107}{7015} Precision = 0.7280 Precision =$$

b. Recall

Recall merupakan perbandingan antara jumlah objek yang berhasil dideteksi dengan benar terhadap seluruh objek yang sebenarnya terdapat pada *dataset*.

$$Recall = \frac{TP}{TP+FN} \quad Recall = \frac{5107}{5107+2331} \quad Recall = \frac{5107}{7438} \quad Recall = 0.6866 \quad Recall = 68.66\%$$

c. *F1-Score*

F1-Score merupakan rata-rata harmonis antara *Precision* dan *Recall*.

$$F1 = \frac{2TP}{2TP+FP+FN} \quad F1 = \frac{2(5107)}{2(5107)+1908+2331} \quad F1 = \frac{10214}{14453} \quad F1 = 0.7067 \quad F1 = 70.67\%$$

4) *Mean Average Precision (mAP@50)*

Mean Average Precision (mAP@50) merupakan rata-rata nilai *Average Precision (AP)* dari seluruh kelas pada ambang batas *Intersection over Union (IoU)* sebesar 0,50.

$$mAP@50 = \frac{\sum_{i=1}^N AP_i}{N} \quad mAP@50 = 68,13\%$$

$$mAP@50 - 95 = 44,71\%$$

Tabel 4.11 Rangkuman *Matrix* Evaluasi

<i>Matrix</i>	Rumus	Nilai
<i>Precision</i>	$TP / (TP + FP)$	72.80%
<i>Recall</i>	$TP / (TP + FN)$	68.66%
<i>F1-Score</i>	$2TP / (2TP + FP + FN)$	70.67%
<i>mAP@50</i>		68.13%

	Rata-rata AP seluruh kelas pada IoU 0,50	
--	---	--

5 Berdasarkan hasil evaluasi, model YOLOv8 memperoleh nilai Precision sebesar 72,80%, Recall sebesar 68,66%, F1-Score sebesar 70,67%, mAP@50 sebesar 68,13%, dan mAP@50-95 sebesar 44,71%. Hasil tersebut menunjukkan bahwa model mampu mendeteksi tujuh kondisi panel surya dengan performa yang cukup baik. Nilai mAP@50 yang mendekati 70% menunjukkan bahwa model telah mampu mengenali objek serta menentukan lokasi objek secara akurat pada sebagian besar citra pengujian.

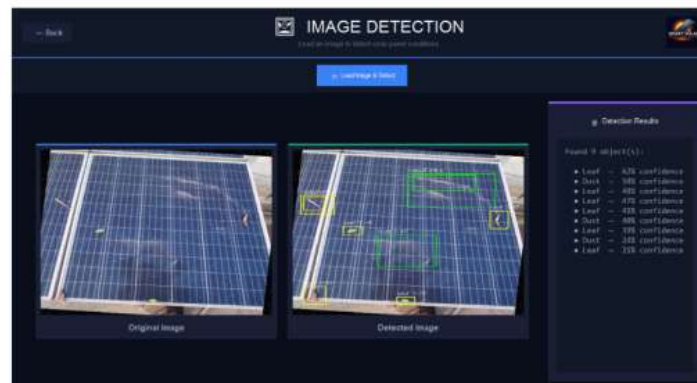
10 Nilai Precision yang lebih tinggi dibandingkan Recall menunjukkan bahwa model relatif jarang menghasilkan deteksi yang salah, namun masih terdapat beberapa objek yang gagal terdeteksi. Hal ini terlihat dari nilai False Negative yang masih cukup tinggi pada beberapa kelas seperti Leaf, Dust, dan Physical Damage.

Secara keseluruhan, model mampu mengenali tujuh kondisi panel surya dengan performa yang cukup baik sehingga dapat digunakan sebagai pendukung sistem monitoring kondisi panel surya berbasis Computer Vision dan YOLOv8.

5. Hasil Implementasi Deteksi Objek

Pengujian implementasi dilakukan untuk mengetahui kemampuan sistem dalam mendeteksi berbagai kondisi panel surya menggunakan model YOLOv8. Hasil deteksi ditampilkan dalam bentuk bounding box, label kelas, dan nilai confidence pada objek yang teridentifikasi.

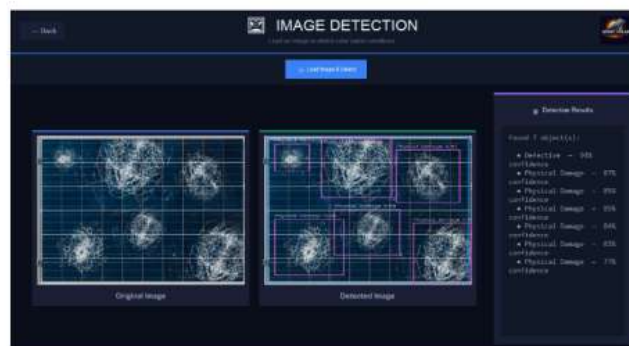
a. Deteksi Objek Dust dan Leaf



Gambar 4.7 Hasil Deteksi Objek *Dust* dan *Leaf*

Berdasarkan Gambar 4.7, sistem berhasil mendeteksi objek *Dust* dan *Leaf* yang berada pada permukaan panel surya. Model *YOLOv8* mampu memberikan label kelas yang sesuai serta menampilkan *bounding box* pada lokasi objek yang terdeteksi. Hasil tersebut menunjukkan bahwa model dapat mengenali objek pengotor panel surya yang berpotensi menurunkan efisiensi penyerapan cahaya matahari.

b. Deteksi Objek *Physical Damage*



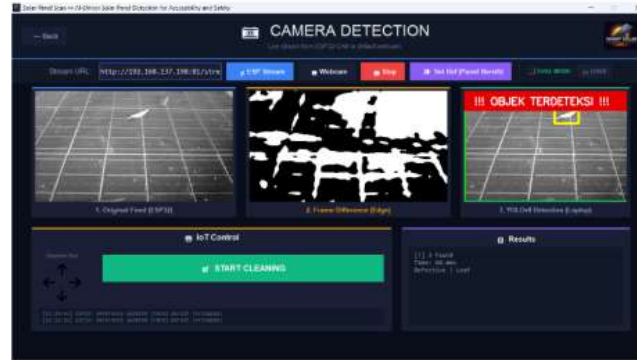
72

Gambar 4.8 Hasil Deteksi *Physical Damage*

Berdasarkan Gambar 4.8, sistem mampu mengidentifikasi kerusakan fisik pada permukaan panel surya. Objek yang terdeteksi diberikan label *Physical Damage* beserta nilai *confidence* sesuai hasil prediksi model. Kemampuan ini menunjukkan bahwa model tidak hanya mampu mendeteksi

objek pengotor, tetapi juga dapat mengenali kerusakan yang berpotensi memengaruhi performa panel surya.

c. Hasil Deteksi *Real-Time* Menggunakan *ESP32-CAM*



Gambar 4.9 *Monitoring Real-Time* Menggunakan *ESP32-CAM*

Pada pengujian *real-time*, *ESP32-CAM* berhasil mengirimkan *streaming* video ke aplikasi *desktop* melalui jaringan lokal. Selanjutnya model *YOLOv8* melakukan proses inferensi secara langsung pada *frame* yang diterima dan menampilkan hasil deteksi dalam bentuk *bounding box* serta label kelas objek. Hasil pengujian menunjukkan bahwa integrasi *ESP32-CAM* dan *YOLOv8* mampu mendukung proses *monitoring* kondisi panel surya secara *real-time*. langsung dan menampilkan hasil identifikasi objek pada *frame* yang diterima.

B. Pembahasan

1. Pembahasan Perancangan dan Integrasi Sistem

Penelitian ini berhasil merancang dan mengintegrasikan sistem deteksi kondisi panel surya berbasis *ESP32-CAM* dan model *YOLOv8*. Sistem dibangun menggunakan dua komponen utama yaitu modul *ESP32-CAM* sebagai perangkat akuisisi citra dan aplikasi *desktop* sebagai pusat pemrosesan data menggunakan model *YOLOv8*.

ESP32-CAM berfungsi untuk menangkap kondisi permukaan panel surya dan mengirimkan data citra melalui jaringan *WiFi* menggunakan protokol *HTTP streaming*. Data yang diterima kemudian diproses oleh

aplikasi *desktop* yang telah terintegrasi dengan model *YOLOv8* untuk melakukan deteksi objek secara otomatis.

Sistem yang dikembangkan memiliki tiga mode deteksi yaitu *Image Detection*, *Video Detection*, dan *Camera Detection*. Pada mode *Image Detection*, pengguna dapat melakukan deteksi objek menggunakan gambar statis. Pada mode *Video Detection*, sistem melakukan deteksi pada setiap *frame* video yang diunggah pengguna. Sedangkan pada mode *Camera Detection*, sistem menerima *streaming* secara langsung dari *ESP32-CAM* dan melakukan deteksi secara *real-time*.

Berdasarkan hasil pengujian fungsional, seluruh fitur utama sistem dapat berjalan sesuai dengan kebutuhan pengguna. Sistem mampu menerima data citra, melakukan proses deteksi, serta menampilkan hasil identifikasi objek dalam bentuk *bounding box*, label kelas, dan nilai *confidence*. Hal ini menunjukkan bahwa proses integrasi antara perangkat *ESP32-CAM* dan aplikasi berbasis *YOLOv8* telah berhasil dilakukan dengan baik.

2. Pembahasan Kinerja Model *YOLOv8*

Evaluasi kinerja model dilakukan menggunakan *confusion matrix* untuk mengetahui kemampuan model dalam mendeteksi tujuh kelas kondisi panel surya yang terdiri dari *Bird Drop*, *Defective*, *Dust*, *Non Defective*, *Physical Damage*, *Snow*, dan *Leaf*.

Berdasarkan hasil pengujian diperoleh nilai *True Positive* (TP) sebanyak 5107, *False Positive* (FP) sebanyak 1908, dan *False Negative* (FN) sebanyak 2331. Dari hasil tersebut diperoleh nilai *Precision* sebesar 72,80%, *Recall* sebesar 68,66%, dan *F1-Score* sebesar 70,67%.

Nilai *precision* menunjukkan bahwa sebagian besar objek yang diprediksi oleh model merupakan objek yang benar. Nilai *recall* menunjukkan kemampuan model dalam menemukan objek yang terdapat pada data uji. Sedangkan nilai *F1-Score* menunjukkan

keseimbangan antara *precision* dan *recall* sehingga dapat digunakan untuk menggambarkan performa model secara keseluruhan.

Selain itu, hasil pelatihan model menunjukkan nilai *Precision* sebesar 73,06%, *Recall* sebesar 64,46%, *mAP50* sebesar 68,13%, dan *mAP50-95* sebesar 44,71%. Nilai *mAP50* menunjukkan bahwa model mampu melakukan deteksi objek dengan cukup baik pada tingkat *Intersection over Union* (IoU) 50%, sedangkan nilai *mAP50-95* menunjukkan performa model pada berbagai tingkat IoU yang lebih ketat.

Berdasarkan hasil *confusion matrix*, kelas *Non Defective* dan *Leaf* memiliki performa deteksi yang relatif lebih baik dibandingkan beberapa kelas lainnya. Sebaliknya, kelas *Bird Drop* dan *Physical Damage* masih menghasilkan kesalahan deteksi yang cukup tinggi. Hal ini disebabkan karena karakteristik visual beberapa objek memiliki kemiripan bentuk maupun warna sehingga mempengaruhi kemampuan model dalam melakukan klasifikasi secara tepat.

Secara keseluruhan, hasil evaluasi menunjukkan bahwa model *YOLOv8* mampu melakukan deteksi kondisi panel surya dengan performa yang cukup baik pada skenario multikelas.

3. Keunggulan dan Keterbatasan Sistem

Berdasarkan hasil pengujian yang telah dilakukan, sistem yang dikembangkan memiliki beberapa keunggulan. Sistem mampu melakukan deteksi kondisi panel surya secara otomatis tanpa memerlukan inspeksi manual. Integrasi dengan *ESP32-CAM* memungkinkan proses *monitoring* dilakukan secara *real-time* sehingga pengguna dapat mengetahui kondisi panel surya secara langsung. Selain itu, sistem dapat digunakan pada berbagai sumber *input* seperti gambar, video, dan *streaming* kamera sehingga memberikan fleksibilitas penggunaan.

Meskipun demikian, sistem masih memiliki beberapa keterbatasan. Beberapa kelas objek masih menghasilkan tingkat kesalahan deteksi

yang cukup tinggi, terutama pada objek yang memiliki karakteristik visual yang mirip. Selain itu, kestabilan *streaming* dari *ESP32-CAM* masih dipengaruhi oleh kualitas jaringan dan kemampuan perangkat keras yang digunakan. Pada kondisi tertentu juga ditemukan keterlambatan proses deteksi akibat beban komputasi yang tinggi saat melakukan inferensi secara *real-time*.

Keterbatasan tersebut menunjukkan bahwa masih ⁵ diperlukan pengembangan lebih lanjut baik pada sisi *dataset*, optimasi model, maupun perangkat keras yang digunakan agar performa sistem dapat ditingkatkan.

4. Kesimpulan Pembahasan

Berdasarkan hasil pembahasan dapat diketahui bahwa sistem deteksi kondisi panel surya berbasis *ESP32-CAM* dan *YOLOv8* telah berhasil dirancang dan diimplementasikan sesuai dengan tujuan penelitian. Integrasi antara *ESP32-CAM* sebagai perangkat akuisisi citra dan *YOLOv8* sebagai model deteksi objek mampu berjalan dengan baik pada mode gambar, video, maupun *streaming real-time*.

Hasil evaluasi ⁵¹ menunjukkan bahwa model *YOLOv8* memiliki kemampuan yang cukup baik dalam mendeteksi tujuh kelas kondisi panel surya dengan ⁴ nilai *Precision* sebesar 72,80%, *Recall* sebesar 68,66%, dan *F1-Score* sebesar ⁷⁶ 70,67%. Dengan demikian, sistem yang dikembangkan dapat digunakan sebagai solusi *monitoring* kondisi panel surya secara otomatis serta berpotensi diterapkan pada lingkungan nyata untuk mendukung proses pemeliharaan panel surya yang lebih efektif.

28
BAB V
PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diperoleh kesimpulan sebagai berikut:

1. Sistem deteksi kondisi panel surya berbasis *ESP32-CAM* dan model *YOLOv8* berhasil dirancang dan diintegrasikan dengan baik. *ESP32-CAM* berfungsi sebagai perangkat akuisisi citra dan *pre-detector* menggunakan metode *Frame Difference*, sedangkan aplikasi *desktop* berbasis Python berperan sebagai penerima data dan pemroses utama menggunakan model *YOLOv8*. Sistem yang dikembangkan mampu menjalankan proses deteksi pada gambar statis, video, maupun *streaming* kamera secara *real-time* sehingga dapat digunakan untuk membantu proses *monitoring* kondisi panel surya secara otomatis.
2. Model *YOLOv8* mampu mendeteksi tujuh kondisi panel surya yang terdiri dari *Bird Drop*, *Defective*, *Dust*, *Non Defective*, *Physical Damage*, *Snow*, dan *Leaf* dengan performa yang cukup baik. Berdasarkan hasil evaluasi *confusion matrix* diperoleh nilai *Precision* sebesar 72,80%, *Recall* sebesar 68,66%, dan *F1-Score* sebesar 70,67%. Selain itu, sistem memiliki waktu inferensi rata-rata sebesar 85,4 ms per *frame* yang menunjukkan bahwa model mampu bekerja pada skenario *real-time* pada perangkat komputasi ringan. Hasil tersebut menunjukkan bahwa *YOLOv8* memiliki potensi yang baik untuk diterapkan pada sistem *monitoring* dan inspeksi kondisi panel surya secara otomatis.

B. Saran

Sebagai pengembangan lebih lanjut, penelitian ini diharapkan dapat menjadi dasar bagi penelitian berikutnya dengan menambahkan jumlah dan variasi *dataset* agar model deteksi memiliki tingkat

generalisasi yang lebih baik. Selain itu, pengembangan sistem dapat diarahkan pada integrasi dengan platform berbasis *web* atau *mobile* untuk memudahkan akses informasi secara jarak jauh.

Penelitian selanjutnya juga disarankan untuk menguji performa sistem pada berbagai kondisi lingkungan, seperti perbedaan pencahayaan dan cuaca, serta melakukan optimasi model agar lebih efisien ketika diterapkan pada perangkat dengan sumber daya terbatas. Dengan pengembangan tersebut, sistem deteksi kondisi panel surya diharapkan dapat diterapkan secara lebih luas pada lingkungan nyata

ORIGINALITY REPORT

12%

SIMILARITY INDEX

9%

INTERNET SOURCES

6%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1 Submitted to Universitas Islam Riau

Student Paper

2 Submitted to UPN Veteran Yogyakarta

Student Paper

3 informatika.unpkediri.ac.id

Internet Source

4 statistics.unl.edu

Internet Source

5 publikasi.dinus.ac.id

Internet Source

6 eprints3.upgris.ac.id

Internet Source

7 journal.lontaradigitech.com

Internet Source

8 repository.unpkediri.ac.id

Internet Source

9 Tri Ahmad Jabar JABAR, Rudi Heriansyah, Evi Purnamasari. "Pengenal Pola Huruf Bahasa Isyarat Menggunakan Framework You Only Look Once (YOLO)", JURNAL FASILKOM, 2025

Publication

10 Submitted to Universitas Pancasila

Student Paper

11 repository.ub.ac.id

Internet Source

12 repository.upiypk.ac.id

Internet Source

13 Submitted to Universitas Tarumanagara

Student Paper

14 cdn.juris.id

Internet Source

-
- 15 eprints.uny.ac.id
Internet Source
-
- 16 Submitted to Universitas Muslim Indonesia
Student Paper
-
- 17 Submitted to Universitas Mataram
Student Paper
-
- 18 Hashri Hayati, Muhammad Riza Alifi. "ANALISIS BRAND LAYANAN AKADEMIK PERGURU. TINGGI INDONESIA MENGGUNAKAN KLASIFIKASI TEKS DI MEDIA SOSIAL", Syntax : Journal of Software Engineering, Computer Science and Information Technology, 2025
Publication
-
- 19 Submitted to STT PLN
Student Paper
-
- 20 Submitted to Universitas Muhammadiyah Surakarta
Student Paper
-
- 21 Zendra Utama, Pahrizal Pahrizal. "PENERAPAN METODE CNN UNTUK ANALISIS SENTIM PENGGUNA APLIKASI DOLA", INTECOMS: Journal of Information Technology and Computer Science, 2026
Publication
-
- 22 spaceweather.astron.nl
Internet Source
-
- 23 www.coursehero.com
Internet Source
-
- 24 Submitted to Abdullah Gul University
Student Paper
-
- 25 Citha Nastitie, Irma Handayani2. "PERBANDINGAN METODE MOBILENETV2 DAN EFFICIENTNETB0 UNTUK KLASIFIKASI JENIS PISANG", INTECOMS: Journal of Information Technology and Computer Science, 2025
Publication
-
- 26 Haffas Zikri Ariyandi, Muis Muhtadi, Dodik Dwi Andreanto. "Metode Frame Difference untuk Deteksi Gerakan Tidur Bayi berbasis Computer Vision", Edumatic: Jurnal Pendidikan Informatika, 2025
Publication
-
- 27 Jaejin Jung, Sungyong Kim, Shinhwan Lim, Kyoungmin Koh, Hae-Seung Lee. "Mitigation Vertical Fixed Pattern Noise in CMOS Image Sensors through Analog Offset Dithering and Dynamic Clock Adjustment", 2025 IEEE International Symposium on Circuits and Systems (ISCAS), 2025
Publication
-

28 repository.uinpalopo.ac.id
Internet Source

29 Submitted to Hankuk University of Foreign Studies
Student Paper

30 doku.pub
Internet Source

31 journalversa.com
Internet Source

32 jurusan.tik.pnj.ac.id
Internet Source

33 repository.uinjambi.ac.id
Internet Source

34 widuri.raharja.info
Internet Source

35 Submitted to SDM Universitas Gadjah Mada
Student Paper

36 Submitted to Universitas Singaperbangsa Karawang
Student Paper

37 journals.gesociety.org
Internet Source

38 klikgss.com
Internet Source

39 Submitted to Fakultas Teknik
Student Paper

40 Submitted to Universitas Indonesia
Student Paper

41 Submitted to Universitas Pendidikan Ganesha
Student Paper

42 Anggi Riantika Dewi, Sri Diana, Moch Alvi Fakhrezi, Nana Awang, Helmi Ma'arif, Dedi Dw Saputra. "SENTIMEN ANALISIS TERHADAP PUAN MAHARANI SEBAGAI KANDIDAT CALON PRESIDEN 2024 BERDASARKAN OPINI TWITTER MENGGUNAKAN METODE NAIVE BAYES DAN ADABOOST", JSil (Jurnal Sistem Informasi), 2023
Publication

43 Submitted to Universitas 17 Agustus 1945 Semarang
Student Paper

44

Submitted to Universitas Muhammadiyah Makassar

Student Paper

45

Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur

Student Paper

46

Submitted to Perpustakaan

Student Paper

47

Submitted to Universitas Merdeka Malang

Student Paper

48

Submitted to Universitas Muhammadiyah Purwokerto

Student Paper

49

jurnal.polibatam.ac.idInternet Source

50

jurnal.stkipppgritulungagung.ac.idInternet Source

51

Chaerul Hidayat, Agung Nugroho, Asep Suprianto. "Prediksi Risiko Stroke Berdasarkan Faktor Klinis Menggunakan Random Forest Dengan Optimasi Threshold dan SHAP", Journal Automation Computer Information System, 2026

Publication

52

journals.arces.orgInternet Source

53

repository.fe.unj.ac.idInternet Source

54

www.slideshare.netInternet Source

55

Istanti, Ike. "Peran Mediasi Etos Kerja Islam Dalam Hubungan Antara Perceived Organizational Support dan Work Engagement Dengan Inovation Behavior", Universitas Islam Sultan Agung (Indonesia), 2024

Publication

56

Submitted to Sriwijaya University

Student Paper

57

journalcenter.orgInternet Source

58

penerbit.stekom.ac.idInternet Source

59

repository.telkomuniversity.ac.idInternet Source

60 repository.uinsu.ac.id
Internet Source

61 repository.usd.ac.id
Internet Source

62 text-id.123dok.com
Internet Source

63 123dok.com
Internet Source

64 Navedha Evanjalín R, Mohaideen Pitchai K, Dhanush Ragavendar S. "Energy-Efficient Target Coverage in Wireless Visual Sensor Networks Using LSTM-Based Trajectory Prediction and Grey Wolf Optimization", 2026 7th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), 2026
Publication

65 Samak, Ibrahim M. A.. "Derin öğrenme tabanlı yolo algoritması ile otomatik çöp sınıflandırma sistemi.", Üzüküdar University (Turkey)
Publication

66 Thomas Brian, Immanuel Freddy Augustino, Parman Parman, Muhamad Sukarno. "SISTEM DETEKSI KERUSAKAN PANEL PLTS APUNG DI EMBUNG SIDOBANDUNG BERBASIS CONVOLUTIONAL NEURAL NETWORK DENGAN VISUALISASI AUGMENTED REALITY", TEKNIMEDIA: Teknologi Informasi dan Multimedia, 2026
Publication

67 Submitted to Universitas Muria Kudus
Student Paper

68 adirasozietty.blogspot.com
Internet Source

69 e-jurnal.lppmunsera.org
Internet Source

70 elib.pnc.ac.id
Internet Source

71 Adnan Buyung Nasution, Ahir Yugo Nugroho Hrp, Muhammad Fauzi, Yudi, Heri Gunawati. "Optimizing Web Access for Hand Disability Sufferers Through Blink Technology and Deep Learning with CNN/Tensor Flow", Indonesian Journal of Interdisciplinary Research in Science and Technology, 2025
Publication

72 Miftahul Qowim, Nina Aini Mahbubah, M. Zainuddin Fathoni. "PENERAPAN 5S PADA DIPTERAKSI GUDANG (STUDI KASUS PT. SUMBER URIP SEJATI)", JUSTI (Jurnal Sistem dan Teknik Industri), 2020
Publication

-
- 73 digilib.iain-palangkaraya.ac.id
Internet Source
-
- 74 digilib.uin-suka.ac.id
Internet Source
-
- 75 eprints.upj.ac.id
Internet Source
-
- 76 es.scribd.com
Internet Source
-
- 77 jurnal.kdi.or.id
Internet Source
-
- 78 repository.dinamika.ac.id
Internet Source
-
- 79 repository.iainpalopo.ac.id
Internet Source
-
- 80 repository.its.ac.id
Internet Source
-
- 81 repository.ittelkom-pwt.ac.id
Internet Source
-
- 82 staff.uny.ac.id
Internet Source
-
- 83 www.scribd.com
Internet Source
-
- 84 Conrad Plaut. "Interactive Linear Algebra", CRC Press, 2026
Publication
-
- 85 Edo Ranov Anjasmara, Farniwati Fattah, Andi Widya Mufila Gaffar. "Implementasi Mode MobileNetV2 pada ESP-32 CAM untuk Klasifikasi Botol Plastik", Jurnal Minfo Polgan, 2026
Publication
-
- 86 Adinda Riska Safitri, Andi Arniaty Arsyad. "Pemodelan Emosi Pengguna Berbasis Ulasan Digital Melalui Integrasi Natural Language Processing dan Ilmu Sosial Komputasional", JURNAL AI-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI, 2026
Publication
-

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

On