

## BAB IV PENGUJIAN DAN ANALISA

### 4.1 Analisis Hasil Pelatihan & Komparasi Model

Berdasarkan hasil *training* diperoleh parameter loss, precision, recall, berupa grafik yang dapat dilakukan analisis komparatif antara Model A dan Model B untuk mengevaluasi performa klasifikasi pada dataset.

#### 4.1.1 Hasil Training Model A

Model A dikonfigurasi dengan dense head 128 neuron, *dropout rate* 0.3, *focal loss* dengan  $\alpha=0.25$  dan  $\gamma=2.0$ , *learning rate* fase pertama sebesar  $1 \times 10^{-3}$ , dan *learning rate* fase *fine-tuning* sebesar  $1 \times 10^{-5}$  dengan 30 lapisan teratas *backbone* yang diaktifkan kembali untuk dilatih ulang (*unfrozen*).

Tabel 4.1 Sample Log Training Model A

Epoch	Fase	Train Acc	Val Acc	Train Loss	Val Loss
1	Phase 1	0.4689	0.5789	0.1351	0.0613
5	Phase 1	0.6430	0.5714	0.0777	0.0557
10	Phase 1	0.6876	0.6767	0.0639	0.0416
15	Phase 1	0.6983	0.7218	0.0612	0.0375
20	Phase 1	0.7177	0.6992	0.0579	0.0360
21	Phase 1	0.7359	0.7594	0.0535	0.0341
25	Phase 1	0.7431	0.6917	0.0520	0.0357
30	Phase 1	0.7488	0.7218	0.0485	0.0328
31	Phase 2	0.5626	0.6842	0.1306	0.0423
35	Phase 2	0.6714	0.6842	0.0682	0.0363
40	Phase 2	0.7258	0.7669	0.0533	0.0311
44	Phase 2	0.7579	0.8120	0.0453	0.0277
47	Phase 2	0.7682	0.8195	0.0430	0.0242
50	Phase 2	0.7720	0.7744	0.0422	0.0250

Berdasarkan data pada Tabel 4.1, proses *training* Model A menunjukkan tren peningkatan yang konsisten. Pada Fase 1 (epoch 1–30), *Training Accuracy* meningkat secara bertahap dari 46.89% pada epoch 1 menjadi 74.88% pada epoch 30. *Validation Accuracy* pada Fase 1 bergerak fluktuatif dalam kisaran 57.14%

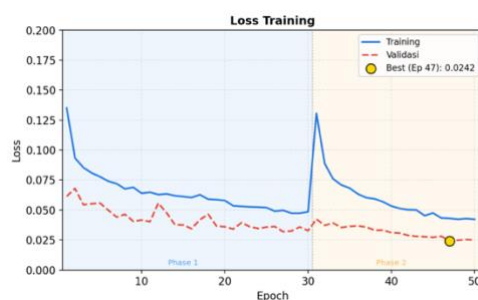
hingga 75.94%, dengan nilai tertinggi Fase 1 tercapai pada epoch 21 sebesar 75.94%. *Training Loss* juga menunjukkan penurunan stabil dari 0.1351 menjadi 0.0485, mengindikasikan model berhasil meminimalkan *error* pada data latih secara progresif.

Transisi ke Fase 2 (*Fine-Tuning*) pada epoch 31 ditandai dengan penurunan sementara pada *Training Accuracy* menjadi 56.26% dan kenaikan Train Loss menjadi 0.1306. Penurunan ini adalah kondisi yang wajar dan dapat diantisipasi, karena saat lapisan *base model* dibuka kembali untuk *fine-tuning*, bobot yang sebelumnya telah disesuaikan untuk mengekstraksi fitur umum mengalami perubahan awal akibat *learning rate* yang berlaku. Meskipun demikian, model dengan cepat pulih dan kembali mengalami tren peningkatan.

Pada Fase 2, kinerja model meningkat secara signifikan melampaui pencapaian Fase 1. *Training Accuracy* terus naik hingga mencapai 77.20% pada epoch 50, sementara *Training Loss* berhasil ditekan hingga 0.0422. Yang paling krusial, *Validation Accuracy* pada Fase 2 menunjukkan peningkatan substansial, dengan nilai terbaik dicapai pada epoch ke-47 sebesar 81.95% disertai *Validation Loss* terendah sebesar 0.0242. Nilai inilah yang menjadi dasar model terbaik yang disimpan dan digunakan untuk evaluasi pada test set.



**Gambar 4.1** Grafik Akurasi Training Model A



**Gambar 4.2** Grafik Loss Training Model A

Gambar 4.1 menunjukkan kurva *training* dan validasi Model A secara keseluruhan. Pada grafik Akurasi *Training*, kurva *Training* (garis biru) dan Validasi (garis merah) bergerak beriringan dengan perbedaan yang tidak terlalu besar sepanjang fase 1 dan fase 2. Hal ini mengindikasikan tidak adanya gejala *overfitting* yang signifikan. Pada grafik *Loss Training*, kurva validasi (merah) cenderung

berada di bawah kurva training (biru) selama sebagian besar proses pelatihan, yang menunjukkan model memiliki kemampuan generalisasi yang baik terhadap data yang belum pernah dilihat sebelumnya. Lonjakan *loss* pada awal Fase 2 (epoch 31) terlihat jelas pada grafik kanan, namun segera stabil dan menurun kembali menuju nilai optimal.

#### 4.1.2 Hasil Training Model B

Model B dikonfigurasi dengan *dense head* 256 neuron, *dropout rate* 0.5, *focal loss* dengan  $\alpha=0.35$  dan  $\gamma=1.5$ , *learning rate* fase pertama sebesar  $5 \times 10^{-4}$ , dan *learning rate* fase *fine-tuning* sebesar  $1 \times 10^{-6}$  dengan lapisan teratas *backbone* yang diaktifkan kembali untuk dilatih ulang (*unfrozen*).

**Tabel 4.2** Sample Log Training Model B

Epoch	Fase	Train Acc	Val Acc	Train Loss	Val Loss
1	Phase 1	0.4744	0.5564	0.2229	0.1632
5	Phase 1	0.6327	0.5639	0.1411	0.1505
10	Phase 1	0.6721	0.7068	0.1263	0.1173
16	Phase 1	0.7056	0.7594	0.1140	0.0964
20	Phase 1	0.7159	0.7519	0.1085	0.0935
23	Phase 1	0.7244	0.7143	0.1032	0.0920
24	Phase 1	0.7171	0.7444	0.1061	0.0998
25	Phase 2	0.4950	0.7594	0.2234	0.0984
26	Phase 2	0.5226	0.7820	0.2002	0.1011
28	Phase 2	0.5588	0.7368	0.1768	0.1084
30	Phase 2	0.5773	0.7068	0.1609	0.1137
34	Phase 2	0.5860	0.6992	0.1553	0.1309

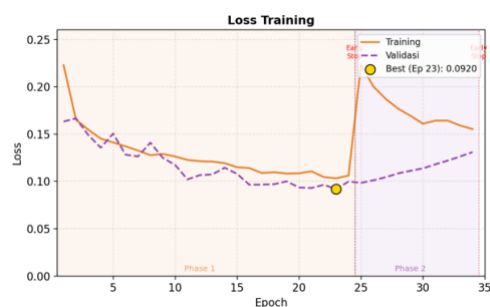
Berdasarkan Tabel 4.2, Fase 1 Model B berlangsung selama 24 epoch dikarenakan *val\_acc* stagnan di angka 0,7594 sebanyak 8 kali sehingga *Early Stopping* menghentikan pelatihan. Selama Fase 1, *Training Accuracy* meningkat dari 47.44% pada epoch 1 menjadi 71.71% pada epoch 24. *Validation Accuracy* bergerak fluktuatif dengan nilai tertinggi pada Fase 1 mencapai 75.94% di epoch 16. *Training Loss* turun dari 0.2229 menjadi 0.1061, sementara *Validation Loss* turun dari 0.1632 menjadi 0.0998. Mekanisme *Early Stopping* menghentikan Fase

1 pada epoch 24 karena tidak ada perbaikan signifikan pada *Validation Loss* selama 8 epoch berturut turut.

Fase 2 dimulai pada epoch 25 dan berlangsung hanya selama 10 epoch (epoch 25–34) sebelum *Early Stopping* kembali mengakhiri pelatihan. Pola yang sama dengan Model A terlihat pada awal Fase 2, yaitu *Training Accuracy* turun drastis ke 49.50% dan *Training Loss* melonjak ke 0.2234. Namun, yang perlu dicatat adalah *Validation Accuracy* justru tetap terjaga dengan baik di angka 75.94% pada epoch pertama Fase 2, mengindikasikan kemampuan generalisasi model yang kuat. Nilai *Validation Accuracy* terbaik Model B dicapai pada epoch 26 sebesar 78.20% dengan *Validation Loss* sebesar 0.1011.



**Gambar 4.3** Grafik Akurasi Training Model B

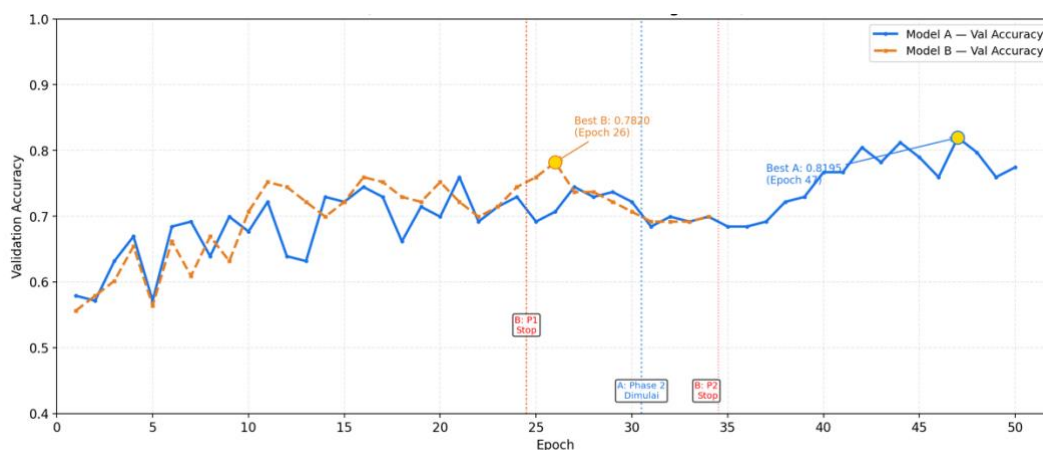


**Gambar 4.4** Grafik Loss Training Model B

Gambar 4.3 menampilkan kurva *training* Model B dengan dua fase yang terpisah. Pada grafik Akurasi *Training*, terdapat pola yang mencolok di mana kurva *Training* (garis oranye) mengalami penurunan tajam pada transisi ke Fase 2 (epoch 25), sementara kurva *Validasi* (garis ungu) relatif stabil. Fenomena ini menunjukkan bahwa meskipun bobot-bobot yang baru dibuka untuk *fine-tuning* memerlukan waktu penyesuaian, representasi fitur yang telah dipelajari model tetap relevan untuk generalisasi. Pada Gambar 4.4 grafik *Loss Training*, pola serupa terlihat dengan lonjakan *Training Loss* di awal Fase 2, yang kemudian distabilkan oleh mekanisme *Early Stopping* sebelum *overfitting* terjadi.

### 4.1.3 Komparasi Validasi Model A dan B

Untuk memberikan gambaran perbandingan performa kedua model selama proses pelatihan, Gambar 4.5 menampilkan kurva *Validation Accuracy* Model A dan Model B secara bersamaan dalam satu grafik komparasi.



**Gambar 4.5** Komparasi Validation Accuracy Model A dan B

Gambar 4.5 memperlihatkan perbandingan *Validation Accuracy* kedua model sepanjang epoch pelatihan. Garis biru solid merepresentasikan Model A, sementara garis oranye putus-putus merepresentasikan Model B. Beberapa poin krusial dapat diidentifikasi dari grafik ini. Pertama, pada 24 epoch pertama, kedua model menunjukkan tren yang sangat mirip dengan fluktuasi yang hampir beriringan, mengindikasikan bahwa kedua model memiliki kemampuan belajar yang sebanding selama Fase *Transfer Learning*.

Kedua, perbedaan mulai terlihat signifikan setelah memasuki Fase 2. Model B mencapai nilai terbaik lebih awal (epoch 26 dengan 78.20%), namun kemudian mengalami stagnasi dan *Early Stopping* menghentikan pelatihan karena nilai *Validation Accuracy* stagnan selama 8 epoch berturut-turut. Model A, di sisi lain, terus meningkat selama Fase 2 dan akhirnya mencapai nilai puncak 81.95% pada epoch 47, melampaui pencapaian Model B sebesar 3.75%.

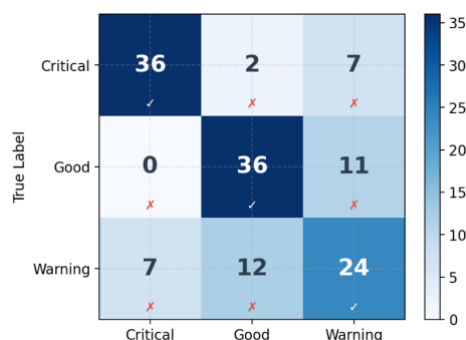
## 4.2 Analisis Hasil Pengujian Test Set

Evaluasi model pada *test set* dilakukan untuk mengukur kemampuan generalisasi model CNN MobileNetV2 yang telah dilatih terhadap data yang belum pernah dilihat sebelumnya. *Test set* terdiri dari 135 sampel citra solder yang terbagi

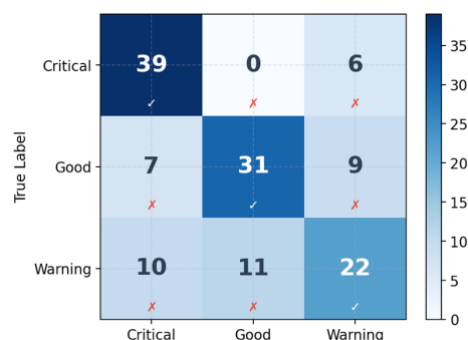
ke dalam tiga kelas: Critical (45 sampel), Good (47 sampel), dan Warning (43 sampel).

#### 4.2.1 Confusion Matriks

*Confusion matrix* digunakan untuk memvisualisasikan secara menyeluruh bagaimana model mengklasifikasikan setiap sampel pada *test set*, termasuk kesalahan prediksi antar kelas. Gambar 4.6 dan Gambar 4.7 menampilkan confusion matrix Model A dan B pada 135 sampel test set.



**Gambar 4.6** Confusion Matriks Testing Set Model A



**Gambar 4.7** Confusion Matriks Testing Set Model B

Sebelum menghitung metrik evaluasi, nilai TP (*True Positive*), FP (*False Positive*), FN (*False Negative*), dan TN (*True Negative*) diturunkan terlebih dahulu dari confusion matrix di atas. Nilai-nilai ini menjadi dasar seluruh perhitungan pada subbagian 4.2.2.

**Tabel 4.3** Derivasi TP, FP, FN, TN per Kelas – Model A

Kelas	TP	FP	FN	TN
Critical	36	7	9	83
Good	36	14	11	74
Warning	24	18	19	74

**Tabel 4.4** Derivasi TP, FP, FN, TN per Kelas – Model B

Kelas	TP	FP	FN	TN
Critical	39	17	6	73
Good	31	11	16	77
Warning	22	15	21	77

### 4.2.2 Metrik Evaluasi Per Kelas

Evaluasi per kelas dilakukan untuk memahami kinerja model secara lebih detail pada masing-masing kategori kualitas solder. Tiga metrik utama yang digunakan adalah *Precision*, *Recall*, dan F1-Score. *Precision* mengukur seberapa akurat prediksi positif model. *Recall* mengukur kemampuan model menemukan seluruh sampel dari suatu kelas. F1-Score merupakan rata-rata harmonik dari *Precision* dan *Recall* yang memberikan ukuran keseimbangan keduanya

#### A. Kelas Critical

Dari nilai TP, FP, TN, FN berdasarkan Tabel 4.3 dan Tabel 4.4, perhitungan lengkap metrik kelas critical untuk kedua model disajikan pada Tabel 4.5 dan Tabel 4.6 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.5** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Critical Model A

Metrik	Perhitungan
Accuracy	$\frac{36 + 83}{135} = 0,8815$
Precision	$\frac{36}{36 + 7} = 0,8372$
Recall	$\frac{36}{36 + 9} = 0,8000$
F1-Score	$\frac{2 \times (0,8372 \times 0,8000)}{0,8372 + 0,8000} = 0,8182$

**Tabel 4.6** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Critical Model B

Metrik	Perhitungan
Accuracy	$\frac{39 + 73}{135} = 0,8296$
Precision	$\frac{39}{39 + 17} = 0,6964$

Recall	$\frac{39}{39 + 6} = 0.8667$
F1-Score	$\frac{2 \times (0,6964 \times 0,8667)}{0,6964 + 0,8667} = 0,7723$

Dalam konteks AOI, kesalahan pada kelas ini memiliki konsekuensi tertinggi karena dapat menyebabkan produk cacat lolos ke tahap berikutnya (*False Negative*) atau sebaliknya produk baik ditolak secara berlebihan (*False Positive*).

Berdasarkan hasil perhitungan pada Tabel 4.5 dan Tabel 4.6, Model A mencapai Precision sebesar 83.72% dengan Recall 80.00% dan F1-Score 81.82%. Ini berarti dari setiap 100 sampel yang Model A prediksi sebagai Critical, sekitar 84 di antaranya memang benar-benar Critical. Model B menunjukkan karakteristik yang berbeda: Recall untuk Critical lebih tinggi (86.67%) namun Precision jauh lebih rendah (69.64%), menghasilkan F1-Score 77.23%. Model B lebih agresif dalam memprediksi Critical sehingga lebih sedikit yang terlewat, namun dengan biaya false alarm yang jauh lebih banyak.

#### B. Kelas Warning

Dari nilai TP, FP, TN, FN berdasarkan Tabel 4.3 dan Tabel 4.4, perhitungan lengkap metrik kelas warning untuk kedua model berdasarkan persamaan 2.12, 2.13, 2.14, 2.15 disajikan pada tabel perhitungan Tabel 4.7 dan Tabel 4.8 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.7** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Warning Model A

Metrik	Perhitungan
Accuracy	$\frac{24 + 74}{135} = 0,7259$
Precision	$\frac{24}{24 + 18} = 0,5714$
Recall	$\frac{24}{24 + 19} = 0.5581$

---

F1-Score	$\frac{2 \times (0,5714 \times 0,5581)}{0,5714 + 0,5581} = 0,5647$
----------	--

---

**Tabel 4.8** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Warning Model B

---

Metrik	Perhitungan
Accuracy	$\frac{22 + 77}{135} = 0,7333$
Precision	$\frac{22}{22 + 15} = 0,5946$
Recall	$\frac{22}{22 + 21} = 0,5116$
F1-Score	$\frac{2 \times (0,5946 \times 0,5116)}{0,5946 + 0,5116} = 0,5500$

---

Sebagaimana terlihat dari hasil perhitungan pada Tabel 4.7 dan 4.8. Model A mencapai F1-Score 56.47% (Precision 57.14%, Recall 55.81%), sementara Model B lebih rendah dengan F1-Score 55.00% (Precision 59.46%, Recall 51.16%). Rendahnya performa pada kelas Warning disebabkan oleh ambiguitas visual antara kelas Warning dengan kelas Good. Solder dalam kondisi Warning sering memiliki karakteristik yang berada di antara kondisi lainnya, sehingga batas keputusan (*decision boundary*) CNN sulit memisahkannya dengan sempurna. Model A sedikit unggul dengan selisih F1-Score +1.47% dibandingkan Model B.

### C. Kelas Good

Dari nilai TP, FP, TN, FN berdasarkan Tabel 4.3 dan Tabel 4.4, perhitungan lengkap metrik kelas warning untuk kedua model disajikan pada tabel perhitungan Tabel 4.9 dan Tabel 4.10 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.9** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Good Model A

---

Metrik	Perhitungan
Accuracy	$\frac{36 + 74}{135} = 0,8148$

---

Precision	$\frac{36}{36 + 14} = 0,7200$
Recall	$\frac{36}{36 + 11} = 0,7660$
F1-Score	$\frac{2 \times (0,7200 \times 0,7660)}{0,7200 + 0,7660} = 0,7423$

**Tabel 4.10** Perhitungan Akurasi, Presisi, Recall dan F1-Score Kelas Warning Model B

Metrik	Perhitungan
Accuracy	$\frac{31 + 77}{135} = 0,8000$
Precision	$\frac{31}{31 + 11} = 0,7381$
Recall	$\frac{31}{31 + 16} = 0,6966$
F1-Score	$\frac{2 \times (0,7381 \times 0,6966)}{0,7381 + 0,6966} = 0,6966$

Berdasarkan hasil pada Tabel 4.9 dan 4.10, Model A unggul pada kelas ini dengan Precision 72.00%, Recall 76.60%, dan F1-Score 74.23%. Model B memiliki Precision sedikit lebih tinggi (73.81%) namun Recall lebih rendah (65.96%) sehingga F1-Score-nya lebih rendah (69.66%). Perbedaan paling mencolok adalah Model B salah mengklasifikasikan 7 sampel Good sebagai Critical, kesalahan yang sama sekali tidak terjadi pada Model A (0 FP untuk Critical dari kelas Good). Kesalahan ini berimplikasi serius karena produk yang sebenarnya bagus akan dianggap cacat serius, menyebabkan pengerjaan ulang yang tidak perlu.

### 4.2.3 Komparasi Metriks

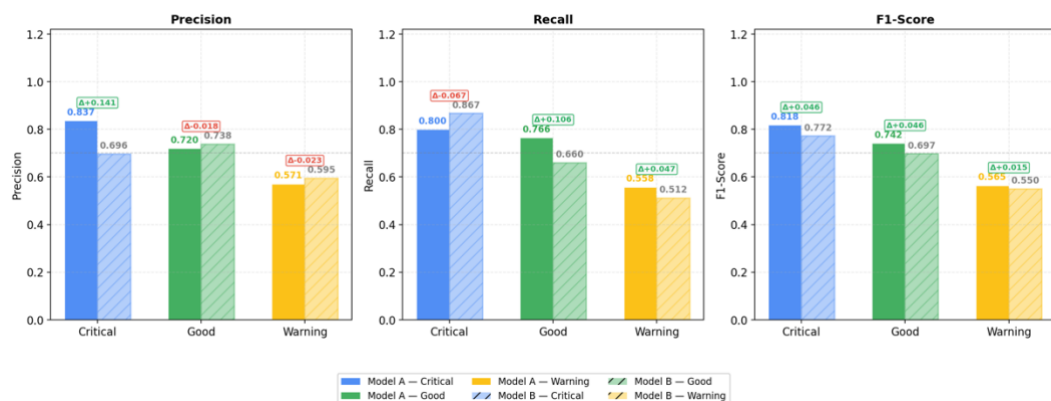
Rekapitulasi perbandingan metrik evaluasi kedua model disajikan pada Tabel 4.11. Berdasarkan Tabel komparasi, Model A menunjukkan performa yang lebih baik secara keseluruhan dibandingkan Model B. Hal ini ditunjukkan oleh nilai F1-Score yang lebih tinggi pada seluruh kelas, yaitu 0,8182 untuk kelas Critical, 0,7423

untuk kelas Good, dan 0,5647 untuk kelas Warning. Selain itu, Model A juga memiliki nilai precision yang lebih tinggi pada kelas Critical serta recall yang lebih baik pada kelas Good dan Warning, yang mengindikasikan kemampuan klasifikasi yang lebih seimbang dan konsisten pada seluruh kategori solder.

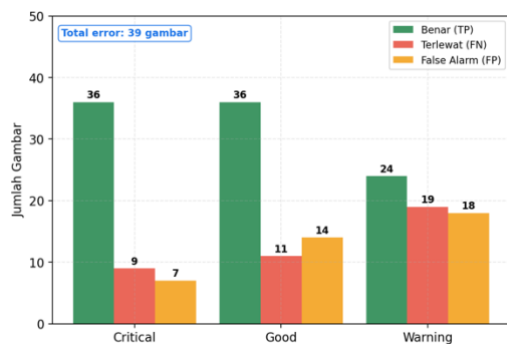
**Tabel 4.11** Rekapitulasi Perbandingan Matriks Evaluasi Model A dan B

Model	Kelas	Accuracy	Precision	Recall	F1-Score
Model A	Critical	0,8815	0.8372	0.8000	0.8182
Model A	Good	0,8148	0.7200	0.7660	0.7423
Model A	Warning	0,7259	0.5714	0.5581	0.5647
Model B	Critical	0,8296	0.6964	0.8667	0.7723
Model B	Good	0,8000	0.7381	0.6596	0.6966
Model B	Warning	0,7333	0.5946	0.5116	0.5500

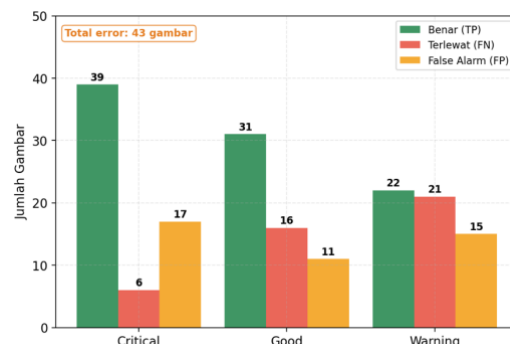
Di sisi lain, Model B memiliki keunggulan pada kemampuan deteksi kelas Critical dengan nilai recall sebesar 0,8667, lebih tinggi dibandingkan Model A yang memperoleh nilai 0,8000. Hasil ini menunjukkan bahwa Model B lebih sensitif dalam mengidentifikasi cacat kritis sehingga berpotensi mengurangi jumlah kasus yang tidak terdeteksi. Namun, nilai precision dan F1-Score yang lebih rendah menunjukkan adanya peningkatan kesalahan klasifikasi pada kelas tersebut. Dengan demikian, berdasarkan keseimbangan kinerja pada seluruh metrik evaluasi, Model A dinilai lebih optimal untuk diterapkan pada sistem inspeksi solder yang dikembangkan.



**Gambar 4.8** Komparasi Grafik Presisi, Recall dan F1-Score per-Kelas



**Gambar 4.9** Grafik Kesalahan Prediksi Model A



**Gambar 4.10** Grafik Kesalahan Prediksi Model B

Gambar 4.9 dan Gambar 4.10 secara visual menunjukkan hasil perhitungan ketiga metrik. Total error Model A adalah 39 gambar, lebih rendah dibandingkan Model B yang mencapai 43 gambar. Kelas Warning pada kedua model konsisten menunjukkan jumlah error tertinggi, mengonfirmasi temuan bahwa kelas ini membutuhkan perhatian khusus dalam pengembangan sistem AOI lebih lanjut.

Dari analisis hasil training pada sub bab 4.1 dan 4.2 dapat disimpulkan bahwa model A lebih unggul secara keseluruhan dibanding model B. Dalam pengujian lanjutan, model A digunakan sebagai model yang akan dijalankan pada sistem AOI untuk klasifikasi cacat solder.

### 4.3 Eksplorasi Internal Model Terpilih

Guna memperoleh pemahaman yang komprehensif terhadap karakteristik model hasil pelatihan, dilakukan eksplorasi mendalam terhadap Model A sebagai model terpilih. Eksplorasi mencakup analisis arsitektur dan statistik parameter bobot (*weights*).

#### 4.3.1 Komposisi Arsitektur dan Parameter Model

Model A tersusun atas arsitektur MobileNetV2 dengan penambahan *classification head* yang dirancang untuk klasifikasi tiga kelas kualitas solder. Pada tingkat struktur paling atas (top-level), model terdiri atas enam komponen utama, yaitu *Input Layer*, lapisan *Functional* (MobileNetV2), *GlobalAveragePooling2D*, *Dense* (128 unit), *Dropout* (0,3), dan *Dense output* (3 unit). Apabila lapisan internal MobilenetV2 diperhitungkan secara keseluruhan, model memiliki 154 lapisan dengan komposisi sebagaimana ditunjukkan pada Tabel 4.12.

Tabel 4.12 Komposisi Layer Model A

Tipe Layer	Jumlah	Total Parameter
Conv2D	35	2.125.536
BatchNormalization	52	68.224
ReLU / ReLU6	35	0
DepthwiseConv2D	17	64.224
Add	10	0
Dense	2	164.355
Dropout	1	0

Secara keseluruhan, model memiliki 2.422.339 total parameter yang terbagi menjadi 1.690.755 parameter *trainable* (69,8%) dan 731.584 parameter *frozen* (30,2%). Parameter *trainable* mencakup seluruh *classification head* serta 30 lapisan teratas backbone (block\_13\_project\_BN hingga Conv\_1), sedangkan parameter *frozen* merupakan bobot pre-trained ImageNet yang dipertahankan pada lapisan block\_1 hingga block\_13.

#### 4.3.2 Tensor Bobot dan Interpretasi Statistik

Setiap lapisan dalam bentuk tensor numerik yang nilainya diperoleh dan diperbarui selama proses pelatihan melalui algoritma *backpropagation*. Setiap tipe lapisan memiliki struktur tensor bobot yang berbeda sesuai dengan operasi matematika yang dilakukannya.. Lapisan Conv2D menyimpan bobot dalam tensor empat dimensi ( $h \times w \times C_{in} \times C_{out}$ ), di mana setiap filter berperan sebagai *sliding window* yang digeser ke seluruh area citra untuk mendeteksi pola lokal melalui operasi perkalian elemen. Sebagai contoh, lapisan Conv1 memiliki *shape* ( $3 \times 3 \times 3 \times 32$ ), artinya terdapat 32 filter berukuran  $3 \times 3$  piksel untuk 3 channel warna (R, G, B) dengan total 864 elemen bobot. Berbeda dengan lapisan konvolusi, lapisan Batch Normalization menyimpan bobot dalam vektor satu dimensi (C,) karena tugasnya bukan mendeteksi pola spasial, melainkan menskalakan ( $\gamma$ ) dan menggeser ( $\beta$ ) distribusi aktivasi setiap channel secara independen. Lapisan block\_13\_project\_BN misalnya memiliki shape (160,), yang berarti terdapat 160

parameter  $\gamma$  dan 160 parameter  $\beta$  masing-masing satu untuk tiap channel. Sementara itu, lapisan Dense menyimpan bobot dalam matriks dua dimensi ( $N_{in} \times N_{out}$ ) yang merepresentasikan koneksi penuh antar neuron; lapisan dense memiliki shape  $(1.280 \times 128)$  dengan total 163.840 elemen bobot.

**Tabel 4.13** Statistik Bobot

Layer	Shape	Mean	Std	Min	Max
Conv1	(3,3,3,32)	-0,004990	0,157948	-0,7747	0,7419
expanded_conv_project	(1,1,32,16)	-0,001358	0,179672	-1,2868	0,9651
block_1_expand	(1,1,16,96)	0,001886	0,144511	-1,1235	0,6553
block_6_project	(1,1,192,64)	-0,001002	0,083647	-0,8963	0,5828
block_12_project	(1,1,576,96)	0,000029	0,043160	-0,4602	0,3667
block_14_expand	(1,1,160,960)	-0,000116	0,040036	-0,2887	0,2803
block_16_project	(1,1,960,320)	0,000013	0,035965	-0,4389	0,5835
Conv_1	(1,1,320,1280)	-0,000040	0,028846	-0,1449	0,1478
dense	(1280,128)	-0,005427	0,048149	-0,3360	0,3872
dense_1	(128,3)	0,007956	0,121217	-0,2714	0,2664

Karena tidak memungkinkan untuk menampilkan seluruh nilai individual tensor bobot, analisis bobot dilakukan melalui statistik deskriptif yang meringkas karakteristik distribusi seluruh elemen dalam satu tensor. Statistik yang dihitung mencakup nilai rata-rata (mean), simpangan baku (standard deviation/std), nilai minimum, nilai maksimum

Berdasarkan Tabel 4.13, seluruh lapisan menunjukkan nilai mean yang sangat mendekati nol, berkisar antara, menunjukkan bahwa distribusi bobot bersifat simetris dan proses pelatihan berlangsung stabil. Penurunan standar deviasi bobot dari 0,1579 pada lapisan Conv1 hingga 0,0288 pada lapisan Conv\_1 mengindikasikan transisi dari ekstraksi fitur universal berkontras tinggi pada lapisan awal, menuju penyesuaian fitur yang lebih halus dan terfokus pada lapisan dalam. Pola ini juga mencerminkan efektivitas penggunaan learning rate kecil ( $1 \times 10^{-5}$ ) pada fase fine-tuning yang membatasi besaran perubahan bobot agar

representasi pre-trained dari ImageNet tidak mengalami distorsi berlebihan, sehingga proses adaptasi terhadap domain solder berlangsung secara bertahap dan stabil.

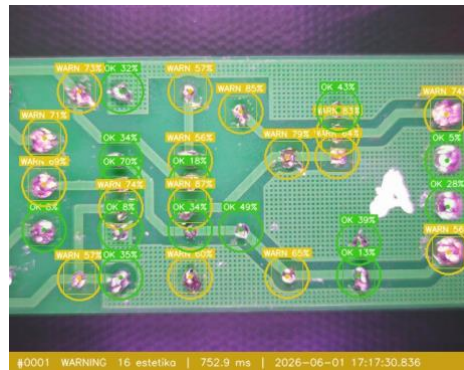
#### **4.4 Pengujian Model Pada RaspberryPI**

Setelah model CNN selesai dilatih dan dievaluasi menggunakan testing set, tahap selanjutnya adalah melakukan pengujian implementasi pada Raspberry Pi 5. Pada tahap ini, conveyor, mekanisme sortir, dan aktuator tidak diaktifkan sehingga pengujian difokuskan pada proses akuisisi citra, pra-pemrosesan, dan inferensi model menggunakan TensorFlow Lite. Pengujian dilakukan menggunakan delapan PCB yang mewakili variasi kelas good, warning, dan critical, dengan masing-masing PCB diuji sebanyak dua kali untuk mengevaluasi konsistensi hasil deteksi. Evaluasi dilakukan pada tingkat titik solder karena model melakukan klasifikasi terhadap setiap ROI solder secara individual. Data hasil klasifikasi akan dipetakan dalam bentuk Confusion Matriks dan dianalisis dalam tiga metrik utama yang digunakan adalah *Precision*, *Recall*, dan F1-Score.

##### **4.4.1 Pengujian Pada Tingkat Titik Solder**

Pengujian pada tingkat titik solder dilakukan untuk mengevaluasi kemampuan model CNN dalam mengklasifikasikan setiap titik solder secara individual ke dalam tiga kelas, yaitu good, warning, dan critical. Pengujian dilakukan menggunakan delapan PCB yang diuji sebanyak dua kali pada Raspberry Pi 5 dengan kondisi kamera dan pencahayaan yang sama seperti saat sistem AOI digunakan. Hasil klasifikasi kemudian disusun dalam bentuk tabel dan dibandingkan dengan kondisi aktual setiap titik solder untuk mengidentifikasi hasil deteksi yang termasuk *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), maupun *False Negative* (FN). Data tersebut selanjutnya digunakan sebagai dasar dalam penyusunan confusion matrix serta perhitungan metrik kinerja model, seperti akurasi, presisi, recall, dan F1-score.

##### **A. PCB A**



**Gambar 4.11** Hasil Pengujian PCB A - Percobaan 1



**Gambar 4.12** Hasil Pengujian PCB A - Percobaan 2

**Tabel 4.14** Confusion Matriks PCB A - Percobaan 1

Aktual/Prediksi	Good	Warning	Critical
Good	0	0	0
Warning	14	16	0
Critical	0	0	0

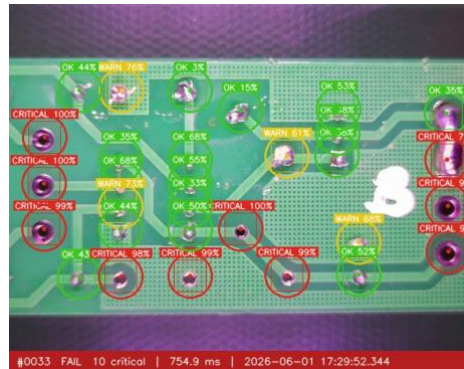
**Tabel 4.15** Confusion Matriks PCB A - Percobaan 2

Aktual/Prediksi	Good	Warning	Critical
Good	0	0	0
Warning	13	17	0
Critical	0	0	0

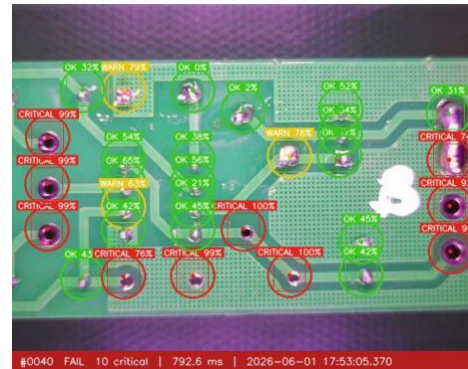
Berdasarkan hasil pengujian pada PCB A, seluruh titik solder yang diuji termasuk dalam kategori warning. Pada percobaan pertama, model berhasil mengklasifikasikan 16 dari 30 titik solder sebagai warning, sedangkan 14 titik lainnya salah diklasifikasikan sebagai good. Pada percobaan kedua, jumlah deteksi yang benar meningkat menjadi 17 titik solder, sementara 13 titik lainnya masih diprediksi sebagai good. Tidak ditemukan kesalahan klasifikasi ke kelas critical pada kedua percobaan. Hasil ini menunjukkan bahwa model cenderung menganggap sebagian solder kategori warning sebagai solder good. Kondisi tersebut mengindikasikan bahwa karakteristik visual kelas warning masih memiliki

kemiripan yang cukup tinggi dengan kelas good, sehingga menjadi salah satu sumber utama kesalahan klasifikasi pada model.

## B. PCB B



**Gambar 4.13** Hasil Pengujian PCB B - Percobaan 1



**Gambar 4.14** Hasil Pengujian PCB B - Percobaan 2

**Tabel 4.16** Confusion Matriks PCB B - Percobaan 1

Aktual/Prediksi	Good	Warning	Critical
Good	12	1	0
Warning	3	3	0
Critical	1	0	10

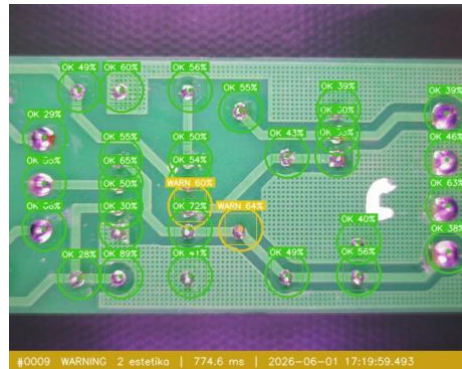
**Tabel 4.17** Confusion Matriks PCB B - Percobaan 2

Aktual/Prediksi	Good	Warning	Critical
Good	12	1	0
Warning	4	2	0
Critical	1	0	10

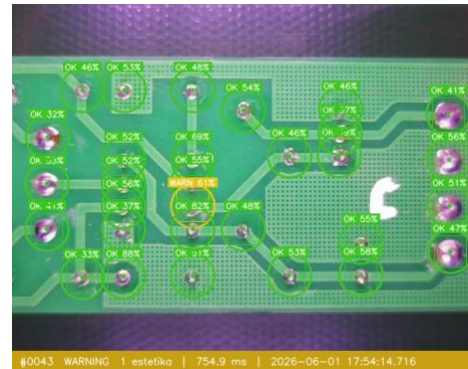
PCB B memiliki kombinasi tiga kelas, yaitu good, warning, dan critical. Pada kedua percobaan, model mampu mendeteksi sebagian besar titik solder kategori good dan critical dengan baik. Seluruh 10 titik critical berhasil dikenali pada kedua percobaan, sedangkan hanya terdapat satu titik critical yang salah diklasifikasikan sebagai good. Di sisi lain, akurasi pada kelas warning masih relatif rendah karena sebagian titik warning diprediksi sebagai good. Hasil ini menunjukkan bahwa model memiliki kemampuan

yang baik dalam mendeteksi cacat kritis, namun masih mengalami kesulitan dalam membedakan kelas warning dengan good.

### C. PCB C



**Gambar 4.15** Hasil Pengujian PCB C - Percobaan 1



**Gambar 4.16** Hasil Pengujian PCB C - Percobaan 2

**Tabel 4.18** Confusion Matriks PCB C - Percobaan 1

Aktual/Prediksi	Good	Warning	Critical
Good	28	2	0
Warning	0	0	0
Critical	0	0	0

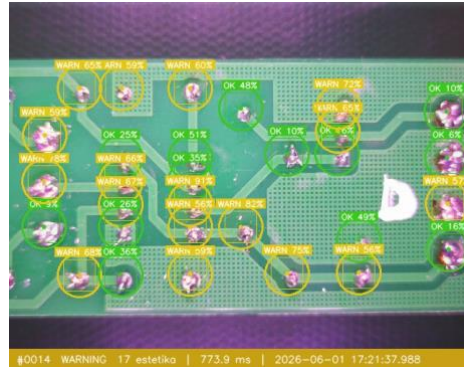
**Tabel 4.19** Confusion Matriks PCB C - Percobaan 2

Aktual/Prediksi	Good	Warning	Critical
Good	29	1	0
Warning	0	0	0
Critical	0	0	0

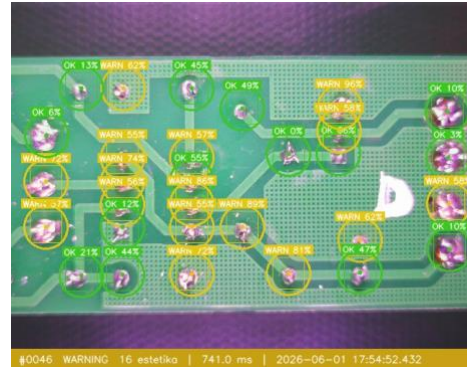
PCB C didominasi oleh titik solder kategori good. Pada percobaan pertama, model berhasil mengidentifikasi 28 dari 30 titik good dengan benar, sedangkan dua titik lainnya terdeteksi sebagai warning. Pada percobaan kedua, jumlah deteksi benar meningkat menjadi 29 titik dan hanya satu titik yang salah diklasifikasikan. Hasil tersebut menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengenali solder normal. Tingkat konsistensi antar percobaan juga menunjukkan

bahwa model mampu mempertahankan performa yang stabil pada kelas good.

#### D. PCB D



**Gambar 4.17** Hasil Pengujian PCB D - Percobaan 1



**Gambar 4.18** Hasil Pengujian PCB D - Percobaan 2

**Tabel 4.20** Confusion Matriks PCB D - Percobaan 1

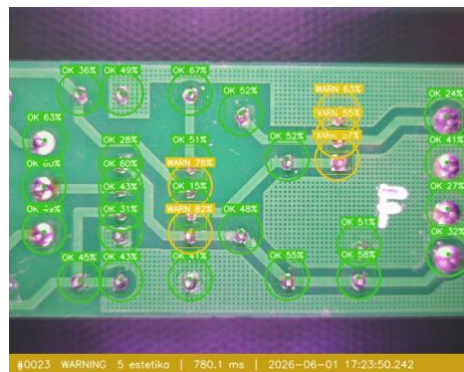
Aktual/Prediksi	Good	Warning	Critical
Good	0	0	0
Warning	13	17	0
Critical	0	0	0

**Tabel 4.21** Confusion Matriks PCB D - Percobaan 2

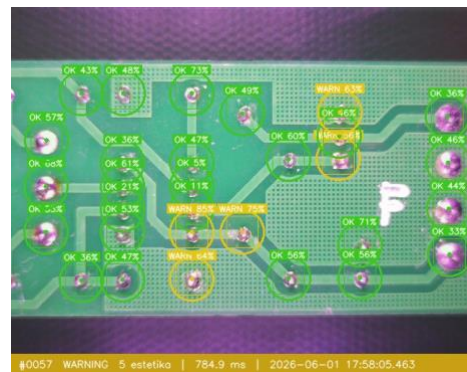
Aktual/Prediksi	Good	Warning	Critical
Good	0	0	0
Warning	14	16	0
Critical	0	0	0

Seluruh titik solder pada PCB D termasuk dalam kategori warning. Hasil pengujian menunjukkan bahwa model berhasil mendeteksi 17 titik warning pada percobaan pertama dan 16 titik pada percobaan kedua. Namun, masih terdapat sejumlah titik yang salah diklasifikasikan sebagai good. Pola kesalahan yang muncul serupa dengan PCB A, yaitu dominasi kesalahan antara kelas warning dan good. Hal ini menunjukkan bahwa model masih mengalami kesulitan dalam mengenali karakteristik cacat estetika yang menjadi ciri utama kelas warning.





**Gambar 4.21** Hasil Pengujian PCB F - Percobaan 1



**Gambar 4.22** Hasil Pengujian PCB F - Percobaan 2

**Tabel 4.24** Confusion Matriks PCB F - Percobaan 1

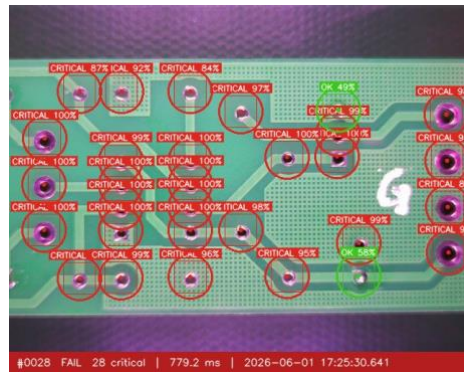
Aktual/Prediksi	Good	Warning	Critical
Good	25	5	0
Warning	0	0	5
Critical	0	0	0

**Tabel 4.25** Confusion Matriks PCB F - Percobaan 2

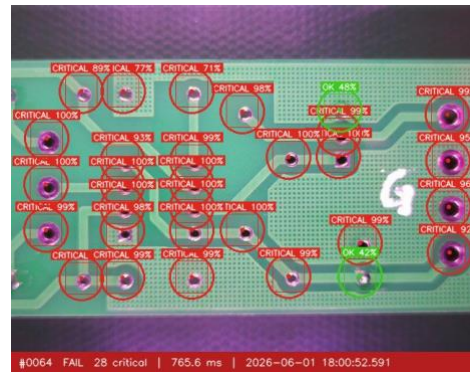
Aktual/Prediksi	Good	Warning	Critical
Good	25	5	0
Warning	0	0	0
Critical	0	0	0

PCB F didominasi oleh kelas good dengan sejumlah kecil titik warning. Pada kedua percobaan, model berhasil mendeteksi seluruh titik good dengan baik, yaitu sebanyak 25 titik. Namun, sebagian titik warning pada percobaan pertama justru diklasifikasikan sebagai critical. Kesalahan ini menunjukkan bahwa beberapa karakteristik visual solder warning memiliki kemiripan dengan solder critical. Meskipun demikian, performa model pada kelas good tetap menunjukkan hasil yang stabil dan konsisten pada kedua percobaan.

## G. PCB G



**Gambar 4.23** Hasil Pengujian PCB G - Percobaan 1



**Gambar 4.24** Hasil Pengujian PCB G - Percobaan 2

**Tabel 4.26** Confusion Matriks PCB G - Percobaan 1

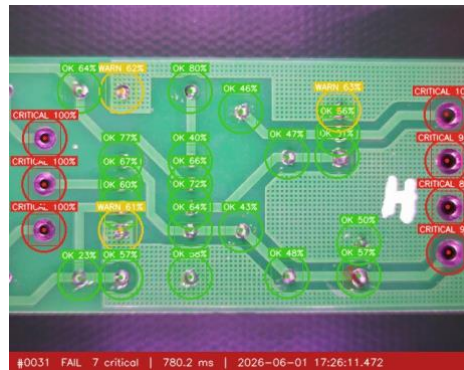
Aktual/Prediksi	Good	Warning	Critical
Good	2	0	0
Warning	0	0	0
Critical	0	0	28

**Tabel 4.27** Confusion Matriks PCB G - Percobaan 2

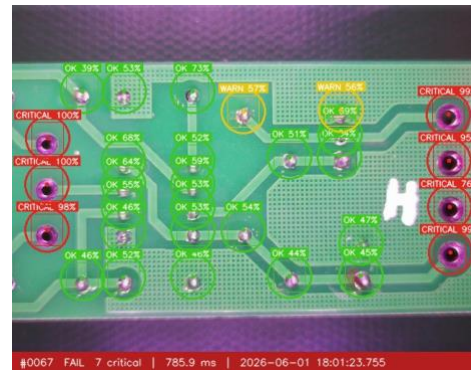
Aktual/Prediksi	Good	Warning	Critical
Good	2	0	0
Warning	0	0	0
Critical	0	0	28

PCB G merupakan PCB yang didominasi oleh kelas critical. Pada percobaan pertama, model berhasil mengidentifikasi 28 dari 30 titik critical, sedangkan dua titik lainnya diprediksi sebagai good. Pada percobaan kedua, seluruh 28 titik critical berhasil dikenali dengan benar tanpa kesalahan klasifikasi. Hasil ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mendeteksi cacat fungsional pada solder. Tingginya tingkat keberhasilan deteksi kelas critical menjadi indikator penting karena kelas ini merupakan fokus utama dalam sistem AOI yang dikembangkan.

## H. PCB H



**Gambar 4.25** Hasil Pengujian PCB H - Percobaan 1



**Gambar 4.26** Hasil Pengujian PCB H - Percobaan 2

**Tabel 4.28** Confusion Matriks PCB H - Percobaan 1

Aktual/Prediksi	Good	Warning	Critical
Good	19	2	0
Warning	1	1	0
Critical	0	0	7

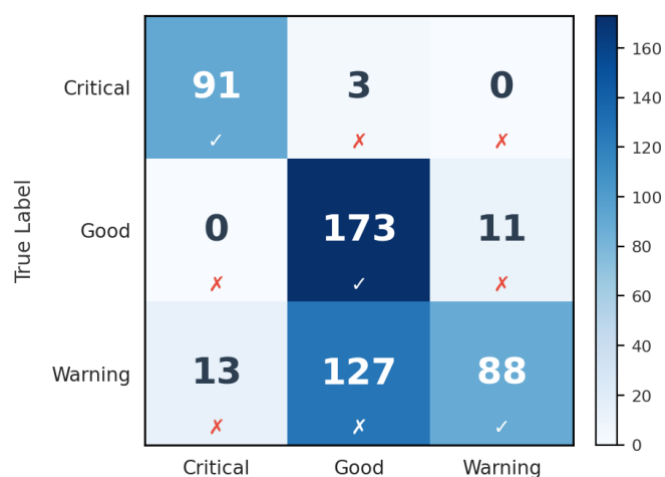
**Tabel 4.29** Confusion Matriks PCB H - Percobaan 2

Aktual/Prediksi	Good	Warning	Critical
Good	19	2	0
Warning	2	0	0
Critical	0	0	7

PCB H terdiri atas kombinasi kelas good, warning, dan critical. Hasil pengujian menunjukkan bahwa seluruh titik critical berhasil terdeteksi dengan benar pada kedua percobaan. Sebagian besar titik good juga berhasil dikenali dengan baik, meskipun masih terdapat beberapa titik yang diprediksi sebagai warning. Pada kelas warning, jumlah deteksi benar masih rendah dan sebagian besar salah diklasifikasikan sebagai good. Pola tersebut kembali menunjukkan bahwa model memiliki performa yang baik pada kelas good dan critical, namun masih menghadapi tantangan dalam membedakan kelas warning dari kelas lainnya

#### 4.4.2 Confusion Matriks

Confusion matrix implementasi diperoleh dengan menjumlahkan seluruh hasil prediksi dari 16 sesi pengujian (8 PCB  $\times$  2 percobaan) pada level titik solder secara individual. Total sampel yang dievaluasi adalah 4 titik solder yang mencakup tiga kelas: Critical, Good, dan Warning. Confusion Matriks gabungan divisualisasikan pada Gambar 4.27.



**Gambar 4.27** Confusion Matriks Gabungan Offline Test

Sebelum menghitung metrik evaluasi, nilai TP, FP, FN, dan TN diturunkan terlebih dahulu dari *confusion matrix* gabungan. Tabel 4.28 menjelaskan derivasi nilai tersebut untuk setiap kelas.

**Tabel 4.30** Derivasi TP, FP, FN, TN Gabungan per-Kelas

Kelas	TP	FP	FN	TN
Critical	91	13	3	83
Good	173	130	11	74
Warning	88	11	140	74

#### 4.4.3 Metrik Evaluasi Per-Kelas

Evaluasi per kelas dilakukan untuk memahami kinerja model secara lebih detail pada masing-masing kategori kualitas solder dalam kondisi operasional pada

perangkat Raspberry Pi 5. Empat metrik utama yang digunakan adalah *Accuracy*, *Precision*, *Recall*, dan F1-Score.

#### A. Critical

Berdasarkan nilai TP, FP, FN, dan TN yang diperoleh dari Tabel 4.30, perhitungan lengkap metrik kelas Critical untuk implementasi Raspberry Pi 5 disajikan pada Tabel 4.31 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.31** Perhitungan, Presisi, Recall dan F1-Score Kelas Critical Gabungan Offline Test

Metrik	Perhitungan
Accuracy	$\frac{91 + 388}{495} = 0,9677$
Precision	$\frac{91}{91 + 13} = 0,8750$
Recall	$\frac{91}{91 + 3} = 0,9681$
F1-Score	$\frac{2 \times (0,8750 \times 0,9681)}{0,8750 + 0,9681} = 0,9676$

Kelas Critical merepresentasikan kondisi solder yang mengalami cacat fungsional dan berpotensi menyebabkan kegagalan kerja PCB. Berdasarkan hasil perhitungan pada Tabel 4.31, implementasi pada Raspberry Pi 5 menghasilkan *Accuracy* sebesar 96,77%, *Precision* sebesar 87.50% dengan *Recall* 96.81% dan F1-Score 91.89%. Ini berarti dari 94 titik solder aktual kelas Critical, hanya 3 titik yang gagal terdeteksi (*False Negative*), sementara 13 titik dari kelas lain salah diprediksi sebagai Critical (*False Positive*). Nilai *Recall* yang mendekati 97% mengindikasikan bahwa model sangat sensitif terhadap cacat fungsional, yang merupakan karakteristik yang paling diutamakan dalam penelitian ini.

#### B. Good

Berdasarkan nilai TP, FP, FN, dan TN yang diperoleh dari Tabel 4.30, perhitungan lengkap metrik kelas Good untuk implementasi Raspberry Pi 5 disajikan pada Tabel 4.32 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.32** Perhitungan, Presisi, Recall dan F1-Score Kelas Good Gabungan Offline Test

Metrik	Perhitungan
Accuracy	$\frac{173 + 181}{495} = 0,7152$
Precision	$\frac{173}{173 + 130} = 0,5710$
Recall	$\frac{173}{173 + 11} = 0,9402$
F1-Score	$\frac{2 \times (0,5710 \times 0,9402)}{0,5710 + 0,9402} = 0,7115$

Kelas Good merepresentasikan kondisi solder yang memenuhi standar kualitas. Implementasi pada Raspberry Pi 5 menghasilkan *Accuracy* sebesar 71,52%, *Recall* 94.02% yang tinggi, artinya hampir seluruh titik solder berkualitas baik berhasil diidentifikasi dengan benar. Namun *Precision* hanya 57.10%, yang mencerminkan tingginya jumlah *False Positive* sebesar 130 titik. Sebagian besar *False Positive* ini berasal dari kelas Warning yang terprediksi sebagai Good (127 dari 130). Kondisi ini menghasilkan F1-Score 71.15% yang moderat akibat ketimpangan antara Precision dan Recall.

### C. Warning

Berdasarkan nilai TP, FP, FN, dan TN yang diperoleh dari Tabel 4.30, perhitungan lengkap metrik kelas warning untuk implementasi Raspberry Pi 5 disajikan pada Tabel 4.33 berdasarkan persamaan 2.12, 2.13, 2.14, 2.15.

**Tabel 4.33** Perhitungan, Presisi, Recall dan F1-Score Kelas Warning Gabungan Offline Test

Metrik	Perhitungan
Accuracy	$\frac{88 + 256}{495} = 0,6949$
Precision	$\frac{88}{88 + 11} = 0,8889$
Recall	$\frac{88}{88 + 140} = 0,3860$

---


$$\text{F1-Score} = \frac{2 \times (0,8889 \times 0,3860)}{0,8889 + 0,3860} = 0,5373$$


---

Kelas Warning merepresentasikan kondisi solder yang mengalami cacat estetika namun masih berfungsi secara elektrik. Sebagaimana terlihat pada Tabel 4.33, kelas ini menjadi yang paling sulit diklasifikasikan dengan benar. Nilai *Accuracy* sebesar 71,52%, *Precision* mencapai 88.89%, namun *Recall* hanya 38.60%, artinya dari 228 titik solder aktual Warning, sebanyak 140 titik terprediksi ke kelas lain. Fenomena ini mengakibatkan F1-Score terendah di antara ketiga kelas, yakni 53.73%. Rendahnya *Recall* Warning disebabkan oleh ambiguitas visual yang antara kelas Warning dengan kelas Good: solder dengan cacat estetika ringan memiliki penampilan yang secara visual dekat dengan solder normal, sehingga batas keputusan (*decision boundary*) CNN sulit memisahkannya secara konsisten pada kondisi pencahayaan dan sudut kamera yang bervariasi di lingkungan operasional.

Tabel 4.34 merangkum seluruh nilai metrik evaluasi per kelas pada implementasi di Raspberry Pi 5

**Tabel 4.34** Rangkuman Perhitungan, Presisi, Recall dan F1-Score Gabungan Offline Test

Kelas	Accuracy	Precision	Recall	F1-Score
Critical	0,9617	0.8750	0.9681	0.9189
Good	0,7152	0.5710	0.9402	0.7115
Warning	0,6949	0.8889	0.3860	0.5373

Hasil implementasi pada offline set menunjukkan bahwa performa model masih konsisten setelah proses deployment pada Raspberry Pi 5. Meskipun demikian, kelas Warning memiliki nilai *Recall* yang lebih rendah (38,60%) dibandingkan kelas Critical (96,81%) dan Good (94,02%). Berdasarkan confusion matrix gabungan, sebagian besar kesalahan berasal dari prediksi Warning menjadi Good, yang menunjukkan bahwa kedua kelas memiliki karakteristik visual yang saling berdekatan sehingga sulit dipisahkan oleh model CNN.

Fenomena tersebut sejalan dengan penelitian Dai dkk. (2020) yang mengembangkan sistem *Automatic Optical Inspection (AOI)* berbasis deep learning

untuk deteksi cacat solder. Penelitian tersebut menyatakan bahwa klasifikasi solder joint tidak hanya dipengaruhi oleh arsitektur CNN, tetapi juga oleh variasi pencahayaan, refleksi permukaan solder, dan perubahan karakteristik visual antar PCB. Faktor-faktor tersebut menyebabkan model lebih sulit mempertahankan performa ketika diterapkan pada kondisi operasional dibandingkan pada data evaluasi laboratorium [45].

Temuan penelitian ini juga konsisten dengan penelitian Gunraj dkk. (2023) melalui sistem SolderNet, yang menunjukkan bahwa meskipun arsitektur CNN modern mampu memperoleh akurasi tinggi pada dataset solder joint, kesalahan klasifikasi tetap didominasi oleh solder dengan karakteristik visual yang berada pada batas antar kelas (*borderline defects*). Oleh karena itu, penelitian tersebut menekankan pentingnya model yang mampu meningkatkan *trustworthiness* dan interpretabilitas agar keputusan klasifikasi terhadap cacat yang bersifat ambigu dapat dipahami dengan lebih baik [46].

Selain itu, penelitian Zhang dan Shen (2021) menunjukkan bahwa inspeksi solder joint sangat dipengaruhi oleh kondisi pencahayaan dan pantulan permukaan solder (*specular reflection*). Meskipun fitur solder dapat diekstraksi dengan baik menggunakan CNN, perubahan kecil pada intensitas cahaya maupun sudut pengambilan citra dapat mengurangi kemampuan model dalam membedakan cacat yang memiliki karakteristik visual serupa [47]. Kondisi tersebut sejalan dengan hasil penelitian ini, di mana kelas Warning yang hanya memiliki cacat estetika ringan lebih sering diprediksi sebagai Good dibandingkan sebagai Warning.

Penelitian yang lebih baru oleh Zhang dkk. (2025) bahkan secara khusus mengatasi masalah variasi lingkungan pada inspeksi solder dengan menambahkan MixStyle untuk mensimulasikan perubahan gaya citra akibat perbedaan pencahayaan dan *Coordinate Attention* untuk memperkuat fokus model pada area cacat solder [48]. Hasil penelitian tersebut menunjukkan bahwa peningkatan kemampuan generalisasi terhadap perubahan kondisi pencahayaan mampu meningkatkan performa klasifikasi pada lingkungan operasional yang berbeda. Temuan ini mendukung hasil penelitian bahwa penurunan Recall kelas Warning

pada offline set kemungkinan dipengaruhi oleh *domain shift* antara data pelatihan dan kondisi implementasi alat.

Secara keseluruhan, hasil penelitian ini menunjukkan kecenderungan yang sama dengan berbagai penelitian AOI berbasis CNN, yaitu bahwa *deployment* pada alat nyata tidak selalu menurunkan performa secara keseluruhan, tetapi kelas yang memiliki karakteristik visual sangat mirip tetap menjadi tantangan utama. Pada penelitian ini, fenomena tersebut ditunjukkan oleh Recall kelas Warning sebesar 38,60%, sedangkan kelas Critical tetap mempertahankan Recall sebesar 96,81%. Hal ini mengindikasikan bahwa MobileNetV2 mampu mempertahankan sensitivitas terhadap cacat fungsional, namun masih mengalami kesulitan dalam membentuk batas keputusan yang jelas antara solder Good dan Warning pada kondisi operasional.

#### 4.4.4 Perbandingan Hasil Pengujian pada Test Set dan Implementasi Raspberry Pi

Komparasi dilakukan untuk menganalisis konsistensi performa model antara tahapan evaluasi menggunakan *test set* dengan kondisi menggunakan *offline set* pada perangkat Raspberry Pi 5. Analisis perbandingan dilakukan berdasarkan hasil klasifikasi dan metrik performa yang diperoleh pada kedua skenario pengujian untuk mengetahui tingkat konsistensi model ketika dijalankan pada lingkungan yang berbeda.

##### A. Critical

Tabel 4.35 menyajikan perbandingan Precision, Recall, dan F1-Score pada kelas critical antara Model A pada test set dan hasil implementasi pada Raspberry Pi 5.

**Tabel 4.35** Perbandingan Hasil Test set dan Offline Test Kelas Critical

Metrik	Model A Test Set	Raspberry Pi 5	$\Delta$ (Raspi – Test Set)
Accuracy	88,15%	96,17%	+8,02%
Precision	83,72%	87,50%	+3,78%
Recall	80,00%	96,80%	+16,80%
F1-Score	81,82%	91,90%	+10,08%

Kelas Critical mencatat peningkatan performa yang paling signifikan pada implementasi Raspberry Pi 5. Recall Critical meningkat dari 80,00% pada test set menjadi 96,80% pada Raspberry Pi 5, menghasilkan selisih positif sebesar +16,80 poin persentase. Dari 94 titik solder aktual Critical, hanya 3 titik yang gagal terdeteksi pada implementasi, dibandingkan 9 sampel yang terlewat pada *test set*. *Precision Critical* juga sedikit meningkat (+3,78%) sehingga F1-Score Critical mencapai 91,90%, naik 10,08 poin dari 81,82% pada test set. Temuan ini mengindikasikan bahwa model mempertahankan sensitivitas terhadap cacat fungsional setelah proses *deployment* ke perangkat *edge computing*.

#### B. Good

Tabel 4.36 menyajikan perbandingan *Accuracy*, *Precision*, *Recall*, dan F1-Score pada kelas good antara Model A pada test set dan hasil implementasi pada Raspberry Pi 5.

**Tabel 4.36** Perbandingan Hasil Test set dan Offline Test Kelas Good

Metrik	Model A Test Set	Raspberry Pi 5	$\Delta$ (Raspi – Test Set)
Accuracy	81,48%	71,52%	-5,04%
Precision	72,00%	57,10%	-14,90%
Recall	76,60%	94,00%	+17,40%
F1-Score	74,23%	71,10%	-3,13%

Pada kelas Good, terjadi pola trade-off yang berlawanan. Recall Good meningkat signifikan dari 76,60% menjadi 94,00% (+17,40%), artinya model pada Raspberry Pi 5 sangat jarang melewatkan solder berkualitas baik. Namun, Precision Good turun dari 72,00% menjadi 57,10% (-14,90%), yang mengindikasikan bahwa lebih banyak titik solder dari kelas lain (khususnya Warning) diprediksi sebagai Good secara keliru. Hal ini konsisten dengan pola misklasifikasi Warning menjadi Good yang dominan pada implementasi sebagaimana terlihat pada *confusion matrix* gabungan. Secara keseluruhan, F1-Score Good tetap relatif stabil dengan selisih -3,13 poin.

#### C. Warning

Tabel 4.37 menyajikan perbandingan Precision, Recall, dan F1-Score pada kelas good antara Model A pada test set dan hasil implementasi pada Raspberry Pi 5.

**Tabel 4.37** Perbandingan Hasil Test set dan Offline Test Kelas Warning

Metrik	Model A Test Set	Raspberry Pi 5	$\Delta$ (Raspi – Test Set)
Accuracy	72,59%	69,49%	-3,10%
Precision	57,14%	88,90%	+31,76%
Recall	55,81%	38,60%	-17,21%
F1-Score	56,47%	53,90%	-2,57%

Kelas Warning menunjukkan pola yang paling kompleks. Precision Warning meningkat drastis dari 57,14% menjadi 88,90% (+31,76%), namun Recall Warning justru turun dari 55,81% menjadi 38,60% (-17,21%). Kondisi ini menggambarkan bahwa pada implementasi, model menjadi lebih selektif dalam memprediksi Warning: ketika model memprediksi suatu titik sebagai Warning, kemungkinan besar prediksi tersebut benar, namun banyak titik Warning yang terlewat dan dikategorikan sebagai Good. Kelas Warning tetap menjadi kelas dengan performa terendah pada kedua kondisi pengujian, dengan F1-Score 56,47% (*test set*) dan 53,90% (*offline set*), menegaskan bahwa ambiguitas visual kelas Warning merupakan tantangan inheren yang tidak bergantung pada platform inferensi.

Perbedaan performa antara test set dan offline set, khususnya pada kelas Warning, juga dapat dikaitkan dengan karakteristik konfigurasi Model A yang digunakan pada implementasi Raspberry Pi 5. Model A menggunakan fungsi Focal Loss dengan parameter  $\alpha = 0,25$  dan  $\gamma = 2,0$ . Menurut Tsung-Yi Lin, nilai  $\gamma$  yang lebih tinggi akan meningkatkan fokus pembelajaran pada sampel yang sulit diklasifikasikan (*hard samples*) dan mengurangi kontribusi sampel yang telah diklasifikasikan dengan benar (*easy samples*) [49]. Dalam penelitian ini, konfigurasi tersebut terbukti efektif untuk meningkatkan sensitivitas terhadap kelas Critical yang merupakan kelas dengan tingkat risiko tertinggi, ditunjukkan oleh nilai Recall Critical sebesar 96,81% pada offline set.

Namun demikian, peningkatan sensitivitas terhadap karakteristik cacat yang jelas juga dapat menyebabkan model membentuk batas keputusan yang lebih tegas antara kelas Critical dan non-Critical. Akibatnya, sampel kelas Warning yang memiliki karakteristik visual berada di antara Good dan Critical cenderung tidak memperoleh representasi fitur yang sekuat dua kelas lainnya. Kondisi ini tercermin pada *confusion matrix* gabungan, di mana sebanyak 127 sampel Warning diprediksi sebagai Good. Fenomena tersebut menyebabkan Recall kelas Warning turun menjadi 38,60%, meskipun Precision mencapai 88,89%.

Selain itu, penggunaan dropout sebesar 0,3 dan dense head berukuran 128 neuron menghasilkan model yang relatif sederhana dibandingkan Model B yang menggunakan 256 neuron. Konfigurasi ini memberikan kemampuan generalisasi yang lebih baik dan terbukti menghasilkan *Validation Accuracy* tertinggi sebesar 81,95% selama proses pelatihan. Akan tetapi, kapasitas representasi fitur yang lebih ringkas dapat membatasi kemampuan model dalam mempelajari variasi visual yang sangat halus pada kelas Warning. Kelas Warning sendiri memiliki karakteristik yang bersifat *fine-grained*, yaitu perbedaan visual yang tipis terhadap kelas Good, seperti ketidaksempurnaan fillet solder, distribusi timah yang kurang merata, atau cacat estetika ringan [50]. Pada kondisi implementasi raspberry pi, variasi pencahayaan dan refleksi permukaan solder semakin memperbesar kemiripan visual kedua kelas tersebut sehingga meningkatkan peluang terjadinya misklasifikasi.

Proses fine-tuning dengan membuka kembali 30 lapisan teratas MobileNetV2 dan menggunakan learning rate yang sangat kecil ( $1 \times 10^{-5}$ ) berhasil meningkatkan kemampuan ekstraksi fitur spesifik terhadap solder joint, yang ditunjukkan oleh peningkatan *Validation Accuracy* dari 75,94% pada akhir fase pertama menjadi 81,95% pada fase kedua. Hasil implementasi pada Raspberry Pi 5 menunjukkan bahwa strategi tersebut sangat efektif dalam mempertahankan kemampuan deteksi kelas Critical dan Good, yang masing-masing memperoleh Recall sebesar 96,81% dan 94,02%. Namun demikian, performa Recall Warning yang lebih rendah menunjukkan bahwa fitur visual kelas Warning masih memiliki tingkat tumpang

tindih (*feature overlap*) yang tinggi dengan kelas Good sehingga belum dapat dipisahkan secara konsisten oleh model [51].

Secara keseluruhan, hasil ini menunjukkan bahwa konfigurasi Model A cenderung menghasilkan model yang lebih konservatif terhadap prediksi Warning. Model lebih sering mengklasifikasikan sampel ambigu ke kelas Good dibandingkan memberikan prediksi Warning yang salah. Huang dkk. (2016) menjelaskan bahwa klasifikasi dengan perbedaan visual yang sangat halus menyebabkan terjadinya *feature overlap* antar kelas sehingga batas keputusan CNN menjadi sulit dipisahkan secara konsisten [50]. Hal ini tercermin dari kombinasi Precision kelas Warning yang tinggi (88,89%) dan Recall Warning yang rendah (38,60%). Dengan kata lain, ketika model memutuskan suatu titik solder termasuk kategori Warning, prediksi tersebut umumnya benar, tetapi masih banyak sampel Warning yang terlewat dan dikategorikan sebagai Good. Fenomena ini mengindikasikan bahwa keterbatasan utama sistem bukan terletak pada proses implementasi pada Raspberry Pi 5, melainkan pada tingkat kemiripan visual antara kelas Good dan Warning yang masih sulit dipisahkan oleh representasi fitur yang dipelajari model CNN.

#### 4.5 Pengujian Waktu inferensi

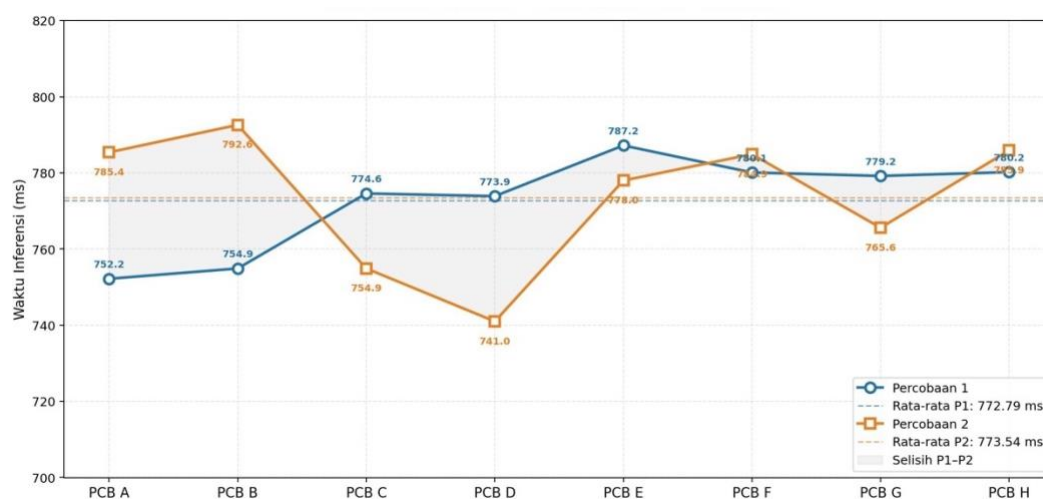
Pengujian waktu inferensi dilakukan untuk mengukur kemampuan sistem dalam melakukan proses klasifikasi kondisi PCB menggunakan model yang dijalankan pada Raspberry Pi 5. Waktu inferensi didefinisikan sebagai durasi yang dibutuhkan sejak citra hasil akuisisi diterima oleh model hingga diperoleh hasil prediksi klasifikasi. Pengukuran dilakukan terhadap delapan sampel PCB dengan dua kali percobaan untuk setiap sampel, hasil inferensi dijabarkan pada Tabel 4.38.

**Tabel 4.38** Data Waktu Inferensi

Objek	Percobaan 1 (ms)	Percobaan 2 (ms)
PCB A	752,2	785,4
PCB B	754,9	792,6
PCB C	774,6	754,9
PCB D	773,9	741,0
PCB E	787,2	778,0
PCB F	780,1	784,9

PCB G	779,2	765,6
PCB H	780,2	785,9

Berdasarkan hasil pengujian, waktu inferensi yang diperoleh berada pada rentang 741,0 ms hingga 792,6 ms. Nilai rata-rata keseluruhan dari 16 pengukuran adalah sebesar 773,16 ms dengan standar deviasi 15,02 ms. Standar deviasi yang relatif kecil menunjukkan bahwa model memiliki performa yang stabil dan konsisten dalam memproses citra PCB. Selain itu, tidak ditemukan perbedaan yang signifikan antara percobaan pertama dan percobaan kedua. Rata-rata waktu inferensi pada percobaan pertama tercatat sebesar 772,79 ms, sedangkan pada percobaan kedua sebesar 773,54 ms. Selisih kedua nilai tersebut hanya sekitar 0,75 ms sehingga dapat diabaikan secara praktis. Perbandingan waktu inferensi antara percobaan 1 dan percobaan 2 disajikan pada Gambar 4.28.



**Gambar 4.28** Tren Waktu Inverensi Percobaan 1 vs Percobaan 2

Variasi waktu inferensi yang muncul antar sampel PCB kemungkinan disebabkan oleh perbedaan kompleksitas fitur visual yang diproses oleh model. Pada beberapa sampel, model memerlukan waktu sedikit lebih lama untuk melakukan ekstraksi fitur dan proses klasifikasi, terutama ketika terdapat variasi pencahayaan, posisi komponen, maupun karakteristik pola solder yang lebih kompleks. Meskipun demikian, rentang variasi yang diperoleh masih tergolong kecil jika dibandingkan dengan rata-rata waktu inferensi secara keseluruhan.

## 4.6 Pengujian Protokol Komunikasi Modbus TCP/IP

Pengujian komunikasi Modbus TCP dilakukan untuk mengevaluasi kinerja pertukaran data antara Raspberry Pi 5 sebagai Modbus TCP Client (Master) dan WT32-ETH01 sebagai Modbus TCP Server (Slave). Analisis dilakukan berdasarkan hasil perekaman paket jaringan menggunakan tcpdump selama satu siklus deteksi mulai dari start hingga keputusan akhir selama 26,88 detik. Parameter yang dievaluasi meliputi proses pembentukan koneksi TCP, mekanisme polling data, operasi penulisan data, latensi komunikasi, dan struktur paket .

```
bismillah@TugasAkhir:~$ sudo tcpdump -i eth0 host 192.168.2.177 and port 502 -w ~/log_modbus.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2768 packets captured
2768 packets received by filter
0 packets dropped by kernel
bismillah@TugasAkhir:~$
```

**Gambar 4.29** Program TCPDUMP untuk merekam paket data

### 4.6.1 Pembentukan Koneksi TCP (*Three Way Handshake*)

Sebelum proses pertukaran data berlangsung, Raspberry Pi dan WT32-ETH01 terlebih dahulu membentuk koneksi TCP melalui mekanisme Three-Way Handshake. Berdasarkan hasil capture tcpdump, proses ini terekam pada frame 1 hingga frame 3. Raspberry Pi menginisiasi komunikasi dengan mengirimkan paket SYN pada frame 1, yang kemudian direspons oleh WT32-ETH01 menggunakan paket SYN-ACK pada frame 2, dan diselesaikan dengan pengiriman paket ACK dari Raspberry Pi pada frame 3.

No.	Time	Source	Destination	Protocol	Length	Info
1		192.168.2.100	192.168.2.177	TCP	74	38064 → 502 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1455819275 TSecr=0 WS=128
2	0.000531	192.168.2.177	192.168.2.100	TCP	62	502 → 38064 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1436 SACK_PERM
3	0.000027	192.168.2.100	192.168.2.177	TCP	54	38064 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.002378	192.168.2.177	192.168.2.100	TCP	60	502 → 45484 [FIN, ACK] Seq=1 Ack=1 Win=5557 Len=0
5	0.000032	192.168.2.100	192.168.2.177	TCP	54	45484 → 502 [ACK] Seq=1 Ack=2 Win=64071 Len=0

**Gambar 4.30** Inisiasi Komunikasi Dengan SYN dan ACK

Nilai Round Trip Time (RTT) yang diperoleh sebesar 0,531 ms dihitung dari WT32-ETH merespon paket SYN dari Raspberry Pi menunjukkan bahwa jaringan lokal memiliki kualitas koneksi yang sangat baik dengan delay transmisi yang sangat rendah.

2767	0.000042	192.168.2.100	192.168.2.177	TCP	54	38064 → 502 [FIN, ACK] Seq=16372 Ack=14353 Win=64071 Len=0
2768	0.000425	192.168.2.177	192.168.2.100	TCP	60	502 → 38064 [ACK] Seq=14353 Ack=16373 Win=5231 Len=0

**Gambar 4.31** Terminasi Komunikasi Dengan FIN

Keberhasilan pembentukan koneksi tersebut menjadi dasar penting bagi komunikasi Modbus TCP karena seluruh transaksi pembacaan maupun penulisan data dilakukan melalui sesi TCP yang sama. Selama pengujian berlangsung tidak

ditemukan retransmission, reset connection, maupun timeout sehingga dapat disimpulkan bahwa koneksi komunikasi memiliki tingkat reliabilitas yang sangat tinggi..

#### 4.6.2 Aktivitas Polling Data

Aktivitas komunikasi selama pengujian didominasi oleh mekanisme polling yang dilakukan secara periodik oleh Raspberry Pi. Polling pertama mulai terdeteksi pada frame 15 dan selanjutnya berlangsung secara kontinu hingga akhir sesi pengujian sesuai pada Gambar 4.32. Pola polling terdiri atas dua jenis fungsi utama, yaitu Function Code 02 (Read Discrete Inputs) dan Function Code 04 (Read Input Registers).

No.	Time	Source	Destination	Protocol	Length/Info
829	0.000086	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 407; Unit: 1, Func: 4: Read Input Registers
830	0.005951	192.168.2.177	192.168.2.100	Modbus/TCP	65 Response: Trans: 407; Unit: 1, Func: 4: Read Input Registers
831	0.020183	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 408; Unit: 1, Func: 2: Read Discrete Inputs
832	0.010809	192.168.2.177	192.168.2.100	Modbus/TCP	64 Response: Trans: 408; Unit: 1, Func: 2: Read Discrete Inputs
833	0.000207	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 409; Unit: 1, Func: 4: Read Input Registers
834	0.005310	192.168.2.177	192.168.2.100	Modbus/TCP	65 Response: Trans: 409; Unit: 1, Func: 4: Read Input Registers
835	0.020175	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 410; Unit: 1, Func: 2: Read Discrete Inputs
836	0.015773	192.168.2.177	192.168.2.100	Modbus/TCP	64 Response: Trans: 410; Unit: 1, Func: 2: Read Discrete Inputs
837	0.000054	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 411; Unit: 1, Func: 4: Read Input Registers
838	0.005988	192.168.2.177	192.168.2.100	Modbus/TCP	65 Response: Trans: 411; Unit: 1, Func: 4: Read Input Registers
839	0.020156	192.168.2.100	192.168.2.177	Modbus/TCP	66 Query: Trans: 412; Unit: 1, Func: 2: Read Discrete Inputs
840	0.010806	192.168.2.177	192.168.2.100	Modbus/TCP	64 Response: Trans: 412; Unit: 1, Func: 2: Read Discrete Inputs

**Gambar 4.32** Aktifitas Pooling Data menggunakan FC04 dan FC02

Secara keseluruhan tercatat sebanyak 668 transaksi FC02 dan 668 transaksi FC04. Jumlah transaksi yang sama menunjukkan bahwa kedua fungsi dijalankan secara berpasangan dalam setiap siklus polling. FC02 digunakan untuk membaca status input digital seperti sensor proximity dan ES, sedangkan FC04 digunakan untuk membaca data numerik yang tersimpan pada register input WT32-ETH01.

Dengan total 1.336 transaksi pembacaan selama 26,88 detik, diperoleh frekuensi polling efektif sebesar 49,7 transaksi per detik atau sangat mendekati target sistem sebesar 50 Hz. Hasil ini menunjukkan bahwa mekanisme polling yang dirancang pada Raspberry Pi mampu mempertahankan interval komunikasi secara konsisten sehingga informasi kondisi sistem dapat diperbarui hampir secara real-time.

Analisis struktur paket juga menunjukkan bahwa seluruh frame menggunakan Protocol Identifier bernilai 0x0000 dan Unit Identifier bernilai 0x01. Hal tersebut membuktikan bahwa implementasi protokol pada kedua perangkat telah sesuai dengan standar Modbus TCP.

### 4.6.3 Latensi Komunikasi

Kecepatan komunikasi dapat dievaluasi melalui parameter *service response time* atau latensi. Latensi dihitung berdasarkan selisih waktu antara paket *query* yang dikirimkan oleh Raspberry Pi dan paket *response* yang diterima kembali dari WT32-ETH01.

Hasil pengujian menunjukkan bahwa transaksi FC04 memiliki rata-rata latensi sebesar 5,95 ms dengan nilai minimum 2,06 ms dan maksimum 35,90 ms. Nilai tersebut merupakan performa terbaik di antara seluruh fungsi yang diuji. Rendahnya waktu tanggap ini menunjukkan bahwa proses pembacaan register pada WT32-ETH01 dapat dilakukan secara cepat karena data telah tersedia pada area memori internal sehingga tidak memerlukan proses akuisisi tambahan.

Sementara itu, transaksi FC02 memiliki rata-rata latensi sebesar 13,57 ms dengan rentang nilai antara 5,24 ms hingga 20,52 ms. Nilai yang lebih tinggi dibandingkan FC04 menunjukkan adanya tambahan waktu pemrosesan ketika WT32-ETH01 harus membaca kondisi GPIO fisik sebelum membentuk paket respons. Meskipun demikian, rata-rata latensi yang masih berada di bawah 15 ms menunjukkan bahwa proses pembacaan status sensor tetap berlangsung dengan cepat dan memenuhi kebutuhan sistem *real-time*.

Nilai standar deviasi FC02 sebesar 2,66 ms dan FC04 sebesar 1,34 ms menunjukkan bahwa variasi latensi relatif kecil. Dengan demikian, komunikasi dapat dikategorikan stabil karena sebagian besar transaksi memiliki waktu respons yang konsisten sepanjang pengujian.

### 4.6.4 Operasi Pengendalian dan Penulisan Data

Selain aktivitas *monitoring*, sistem juga melakukan operasi pengendalian melalui beberapa fungsi *write*. Aktivitas ini dimulai sejak awal komunikasi ketika Raspberry Pi melakukan *inisialisasi* kondisi output menggunakan *Function Code 05 (Write Single Coil)*. Transaksi tersebut terdeteksi pada frame 6 hingga frame 14 yang digunakan untuk mengatur kondisi awal coil sebelum proses inspeksi berjalan.

Secara keseluruhan tercatat 20 transaksi FC05 dengan rata-rata latensi sebesar 11,90 ms. Nilai tersebut menunjukkan bahwa pengendalian aktuator digital dapat dilakukan dengan cepat dan responsif. Aktivitas ini muncul kembali pada

beberapa waktu tertentu selama pengujian yang mengindikasikan adanya perubahan status aktuator berdasarkan kondisi hasil inspeksi. Transaksi FC05 ditunjukkan pada Gambar 4.33.

No.	Time	Source	Destination	Protocol	Length	Info
9	0.007250	192.168.2.100	192.168.2.177	Modbus/TCP	66	Query: Trans: 2; Unit: 1, Func: 5: Write Single Coil
10	0.013321	192.168.2.177	192.168.2.100	Modbus/TCP	66	Response: Trans: 2; Unit: 1, Func: 5: Write Single Coil
11	0.000129	192.168.2.100	192.168.2.177	Modbus/TCP	66	Query: Trans: 3; Unit: 1, Func: 5: Write Single Coil
12	0.005887	192.168.2.177	192.168.2.100	Modbus/TCP	66	Response: Trans: 3; Unit: 1, Func: 5: Write Single Coil
13	0.000094	192.168.2.100	192.168.2.177	Modbus/TCP	66	Query: Trans: 4; Unit: 1, Func: 5: Write Single Coil
14	0.005911	192.168.2.177	192.168.2.100	Modbus/TCP	66	Response: Trans: 4; Unit: 1, Func: 5: Write Single Coil

**Gambar 4.33** Paket Data Untuk Mengontrol Coil Menggunakan FC05

Selain FC05, sistem juga melakukan enam transaksi FC06 (*Write Single Register*) ditunjukkan pada Gambar 4.32 dengan rata-rata latensi sebesar 8,11 ms. Operasi ini digunakan untuk mengubah nilai register kontrol yang berkaitan dengan proses *stop-and-go* konveyor. Waktu respons yang rendah menunjukkan bahwa parameter kendali dapat dikirim dan diterapkan hampir secara langsung.

No.	Time	Source	Destination	Protocol	Length	Info
1493	0.000754	192.168.2.100	192.168.2.177	Modbus/TCP	66	Query: Trans: 735; Unit: 1, Func: 6: Write Single Register
1494	0.005255	192.168.2.177	192.168.2.100	Modbus/TCP	66	Response: Trans: 735; Unit: 1, Func: 6: Write Single Register

**Gambar 4.34** Paket Data Untuk Mengubah Register Menggunakan FC06

Pada pertengahan sesi pengujian juga terdeteksi satu transaksi FC16 (*Write Multiple Registers*) yang digunakan untuk mengirimkan tujuh register secara simultan. Meskipun ukuran *payload* lebih besar dibandingkan fungsi *write* lainnya, latensi yang diperoleh hanya sebesar 15,28 ms. Hasil ini menunjukkan bahwa implementasi Modbus TCP tetap mampu mempertahankan performa yang baik meskipun menangani transfer data yang lebih kompleks. Transaksi FC16 dicontohkan pada Gambar 4.35.

No.	Time	Source	Destination	Protocol	Length	Info
1499	0.015988	192.168.2.100	192.168.2.177	Modbus/TCP	81	Query: Trans: 738; Unit: 1, Func: 16: Write Multiple Registers
1500	0.015279	192.168.2.177	192.168.2.100	Modbus/TCP	66	Response: Trans: 738; Unit: 1, Func: 16: Write Multiple Registers

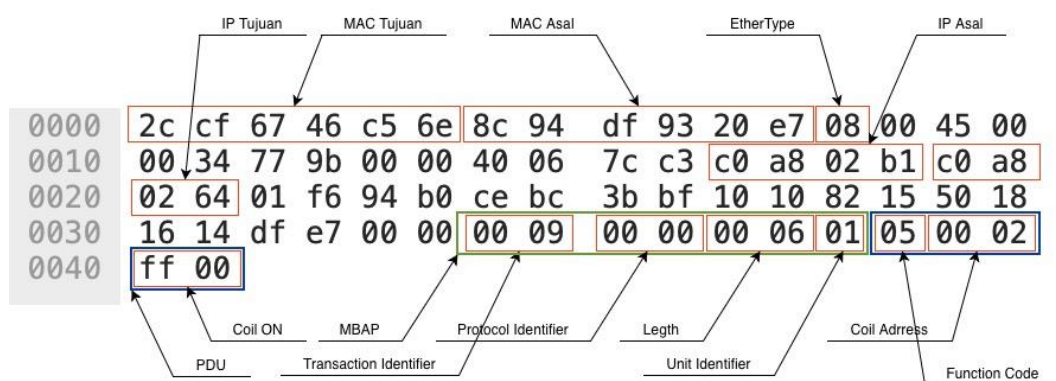
**Gambar 4.35** Paket Data Untuk Mengubah Beberapa Register Bersamaan dengan FC16

Seluruh operasi *write* berlangsung di sela-sela aktivitas *polling* tanpa menyebabkan penurunan performa komunikasi maupun kegagalan transaksi. Hasil ini menunjukkan bahwa implementasi sistem pada WT32-ETH01 berhasil memanfaatkan arsitektur dual-core WT32-ETH01 untuk memisahkan tugas komunikasi Modbus TCP dan tugas kontrol perangkat. Dengan mekanisme tersebut, proses pembacaan dan penulisan data dapat berjalan secara simultan tanpa menimbulkan konflik sumber daya maupun peningkatan latensi yang signifikan.

#### 4.6.5 Struktur Frame Modbus TCP/IP

Untuk memastikan implementasi komunikasi Modbus TCP telah sesuai dengan spesifikasi protokol, dilakukan analisis terhadap paket 24 yakni respons *Function Code 05 (Write Single Coil)*. Paket yang dianalisis merupakan respons dari WTP32-ETH01 terhadap perintah pengaktifan coil yang dikirimkan oleh Raspberry Pi.

Berdasarkan hasil tangkapan struktur paket data yang ditunjukkan pada Gambar 4.36, paket respons tersebut memiliki panjang total 64-byte dan terdiri atas empat lapisan protokol, yaitu Ethernet, IPv4, TCP, serta Modbus TCP.



**Gambar 4.36** Struktur Paket Data 24

Pada lapisan Ethernet, alamat MAC sumber teridentifikasi sebagai WT32-ETH01 (8C:94:DF:93:20:E7), sedangkan alamat MAC tujuan merupakan Raspberry Pi (2C:CF:67:46:C5:6E). Nilai EtherType sebesar 0x0800 menunjukkan bahwa *payload* yang dibawa merupakan paket IPv4.

Pada lapisan jaringan (IPv4), alamat IP sumber adalah C0.A8.02.B1 (192.168.2.177) yang merupakan alamat WT32-ETH01, sedangkan alamat tujuan adalah C0.A8.02.64 (192.168.2.100) yang merupakan Raspberry Pi. Nilai Time To Live (TTL) sebesar 64 menunjukkan bahwa paket dikirim langsung melalui jaringan lokal tanpa melewati banyak perangkat perantara.

Lapisan transport menggunakan protokol TCP dengan port sumber 502 yang merupakan port standar Modbus TCP dan port tujuan 38064 yang digunakan oleh aplikasi client pada Raspberry Pi. Flag TCP yang terdeteksi adalah PSH dan ACK. Penggunaan flag PSH menunjukkan bahwa WT32-ETH01 menginstruksikan sistem operasi untuk segera mengirimkan data ke buffer aplikasi penerima tanpa

menunggu akumulasi data tambahan. Mekanisme ini umum digunakan pada komunikasi Modbus TCP karena ukuran data yang relatif kecil dan membutuhkan respons cepat.

Bagian utama komunikasi Modbus TCP terdapat pada MBAP (*Modbus Application Protocol*) Header dan PDU (*Protocol Data Unit*). Hasil dekode paket ditunjukkan pada Tabel 4.39 dan Tabel 4.40.

**Tabel 4.39** Struktur MBAP Paket 24

<b>Field</b>	<b>Nilai Hex</b>	<b>Keterangan</b>
Transaction Identifier	00 09	ID Transaksi
Protocol Identifier	00 00	Modbus TCP
Length	00 06	Panjang data
Unit Identifier	01	Alamat Slave

**Tabel 4.40** Struktur PDU Paket 24

<b>Field</b>	<b>Nilai Hex</b>	<b>Keterangan</b>
Function Code	05	Write Single Coil
Reference Number	00 02	Coil 2
Output Value	FF 00	ON

Nilai Protocol Identifier sebesar 0x0000 menunjukkan bahwa paket telah mengikuti standar Modbus TCP. Sementara itu, Unit Identifier bernilai 0x01 yang mengidentifikasi perangkat WT32-ETH01 sebagai slave tujuan komunikasi.

Pada bagian PDU terdeteksi Function Code 0x05 yang menunjukkan operasi Write Single Coil. Alamat coil yang dituju adalah coil nomor 2 dengan nilai FF00 yang menurut spesifikasi Modbus merepresentasikan kondisi ON. Sesuai karakteristik Function Code 05, paket respons mengembalikan (echo) seluruh informasi yang terdapat pada request sebagai bentuk konfirmasi bahwa perintah telah berhasil dieksekusi.

Hasil analisis tcpdump menunjukkan bahwa paket respons diterima dalam waktu 5,937 ms setelah request dikirimkan. Tidak ditemukan exception code maupun malformed packet pada transaksi tersebut. Dengan demikian dapat

disimpulkan bahwa implementasi Function Code 05 pada WT32-ETH01 telah berjalan sesuai dengan spesifikasi Modbus TCP dan mampu memberikan respons secara cepat serta akurat.

#### 4.7 Pengujian Hardware

Pengujian hardware dilakukan untuk memastikan seluruh komponen utama sistem inspeksi PCB bekerja sesuai dengan spesifikasi perancangan. Pengujian meliputi karakteristik mekanik conveyor, konsumsi daya motor DC, performa catu daya (power supply unit), serta karakteristik aktuator servo yang digunakan sebagai mekanisme pemilah PCB

##### 4.7.1 Pengujian Kecepatan Conveyor

Pengujian kecepatan conveyor dilakukan dengan mengukur waktu yang dibutuhkan conveyor untuk memindahkan objek sejauh 50 cm. Hasil pengukuran menunjukkan bahwa conveyor mampu menempuh jarak tersebut dalam waktu 20,22 detik. Kecepatan Conveyor dapat dihitung

$$v = \frac{50}{20,22} = 2,47 \text{ cm/s}$$

Hasil tersebut menunjukkan bahwa conveyor bergerak relatif lambat sehingga memberikan waktu yang cukup bagi sistem untuk melakukan akuisisi citra, proses inferensi kecerdasan buatan, dan pengambilan keputusan sebelum PCB mencapai area pemilah (*ejector*).

Hasil tersebut menunjukkan bahwa conveyor bergerak relatif lambat sehingga memberikan waktu yang cukup bagi sistem untuk melakukan akuisisi citra, proses inferensi kecerdasan buatan, dan pengambilan keputusan sebelum PCB mencapai area pemilah (*ejector*).

##### 4.7.2 Pengujian Tegangan dan Arus Motor

Pengujian dilakukan untuk mengetahui karakteristik konsumsi daya motor conveyor saat beroperasi pada kondisi normal. Hasil pengukuran pada Gambar 4.35 dan Gambar 4.36 menunjukkan bahwa motor menerima tegangan sebesar 11,98 V dengan arus operasi sebesar 0,098 A.



**Gambar 4.37** Tegangan Motor Saat Aktif



**Gambar 4.38** Arus Motor Saat Aktif

### 4.7.3 Pengujian Power Supply

Pengujian dilakukan untuk mengetahui karakteristik konsumsi daya motor conveyor saat beroperasi pada kondisi normal. Hasil pengukuran pada Gambar 4.39, Gambar 4.40 dan Gambar 4.41 menunjukkan bahwa power supply menghasilkan tegangan nominal sebesar 12,00 V dengan arus nominal 0,178 A dan maksimal sebesar 0,775 A.



**Gambar 4.39** Arus Nominal Catu Daya



**Gambar 4.40** Arus Maksimal Catu Daya



**Gambar 4.41** Tegangan Catu Daya

Daya pada kondisi tanpa beban dihitung sebagai berikut

$$P_{nominal} = 12 \times 0,178 = 2,14 \text{ Watt}$$

Sedangkan daya maksimum ketika semua beban menyala

$$P_{max} = 12 \times 0,776 = 9,31 \text{ Watt}$$

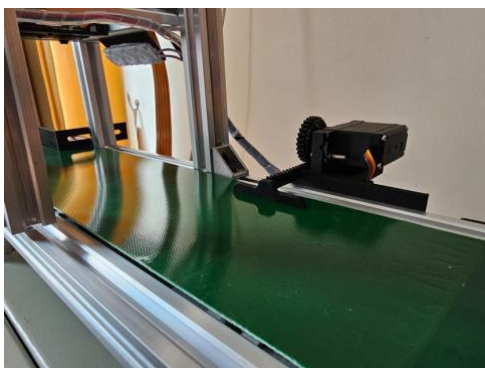
Perbandingan antara kondisi normal dan kondisi puncak menunjukkan bahwa konsumsi arus meningkat ketika beberapa aktuator bekerja secara bersamaan, terutama saat servo melakukan proses pemilahan PCB. Namun demikian, tegangan keluaran tetap stabil pada 12,00 V sehingga menunjukkan bahwa kapasitas catu daya masih mencukupi untuk mendukung seluruh komponen sistem.

Stabilitas tegangan ini penting karena fluktuasi tegangan dapat memengaruhi performa WT32-ETH01, Raspberry Pi, maupun aktuator yang digunakan dalam proses inspeksi.

#### 4.7.4 Pengujian Aktuator Servo Ejecto

Aktuator servo digunakan sebagai mekanisme pemilah (*ejector*) untuk mengarahkan PCB sesuai hasil klasifikasi sistem inspeksi. Pada penelitian ini servo dikendalikan menggunakan metode berbasis waktu (*time-based control*), yaitu dengan menentukan durasi aktivasi servo untuk menghasilkan posisi yang diinginkan.

Hasil pengujian pada Gambar 4.42 dan Gambar 4.43 menunjukkan bahwa servo mampu merespons perintah kendali secara stabil dan dapat mencapai titik maksimum dan minimum juga tidak ditemukan kegagalan aktuasi selama proses pengujian berlangsung. Hal ini menunjukkan bahwa mekanisme pemilah telah bekerja sesuai dengan kebutuhan sistem inspeksi PCB otomatis.



Gambar 4.42 Posisi Minimum Ejector

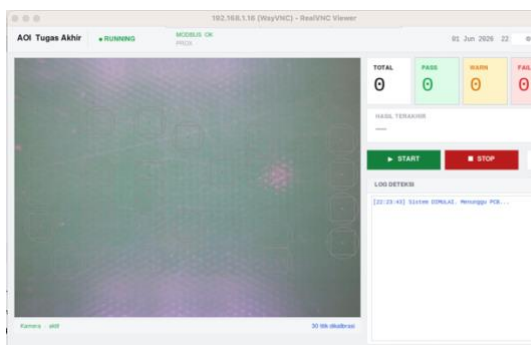


Gambar 4.43 Posisi Maksimum Ejector

#### 4.8 Integrasi Sistem

Pengujian integrasi sistem dilakukan untuk memastikan seluruh subsistem yang terdiri dari HMI, WT32-ETH01, Raspberry Pi 5, kamera, conveyor, aktuator

dan ejector sesuai dengan skenario operasi yang dirancang. Pada kondisi awal ,sistem berada dalam keadaan *idle* yang ditandai dengan aktifnya *pilot lamp* berwarna kuning pada panel kontrol. Pada kondisi ini motor conveyor, kipas pendingin, dan aktuator ejector belum beroperasi. HMI juga menampilkan status sistem siap dijalankan. Gambar 4.44 menunjukkan kondisi sistem pada mode *idle*.

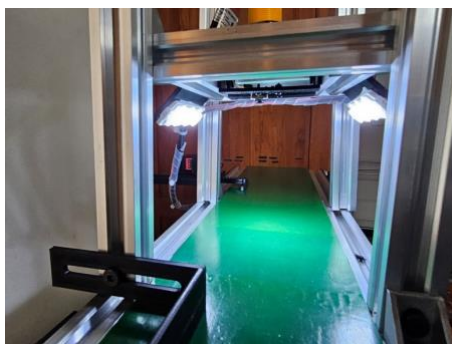


**Gambar 4.44** Tampilan HMI saat idle



**Gambar 4.45** Kondisi Pilot Lamp Saat Sistem Idle

Ketika tombol *Start* ditekan melalui HMI, sistem memasuki mode operasi. Motor conveyor, kipas pendingin, dan lampu penerangan aktif secara bersamaan, sementara *pilot lamp* hijau menyala sebagai indikator bahwa sistem sedang beroperasi. Selanjutnya sensor proximity mendeteksi keberadaan PCB dan conveyor berhenti sementara menggunakan metode *stop-and-go* untuk proses akuisisi citra dan inferensi model. Gambar 4.46 menunjukkan kondisi sistem saat proses operasi berlangsung.



**Gambar 4.46** Kondisi Sistem Saat Sistem Aktif

Hasil klasifikasi yang diperoleh dari model CNN digunakan sebagai dasar pengendalian aktuator ejector. Berdasarkan pengujian sebanyak tiga kali menggunakan delapan sampel PCB pada setiap pengujian, sistem mampu

menjalankan proses deteksi, komunikasi Modbus TCP, dan aktuasi ejector secara otomatis. Namun demikian, masih ditemukan beberapa kasus kegagalan deteksi yang disebabkan oleh posisi PCB yang tidak selalu berhenti tepat pada *Region of Interest* (ROI) kamera.



Gambar 4.47 Tampilan HMI saat proses klasifikasi

Pengujian juga dilakukan terhadap kondisi kegagalan sistem dengan menghapus berkas model CNN yang digunakan untuk proses inferensi. Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi kondisi tersebut dan menampilkan pesan kesalahan pada *log* aplikasi. Selain itu, *pilot lamp* merah menyala sebagai indikator terjadinya gangguan sistem. Kondisi ini menunjukkan bahwa mekanisme *error handling* telah berjalan dengan baik. Gambar 4.49 memperlihatkan tampilan pesan kesalahan ketika model CNN gagal dimuat.



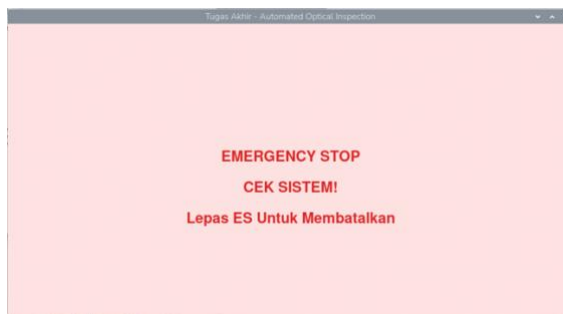
Gambar 4.48 Kondisi Pilot Lamp Ketika Sistem Error



Gambar 4.49 Log Deteksi Ketika Sistem Error

Pengujian fungsi keselamatan dilakukan menggunakan tombol *Emergency Stop* (ES). Ketika tombol ES ditekan, seluruh aktuator termasuk motor conveyor, kipas pendingin, lampu penerangan, dan ejector langsung berhenti beroperasi. Pada saat yang sama HMI menampilkan pesan peringatan bahwa sistem berada dalam

kondisi darurat. Hasil ini menunjukkan bahwa fungsi keselamatan telah bekerja sesuai dengan tujuan perancangan. Gambar 4.50 menunjukkan tampilan peringatan pada HMI ketika tombol Emergency Stop diaktifkan.



**Gambar 4.50** Tampilan HMI Ketika ES aktif

Selama pengujian integrasi, sistem berhasil menjalankan proses deteksi, komunikasi Modbus TCP, dan aktuasi ejector secara otomatis. Pengujian dilakukan sebanyak tiga kali menggunakan delapan sampel PCB pada setiap percobaan. Secara umum, sistem mampu menyelesaikan seluruh siklus inspeksi mulai dari deteksi objek, akuisisi citra, inferensi model, hingga proses pemilahan PCB sesuai hasil klasifikasi.

Meskipun demikian, masih ditemukan beberapa kasus salah deteksi (*misclassification*) selama pengujian berlangsung. Berdasarkan hasil observasi, kesalahan tersebut tidak disebabkan oleh kegagalan model CNN maupun komunikasi Modbus TCP, melainkan karena posisi PCB tidak selalu berhenti tepat pada *Region of Interest* (ROI) yang telah ditentukan. Kondisi ini terjadi karena sistem conveyor menggunakan metode *stop-and-go* tanpa dilengkapi mekanisme *stopper* fisik sehingga posisi akhir PCB dipengaruhi oleh variasi waktu deteksi sensor proximity, interval polling sebesar 20 ms, serta karakteristik inersia motor conveyor ketika menerima perintah berhenti.

Akibat pergeseran posisi tersebut, sebagian area solder yang menjadi objek inspeksi terkadang tidak berada tepat di tengah ROI kamera. Perubahan posisi ini menyebabkan citra yang diperoleh memiliki sudut pandang dan komposisi objek yang sedikit berbeda dibandingkan data pelatihan model, sehingga berpotensi menurunkan akurasi klasifikasi. Pada beberapa kasus, PCB yang seharusnya

teridentifikasi sebagai kategori tertentu dapat menghasilkan prediksi yang berbeda akibat pergeseran posisi tersebut.

Namun demikian, ketika PCB berhasil berhenti pada posisi yang sesuai dengan ROI, sistem mampu melakukan klasifikasi dan pemilahan menggunakan aktuator ejector dengan baik. Hasil ini menunjukkan bahwa performa model CNN dan sistem komunikasi yang digunakan telah bekerja sesuai perancangan, sedangkan kendala utama yang masih ditemukan berasal dari aspek mekanis sistem conveyor. Oleh karena itu, pengembangan selanjutnya dapat dilakukan dengan menambahkan mekanisme *stopper*, sensor posisi tambahan, atau metode kendali posisi yang lebih presisi untuk meningkatkan konsistensi posisi PCB pada area inspeksi.