

BAB IV

PENGUJIAN DAN ANALISA

Bab ini menyajikan hasil pengujian persepsi, *Visual Servoing*, dan integrasi pemetikan. Analisis memakai data yang telah direkam selama pengembangan sistem. Lima pengulangan pada cabai yang sama menjadi pengujian utama pengendali, sedangkan pengujian lainnya digunakan untuk menjelaskan kemampuan dan keterbatasan sistem.

4.1 Analisa Hasil *Training* YOLOv8

Proses pelatihan pada penelitian ini dilakukan terhadap dua konfigurasi model YOLOv8n-Seg, yaitu Model A dan Model B. Kedua model menggunakan *optimizer* Stochastic Gradient Descent (SGD), namun memiliki perbedaan pada beberapa *hyperparameter* pelatihan. Model A menggunakan *learning rate* awal sebesar 0,01, *batch size* 16, target pelatihan 120 *epoch*, dan *patience* 35 *epoch*. Sementara itu, Model B merupakan hasil penyesuaian *hyperparameter* dengan *learning rate* awal sebesar 0,0003, *batch size* 32, target pelatihan 120 *epoch*, dan *patience* 25 *epoch*. Model A memang dikonfigurasi sebagai YOLOv8n-Seg dengan SGD, *batch size* 16, target 120 *epoch* dan *patience* 35.

4.1.1 Hasil *Training* Model A

Model A dilatih menggunakan *optimizer* SGD dengan *batch size* 16 dan *initial learning rate* sebesar 0,01. Pelatihan dikonfigurasi untuk berlangsung maksimum 150 *epoch* dengan nilai *patience* sebesar 35. Meskipun demikian, proses pelatihan berhenti pada *epoch* ke-107 akibat mekanisme *Early stopping*.

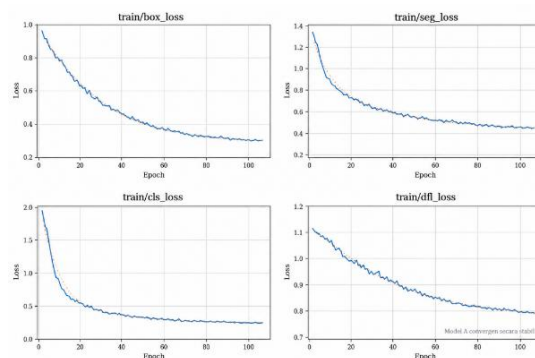
Tabel 4.1 Sample Log *Training loss* Model A

<i>Epoch</i>	<i>Train Box loss</i>	<i>Train Seg Loss</i>	<i>Train Class Loss</i>	<i>Train DFL Loss</i>
1	0,9599	1,3349	1,9585	1,1060
10	0,7484	0,8570	0,6814	0,9698
20	0,6144	0,7028	0,5286	0,9077
30	0,5450	0,6432	0,4503	0,8782

<i>Epoch</i>	<i>Train Box loss</i>	<i>Train Seg Loss</i>	<i>Train Class Loss</i>	<i>Train DFL Loss</i>
50	0,4696	0,5703	0,3810	0,8450
72	0,4085	0,5153	0,3298	0,8244
90	0,3699	0,4796	0,2943	0,8125
107	0,3324	0,4515	0,2659	0,8047

Berdasarkan **Tabel 4.1** , seluruh komponen *training loss* Model A menunjukkan penurunan yang konsisten sepanjang proses pelatihan. *Train Box loss* turun dari 0,9599 pada *epoch* pertama menjadi 0,3324 pada *epoch* ke-107 atau mengalami penurunan sebesar 65,37%. *Train Segmentation loss* turun dari 1,3349 menjadi 0,4515 atau sebesar 66,17%, sedangkan *Train Classification loss* mengalami penurunan terbesar, yaitu dari 1,9585 menjadi 0,2659 atau sebesar 86,42%. *Train DFL Loss* juga mengalami penurunan dari 1,1060 menjadi 0,8047 atau sebesar 27,24%.

Penurunan tersebut menunjukkan bahwa proses optimasi Model A berlangsung secara progresif. Penurunan *Box loss* menunjukkan semakin kecilnya kesalahan model dalam menentukan lokasi objek cabai, sedangkan penurunan *Segmentation loss* menunjukkan bahwa bentuk *mask* hasil prediksi semakin mendekati *mask ground truth*. Penurunan *Classification loss* menunjukkan bahwa model semakin mampu membedakan Cabai Matang dan Cabai Tidak Matang, sementara penurunan *DFL Loss* menunjukkan perbaikan pada proses regresi lokasi objek.



Gambar 4.1 Kurva *Training loss* Model A

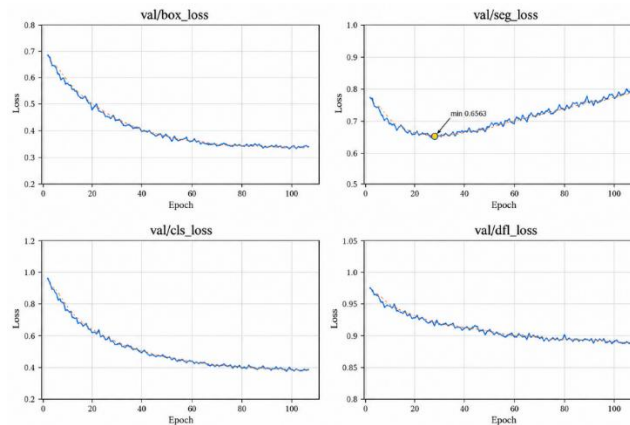
Gambar 4.1 memperlihatkan bahwa penurunan *loss* paling besar terjadi pada *epoch* awal hingga pertengahan proses *training*. Seiring bertambahnya *epoch*, kemiringan masing-masing kurva semakin kecil dan bergerak menuju kondisi yang lebih stabil. Pola tersebut menunjukkan bahwa sebagian besar karakteristik visual objek telah dipelajari pada tahap awal hingga pertengahan *training*, sedangkan *epoch* berikutnya memberikan perbaikan yang semakin kecil terhadap nilai *loss*.

Tabel 4.2 Sample Log Validation loss Model A

<i>Epoch</i>	<i>Val Box loss</i>	<i>Val Box Loss</i>	<i>Val Box Loss</i>	<i>Val Box Loss</i>
1	0,6843	0,7741	0,9407	0,9740
10	0,6030	0,7816	0,5460	0,9450
20	0,4962	0,7719	0,4756	0,9080
30	0,4640	0,6563	0,4415	0,9032
50	0,3930	0,7797	0,3802	0,8941
72	0,3672	0,7427	0,3607	0,8979
90	0,3471	0,7773	0,3450	0,8959
107	0,3401	0,7960	0,3386	0,8970

Pada data validasi, *Val Box loss* turun dari 0,6843 menjadi 0,3401 atau berkurang sebesar 50,30%, sedangkan *Val Classification loss* menurun dari 0,9407 menjadi 0,3386 atau sebesar 64,00%. *Val DFL Loss* juga mengalami penurunan dari 0,9740 menjadi 0,8970 atau sekitar 7,90%.

Pola berbeda terlihat pada *Val Segmentation loss*. Nilainya sebesar 0,7741 pada *epoch* pertama kemudian turun hingga mencapai nilai minimum 0,6563 pada *epoch* ke-30, atau mengalami perbaikan sebesar sekitar 15,21% dibandingkan kondisi awal. Setelah *epoch* tersebut, *Val Segmentation loss* kembali meningkat dan berfluktuasi hingga mencapai 0,7960 pada *epoch* ke-107. Dibandingkan nilai minimumnya, terjadi peningkatan *loss* sekitar 21,28%.



Gambar 4.2 Kurva *Validation loss* Model A

Gambar 4.2 menunjukkan bahwa kemampuan generalisasi pada komponen lokalisasi dan klasifikasi masih mengalami perbaikan selama proses pelatihan, yang ditunjukkan oleh penurunan *Val Box loss* dan *Val Classification loss*. Sebaliknya, *Val Segmentation loss* mencapai nilai minimum lebih awal dan kemudian kembali meningkat, sementara *Train Segmentation loss* masih terus menurun. Kondisi tersebut menunjukkan mulai terbentuknya kesenjangan antara optimasi pada data *training* dan data validasi pada cabang segmentasi.

Meskipun demikian, peningkatan *Val Segmentation loss* tidak disertai dengan penurunan drastis pada metrik *mask*. Oleh karena itu, kondisi tersebut lebih tepat menunjukkan bahwa kemampuan generalisasi segmentasi mulai mengalami plateau atau diminishing improvement, bukan menunjukkan terjadinya *overfitting* berat.

Meskipun Model A direncanakan untuk dilatih hingga 150 *epoch*, proses *training* berhenti pada *epoch* ke-107 akibat mekanisme *Early stopping*. *Checkpoint* terbaik keseluruhan Model A diperoleh pada *epoch* ke-72. Setelah *checkpoint* tersebut tidak terdapat peningkatan fitness yang mampu menghasilkan *checkpoint* baru selama 35 *epoch* sesuai nilai *patience*.

4.1.2 Hasil *Training* Model B

Model B merupakan konfigurasi hasil penyesuaian *hyperparameter* terhadap Model A. *Initial learning rate* diturunkan dari 0,01 menjadi 0,0003 atau sebesar **70%**, sedangkan *batch size* ditingkatkan dari 16 menjadi 32 atau sebesar **100%**.

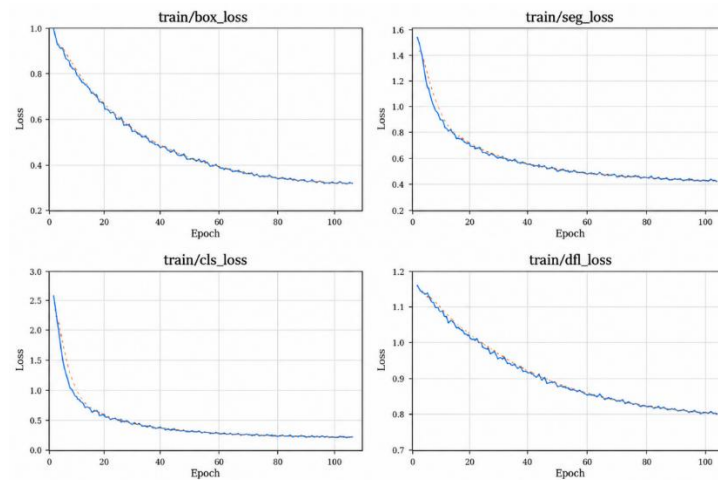
Selain itu, target maksimum *training* dikurangi dari 150 menjadi 120 *epoch* dan nilai *patience* dari 35 menjadi 25. Penyesuaian tersebut digunakan untuk mengevaluasi pengaruh konfigurasi *training* terhadap proses konvergensi dan kemampuan generalisasi model.

Tabel 4.3 Sample Log *Training loss* Model B

<i>Epoch</i>	<i>Train Box loss</i>	<i>Train Seg Loss</i>	<i>Train Class Loss</i>	<i>Train DFL Loss</i>
1	1,0016	1,5298	2,5799	1,1556
10	0,6255	0,7463	0,6335	0,9079
20	0,5330	0,6387	0,4866	0,8674
40	0,4486	0,5557	0,3836	0,8345
60	0,3904	0,5029	0,3261	0,8155
78	0,3474	0,4666	0,2892	0,8043
90	0,3213	0,4449	0,2695	0,7978
103	0,3001	0,4290	0,2551	0,7937

Berdasarkan **Tabel 4.3** , seluruh komponen *training loss* Model B menunjukkan penurunan secara konsisten. *Train Box loss* turun dari 1,0016 menjadi 0,3001 atau sebesar 70,04%, *Train Segmentation loss* dari 1,5298 menjadi 0,4290 atau sebesar 71,96%, dan *Train Classification loss* dari 2,5799 menjadi 0,2551 atau sebesar 90,11%. Sementara itu, *Train DFL Loss* mengalami penurunan dari 1,1556 menjadi 0,7937 atau sebesar 31,32%.

Persentase penurunan *training loss* Model B lebih besar dibandingkan Model A pada seluruh komponen. Hal tersebut menunjukkan bahwa konfigurasi Model B menghasilkan tingkat optimasi terhadap data *training* yang lebih tinggi. Namun, penurunan *training loss* yang lebih besar tidak secara langsung menunjukkan bahwa Model B memiliki generalisasi yang lebih baik, sehingga hasil tersebut perlu dianalisis bersama *validation loss* dan metrik segmentasi.



Gambar 4.3 Kurva *Training loss* Model B

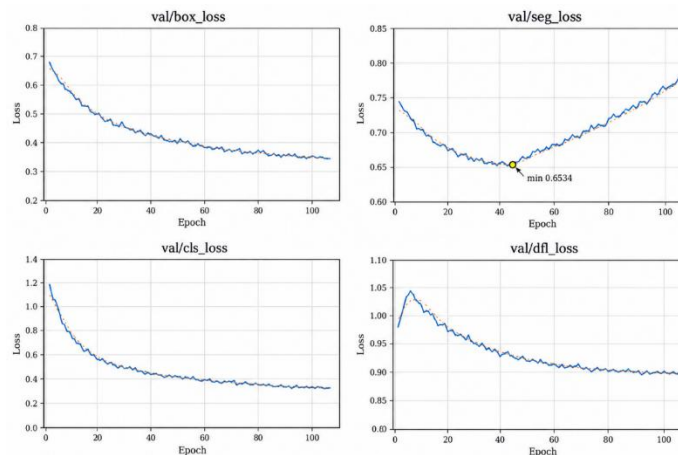
Gambar 4.3 memperlihatkan pola penurunan yang serupa dengan Model A, yaitu penurunan tajam pada *epoch* awal kemudian melandai ketika jumlah *epoch* bertambah. *Classification loss* mengalami penurunan terbesar, sedangkan DFL *Loss* memiliki perubahan yang lebih gradual. Menjelang akhir *training*, keempat kurva bergerak menuju kondisi yang lebih stabil sehingga menunjukkan bahwa Model B telah mendekati kondisi konvergen terhadap data *training*.

Tabel 4.4 Sample Log *Validation loss* Model B

<i>Epoch</i>	<i>Train Box loss</i>	<i>Train Seg Loss</i>	<i>Train Class Loss</i>	<i>Train DFL Loss</i>
1	0,6657	0,7428	1,1765	0,9801
10	0,5049	0,7328	0,5537	0,9064
20	0,4564	0,7088	0,4825	0,9048
40	0,3977	0,6874	0,4009	0,8958
60	0,3666	0,7548	0,3981	0,9007
78	0,3516	0,7405	0,3835	0,8951
90	0,3457	0,8130	0,3843	0,8987
103	0,3433	0,7813	0,3875	0,8959

Pada data validasi, Val *Box loss* Model B turun dari 0,6657 menjadi 0,3433 atau sebesar 48,44%, sedangkan Val *Classification loss* turun dari 1,1765 menjadi 0,3875 atau sebesar 67,06%. Val DFL *Loss* juga mengalami penurunan dari 0,9801 menjadi 0,8959 atau sekitar 8,60%.

Val *Segmentation loss* menurun dari 0,7428 pada awal *training* hingga mencapai nilai minimum 0,6534 pada *epoch* ke-44, atau mengalami perbaikan sekitar 12,04%. Setelah titik tersebut, nilai Val *Segmentation loss* kembali meningkat dan mencapai 0,7813 pada *epoch* ke-103. Dibandingkan titik minimumnya, terjadi peningkatan sekitar 19,58%.



Gambar 4.4 Kurva *Validation loss* Model B

Gambar 4.4 memperlihatkan bahwa pola validasi Model B memiliki karakteristik serupa dengan Model A. Val *Box loss* dan Val *Classification loss* cenderung terus menurun, sedangkan Val *Segmentation loss* mencapai minimum pada pertengahan *training* kemudian meningkat kembali. Dengan kata lain, penurunan *Training Segmentation loss* setelah *epoch* ke-44 tidak lagi diikuti oleh penurunan *Validation Segmentation loss* secara searah. Kondisi tersebut menunjukkan bahwa peningkatan kemampuan generalisasi *mask* mulai mengalami kejenuhan.

Model B memperoleh *checkpoint* terbaik keseluruhan pada *epoch* ke-78. Dengan nilai *patience* sebesar 25, tidak terdapat peningkatan fitness yang mampu menghasilkan *checkpoint* baru selama 25 *epoch* berikutnya. Oleh karena itu:

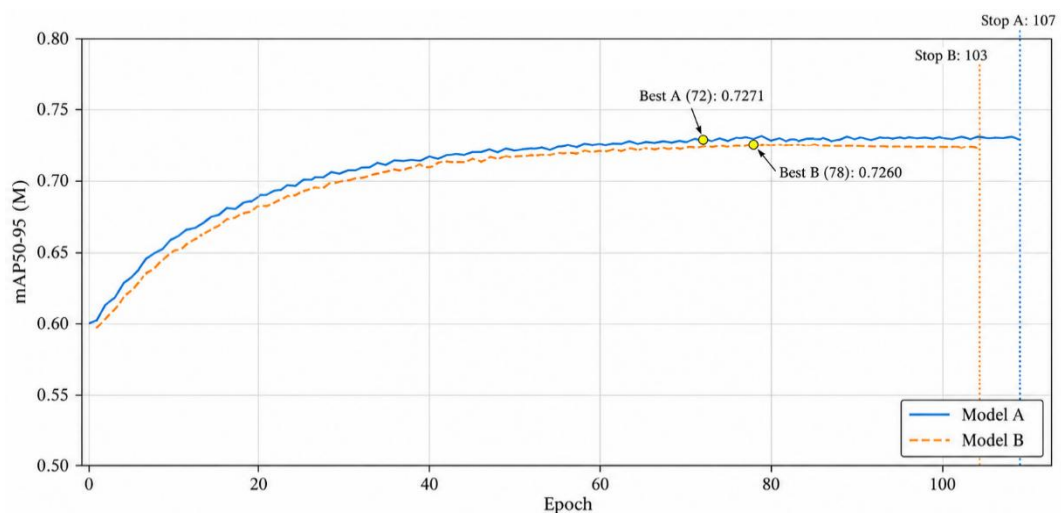
4.1.3 Komparasi Hasil Pelatihan Model A dan Model B

Gambaran perbandingan proses pelatihan kedua model, dilakukan analisis terhadap karakteristik konvergensi, nilai *validation loss*, perkembangan mAP50-95 *mask*, *checkpoint* terbaik, serta mekanisme *Early*

stopping. Perbedaan konfigurasi *hyperparameter* menyebabkan kedua model memiliki karakteristik optimasi yang berbeda.

Secara umum, Model B menghasilkan penurunan *training loss* yang lebih besar dibandingkan Model A. Penurunan Train Box loss meningkat dari 65,37% pada Model A menjadi 70,04% pada Model B, sedangkan Train Segmentation loss meningkat dari 66,17% menjadi 71,96%. Pada Classification loss, Model A mengalami penurunan sebesar 86,42% dan Model B sebesar 90,11%. Hasil tersebut menunjukkan bahwa konfigurasi Model B mampu menghasilkan optimasi terhadap data *training* yang lebih kuat.

Namun, pada *validation set*, perbedaan kedua model tidak sebesar perbedaan *training loss*. Minimum Val Segmentation loss Model A adalah 0,6563 pada *epoch* ke-30, sedangkan Model B memperoleh nilai minimum yang sedikit lebih rendah, yaitu 0,6534 pada *epoch* ke-44. Selisih kedua nilai tersebut hanya sebesar 0,0029. Hal ini menunjukkan bahwa penurunan *training loss* Model B yang lebih besar tidak secara langsung menghasilkan peningkatan yang sama besar terhadap kemampuan generalisasi segmentasi.



Gambar 4.5 Komparasi mAP50-95 Mask Model A dan Model B

Gambar 4.5 menunjukkan perkembangan mAP50-95 *mask* kedua model selama proses pelatihan. Pada *epoch* awal, kedua model mengalami peningkatan performa yang relatif cepat. Namun, ketika jumlah *epoch* semakin besar, kenaikan mAP50-95 semakin kecil dan kedua kurva bergerak menuju nilai yang relatif stabil pada kisaran 0,72–0,73. Kondisi tersebut menunjukkan bahwa kedua model telah memasuki daerah konvergensi dan penambahan *epoch* selanjutnya hanya menghasilkan peningkatan performa yang terbatas.

Model A mencapai nilai maksimum individual mAP50-95 *mask* sekitar 0,7276, sedangkan Model B juga mencapai nilai maksimum sekitar 0,7276, sehingga selisih performa maksimum keduanya sangat kecil. Meskipun demikian, *checkpoint* terbaik keseluruhan dipilih pada *epoch* ke-72 untuk Model A dan *epoch* ke-78 untuk Model B. Model A kemudian dihentikan pada *epoch* ke-107 dengan *patience* 35, sedangkan Model B dihentikan pada *epoch* ke-103 dengan *patience* 25.

Berdasarkan hasil tersebut, penyesuaian *hyperparameter* pada Model B menghasilkan proses optimasi yang lebih efisien terhadap data *training* dan mencapai *checkpoint* terbaik dalam 78 *epoch*. Akan tetapi, peningkatan tersebut tidak menghasilkan kenaikan yang besar pada mAP50-95 *mask* dibandingkan Model A. Dengan demikian, pengaruh utama konfigurasi Model B terlihat pada efisiensi proses optimasi dan keseimbangan performa keseluruhan, sedangkan kualitas *mask* maksimum kedua model relatif setara.

4.2 Analisis Hasil Pengujian Model pada Test set

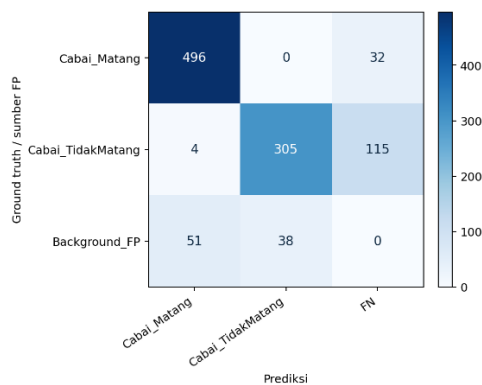
Setelah proses pelatihan selesai, Model A dan Model B dievaluasi menggunakan *test set* yang tidak digunakan dalam proses pembaruan bobot selama *training*. Pengujian ini bertujuan untuk mengetahui kemampuan generalisasi kedua model dalam mendeteksi, mengklasifikasikan, dan mensegmentasi objek cabai pada data yang belum pernah digunakan selama proses pembelajaran.

Evaluasi dilakukan terhadap 952 anotasi objek, yang terdiri atas 528 anotasi cabai matang dan 424 anotasi cabai tidak matang. Evaluasi dilakukan menggunakan *confusion matrix*, *Precision*, *Recall*, Skor F1, *Precision–Recall Curve*, mAP@50,

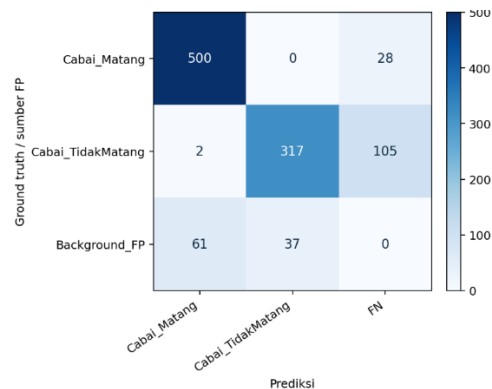
Mean IoU, dan *Counting error*. Metrik tersebut digunakan secara bersama-sama karena masing-masing memberikan informasi yang berbeda mengenai kemampuan model. *Precision* menunjukkan ketepatan prediksi positif, *Recall* menunjukkan kemampuan menemukan seluruh objek aktual, Skor F1 menunjukkan keseimbangan *Precision* dan *Recall*, sedangkan Mean IoU digunakan untuk mengevaluasi kualitas tumpang tindih *mask* hasil segmentasi terhadap *ground truth*. *Counting error* digunakan untuk mengetahui kesesuaian jumlah objek yang berhasil dideteksi terhadap jumlah objek sebenarnya.

4.2.1 Confusion matrix

Confusion matrix digunakan untuk menganalisis distribusi prediksi benar maupun kesalahan yang dihasilkan Model A dan Model B pada *test set*. Berbeda dengan klasifikasi citra sederhana, pada deteksi dan segmentasi objek terdapat pula kategori *background*, karena suatu objek dapat tidak berhasil dideteksi atau model dapat menghasilkan prediksi pada area yang tidak memiliki objek *ground truth*. Oleh karena itu, nilai False Positive dan False Negative tidak hanya berasal dari kesalahan klasifikasi antara Cabai Matang dan Cabai Tidak Matang, tetapi juga dapat berasal dari kesalahan terhadap *background*.



Gambar 4.6 Confusion Matriks *Testing set* Model A



Gambar 4.7 Confusion Matriks *Testing set* Model B

Berdasarkan **Gambar 4.6**, Model A berhasil mendeteksi dengan benar sebanyak 496 dari 528 anotasi Cabai Matang dan 305 dari 424 anotasi Cabai Tidak Matang. Pada kelas Cabai Matang terdapat 32 objek yang tidak berhasil

dideteksi dengan benar, sedangkan pada kelas Cabai Tidak Matang terdapat 119 objek yang tidak berhasil dideteksi dengan benar. Dengan demikian, jumlah True Positive Model A dari kedua kelas adalah sebanyak 801 objek.

Pada **Gambar 4.7**, jumlah prediksi benar mengalami peningkatan. Model B berhasil mendeteksi dengan benar sebanyak 500 dari 528 anotasi Cabai Matang dan 317 dari 424 anotasi Cabai Tidak Matang. Jumlah objek Cabai Matang yang tidak berhasil dideteksi turun menjadi 28 objek, sedangkan Cabai Tidak Matang yang tidak berhasil dideteksi turun menjadi 107 objek. Total True Positive Model B menjadi 817 objek, atau meningkat sebanyak 16 objek dibandingkan Model A.

Sebelum menghitung metrik evaluasi, nilai TP, FP, dan FN diturunkan terlebih dahulu dari *confusion matrix*. Nilai tersebut menjadi dasar perhitungan *Precision*, *Recall*, dan Skor F1 pada subbab berikutnya.

Tabel 4.5 Derivasi TP,FP, dan FN per Kelas pada Model A

Kelas	TP	FP	FN
Cabai Matang	500	63	28
Cabai Tidak Matang	317	37	107

Tabel 4.6 Derivasi TP,FP, dan FN per Kelas – Model B

Kelas	TP	FP	FN
Cabai Matang	496	55	32
Cabai Tidak Matang	305	38	119

Berdasarkan **Tabel 4.5**, Model A memiliki total False Negative sebanyak 151 objek, sedangkan Model B pada **Tabel 4.6** memiliki 135 objek. Hal ini menunjukkan bahwa Model B mampu mengurangi jumlah objek yang tidak berhasil dideteksi sebanyak 16 objek. Namun, total *False Positive* Model B meningkat dari 93 prediksi pada Model A menjadi 100 prediksi. Nilai False Positive pada Tabel tersebut merupakan keseluruhan prediksi suatu kelas yang tidak sesuai dengan kelas *ground truth*. Nilai ini dapat berasal dari kesalahan klasifikasi antar kelas maupun prediksi terhadap *background*. Sebagai contoh, pada Model A terdapat 51 prediksi Cabai Matang yang berasal dari *background* serta 4 objek Cabai Tidak Matang yang salah

diprediksi sebagai Cabai Matang. Oleh karena itu, jumlah False Positive kelas Cabai Matang pada Model A menjadi 55. Pendekatan derivasi yang sama digunakan untuk memperoleh nilai TP, FP, dan FN pada kelas dan model lainnya.

Secara keseluruhan, Model B menghasilkan lebih banyak deteksi benar dan lebih sedikit objek yang terlewat dibandingkan Model A. Peningkatan tersebut paling terlihat pada kelas Cabai Tidak Matang, di mana jumlah True Positive meningkat dari 305 menjadi 317 dan False Negative berkurang dari 119 menjadi 107.

4.2.2 Metrik Evaluasi Per Kelas

Evaluasi per kelas dilakukan untuk mengetahui performa Model A dan Model B secara lebih rinci pada masing-masing tingkat kematangan cabai. Tiga metrik utama yang digunakan adalah *Precision*, *Recall*, dan Skor F1. *Precision* menunjukkan ketepatan model dalam memberikan prediksi pada suatu kelas, *Recall* menunjukkan kemampuan model menemukan seluruh objek aktual pada kelas tersebut, sedangkan Skor F1 menunjukkan keseimbangan antara *Precision* dan *Recall*.

A. Kelas Cabai Matang

Dari nilai TP, FP, dan FN yang diperoleh dari *confusion matrix*, perhitungan *Precision*, *Recall*, dan Skor F1 kelas Cabai Matang untuk Model A dan Model B disajikan pada **Tabel 4.7** dan **Tabel 4.8**.

Tabel 4.7 Perhitungan Akurasi, Presisi, *Recall*, dan Skor F1 Kelas Model A

Metrik	Perhitungan
<i>Precision</i>	$\frac{496}{496 + 55} = 0,9002$
<i>Recall</i>	$\frac{496}{496 + 32} = 0,9394$
Skor F1	$\frac{2 \times (0,9002 \times 0,9394)}{0,9002 + 0,9394} = 0,9194$

Tabel 4.8 Perhitungan Akurasi, Presisi, *Recall*, dan Skor F1 Kelas Model B

Metrik	Perhitungan
<i>Precision</i>	$\frac{500}{500 + 63} = 0,8881$
<i>Recall</i>	$\frac{500}{500 + 28} = 0,9470$
Skor F1	$\frac{2 \times (0,8881 \times 0,9470)}{0,8881 + 0,9470} = 0,9166$

Berdasarkan hasil perhitungan, Model A memperoleh *Precision* sebesar 90,02%, *Recall* sebesar 93,94%, dan Skor F1 sebesar 91,94%. Nilai *Recall* sebesar 93,94% menunjukkan bahwa dari seluruh Cabai Matang pada *test set*, sebagian besar berhasil dideteksi dengan benar oleh Model A.

Model B memperoleh *Precision* sebesar 88,81%, *Recall* sebesar 94,70%, dan Skor F1 sebesar 91,66%. Dibandingkan Model A, *Recall* Model B meningkat sebesar 0,76 poin persentase, sedangkan *Precision* menurun sebesar 1,21 poin persentase. Skor F1 kedua model memiliki selisih yang relatif kecil, yaitu sebesar 0,28 poin persentase.

Hasil tersebut menunjukkan bahwa performa kedua model pada kelas Cabai Matang relatif seimbang. Model A memiliki *Precision* dan Skor F1 sedikit lebih tinggi, sedangkan Model B memiliki *Recall* yang lebih tinggi. *Recall* Model B yang lebih tinggi juga sesuai dengan jumlah False Negative yang lebih kecil, yaitu 28 objek dibandingkan 32 objek pada Model A.

B. Kelas Tidak Matang

Perhitungan *Precision*, *Recall*, dan Skor F1 pada kelas Cabai Tidak Matang dilakukan menggunakan nilai TP, FP, dan FN yang diperoleh dari *confusion matrix*. Hasil perhitungan kedua model disajikan pada **Tabel 4.9** dan **Tabel 4.10**.

Tabel 4.9 Perhitungan Akurasi, Presisi, *Recall*, dan Skor F1 Kelas Model A

Metrik	Perhitungan
<i>Precision</i>	$\frac{305}{305 + 38} = 0,8892$
<i>Recall</i>	$\frac{305}{305 + 119} = 0,7193$
Skor F1	$\frac{2 \times (0,8892 \times 0,7193)}{0,8892 + 0,7193} = 0,7935$

Tabel 4.10 Perhitungan Akurasi, Presisi, *Recall*, dan Skor F1 Kelas Model B

Metrik	Perhitungan
<i>Precision</i>	$\frac{317}{317 + 37} = 0,8955$
<i>Recall</i>	$\frac{317}{317 + 107} = 0,7476$
Skor F1	$\frac{2 \times (0,8955 \times 0,7476)}{0,8955 + 0,7476} = 0,8149$

Pada kelas Cabai Tidak Matang, Model A memperoleh *Precision* sebesar 88,92%, *Recall* sebesar 71,93%, dan Skor F1 sebesar 79,53%. Nilai *Recall* yang lebih rendah dibandingkan kelas Cabai Matang menunjukkan bahwa kelas Cabai Tidak Matang lebih sulit dideteksi. Dari 424 anotasi aktual, sebanyak 305 objek berhasil dideteksi dengan benar, sedangkan 119 objek tidak berhasil dideteksi.

Model B memperoleh *Precision* sebesar 89,55%, *Recall* sebesar 74,76%, dan Skor F1 sebesar 81,49%. Dibandingkan Model A, *Precision* meningkat sebesar 0,63 poin persentase, *Recall* meningkat sebesar 2,83 poin persentase, dan Skor F1 meningkat sebesar 1,96 poin persentase.

Peningkatan pada ketiga metrik menunjukkan bahwa perbaikan Model B lebih terlihat pada kelas Cabai Tidak Matang. Jumlah prediksi benar meningkat dari 305 menjadi 317 objek, sedangkan jumlah False Negative menurun dari 119 menjadi 107 objek. Meskipun *Recall* kelas Cabai Tidak Matang masih lebih

rendah dibandingkan kelas Cabai Matang, Model B memberikan hasil yang lebih baik dibandingkan Model A.

4.2.3 Evaluasi Metrik Secara Keseluruhan

Untuk mengetahui performa kedua model secara keseluruhan, dilakukan perhitungan *Macro Precision*, *Macro Recall*, dan *Macro Skor F1* dengan mengambil nilai rata-rata dari kedua kelas. Selain itu, *Mean IoU* digunakan untuk mengukur tingkat kesesuaian *mask* prediksi terhadap *mask ground truth*, sedangkan *Counting error* digunakan untuk mengetahui selisih jumlah objek yang berhasil dideteksi terhadap jumlah objek aktual.

Tabel 4.11 Perhitungan *Macro Precision*, *Macro Recall* dan *Macro Skor F1* Model A

Metrik	Perhitungan
Makro <i>Precision</i>	$\frac{0.9002 + 0.8892}{2} = 0.8947$
Makro <i>Recall</i>	$\frac{0.9394 + 0.7193}{2} = 0.8294$
Macro Skor F1	$\frac{0.9194 + 0.7953}{2} = 0.8573$
Mean IoU	$\frac{737,32}{801} = 0.9205$
<i>Counting error</i>	$\frac{ 894 + 952 }{952} \times 100\% = 6.09\%$

Tabel 4.12 Perhitungan *Macro Precision*, *Macro Recall* dan *Macro Skor F1* Model B

Metrik	Perhitungan
Makro <i>Precision</i>	$\frac{0.8881 + 0.8955}{2} = 0.8918$
Makro <i>Recall</i>	$\frac{0.9470 + 0.7476}{2} = 0.8473$
Macro Skor F1	$\frac{0.9166 + 0.8149}{2} = 0.8658$
Mean IoU	$\frac{754,25}{817} = 0.9232$
<i>Counting error</i>	$\frac{ 917 + 952 }{952} \times 100\% = 3.68\%$

Berdasarkan hasil perhitungan, Model A memperoleh Macro *Precision* sebesar 89,47%, Macro *Recall* sebesar 82,94%, dan Macro Skor F1 sebesar 85,73%. Sementara itu, Model B memperoleh Macro *Precision* sebesar 89,18%, Macro *Recall* sebesar 84,73%, dan Macro Skor F1 sebesar 86,58%.

Macro *Precision* Model A lebih tinggi sebesar 0,29 poin persentase, sedangkan Model B memiliki Macro *Recall* lebih tinggi sebesar 1,79 poin persentase. Model B juga memperoleh Macro Skor F1 sebesar 86,58%, lebih tinggi 0,85 poin persentase dibandingkan Model A. Hasil tersebut menunjukkan bahwa Model B memiliki keseimbangan *Precision* dan *Recall* yang sedikit lebih baik secara keseluruhan.

Pada evaluasi hasil segmentasi, Model A memperoleh Mean IoU sebesar 92,05%, sedangkan Model B memperoleh 92,32%. Terdapat peningkatan sebesar 0,27 poin persentase pada Model B. Nilai Mean IoU kedua model yang berada di atas 90% menunjukkan bahwa *mask* hasil prediksi kedua model memiliki tingkat kesesuaian yang tinggi terhadap *mask ground truth*.

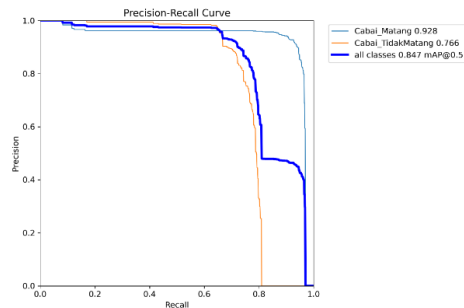
Perbedaan yang lebih besar terlihat pada *Counting error*. Model A menghasilkan *Counting error* sebesar 6,09%, sedangkan Model B menghasilkan 3,68%. Dengan demikian, Model B menurunkan *Counting error* sebesar 2,41 poin persentase dibandingkan Model A.

Pada Model A, jumlah objek hasil prediksi adalah 894 objek dari 952 objek aktual. Sementara itu, Model B menghasilkan 917 objek dari 952 objek aktual. Selisih jumlah prediksi terhadap *ground truth* pada Model B lebih kecil, yang menunjukkan bahwa Model B mampu mendeteksi jumlah objek secara lebih mendekati jumlah sebenarnya.

4.2.4 Evaluasi berbasis *Precision-Recall Curve*

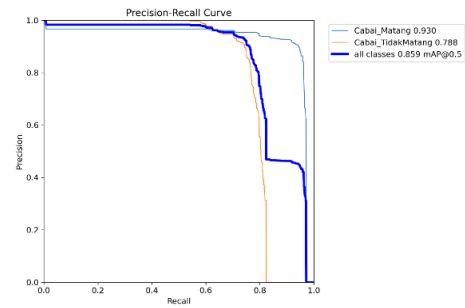
Selain evaluasi berdasarkan *confusion matrix*, performa segmentasi kedua model juga dievaluasi menggunakan *Precision-Recall Curve*. Kurva ini digunakan untuk menunjukkan hubungan antara *Precision* dan *Recall* pada berbagai nilai *confidence threshold*. Nilai luas di bawah kurva pada suatu

kelas dinyatakan sebagai *Average precision* (AP), sedangkan rata-rata AP dari seluruh kelas dinyatakan sebagai mean *Average precision* (mAP). Pada penelitian ini digunakan *Mask* mAP@50, yaitu nilai rata-rata AP pada *mask* dengan kriteria Intersection over Union sebesar 0,50.



Gambar 4.8 Kurva *Precision-Recall*

Model A



Gambar 4.9 Kurva *Precision-Recall*

Model B

Berdasarkan hasil *Precision-Recall Curve*, Model A memperoleh AP@50 sebesar 0,928 pada kelas Cabai Matang dan 0,766 pada kelas Cabai Tidak Matang. Dari kedua kelas tersebut diperoleh *Mask* mAP@50 Model A sebesar 0,8469 atau 84,69%. Model B memperoleh AP@50 sebesar 0,930 pada kelas Cabai Matang dan 0,788 pada kelas Cabai Tidak Matang, sehingga menghasilkan *Mask* mAP@50 sebesar 0,8591 atau 85,91%. Dibandingkan Model A, *Mask* mAP@50 Model B meningkat sebesar 0,0122 atau 1,22 poin persentase. Pada kelas Cabai Matang, AP@50 meningkat dari 0,928 menjadi 0,930 atau sebesar 0,20 poin persentase. Peningkatan yang lebih besar terdapat pada kelas Cabai Tidak Matang, yaitu dari 0,766 menjadi 0,788 atau sebesar 2,20 poin persentase. Hasil tersebut menunjukkan bahwa Model B memiliki performa segmentasi yang lebih baik berdasarkan *Mask* mAP@50. Sejalan dengan hasil metrik per kelas sebelumnya, peningkatan yang paling terlihat kembali terdapat pada kelas Cabai Tidak Matang.

Tabel 4.13 Perhitungan *Mask* mAP@50 Model A

Metrik	Perhitungan
AP@50 Cabai Matang	0.928
AP@50 Cabai Tidak Matang	0.766

Metrik	Perhitungan
Mask mAP@50	$\frac{0.928 + 0.766}{2} = 0.8469$

Berdasarkan **Tabel 4.13** , Model A memperoleh AP@50 sebesar 0,928 pada kelas Cabai Matang dan 0,766 pada kelas Cabai Tidak Matang. Rata-rata *Average precision* kedua kelas menghasilkan *Mask mAP@50* sebesar 0,8469 atau 84,69%. Nilai AP@50 kelas Cabai Matang yang lebih tinggi menunjukkan bahwa Model A memiliki kemampuan segmentasi yang lebih baik pada kelas Cabai Matang dibandingkan Cabai Tidak Matang.

Tabel 4.14 Perhitungan *Mask mAP@50* Model B

Metrik	Perhitungan
AP@50 Cabai Matang	0.930
AP@50 Cabai Tidak Matang	0.788
Mask mAP@50	$\frac{0.930 + 0.788}{2} = 0.859$

Berdasarkan **Tabel 4.14** , Model B memperoleh AP@50 sebesar 0,930 pada kelas Cabai Matang dan 0,788 pada kelas Cabai Tidak Matang. Nilai tersebut menghasilkan *Mask mAP@50* sebesar 0,8591 atau 85,91%. Dibandingkan Model A, peningkatan terbesar terjadi pada kelas Cabai Tidak Matang, yaitu dari AP@50 0,766 menjadi 0,788.

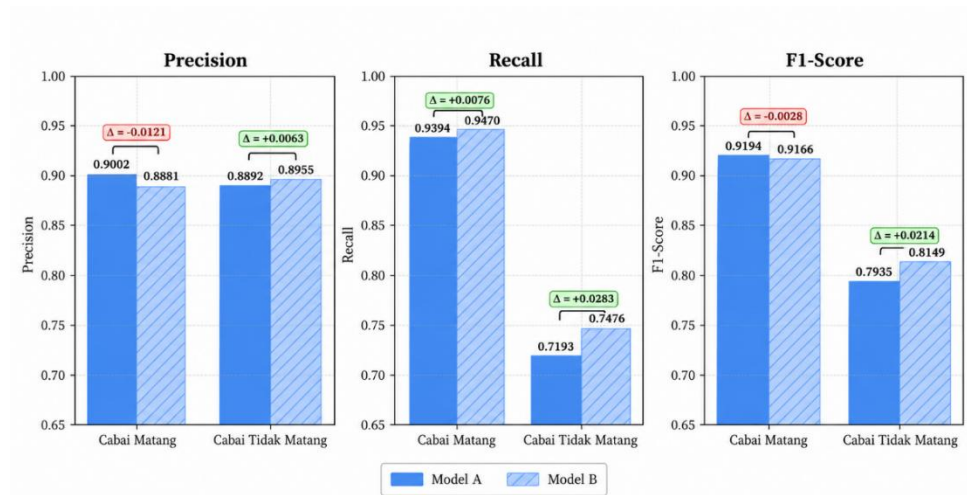
Tabel 4.15 Komparasi Metrik Keseluruhan

Metrik	Model A	Model B	Selisih B – A
AP@50 Cabai Matang	0,928	0,930	+0,002
AP@50 Cabai Tidak Matang	0,766	0,788	+0,022
Mask mAP@50	0,8469	0,8591	+0,0122

Berdasarkan hasil komparasi, Model B memperoleh *Mask mAP@50* sebesar 0,8591, lebih tinggi dibandingkan Model A sebesar 0,8469. Peningkatan yang diperoleh sebesar 0,0122 atau 1,22 poin persentase. Pada kelas Cabai Matang, peningkatan AP@50 hanya sebesar 0,002, sedangkan pada kelas Cabai Tidak Matang terjadi peningkatan sebesar 0,022. Dengan demikian, peningkatan performa segmentasi Model B terutama terlihat pada kemampuan segmentasi kelas Cabai Tidak Matang.

4.2.5 Komparasi Metrik Model A dan Model B

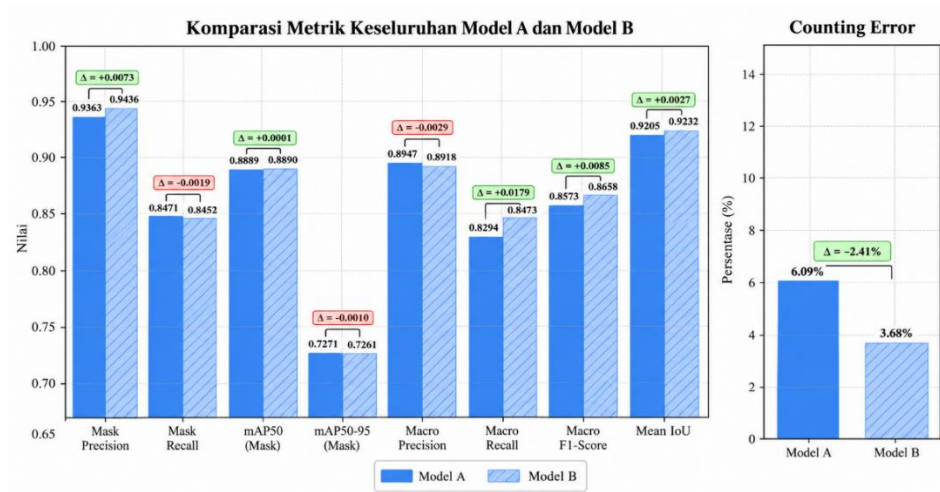
Untuk memberikan gambaran perbandingan performa Model A dan Model B secara keseluruhan, hasil *Precision*, *Recall*, Skor F1, Mean IoU, *Mask* mAP@50, dan *Counting error* dibandingkan berdasarkan hasil pengujian pada *test set*.



Gambar 4.10 Komparasi Grafik Presisi, *Recall*, dan FI-Score per-Kelas

Gambar 4.10 memperlihatkan bahwa pada kelas Cabai Matang, kedua model memiliki performa yang relatif serupa. Model A memiliki *Precision* sebesar 90,02% dan Skor F1 sebesar 91,94%, sedikit lebih tinggi dibandingkan Model B yang memperoleh *Precision* 88,81% dan Skor F1 91,66%. Sebaliknya, *Recall* Model B sebesar 94,70% lebih tinggi dibandingkan Model A sebesar 93,94%.

Perbedaan yang lebih besar terlihat pada kelas Cabai Tidak Matang. Model B memperoleh *Precision* sebesar 89,55%, *Recall* sebesar 74,76%, dan Skor F1 sebesar 81,49%, seluruhnya lebih tinggi dibandingkan Model A. Hal tersebut menunjukkan bahwa peningkatan performa Model B terutama terjadi pada kelas Cabai Tidak Matang.



Gambar 4.11 Komparasi Metrik Keseluruhan Model A dan Model B

Gambar 4.11 menunjukkan perbandingan metrik keseluruhan Model A dan Model B. Model A memiliki *Macro Precision* sebesar 89,47%, sedikit lebih tinggi dibandingkan Model B sebesar 89,18%. Namun, Model B memiliki *Macro Recall* sebesar 84,73% dan *Macro Skor F1* sebesar 86,58%, lebih tinggi dibandingkan Model A yang masing-masing memperoleh 82,94% dan 85,73%.

Model B juga memperoleh *Mean IoU* sebesar 92,32%, lebih tinggi dibandingkan Model A sebesar 92,05%. Pada evaluasi *Precision-Recall Curve*, *Mask mAP@50* Model B sebesar 85,91%, sedangkan Model A sebesar 84,69%. Selain itu, *Counting error* Model B sebesar 3,68%, lebih rendah dibandingkan Model A sebesar 6,09%.

Berdasarkan keseluruhan hasil pengujian, Model B menunjukkan performa yang lebih baik pada sebagian besar metrik evaluasi. Meskipun *Macro Precision* Model A sedikit lebih tinggi, Model B memiliki *Recall*, *Skor F1*, *Mean IoU*, *Mask mAP@50*, dan *Counting error* yang lebih baik. Model B juga berhasil meningkatkan jumlah objek yang terdeteksi dengan benar dari 801 menjadi 817 objek.

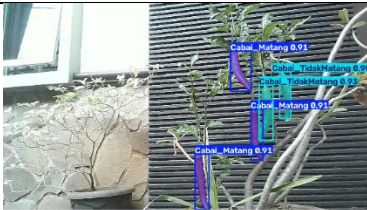
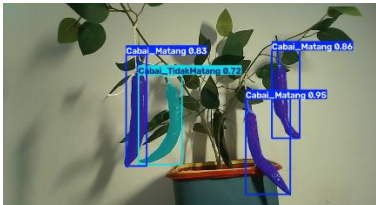
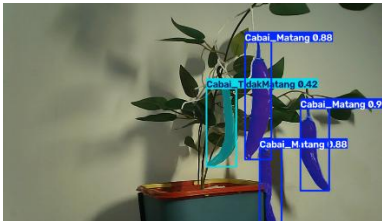
Dengan mempertimbangkan hasil tersebut, Model B dipilih sebagai model yang digunakan pada tahap implementasi sistem berikutnya. Pemilihan Model B terutama didasarkan pada kemampuan *Recall* yang lebih tinggi, kualitas *mask* yang sedikit lebih baik, serta *Counting error* yang lebih rendah, sehingga jumlah cabai yang berhasil dideteksi lebih mendekati jumlah objek aktual.

4.2.6 Pengujian Model Final Pada *Test set Baru*

Berdasarkan **Tabel 4.16** , pengujian model dilakukan sebanyak 10 kali dengan jumlah objek yang berbeda pada setiap data pengujian. Dari total 45 objek cabai aktual, sebanyak 38 objek berhasil dideteksi oleh model, sehingga diperoleh tingkat deteksi keseluruhan sebesar 84,44%. Pengujian A, B, C, D, dan I memperoleh tingkat deteksi sebesar 100%, sedangkan nilai terendah terdapat pada pengujian F sebesar 50%. Objek yang tidak terdeteksi pada pengujian E, F, G, H, dan J merupakan cabai tidak matang. Hal ini menunjukkan bahwa pada kondisi pengujian yang lebih terang, model lebih konsisten mendeteksi cabai matang dan tidak matang, sedangkan pada kondisi pencahayaan yang redup, cabai matang terdeteksi namun cabai tidak matang tidak terdeteksi.

Tabel 4. 16 Pengujian model final pada data pengujian

Pengujian	Data Pengujian	Total Deteksi	Total Aktual	Tingkat deteksi
A		4	4	100%
B		5	5	100%

Pengujian	Data Pengujian	Total Deteksi	Total Aktual	Tingkat deteksi
C		5	5	100%
D		5	5	100%
E		3	4	75%
F		2	4	50%
G		3	4	75%
H		4	5	80%
I		4	4	100%

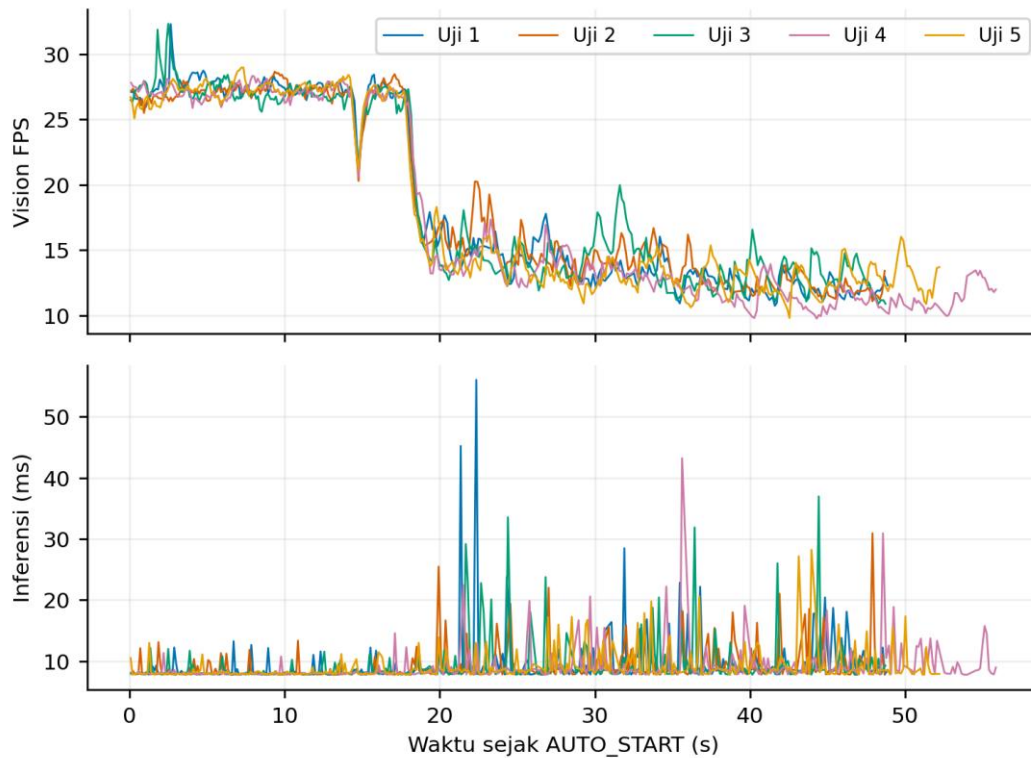
Pengujian	Data Pengujian	Total Deteksi	Total Aktual	Tingkat deteksi
J		3	5	60%
Total		38	45	84,44%

4.2.7 Pengujian Optimasi Model Terhadap *Latency* dan FPS

Pengujian dilakukan dengan kamera menghasilkan citra 640×480 piksel, model diproses pada ukuran masukan 416×416 , dan pendinginan aktif. Kolom vision_FPS, infer_ms, dan loop_ms berasal dari pencatat aplikasi. Rata-rata keseluruhan dihitung dari rata-rata setiap ulangan agar tiap ulangan memiliki bobot yang sama.

Tabel 4. 17 Kinerja *Runtime* pada Lima Pengulangan

Uji	Sampel vision	Vision FPS	Inferensi (ms)	Loop (ms)
1	347	19,61	9,32	61,49
2	337	19,97	9,00	60,06
3	335	19,98	9,19	59,41
4	374	18,51	9,01	65,29
5	352	19,17	9,02	63,43
Rata-rata	—	19,45	9,11	61,94



Gambar 4. 12 Vision FPS dan Waktu Inferensi selama Pengendalian

Rata-rata vision FPS sebesar 19,45 dengan waktu inferensi 9,11 ms. Waktu *loop* rata-rata 61,94 ms mencakup proses di luar eksekusi jaringan. Perbedaan ini menunjukkan bahwa kecepatan sistem tidak dapat dihitung hanya dari kebalikan waktu inferensi. Nilai FPS dan *loop* juga memiliki jendela pengukuran aplikasi masing-masing, sehingga rata-rata keduanya tidak harus saling berbalikan.

Data tersebut membuktikan bahwa model TensorRT dapat dijalankan bersama pengolahan fitur, antarmuka, dan komunikasi robot pada konfigurasi pengujian. Besar percepatan terhadap PyTorch tidak dihitung karena tidak tersedia pasangan *benchmark* dengan kondisi pengujian yang sama.

4.3 Pengujian Kinerja Kendali *Visual Servoing*

4.3.1 Pengujian Respons Jacobian Empiris

Arsip data memuat 93 pasangan gerak *probe* pada lima kondisi ukuran *mask*. Pengujian ini digunakan untuk melihat perubahan sensitivitas fitur terhadap gerak, dan tidak digabungkan dengan lima ulangan V23. Koefisien diperoleh melalui *least*

squares tanpa *intercept* dari perpindahan aktual XYZ terhadap perubahan u , v , dan $100 \ln(A_1/A_0)$. Residual dihitung pada data yang sama dengan pembentukan model.

Tabel 4.18 Perubahan Sensitivitas Lokal pada Data *Probe*

Kondisi	n pasangan	Median <i>mask</i> (px ²)	$\partial u/\partial Y$ (px/mm)	$\partial v/\partial Z$ (px/mm)	RMSE $u-v$ (px)
Jauh	18	2.769	2,522	3,299	4,49
Menengah	22	3.416	2,429	2,610	6,15
Dekat	18	7.259	1,135	3,807	10,42
Sangat dekat	18	14.717	1,326	3,814	6,84
Paling dekat	17	24.639	1,673	-0,083	9,49

Sensitivitas Y terhadap u berubah dari sekitar 2,522 menjadi 1,673 piksel/mm antara kondisi jauh dan paling dekat. Pada kondisi paling dekat, sensitivitas Z terhadap v menjadi $-0,083$ piksel/mm, berbeda dari nilai positif 2,610–3,814 pada empat kondisi lainnya. Temuan ini menunjukkan bahwa satu Jacobian tidak dapat diasumsikan tetap mewakili seluruh daerah gerak. Perubahan tersebut dapat berkaitan dengan proyeksi, *mask*, dan respons gerak; data ini belum memisahkan penyebabnya.

Hasil lima pengulangan menilai gabungan Jacobian, pendekatan X , *deadband*, dan aturan pemulihan. Karena tidak ada pembandingan dengan komponen kontrol dinonaktifkan pada kondisi sama, kontribusi masing-masing komponen tidak dihitung secara terpisah.

4.3.2 Pengujian Koreksi Error pada Pengulangan Cabai yang Sama

Lima ulangan diambil dari satu batch cabai yang sama dengan konfigurasi dan pose awal identik. Uji 1–5 berturut-turut memakai *run* pukul 21:33:35, 21:35:36, 21:37:28, 21:39:18, dan 21:41:37 tanggal 14 September 2026. Pemilihan batch berdasarkan kesamaan protokol, bukan pemilihan lima *error* terkecil dari seluruh arsip.

Titik acuan adalah (387, 36) piksel dengan batas $u = 362\text{--}412$ dan $v = 11\text{--}61$ piksel. READY memerlukan posisi *cut-point* di dalam kedua batas tersebut dan kondisi *bottom* yang dikonfirmasi pengendali. Norma error

dipakai untuk merangkum penurunan, sedangkan keberhasilan penyelarasan diperiksa pada kedua komponen *error*.

Tabel 4. 19 Error Awal dan Error pada posisi READY

Pengujian	Error Awal (px)	Error Akhir (px)	Error Terkoreksi (%)	Waktu READY (s)	Hasil
1	180,62	15,30	91,53	48,89	Sukses
2	199,22	5,39	97,30	48,70	Sukses
3	199,01	16,49	91,71	48,75	Sukses
4	198,49	22,14	88,85	55,87	Kurang sukses
5	207,39	5,83	97,19	52,23	Sukses
Rata-rata	196,95	13,03	93,32	50,89	4/5 sukses

Data lima pengujian menunjukkan bahwa *error* awal rata-rata sebesar 196,95 piksel dapat diturunkan menjadi 13,03 piksel ketika sistem mencapai kondisi *READY*. Jika dihitung berdasarkan persentase koreksi masing-masing percobaan, sistem menghasilkan rata-rata koreksi *error* sebesar 93,32%.

Nilai rata-rata tersebut dapat diverifikasi menggunakan perhitungan berikut.

$$e_{akhir,rata-rata} = \frac{15,30 + 5,39 + 16,49 + 22,14 + 5,83}{5} = 13,03 \text{ px}$$

$$K_{e,rata-rata} = \frac{91,53 + 97,30 + 91,71 + 88,85 + 97,19}{5} = 93,32\%$$

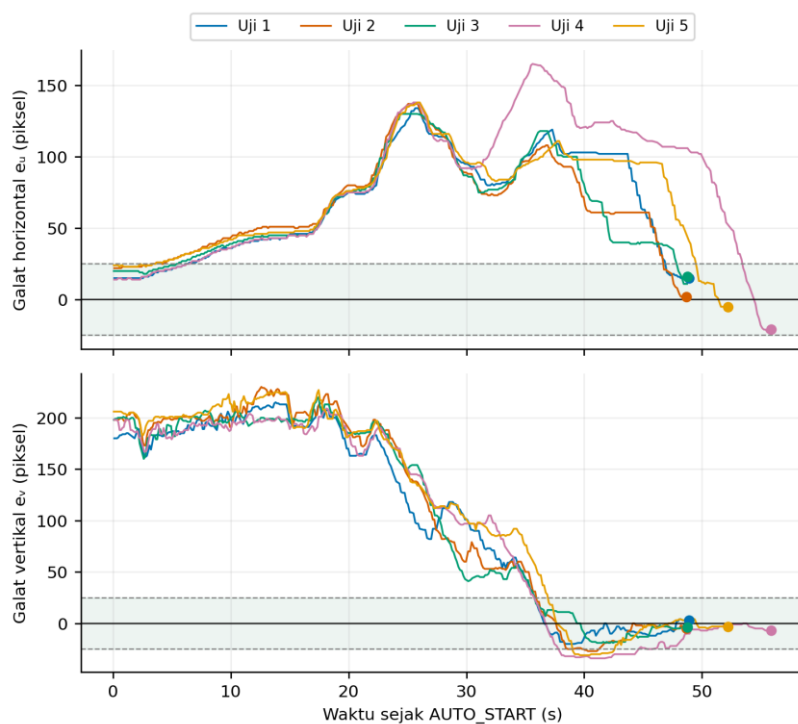
$$t_{READY,rata-rata} = \frac{48,89 + 48,70 + 48,75 + 55,87 + 52,23}{5} \\ = 50,89 \text{ s}$$

$$Keberhasilan \text{ fisik} = \frac{4}{5} \times 100\% = 80\%$$

Kelima pengujian berhasil mencapai kondisi *READY*, sedangkan berdasarkan hasil pengamatan proses fisik terdapat empat pengujian yang dinyatakan sukses dan satu pengujian, yaitu pengujian keempat, dinyatakan kurang sukses karena posisi akhir robot terlalu dekat terhadap objek. Dengan demikian, tingkat keberhasilan fisik pada lima pengujian adalah 80%.

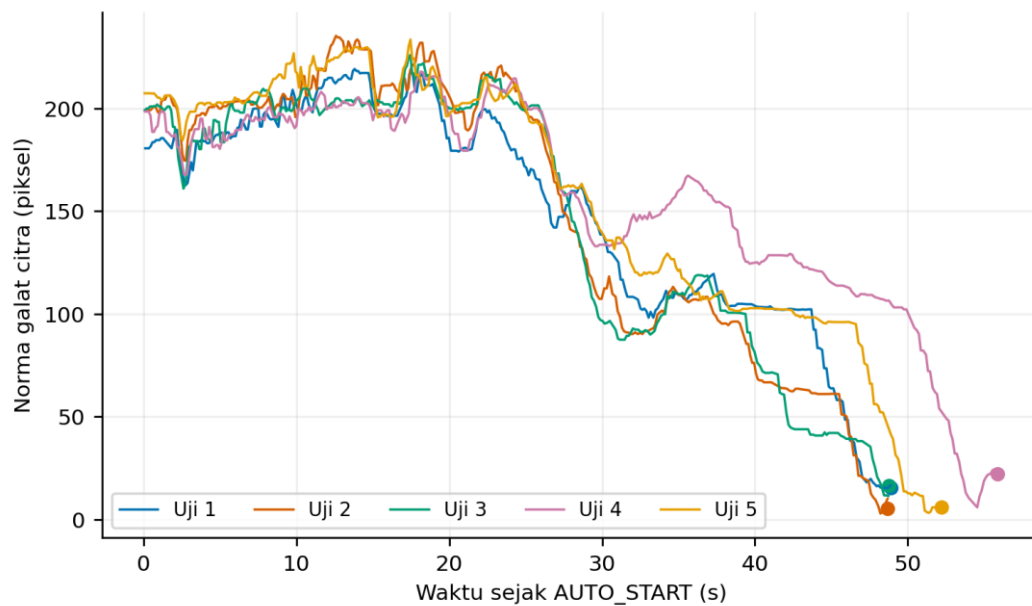
Hasil tersebut memperlihatkan bahwa pencapaian kondisi *READY* belum selalu menjamin keberhasilan proses fisik. Pada pengujian keempat, *error* akhir masih sebesar 22,14 piksel dan secara visual masih berada dalam batas *deadband*, sehingga sistem mengizinkan perpindahan ke tahap berikutnya. Namun, posisi longitudinal robot terhadap cabai terlalu dekat sehingga hasil fisiknya dinilai kurang baik. Hal ini menunjukkan bahwa *error cut-point* pada bidang citra terutama mengevaluasi keberhasilan *alignment* visual, sedangkan keberhasilan pemetikan juga dipengaruhi oleh jarak longitudinal robot terhadap objek.

Berdasarkan lima pengujian tersebut, metode *uncalibrated image-based visual servoing* yang diterapkan mampu mengurangi *error cut-point* terhadap *desired point* rata-rata sebesar 93,32%, dengan rata-rata *error* akhir sebesar 13,03 piksel dan rata-rata waktu menuju kondisi *READY* sebesar 50,89 detik. Hasil ini menunjukkan bahwa *empirical Jacobian* dan kontrol visual yang digunakan mampu menghasilkan koreksi Kartesian sehingga posisi *cut-point* bergerak mendekati *desired point*.



Gambar 4. 13 *Error* Horizontal dan Vertikal pada Lima Pengulangan

Kelima ulangan berakhir di dalam toleransi kedua sumbu. Rata-rata norma *error* turun dari $196,95 \pm 9,84$ menjadi $13,03 \pm 7,25$ piksel. Rata-rata reduksi per ulangan adalah $93,32 \pm 3,76$ poin persentase. Tanda $\pm$ menyatakan simpangan baku sampel antarulangan. Nilai akhir dihitung tepat pada event READY, sehingga berbeda dari rekap lama yang memakai rata-rata sepuluh *frame* terakhir sebesar 14,56 piksel.



Gambar 4. 14 Norma *Error* Citra terhadap Waktu

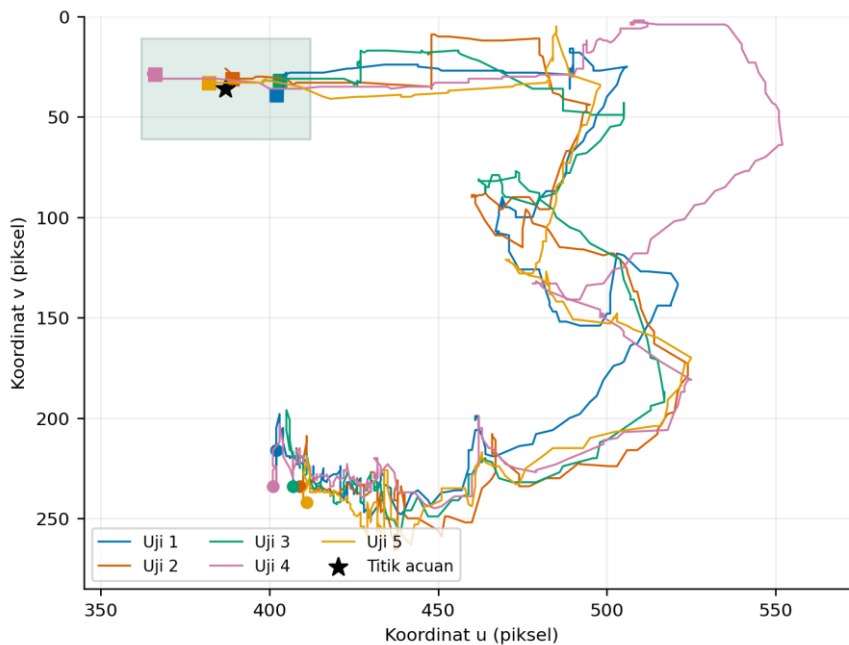
Error tidak menurun secara monoton. Gerak pendekatan X mengubah proyeksi objek sehingga *error* dapat meningkat sementara, kemudian dikoreksi pada pengamatan berikutnya. Oleh sebab itu, keberhasilan yang didukung data adalah tercapainya toleransi pada kelima ulangan, bukan *error* yang selalu turun pada setiap langkah atau *error* akhir yang tepat nol.

Tabel 4. 20 Waktu READY dan Jumlah Langkah Pengendalian

Uji	Waktu READY (s)	Langkah selesai	Pemulihan tepi	Bottom READY
1	48,89	13	3	Ya
2	48,70	13	2	Ya
3	48,75	13	2	Ya
4	55,87	16	6	Ya

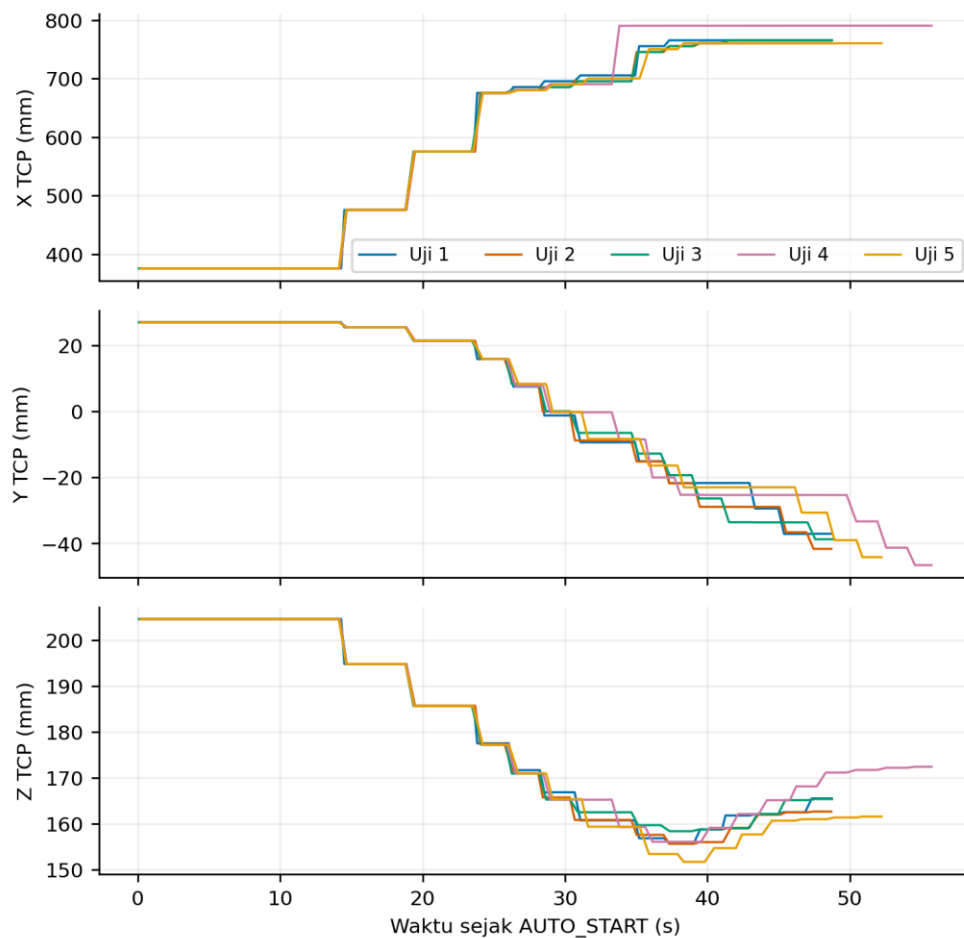
Uji	Waktu READY (s)	Langkah selesai	Pemulihan tepi	Bottom READY
5	52,23	14	3	Ya
Rata-rata	50,89	13,8	3,2	5/5

Waktu menuju READY adalah $50,89 \pm 3,16$ detik, dengan rentang 48,70–55,87 detik. Koefisien variasi waktunya sekitar 6,21%. Uji keempat membutuhkan 16 langkah dan enam pemulihan tepi, lebih banyak daripada ulangan lainnya. Hal ini sejalan dengan waktu pengendalian terlama pada uji tersebut.



Gambar 4. 15 Lintasan *Cut-point* Menuju Daerah Toleransi

Lingkaran menandai posisi awal, persegi posisi READY, dan tanda bintang titik acuan. Arah v mengikuti koordinat citra yang membesar ke bawah. Lintasan menunjukkan bahwa pengendali membawa fitur dari posisi awal yang berbeda menuju daerah toleransi yang sama, meskipun lintasan antarulangan tidak berimpit.



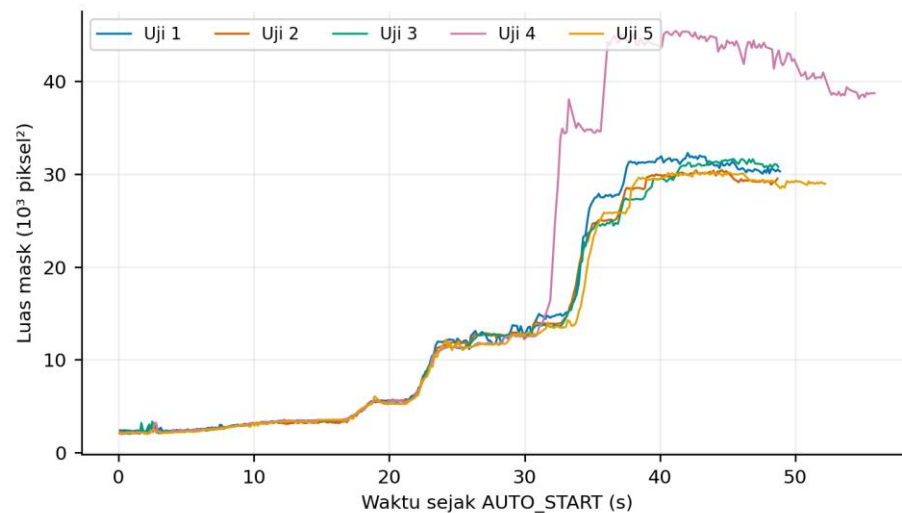
Gambar 4. 16 Respons Posisi TCP Selama *Visual Servoing*

Tabel 4. 21 Pose TCP pada READY

Uji	X (mm)	Y (mm)	Z (mm)
1	765,70	-37,06	165,50
2	760,72	-41,63	162,63
3	765,65	-38,73	165,36
4	790,63	-46,56	172,40
5	760,51	-44,14	161,55
Rata-rata	768,64	-41,62	165,49
SD sampel	12,55	3,87	4,23

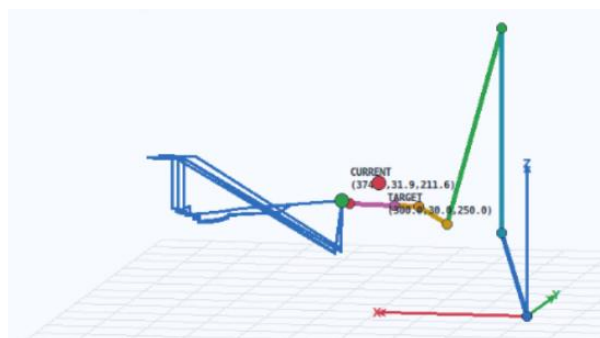
Perpindahan TCP mengonfirmasi adanya pendekatan pada X sekaligus koreksi Y–Z. Namun, keberhasilan di ruang citra tidak menghasilkan pose akhir Kartesian yang identik. Sebagai contoh, X akhir uji keempat mencapai 790,63

mm, sedangkan empat ulangan lainnya berada sekitar 760,51–765,70 mm. Angka ini menjelaskan mengapa posisi visual dalam *deadband* masih dapat disertai kedekatan fisik yang berbeda. Simpangan pose tersebut merupakan variasi hasil pada skenario ini, bukan spesifikasi *repeatability* mekanik robot.



Gambar 4. 17 Perubahan Luas Mask pada Lima Pengulangan

Luas *mask* meningkat selama pendekatan, sejalan dengan perpindahan X robot. *Mask* tetap dapat berubah akibat orientasi, pemotongan oleh batas citra, dan variasi segmentasi. Oleh karena itu, *mask* dipakai sebagai petunjuk kedekatan bersama fitur posisi dan kondisi *bottom*, bukan satu-satunya ukuran jarak.



Gambar 4.18 3D Trajectory pengulangan cabai yang sama posisinya

Dari Gambar **4.18** menunjukkan konsistensi rute yang diambil untuk mencapai cabai yang sama di posisi yang sama.

4.3.3 Pengujian Pendekatan pada Cabai dengan Ukuran Berbeda

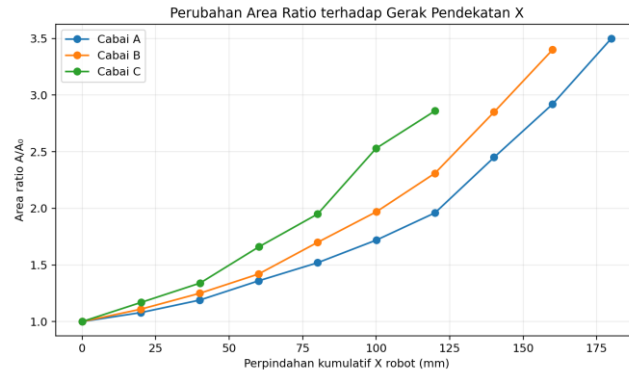
Pengujian cabai A, B, dan C membandingkan perubahan luas *mask* selama pendekatan pada objek berukuran berbeda. Jarak referensi diukur secara manual dari lensa RGB menuju titik potong sebagai *line-of-sight* (LOS), sehingga berbeda makna dengan perpindahan pada sumbu X robot.

Tabel 4.22 Perubahan Jarak dan Luas *Mask* pada Tiga Cabai

Cabai	Kondisi	Panjang (mm)	Ground truth LOS (mm)	ΔX kumulatif robot (mm)	Mask (px ²)	Area ternormalisasi A/A_0
A	Awal	80	360	0	2.553	1,00
A	Akhir	80	180	180	8.938	3,50
B	Awal	110	320	0	5.615	1,00
B	Akhir	110	170	160	19.077	3,40
C	Awal	110	260	0	7.630	1,00
C	Akhir	110	140	120	21.843	2,86

Tabel 4.22 titik awal dan akhir pada Tabel menunjukkan bahwa luas *mask* meningkat ketika jarak LOS antara kamera dan titik potong berkurang. Pada cabai A, luas *mask* meningkat dari 2.553 menjadi 8.938 piksel², cabai B dari 5.615 menjadi 19.077 piksel², dan cabai C dari 7.630 menjadi 21.843 piksel². Rasio luas dinormalisasi terhadap kondisi awal masing-masing target sehingga perubahan skala proyeksi dapat dibandingkan meskipun ukuran fisik dan luas awal ketiga cabai berbeda.

$$\text{Rasio luas cabai A} = A_{\text{akhir}} / A_{\text{awal}} = 8.938 / 2.553 = 3,50 \quad (4.7)$$



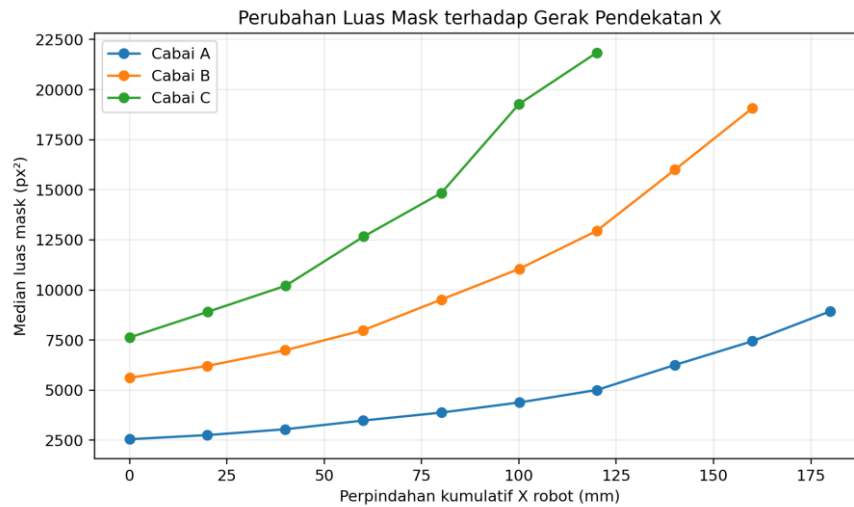
Gambar 4. 19 Perubahan rasio luas *mask* selama pendekatan

Gambar 4.18 menunjukkan rasio luas meningkat pada ketiga cabai, dengan nilai akhir sekitar 3,50; 3,40; dan 2,86. Kecenderungan ini sejalan dengan hubungan proyeksi perspektif pada Bab II tentang *active monocular range*: objek yang sama tampak lebih besar ketika kamera mendekat. Nilai awal luas tidak sama karena ukuran, orientasi, dan jarak awal target berbeda. Oleh sebab itu, satu ambang luas absolut tidak dapat langsung menyatakan jarak yang sama untuk seluruh cabai.

Persamaan (2.39) dapat digunakan sebagai ilustrasi model ideal perubahan skala. Untuk cabai A, jika perpindahan kamera dianggap segaris dengan arah pendekatan sebesar 180 mm dan bentuk proyeksi tetap, perhitungan memberikan estimasi berikut.

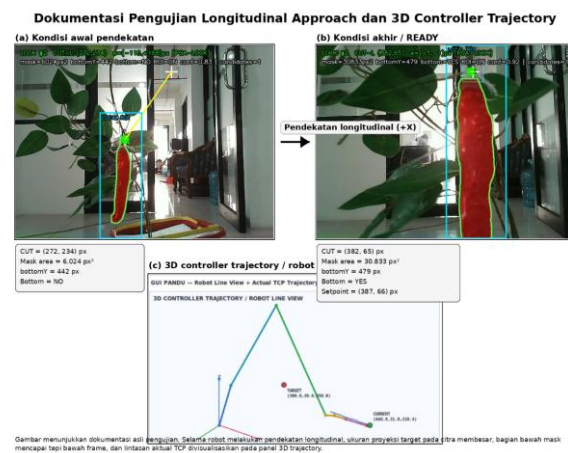
$$Z_2 = 180 / (\sqrt{(8.938 / 2.553)} - 1) \approx 206,64 \text{ mm}$$

Hasil pendekatan ideal tersebut berbeda dari LOS akhir terukur sebesar 180 mm. Selisih sekitar 26,64 mm menunjukkan bahwa asumsi proyeksi tetap dan kesegarisan gerak perlu diperiksa sebelum rumus digunakan sebagai pengukur jarak. LOS diukur dari lensa menuju titik potong, sedangkan ΔX berasal dari sumbu robot; kedua besaran tidak selalu berimpit. Contoh ini merupakan perhitungan ulang dari titik awal dan akhir, bukan hasil validasi estimator jarak yang berdiri sendiri.



Gambar 4. 20 Perubahan luas *mask* terhadap perpindahan X

Gambar 4.19 memperlihatkan luas absolut *mask* yang berbeda antarcabai. Nilai luas cabai yang lebih besar tidak langsung berarti jaraknya lebih dekat karena geometri objek juga memengaruhi proyeksi. Jarak acuan yang terukur pada analisis utama adalah titik awal dan akhir pada **Tabel 4.22** ; titik antara yang diinterpolasi tidak dipakai sebagai pengukuran LOS independen.



Gambar 4. 21 Dokumentasi kondisi awal dan akhir pendekatan

Gambar 4.20 memperlihatkan perubahan proyeksi cabai dan tampilan lintasan pengendali antara awal pendekatan dan kondisi akhir. Dokumentasi mendukung adanya gerak mendekat dan perubahan *mask*, sedangkan besar LOS tetap merujuk pada pengukuran manual.

Tabel 4. 23 Kecocokan Model Empiris terhadap Data Kalibrasi LOS

Target	Kondisi	Panjang fisik (mm)	Mask (px ²)	Ground truth LOS (mm)	Estimasi LOS Model (mm)	Error absolut (mm)	Error (%)
Cabai A	Awal	80	2.553	360	350,85	9,15	2,54%
Cabai A	Akhir	80	8.938	180	173,27	6,73	3,74%
Cabai B	Awal	110	5.615	320	323,07	3,07	0,96%
Cabai B	Akhir	110	19.077	170	161,28	8,72	5,13%
Cabai C	Awal	110	7.630	260	272,80	12,80	4,92%
Cabai C	Akhir	110	21.843	140	148,72	8,72	6,23%

Tabel 4.22 memuat enam titik yang digunakan pada pencocokan hubungan ukuran fisik, luas *mask*, dan jarak LOS dalam draft. Rata-rata *error* absolut dihitung sebagai $(9,15 + 6,73 + 3,07 + 8,72 + 12,80 + 8,72)/6 = 8,20$ mm setelah pembulatan. Untuk cabai A pada kondisi awal, *error* persentase adalah $|350,85 - 360|/360 \times 100\% = 2,54\%$.

Nilai tersebut menggambarkan kecocokan terhadap data kalibrasi yang sama. Hasil tidak digunakan untuk menyatakan akurasi pada cabai baru karena data pembentukan dan pemeriksaan model belum dipisahkan. Selain itu, model referensi yang memakai panjang fisik cabai tidak sama dengan estimasi *active monocular range* berdasarkan perubahan skala target yang sama. Kedua pendekatan tidak digabungkan sebagai satu hasil validasi.

4.4 Pengujian Integrasi Sistem Pemetikan Cabai Otomatis

4.4.1 Pengujian Urutan Kerja dan Durasi Siklus Pemetikan

Pengujian integrasi menjawab rumusan masalah ketiga melalui urutan persepsi, pemilihan target, koreksi visual, pendekatan, READY, dan pelaksanaan siklus pemetikan. Tiga tingkat hasil dibedakan: READY menandakan syarat pengendali terpenuhi; siklus program selesai menandakan urutan perintah selesai; keberhasilan fisik menandakan cabai dinyatakan berhasil terambil pada catatan pengujian. Penyelesaian perintah saja belum membuktikan objek benar-benar terpotong atau masuk ke wadah.

Kelompok pertama berisi sepuluh siklus A–J yang telah direkap sebagai siklus selesai. Kelompok kedua berisi tiga percobaan multi-cabai dengan tiga target per percobaan, termasuk kegagalan. Kedua kelompok tidak dijumlahkan karena daftar identitas *run* tidak cukup untuk membuktikan bahwa seluruh episodenya berbeda dan menggunakan kondisi identik.

Tabel 4.24 Error visual dan durasi sepuluh siklus integrasi

Uji	Err awal (px)	Error READY (px)	Terkoreksi (%)	Waktu READY (s)	Siklus (s)
A	103,39	19,42	81,22%	65,11	110,79
B	190,44	17,49	90,81%	38,30	84,55
C	107,04	20,12	81,20%	83,17	114,92
D	238,05	21,19	91,10%	37,21	68,90
E	222,73	2,00	99,10%	56,85	89,53
F	167,60	6,71	96,00%	38,30	74,40
G	288,92	14,42	95,01%	44,60	81,55
H	239,92	5,66	97,64%	52,81	91,02
I	292,10	28,30	90,31%	103,89	137,09
J	171,17	21,19	87,62%	39,49	72,74
Rata-rata	202,14	15,65	91,00%	55,97	92,55

Tabel 4.24 menunjukkan rata-rata error awal 202,14 piksel turun menjadi 15,65 piksel, dengan rata-rata koreksi 91,00%. Perhitungan memakai Persamaan (4.6) pada setiap siklus. Sebagai contoh, uji A menghasilkan $((103,39 - 19,42)/103,39) \times 100\% \approx 81,22\%$. Rentang koreksi 81,20–99,10% menunjukkan bahwa seluruh siklus yang dicantumkan mengurangi *error*, tetapi besarnya perbaikan bergantung pada kondisi awal dan *error* akhirnya.

Uji I memerlukan waktu READY terlama, yaitu 103,89 s, sedangkan uji D paling singkat sebesar 37,21 s. Data langkah pada Lampiran 6 menunjukkan uji I memerlukan 46 langkah dan sembilan pemulihan tepi, sedangkan uji D memerlukan sepuluh langkah tanpa pemulihan tepi. Perbandingan ini mendukung hubungan

antara banyaknya koreksi/pemulihan dan durasi, tetapi belum memisahkan pengaruh jarak awal serta kondisi *mask*.

$$\text{Waktu sesudah READY rata - rata} = 92,55 - 55,97 = 36,58 \text{ s} \quad (4.8)$$

Rata-rata siklus total adalah 92,55 s, sedangkan tahap menuju READY membutuhkan 55,97 s atau sekitar 60,47% dari rasio kedua rata-rata tersebut. Sisa rata-rata 36,58 s mencakup tahap setelah READY sampai siklus dinyatakan selesai. Dengan demikian, percepatan pemetikan memerlukan perhatian pada koreksi visual dan gerak sesudah READY; peningkatan FPS model saja tidak menjelaskan keseluruhan waktu siklus.

Kesepuluh entri pada kelompok ini tercatat selesai, tetapi data disajikan sebagai kumpulan siklus yang selesai. Nilai 10/10 tidak digunakan sebagai estimasi keberhasilan seluruh percobaan tanpa daftar seluruh upaya, termasuk upaya yang gagal. Keberhasilan fisik juga tidak dihitung dari kolom status program pada tabel ini.

Untuk mengetahui penyebab lamanya waktu siklus secara lebih rinci, dilakukan analisis terhadap raw log finite state machine (FSM) pada berkas fsm_experiment_20260920_015236.csv. Siklus representatif dimulai saat robot telah berada pada *scan pose*, dilanjutkan dengan pencarian target, *Visual Servoing* hingga READY, proses pemetikan, drop–release–home, dan gerakan kembali menuju *scan pose*. Data ini digunakan untuk melihat distribusi waktu setiap tahap, sedangkan nilai rata-rata 92,55 s pada **Tabel 4.23** tetap digunakan sebagai gambaran kinerja sepuluh siklus yang telah selesai.

Tabel 4.25 Analisis Durasi Tahapan Satu Siklus Pemetikan Representatif

Tahapan proses	Waktu awal (s)	Waktu akhir (s)	Durasi (s)	Kategori proses
Pencarian dan lock target	0,0477	0,1507	0,103	Pemrosesan visi
<i>Visual Servoing</i> hingga READY	0,1507	24,8910	24,740	Visi, komunikasi, dan gerak robot

Tahapan proses	Waktu awal (s)	Waktu akhir (s)	Durasi (s)	Kategori proses
Pembukaan <i>gripper</i>	24,8910	26,3932	1,502	Aktuator dan umpan balik
Gerak Z pasca-READY	26,3932	29,6892	3,296	Gerak robot
Final X insertion	29,6892	32,6302	2,941	Perencanaan, komunikasi, dan gerak
Penutupan <i>gripper</i>	32,6302	35,4318	2,802	Aktuator dan umpan balik
Retract sumbu X	35,4318	39,0419	3,610	Komunikasi dan gerak robot
Persiapan drop/home	39,0419	39,3242	0,282	FSM dan komunikasi
STM32 drop–release–home	39,3242	50,1062	10,782	Gerak fisik sekuensial
Kembali menuju <i>scan pose</i>	50,1063	67,3945	17,288	Gerak robot
Verifikasi <i>scan pose</i>	67,3945	67,6592	0,265	Verifikasi
Total siklus representatif	0,0477	67,6592	67,612	—

Berdasarkan **Tabel 4.25** , durasi terbesar pada siklus representatif terdapat pada fase *Visual Servoing* hingga READY sebesar 24,74 s, gerakan kembali menuju *scan pose* sebesar 17,29 s, dan proses drop–release–home sebesar 10,78 s. Sebaliknya, pencarian dan penguncian target hanya membutuhkan sekitar 0,10 s. Hasil tersebut menunjukkan bahwa proses deteksi target bukan merupakan penyumbang utama terhadap lamanya waktu siklus.

Fase *Visual Servoing* sebesar 24,74 s masih merupakan gabungan beberapa proses. Pada konfigurasi pengujian ini, *controller* menggunakan mekanisme *step-and-look* dengan tujuh *fresh observations* pada setiap iterasi. Setelah koreksi Cartesian dikirim, *controller* menunggu acknowledgement, status BUSY, dan MOTION_DONE sebelum mengambil observasi baru. Oleh karena itu, waktu *Visual Servoing* dipisahkan menjadi komponen gerak fisik robot, sinkronisasi perintah, proses pengamatan, dan konfirmasi READY.

Tabel 4.26 Distribusi Waktu pada Fase *Visual Servoing*

Komponen proses	Total waktu (s)	Persentase (%)
Gerak fisik robot (MOTION_BUSY → MOTION_DONE)	12,518	50,60
Sinkronisasi perintah (MOTION_TX → MOTION_ACK)	7,588	30,70
<i>Fresh frame</i> , settling, dan keputusan <i>controller</i>	4,074	16,50
Konfirmasi kondisi READY	0,560	2,30
Transisi ACK → BUSY	±0,001	±0,00
Total	24,740	100,00

Tabel 4.26 menunjukkan bahwa waktu *Visual Servoing* tidak seluruhnya digunakan untuk menghitung *error* citra dan *empirical Jacobian*. Sebesar 50,60% waktu digunakan ketika robot melakukan gerakan fisik pada interval MOTION_BUSY hingga MOTION_DONE, sedangkan 30,70% digunakan pada interval MOTION_TX hingga MOTION_ACK. Pengumpulan *fresh frame*, settling, dan pengambilan keputusan *controller* memberikan kontribusi sebesar 16,50%. Dengan demikian, sekitar 81,30% waktu fase *Visual Servoing* berkaitan dengan eksekusi gerak dan sinkronisasi perintah dengan STM32.

Interval MOTION_BUSY hingga MOTION_DONE menunjukkan waktu ketika manipulator secara fisik masih bergerak menuju target Cartesian. MOTION_DONE bukan penyebab keterlambatan, melainkan penanda bahwa gerakan telah selesai. Interval MOTION_TX hingga MOTION_ACK menunjukkan waktu sejak Jetson mengirim perintah sampai STM32 mengakui perintah. Interval ini tidak dapat langsung dinyatakan sebagai waktu transmisi UART saja karena masih dapat mencakup pemrosesan perintah dan sinkronisasi pada sisi STM32.

Optimasi lanjutan dilakukan pada mekanisme *step-and-look* dengan mengurangi jumlah *fresh observations* dan *settling time*. Pengujian ini merupakan pengujian konfigurasi *controller* dan tidak digabungkan ke dalam rerata sepuluh siklus A–J pada **Tabel 4.23**.

Tabel 4. 27 Perbandingan Respons *Step-and-look* Sebelum dan Sesudah Optimasi

Parameter	Konfigurasi awal	Konfigurasi optimasi	Perubahan
<i>Fresh observations</i> per iterasi	7 frame	3 frame	-57,14%
<i>Settling time</i>	250 ms	100 ms	-60,00%
Median MOTION_DONE → next MOTION_TX	0,477 s	0,186 s	-61,01%
Arsitektur kontrol	<i>Step-and-look</i>	<i>Step-and-look</i>	Tetap
Konfirmasi gerak	ACK → BUSY → DONE	ACK → BUSY → DONE	Tetap

Pengurangan *fresh observations* dari tujuh menjadi tiga frame dan *settling time* dari 250 ms menjadi 100 ms menurunkan median waktu MOTION_DONE hingga perintah berikutnya dari sekitar 0,477 s menjadi 0,186 s atau sekitar 61,01%. Hasil ini menunjukkan bahwa waktu tunggu pada tahap pengamatan dapat dikurangi tanpa mengubah arsitektur dasar *step-and-look*. Meskipun demikian, waktu gerak fisik manipulator dan sinkronisasi perintah tetap menjadi faktor yang lebih dominan.

Secara keseluruhan, kecepatan pemrosesan visi secara *real-time* tidak secara langsung menghasilkan siklus pemetikan yang cepat. *Bottleneck* utama sistem terletak pada gerakan fisik manipulator, sinkronisasi perintah Jetson–STM32, pengulangan proses *step-and-look*, serta urutan drop/home dan return-to-scan. Temuan ini menjadi dasar pengembangan selanjutnya berupa optimasi profil gerak manipulator, pengurangan waktu respons perintah STM32, dan penerapan kontrol visual yang lebih kontinu sehingga koreksi dapat diperbarui tanpa menunggu MOTION_DONE pada setiap iterasi.

4.4.2 Pengujian pemetikan multi-cabai

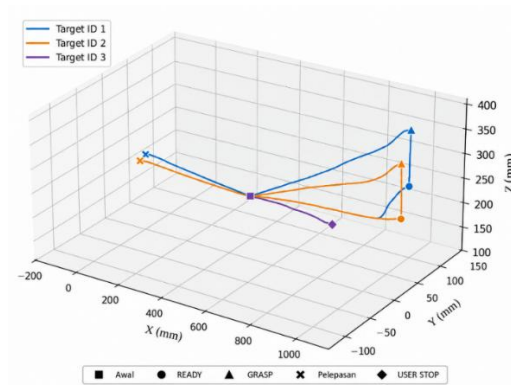
Pengujian ini dilakukan dengan 3 buah cabai untuk mengevaluasi kemampuan sistem terintegrasi dalam mendeteksi target, melakukan penyelarasan *Visual Servoing*, mencapai kondisi READY, dan menjalankan urutan pemetikan pada

beberapa cabai dengan ukuran berbeda. Tiga kali pengujian digunakan, yaitu Trial 1, Trial 2 dan Trial 3. Setiap pengujian melibatkan tiga cabai dengan variasi ukuran kecil, sedang, dan panjang. Berdasarkan catatan pengujian, ukuran tersebut masing-masing adalah 80 cm, 90 cm, dan 100 cm. Setiap target diproses secara berurutan dari kondisi awal, tahap *approach*, penyeselarasan Y-Z, kondisi READY, hingga urutan GRASP dan pemindahan apabila kondisi pengendalian dapat diselesaikan.

Keberhasilan pada pengujian ini dibedakan menjadi dua. Keberhasilan program menyatakan bahwa target mencapai READY dan urutan kendali mencapai INTEGRATED_CYCLE_DONE. Keberhasilan pemetikan fisik menyatakan bahwa cabai benar-benar dapat dijangkau dan terambil oleh *gripper* berdasarkan dokumentasi pengujian. Pemisahan kedua indikator ini diperlukan karena satu siklus program dapat selesai meskipun secara fisik cabai tidak berhasil terambil.

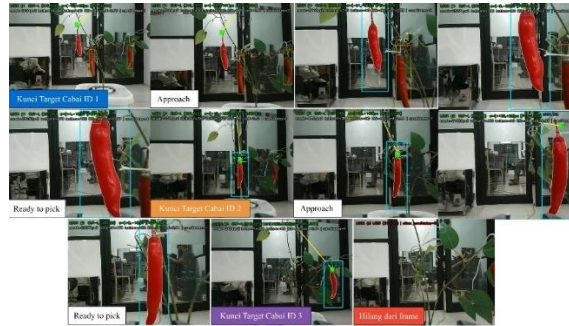
a. Hasil Trial 1

.



Gambar 4. 22 Trajectory Multi-Cabai Trial 1

Gambar 4.19 memperlihatkan tiga lintasan target dengan penanda awal, READY, GRASP, pelepasan, dan penghentian pengguna. Dua target mencapai rangkaian tahap pemetikan, sedangkan lintasan target ketiga berakhir pada penghentian. Lintasan berasal dari pose yang direkam sistem dan menunjukkan urutan gerak; gambar tersebut tidak mengukur kesalahan posisi TCP terhadap pelacakan eksternal.

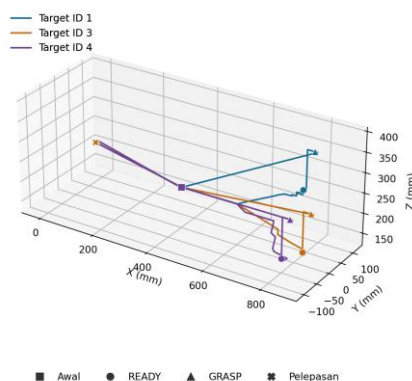


Gambar 4. 23 Tahapan *Visual Servoing* pada Pengujian Pemetikan Multi-Target Cabai Trial 1

Gambar 4.20 memperlihatkan dua target pertama melalui penguncian, pendekatan, dan READY. Pada target ketiga, dokumentasi berakhir pada kondisi hilang dari bingkai. Catatan akhir menyatakan *error* horizontal -104 piksel, *EDGE_X_HOLD* aktif, kemudian *LOST_HOLD* dan *USER_STOP* sebelum READY. Urutan ini menunjukkan bahwa kehilangan visibilitas menghentikan proses sebelum syarat pemetikan terpenuhi.

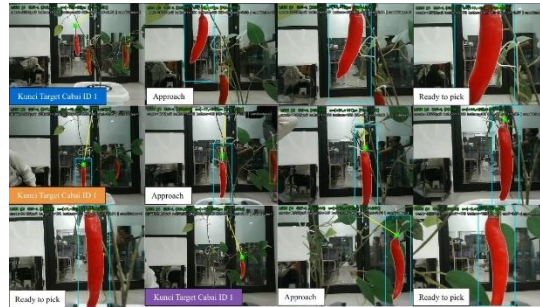
Dua dari tiga target dinyatakan berhasil terambil pada catatan percobaan pertama, sehingga keberhasilan fisiknya $(2/3) \times 100\% = 66,67\%$. Hasil ini memperlihatkan perlunya koreksi yang menjaga target tetap berada dalam bidang pandang saat gerak maju. Kegagalan tersebut tidak dapat disebut kesalahan deteksi READY karena target ketiga belum mencapai status itu.

b. Hasil Trial-2



Gambar 4. 24 Trajectory Multi-Cabai Trial 2

Gambar 4.21 memperlihatkan lintasan target ID 1, 3, dan 4 dengan penanda tahapan yang lengkap. Perbedaan ID tidak berarti urutan target harus bernomor berdekatan; identitas mengikuti hasil asosiasi objek dalam sistem. Ketiga lintasan mendukung bahwa pengendali menjalankan pendekatan dan transisi tahap untuk tiga target yang berbeda.

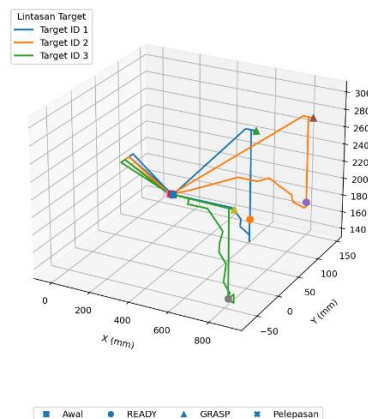


Gambar 4. 25 Tahapan *Visual Servoing* pada Pengujian Pemetikan Multi-Target Cabai Trial 2

Gambar 4.22 memperlihatkan urutan penguncian, pendekatan, dan READY untuk ketiga target. Bersama lintasan pada **Gambar 4.21**, dokumentasi ini menunjukkan keberulangan urutan persepsi dan gerak dalam satu percobaan multi-target. Rekap hasil mencatat tiga READY, tiga siklus program selesai, dan tiga hasil pemetikan fisik berhasil, sehingga rasio hasil fisik menurut catatan adalah 100% pada tiga target tersebut.

Angka 100% berlaku untuk percobaan kedua dengan ukuran sampel tiga target. Gambar antarmuka terutama membuktikan tahapan visual dan gerak, sehingga keberhasilan fisik tetap merujuk pada rekap pengamatan. Kemandirian pemotongan tanpa bantuan operator tidak dapat dihitung hanya dari cuplikan READY; penilaian tersebut memerlukan dokumentasi lengkap selama pemotongan dan pelepasan.

c. Trial 3



Gambar 4. 26 Trajectory Multi-Cabai Trial 3

Gambar 4.23 memperlihatkan tiga lintasan yang mencapai tahapan READY, GRASP, dan pelepasan. Secara logika program, ketiga target melanjutkan siklus sampai selesai. Posisi akhir lintasan berbeda antartarget, sesuai kebutuhan gerak menuju target yang berbeda; keberadaan penanda GRASP atau pelepasan tetap merupakan status sistem, bukan pengukuran langsung bahwa tangkai telah terpotong.



Gambar 4. 27 Tahapan *Visual Servoing* pada Pengujian Pemetikan Multi-Target Cabai Trial 3

Gambar 4.24 menunjukkan ketiga target mencapai tampilan READY. Meskipun demikian, rekam pengamatan menyatakan hanya dua cabai berhasil terambil. Catatan kegagalan mengaitkan hasil cabai panjang dengan posisi longitudinal *gripper* yang tidak sesuai terhadap titik potong. Dengan demikian, kesesuaian fitur dua dimensi pada citra belum cukup untuk memastikan kesesuaian posisi mekanis pemotong pada arah kedalaman.

Tabel 4.28 Rekapitulasi Pengujian Multi-Cabai

Trial	Jumlah target	READY	Siklus program selesai	Pemetikan fisik berhasil	Keberhasilan fisik
1	3	2	2	2	66,67%
2	3	3	3	3	100,00%
3	3	3	3	2	66,67%
Total	9	8	8	7	77,78%

Dari sembilan episode pada tiga pengujian, delapan episode mencapai READY dan menyelesaikan siklus program, sehingga tingkat penyelesaian program sebesar 88,89% dan kegagalan program sebesar 11,11%. Jika keberhasilan dinilai berdasarkan hasil pemetikan fisik, tujuh dari sembilan cabai berhasil terambil. Dengan demikian, keberhasilan pemetikan fisik sebesar 77,78% dan persentase kegagalan fisik sebesar 22,22%.

Dua kegagalan memiliki karakter yang berbeda. Kegagalan Trial 1 merupakan kegagalan persepsi dan pengendalian karena target keluar dari bidang pandang sebelum mencapai READY. Sebaliknya, kegagalan trial 3 terjadi setelah pengendali berhasil mencapai READY dan menyelesaikan siklus program, tetapi posisi longitudinal *gripper* tidak sesuai dengan titik potong cabai panjang. Perbedaan tersebut menunjukkan bahwa peningkatan sistem tidak hanya memerlukan mekanisme *reacquire* dan penjagaan target yang lebih kuat, tetapi juga memerlukan penentuan kedalaman atau langkah X yang mempertimbangkan ukuran dan geometri cabai.