EVALUASI PENGARUH VARIASI TINGKAT ANOTASI TERHADAP PERFORMA DAN EFISIENSI MODEL YOLOv11 PADA DETEKSI MULTI-OBJEK BUAH



Disusun Oleh:

N a m a : Nugroho Fadillah Yudha Putra
NIM : 18523156

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

EVALUASI PENGARUH VARIASI TINGKAT ANOTASI TERHADAP PERFORMA DAN EFISIENSI MODEL YOLOv11 PADA DETEKSI MULTI-OBJEK BUAH

TUGAS AKHIR

Disusun Oleh:

N a m a : Nugoho Fadillah Yudha Putra

NIM : 18523156

Yogyakarta, 4 Agustus 2025

Pembimbing,

(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

EVALUASI PENGARUH VARIASI TINGKAT ANOTASI TERHADAP PERFORMA DAN EFISIENSI MODEL YOLOv11 PADA DETEKSI MULTI-OBJEK BUAH

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 4 Agustus 2025

Tim Penguji

Arrie Kurniawardhani, S.Si., M.Kom. _____

Anggota 1

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. _____

Anggota 2

Rahadian Kurniawan, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nugroho Fadillah Yudha Putra

NIM : 18523156

Tugas akhir dengan judul:

EVALUASI PENGARUH VARIASI TINGKAT ANOTASI TERHADAP PERFORMA DAN EFISIENSI MODEL YOLOv11 PADA DETEKSI MULTI-OBJEK BUAH

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 4 Agustus 2025



(Nugroho Fadillah Yudha Putra)

HALAMAN PERSEMBAHAN

Alhamdulillahi rabbil ‘alamin, segala puji penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, kesehatan, serta kemudahan yang diberikan selama proses penyusunan hingga penyelesaian tugas akhir ini. Semoga karya sederhana ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi para pembaca.

Dengan penuh rasa cinta dan hormat, karya ini penulis persembahkan kepada:

1. **Ayah dan Ibu tercinta, Bapak Zakaria dan Ibu Yuliana**, yang tiada henti memberikan doa, dukungan, dan kasih sayang. Setiap nasihat, pengorbanan, dan kesabaran yang telah diberikan menjadi kekuatan utama bagi penulis untuk terus berjuang meraih cita-cita.
2. **Dosen Pembimbing, Ibu Arrie Kurniawardhani**, atas segala waktu, tenaga, serta bimbingan berharga yang diberikan. Segala nasihat dan arahan Ibu sangat berarti dalam penyelesaian penelitian ini.
3. **Khamidah Ferista dan Muhammad Lutfi Sirajul Huda**, yang senantiasa memberikan dukungan, semangat, dan kepercayaan, serta menemani penulis melewati masa-masa sulit dalam menyelesaikan perjalanan akademik ini.
4. **Rekan-rekan seperjuangan**, yang selalu hadir dengan dukungan, tawa, dan cerita, serta menjadi bagian dari perjalanan yang penuh warna ini.
5. **Yang Terakhir,** Kepada diri saya sendiri yang selalu percaya dan tetap berusaha, walaupun perasaan malas yang sering melanda. Tapi setidaknya tidak berhenti untuk berusaha.

Berikut halaman persembahan ini dibuat semoga dengan adanya penelitian yang ditulis oleh penulis dapat bermanfaat bagi para pembaca sekalian, dan dapat memberikan ilmu pengetahuan yang baru terhadap dunia pendidikan maupun penelitian.

HALAMAN MOTO

"*The world is full of lonely people afraid to make the first move.*"
— Tony Lip (*The Green Book*)

"*Courage is knowing the right thing to do and doing it, even when it's hard.*"
— Dr. Don Shirley (*The Green Book*)

"Each time I find myself flat on my face, I pick myself up and get back in the race."
— Frank Sinatra (*That's Life*)

KATA PENGANTAR

Alhamdulillahi rabbil ‘alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, karunia, dan kemudahan yang diberikan sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“Evaluasi Pengaruh Variasi Tingkat Anotasi Terhadap Performa Dan Efisiensi Model Yolov11 Pada Deteksi Multi-Objek Buah”**. Penulisan skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika.

Dalam penyusunan laporan ini, penulis menyadari bahwa penyelesaian skripsi ini tidak terlepas dari doa, dukungan, serta bantuan berbagai pihak. Oleh karena itu, dengan penuh rasa hormat, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. **Allah SWT**, atas limpahan rahmat, karunia, dan petunjuk-Nya yang senantiasa menguatkan penulis dalam menghadapi berbagai tantangan selama penyusunan skripsi ini.
2. **Kedua orang tua tercinta, Bapak Zakaria dan Ibu Yuliana**, atas doa, kasih sayang, pengorbanan, dan dukungan tanpa batas yang menjadi sumber kekuatan terbesar dalam perjalanan hidup penulis.
3. **Ibu Arrie Kurniawardhani**, selaku dosen pembimbing sekaligus dosen pembimbing akademik, yang dengan sabar memberikan bimbingan, arahan, dan motivasi hingga skripsi ini dapat terselesaikan.
4. **Khamidah Ferista dan Muhammad Lutfi Sirajul Huda**, yang selalu memberikan semangat, dukungan moral, serta menjadi tempat berbagi di saat-saat sulit maupun bahagia.
5. **Teman-teman Discord Avocado dan KIT(Kita Ini Teman)**, atas kebersamaan, candaan, dan dukungan yang menghibur serta memberi motivasi dalam perjalanan penyusunan tugas akhir ini.
6. **Ibu Sulis dan Mas Dhoni selaku Staff Jurusan Informatika,** yang membantu penulis dalam hal administratif, yang membuat penulis dapat mengikuti sidang.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat penulis harapkan. Semoga skripsi ini dapat memberikan manfaat, baik bagi penulis sendiri maupun pembaca yang tertarik pada bidang deteksi objek berbasis *computer vision*.

Yogyakarta, 4 Agustus 2025



(Nugroho Fadillah Yudha Putra)

SARI

Perkembangan teknologi *computer vision* telah mendorong pemanfaatan model deteksi objek di berbagai bidang, termasuk pada sektor pertanian digital. Salah satu algoritma deteksi objek yang banyak digunakan adalah YOLO (*You Only Look Once*), yang dikenal karena kecepatan inferensi tinggi dan akurasi yang kompetitif. Kinerja model ini sangat dipengaruhi oleh kualitas serta kelengkapan anotasi pada data pelatihan. Namun, proses anotasi penuh pada dataset berukuran besar sering kali menjadi tantangan karena sifatnya yang repetitif, memakan waktu, dan membutuhkan sumber daya manusia yang signifikan.

Penelitian ini bertujuan untuk mengevaluasi pengaruh variasi tingkat anotasi terhadap kinerja dan efisiensi komputasi model deteksi objek berbasis YOLOv11. Eksperimen dilakukan dengan melatih model pada lima variasi tingkat anotasi, yaitu 100%, 90%, 80%, 70%, dan 60%, menggunakan dataset enam kelas buah (anggur, apel, jeruk, rambutan, salak, dan stroberi) yang dikembangkan secara mandiri. Dua skenario pengurangan anotasi diterapkan, yaitu: (a) penghapusan anotasi pada objek yang jelas dan utuh (Skenario A), serta (b) penghapusan anotasi pada objek yang tertumpuk atau terpotong (Skenario B). Evaluasi dilakukan menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*, disertai analisis hasil deteksi visual dengan aplikasi berbasis Streamlit. Selain itu, waktu pelatihan (*training time*) juga dianalisis untuk menilai aspek efisiensi komputasi.

Hasil pengujian menunjukkan bahwa penurunan tingkat anotasi berdampak langsung pada penurunan performa model, baik dari sisi metrik maupun hasil deteksi visual. Model dengan tingkat anotasi ≥80% masih mempertahankan performa yang stabil dengan penurunan <10% dari baseline, sedangkan model dengan anotasi 60% menunjukkan degradasi signifikan pada jumlah deteksi dan tingkat kepercayaan. Dari segi efisiensi komputasi, model dengan tingkat anotasi 80% pada Skenario B terbukti mampu menghemat waktu pelatihan sekitar 9% dibanding baseline, sekaligus menjaga performa metrik yang tetap berada di atas ambang batas toleransi. Temuan ini menegaskan bahwa strategi anotasi parsial sebesar ≥80%, khususnya dengan pendekatan Skenario B, dapat dipertimbangkan sebagai alternatif efisien dalam mengurangi beban kerja anotasi tanpa mengorbankan akurasi secara signifikan.

Kata kunci: YOLOv11, deteksi objek, anotasi dataset, *computer vision*, efisiensi komputasi, evaluasi model.

GLOSARIUM

Glosarium memuat daftar kata tertentu yang digunakan dalam laporan dan membutuhkan penjelasan, misalnya kata serapan yang belum lazim digunakan. Urutkan sesuai abjad. Contoh penulisannya seperti di bawah ini:

Anotasi	Proses pelabelan objek pada gambar dengan memberi tanda, seperti *bounding box*, untuk menandai lokasi dan kelas objek.
API	Antarmuka yang memungkinkan aplikasi berinteraksi dengan layanan atau perangkat lunak lain melalui serangkaian perintah terstruktur.
Bounding Box	Kotak persegi panjang yang digunakan untuk menandai area objek tertentu pada gambar untuk keperluan deteksi atau pelatihan model.
Confidence Score	Nilai probabilitas yang menunjukkan tingkat keyakinan model bahwa deteksi yang dihasilkan sesuai dengan kelas objek tertentu.
Confusion Matrix	Tabel yang menunjukkan perbandingan antara hasil prediksi model dengan label sebenarnya, digunakan untuk menilai akurasi model klasifikasi atau deteksi.
Dataset	Kumpulan data (misalnya gambar atau teks) yang digunakan untuk melatih, memvalidasi, dan menguji model pembelajaran mesin.
Deep Learning	Cabang dari pembelajaran mesin (*machine learning*) yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari pola data secara kompleks.
Google Colab	Platform berbasis *cloud* yang digunakan untuk menjalankan kode Python dengan dukungan GPU secara gratis, sering digunakan untuk eksperimen AI dan *deep learning*.
Roboflow	Platform *web-based* yang menyediakan layanan anotasi gambar, konversi format dataset, serta integrasi dengan model deteksi objek.
Streamlit	Framework berbasis Python untuk membangun aplikasi *web* interaktif yang sering digunakan dalam visualisasi data dan hasil model AI.
YOLO	Arsitektur deteksi objek berbasis *deep learning* yang terkenal karena kecepatan inferensinya yang tinggi dan akurasi kompetitif.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiv
DAFTAR GAMBAR xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 6
1.5 Manfaat Penelitian 6
1.6 Metodologi Penelitian 8
1.7 Sistematika Penulisan 10
BAB II TINJAUAN PUSTAKA 11
2.1 Definisi dan Evolusi Deteksi Objek: Fokus YOLO dan YOLOv11 11
2.2 Arsitektur YOLOv11 12
2.3 YOLO untuk Deteksi Buah 14
2.4 Teknologi yang Digunakan 15
2.5 Penelitian Terdahulu dan Gap Penelitian 17
2.6 Kontribusi Penelitian 18
BAB III METODOLOGI PENELITIAN 21
3.1 Metode Penelitian 21
3.2 Alur Penelitian 21
3.3 Variabel Penelitian 25
3.3.1 Variabel Bebas (*Independent Variable*) 25
3.3.2 Variabel Terikat (*Dependent Variables*) 26
3.3.3 Variabel Kendali (Control Variables) 26
3.3.4 Definisi Operasional *Threshold* Performa 27
3.4 Dataset dan Objek Penelitian 28
3.4.1 Kelas Objek 29
3.4.2 Sumber dan Jumlah Data 29
3.4.3 Proses Anotasi 30
3.4.4 Variasi Tingkat Anotasi 30
3.4.5 Distribusi Dataset Awal 30
3.4.6 Integrasi Platform Eksperimen 31
3.4.7 Relevansi Dataset terhadap Tujuan Penelitian 31
3.5 Teknik Pengumpulan Data 31
3.5.1 Pengumpulan Gambar 32
3.5.2 Proses Anotasi dan Pembuatan Skenario 32
3.5.3 Split Dataset 34

3.5.4 Preprocessing 35
3.5.5 Ekspor Dataset 36
3.6 Teknik Pengurangan Anotasi 37
3.6.1 Skenario Pengurangan Anotasi 38
3.6.2 Prinsip Pengurangan Proporsional 40
3.6.3 Contoh Ilustratif 40
3.6.4 Validasi Proporsi 40
3.6.5 Catatan Bias dan Relevansi Metodologis 41
3.7 Pemilihan Model 41
3.8 Teknik Pelatihan Model 44
3.8.1 Hyperparameter Pelatihan 45
3.8.2 Output Pelatihan 45
3.9 Teknik Analisis Data 46
3.10 Penyimpanan Model 47
3.11 Evaluasi Model 48
3.11.1 Metrik Evaluasi Kuantitatif 48
3.11.2 Prinsip Penentuan Ambang Batas (Threshold) 49
3.11.3 Relevansi Evaluasi terhadap Strategi Pengurangan Anotasi 50
3.11.4 Rangkuman Evaluasi 51
3.12 Pengujian Model 51
3.13 Analisis Hasil 52
3.13.1 Fokus Analisis 52
3.13.2 Metrik Evaluasi yang Digunakan 52
3.13.3 Visualisasi Hasil Evaluasi 53
3.13.4 Integrasi Temuan ke Pembahasan 53
3.14 Proof of Concept 54
3.14.1 Inisialisasi dan Import Library 54
3.14.2 Mapping Model YOLOv11 55
3.14.3 Fungsi *Load Model* 55
3.14.4 Upload Gambar dan Tampilan Awal 55
3.14.5 Proses Deteksi dan Visualisasi 56
3.14.6 Keunggulan PoC 57
BAB IV HASIL DAN PEMBAHASAN 58
4.1 Hasil Pelatihan Model 58
4.1.1 Visualisasi Hasil Pelatihan 58
4.1.2 Tren Penurunan Performansi 62
4.1.3 Rekapitulasi Nilai Metrik Pelatihan 64
4.1.4 Waktu Pelatihan Model 66
4.2 Hasil Pengujian Model 68
4.2.1 Analisis Confusion Matrix 68
4.2.2 Hasil Uji Deteksi pada Aplikasi 89
4.3 Analisis Penurunan Performa Model 100
4.3.1 Analisis Penurunan Performa Model Berdasarkan Metrik Evaluasi 100
4.3.2 Kelas Buah yang Paling Terdampak 105
4.3.3 Penentuan Ambang Batas Efektif Anotasi 110
4.4 Rekomendasi Model dan Skenario Terbaik 114
BAB V KESIMPULAN DAN SARAN 117
5.1 Kesimpulan 117
5.2 Saran 119
LAMPIRAN 121

DAFTAR PUSTAKA 122

DAFTAR TABEL

Tabel 3.1 Ringkasan Variabel Penelitian 27
Tabel 3.2 Distribusi Dataset Awal per Kelas 30
Tabel 3.3 Tabel perbandingan performa dan spesifikasi empat varian YOLOv11 43
Tabel 4.1 Ringkasan Nilai *Precision, Recall, mAP@0.5,* dan *mAP@0.5:0.95* pada Setiap Tingkat Anotasi 64
Tabel 4.2 Waktu Pelatihan Model YOLOv11 pada Berbagai Tingkat Anotasi 66
Tabel 4.3 Rekapitulasi Prediksi Benar (True Positive) pada Confusion Matrix 87
Tabel 4.4 Jumlah Deteksi dan Rata-rata Confidence dari Masing-masing Model pada Satu Gambar Uji 98
Tabel 4.5 Ringkasan Nilai Precision, Recall, mAP@0.5, mAP@0.5:0.95, Jumlah Deteksi, dan Confidence Rate 101
Tabel 4.6 Ringkasan Akurasi Class-wise (Anggur, Jeruk, Stroberi) pada Variasi Tingkat Anotasi 106
Tabel 4.7 Nilai mAP@0.5 Tiap Model dan Evaluasi terhadap Ambang Batas 111
Tabel 4.8 Ringkasan Kandidat dan Model Rekomendasi 114

DAFTAR GAMBAR

Gambar 2.1 Arsitektur YOLOv11 14
Gambar 3.1 Diagram Alur Penelitian 23
Gambar 3.2 Diagram Integrasi Platform Roboflow, Google Colab, dan VSCode dalam Alur Penelitian. 31
Gambar 3.3 Anotasi untuk Dataset 100% 32
Gambar 3.4 Anotasi dataset 60 % Skenario A 34
Gambar 3.5 Anotasi dataset 60 % Skenario B 34
Gambar 3.6 Pembagian Dataset pada Platform Roboflow (Train/Valid/Test). 35
Gambar 3.7 Proses Pemilihan Metode *Preprocessing* di *Roboflow* 35
Gambar 3.8 Cuplikan kode dari Roboflow untuk mengakses dataset dalam format YOLOv11, yang digunakan untuk integrasi langsung ke Google Colab. 37
Gambar 3.9 Kode Python yang digunakan untuk mengunduh dataset dari *Roboflow* ke *Google Colab* menggunakan API. 37
Gambar 3.10 Antarmuka pemilihan model YOLOv11 di Roboflow sebagai bagian dari konfigurasi ekspor dataset 43
Gambar 3.11 Perintah pelatihan YOLOv11 dengan *hyperparameter* penelitian. 45
Gambar 3.12 Kode program untuk penyimpanan model 48
Gambar 3.13 Kode menampilkan *confusion matrix* pada Google Colab 49
Gambar 3.14 Kode menampilkan *chart* hasil *training model* 49
Gambar 3.15 Kode untuk melakukan *testing* model 51
Gambar 3.16 Import *Library* yang Diperlukan Sistem 54
Gambar 3.17 Mapping Model yang Digunakan pada Pengujian di Aplikasi. 55
Gambar 3.18 Fungsi untuk *Load Model* YOLOv11. 55
Gambar 3.19 Tampilan Awal dan *Upload* Gambar oleh Pengguna 56
Gambar 3.20 Proses Deteksi dengan Semua Model 57
Gambar 4.1 Grafik hasil pelatihan model baseline 100% anotasi. Menunjukkan konvergensi loss yang stabil dan capaian metrik tertinggi. 58
Gambar 4.2 Grafik hasil pelatihan model Skenario A - 90%. Masih menunjukkan performa mendekati baseline. 59
Gambar 4.3 Grafik hasil pelatihan model Skenario A - 80%. Terlihat penurunan metrik dan fluktuasi nilai loss mulai meningkat. 59

Gambar 4.4 Grafik hasil pelatihan model Skenario A - 70%. Penurunan performa mulai signifikan. 60
Gambar 4.5 Grafik hasil pelatihan model Skenario A - 60%. Terdapat penurunan tajam pada seluruh metrik evaluasi. 60
Gambar 4.6 Grafik hasil pelatihan model Skenario B - 90%. Menunjukkan metrik precision tertinggi dibandingkan seluruh model. 61
Gambar 4.7 Grafik hasil pelatihan model Skenario B - 80%. Performa masih stabil dan berada dalam ambang toleransi. 61
Gambar 4.8 Grafik hasil pelatihan model Skenario B - 70%. Metrik menurun secara bertahap tetapi belum drastis. 62
Gambar 4.9 Grafik hasil pelatihan model Skenario B - 60%. Penurunan mulai menyentuh batas toleransi 10%. 62
Gambar 4.10 Grafik Perbandingan Nilai *Precision*, *Recall*, *mAP@0.5*, dan *mAP@0.5:0.95* pada Setiap Model YOLOv11 dengan Variasi Tingkat Anotasi 65
Gambar 4.11 Grafik Tren Waktu Pelatihan Model YOLOv11 pada Berbagai Tingkat Anotasi 67
Gambar 4.12 Confusion Matrix Model YOLOv11 dengan Anotasi 100%. 70
Gambar 4.13 Confusion Matrix Model YOLOv11 dengan Anotasi 90% (Skenario A). 72
Gambar 4.14 Confusion Matrix Model YOLOv11 dengan Anotasi 80% (Skenario A). 74
Gambar 4.15 Confusion Matrix Model YOLOv11 dengan Anotasi 70% (Skenario A). 76
Gambar 4.16 Confusion Matrix Model YOLOv11 dengan Anotasi 60% (Skenario A). 78
Gambar 4.17 Confusion Matrix Model YOLOv11 dengan Anotasi 90% (Skenario B). 80
Gambar 4.18 Confusion Matrix Model YOLOv11 dengan Anotasi 80% (Skenario B). 82
Gambar 4.19 Confusion Matrix Model YOLOv11 dengan Anotasi 70% (Skenario B). 84
Gambar 4.20 Confusion Matrix Model YOLOv11 dengan Anotasi 60% (Skenario B). 86
Gambar 4.21 Barchart Prediksi Benar (*True Positive*) Antar Model 88
Gambar 4.22 Gambar uji yang digunakan untuk pengujian model YOLOv11 89
Gambar 4.23 Deteksi oleh model YOLOv11_100 (Baseline) 90
Gambar 4.24 Deteksi oleh model YOLOv11_90 (Skenario A) 91
Gambar 4.25 Deteksi oleh model YOLOv11_80 (Skenario A) 92
Gambar 4.26 Deteksi oleh model YOLOv11_70 (Skenario A) 93
Gambar 4.27 Deteksi oleh model YOLOv11_60 (Skenario A) 94
Gambar 4.28 Deteksi oleh model YOLOv11_90 (Skenario B) 95
Gambar 4.29 Deteksi oleh model YOLOv11_80 (Skenario B) 96

Gambar 4.30 Deteksi oleh model YOLOv11_70 (Skenario B) 97
Gambar 4.31 Deteksi oleh model YOLOv11_60 (Skenario B) 98
Gambar 4.32 Grafik Tren Penurunan Precision, Recall, mAP@0.5, Avg Confidence, dan Training Time Model YOLOv11 Berdasarkan Variasi Tingkat Anotasi (Skenario A dan B) 104
Gambar 4.33 Tren Penurunan Akurasi per Kelas terhadap Tingkat Anotasi pada Skenario A dan Skenario B 108
Gambar 4.35 Grafik Penentuan Ambang Batas Efektif Anotasi Berdasarkan mAP@0.5 112

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi *computer vision* telah melahirkan berbagai inovasi dalam pengenalan visual, seperti deteksi objek, klasifikasi gambar, dan pelacakan objek secara otomatis. Teknologi ini memiliki peran penting di berbagai sektor, mulai dari industri manufaktur, transportasi, kesehatan, hingga pertanian. Khusus pada sektor pertanian, deteksi buah secara otomatis dari citra digital menjadi bagian integral dari sistem otomasi modern yang digunakan dalam proses panen, klasifikasi mutu produk, dan pemantauan hasil produksi. Salah satu algoritma yang menonjol dalam bidang ini adalah *You Only Look Once* (YOLO), yaitu pendekatan deteksi objek berbasis *deep learning* yang dikenal karena keunggulannya dalam kecepatan inferensi serta akurasi yang kompetitif (Bochkovskiy, Wang, & Liao, 2020).

Kinerja algoritma deteksi objek sangat bergantung pada kualitas data pelatihan, khususnya kelengkapan dan konsistensi anotasi objek dalam gambar. Anotasi dalam konteks ini merujuk pada proses pemberian label dan *bounding box* terhadap objek-objek yang relevan. Proses tersebut umumnya dilakukan secara manual dan membutuhkan ketelitian, konsistensi, serta alokasi waktu yang besar. Aktivitas anotasi yang repetitif sering kali menimbulkan kelelahan pada anotator, sehingga berpotensi menghasilkan ketidakkonsistenan atau ketidakakuratan dalam pelabelan objek (Ibrahem, Salem, & Kang, 2021). Kondisi ini dapat menyebabkan model kehilangan sebagian informasi penting pada tahap pelatihan, yang berdampak langsung pada kemampuan deteksi objek.

Dalam penelitian ini, dataset terlebih dahulu dianotasi secara penuh (100%). Selanjutnya, variasi tingkat anotasi (90%, 80%, 70%, dan 60%) diperoleh melalui pengurangan sebagian anotasi dari dataset lengkap tersebut. Dengan demikian, penelitian ini tidak secara langsung mengukur efisiensi waktu anotasi manual, melainkan menitikberatkan pada dampak pengurangan anotasi terhadap performa model. Di samping itu, pengurangan jumlah anotasi juga berimplikasi pada efisiensi komputasi. Semakin sedikit jumlah *bounding box* yang diproses selama pelatihan, semakin singkat pula waktu yang diperlukan untuk menyelesaikan training. Aspek ini menjadi penting, terutama dalam konteks keterbatasan sumber daya GPU yang umum terjadi pada platform berbasis *cloud computing* seperti Google Colab.

Untuk memperoleh gambaran yang lebih komprehensif, penelitian ini merancang dua skenario pengurangan anotasi. Skenario A dilakukan dengan menghapus anotasi pada objek

yang tampak jelas, utuh, dan tidak tertutup, sehingga objek yang saling bertumpuk atau terpotong tetap dipertahankan. Sebaliknya, Skenario B menghapus anotasi pada objek yang saling menutupi, tertumpuk, atau hanya terlihat sebagian, sementara objek yang jelas tetap dipertahankan. Kedua skenario ini mencerminkan situasi yang umum dijumpai dalam anotasi manual, khususnya pada citra dengan banyak objek atau posisi objek yang kurang ideal. Analisis terhadap kedua pendekatan tersebut diharapkan mampu memberikan wawasan yang lebih luas mengenai strategi anotasi yang efisien namun tetap efektif.

Objek penelitian mencakup enam kelas buah, yakni jeruk, apel, stroberi, anggur, rambutan, dan salak. Pemilihan keenam kelas buah ini didasarkan pada dua alasan utama. Pertama, buah-buahan tersebut mudah dijumpai di pasar lokal Indonesia, sehingga relevan dengan konteks implementasi praktis di lapangan. Kedua, keenam buah tersebut memiliki kemiripan visual yang cukup tinggi dari segi ukuran, warna, maupun bentuk. Misalnya, anggur dan rambutan memiliki ukuran kecil, bentuk bulat, dan sering kali muncul dalam jumlah banyak dalam satu citra. Sementara itu, stroberi dan rambutan sama-sama berwarna merah mencolok dengan ukuran yang hampir serupa. Kompleksitas visual ini menjadi indikator yang baik dalam menguji kemampuan generalisasi model terhadap data dengan ciri-ciri beragam (Marcelleno & Putra, 2025). Oleh karena itu, pemilihan kelas buah dalam penelitian ini dinilai relevan secara praktis sekaligus penting secara teknis.

Evaluasi performa model dalam penelitian ini dilakukan menggunakan metrik kuantitatif yang umum digunakan dalam deteksi objek, yaitu *precision*, *recall*, serta *mean Average Precision* (mAP) pada ambang 0.5 dan rentang 0.5–0.95. Seluruh metrik tersebut digunakan untuk membandingkan hasil deteksi model pada setiap variasi tingkat anotasi terhadap baseline dengan anotasi lengkap. Selain metrik akurasi, penelitian ini juga mencatat durasi pelatihan model sebagai indikator tambahan untuk menilai efisiensi komputasi. Dengan demikian, penelitian tidak hanya menelaah aspek ketepatan deteksi, tetapi juga mempertimbangkan implikasi praktis berupa penghematan waktu pelatihan.

Urgensi penelitian ini terletak pada kebutuhan akan keseimbangan antara kualitas anotasi dan efisiensi sumber daya dalam pengembangan model deteksi objek. Proses anotasi manual yang memakan waktu dan biaya tinggi sering kali menjadi hambatan dalam skala industri maupun riset. Di sisi lain, performa model akan menurun apabila anotasi dilakukan secara terbatas tanpa strategi yang tepat. Belum adanya kajian yang secara sistematis mengevaluasi dampak pengurangan anotasi terhadap performa YOLOv11, baik dari sisi akurasi maupun efisiensi komputasi, menjadikan penelitian ini penting dilakukan. Dengan menelaah ambang

batas minimal anotasi yang masih layak digunakan, penelitian ini memberikan kontribusi signifikan dalam menjawab kebutuhan praktis sekaligus memperkaya kajian akademik terkait efisiensi anotasi dan pembelajaran berbasis *deep learning*.

Lebih jauh, penelitian ini juga relevan dengan arah pengembangan metode pembelajaran modern, seperti *semi-supervised learning* dan *active learning*, yang menekankan efisiensi dalam proses pelabelan data. Pemahaman mengenai hubungan antara jumlah anotasi, waktu komputasi, dan performa model diharapkan mampu memberikan dasar bagi pengembangan sistem deteksi objek yang hemat sumber daya tanpa mengorbankan akurasi (Xu, Chen, Guan, & Hu, 2023). Selain itu, hasil penelitian ini diharapkan berkontribusi pada pengambilan keputusan strategis dalam pengumpulan dan pelabelan dataset, baik untuk keperluan riset maupun aplikasi praktis di dunia nyata.

Pendekatan yang diadopsi dalam penelitian ini juga sejalan dengan studi-studi terdahulu mengenai efisiensi implementasi model deteksi objek pada perangkat dengan sumber daya terbatas, seperti ESP32-CAM (Wibowo, Tullah, & Ricesa, 2025). Dengan mempertimbangkan keterbatasan umum yang terjadi pada dunia akademik maupun industri, penelitian ini diharapkan mampu memberikan landasan praktis bagi pengembangan model deteksi objek yang efisien, sekaligus tetap andal dalam menjalankan tugas deteksi.

Dengan demikian, penelitian ini tidak hanya berfokus pada analisis penurunan performa akibat pengurangan anotasi, tetapi juga diarahkan untuk memberikan rekomendasi model YOLOv11 terbaik beserta skenario pengurangan anotasi yang paling efektif. Rekomendasi ini mempertimbangkan keseimbangan antara akurasi deteksi, dampak pada kelas buah tertentu, ambang batas anotasi efektif, serta efisiensi waktu komputasi. Hasil penelitian diharapkan dapat menjadi rujukan praktis bagi penelitian selanjutnya maupun implementasi nyata dalam pengembangan sistem pertanian digital berbasis visi komputer.

1.2 Rumusan Masalah

Penelitian ini berfokus pada analisis pengaruh pengurangan tingkat anotasi terhadap performa model deteksi objek berbasis YOLOv11. Untuk menilai dampaknya secara menyeluruh, digunakan dua pendekatan pengurangan anotasi dengan karakteristik berbeda, yaitu:

- **Skenario A**: menghapus sebagian anotasi dari objek buah yang jelas, utuh, dan tidak tertutup, dengan asumsi bahwa anotasi pada objek yang lebih kompleks tetap dipertahankan.

- **Skenario B**: menghapus anotasi dari objek buah yang bertumpuk, tertutup sebagian, atau tidak utuh, sehingga model hanya dilatih pada objek-objek yang relatif mudah dikenali.

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana pengaruh pengurangan tingkat anotasi terhadap performa model YOLOv11 dalam mendeteksi objek pada dua skenario yang berbeda?
2. Bagaimana perbandingan kinerja model YOLOv11 dalam mendeteksi enam kelas buah (jeruk, apel, stroberi, anggur, rambutan, dan salak) pada dataset dengan variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) di bawah Skenario A dan Skenario B?
3. Seberapa besar penurunan performa model deteksi objek pada masing-masing skenario berdasarkan metrik evaluasi yang digunakan (*precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*)?
4. Sampai sejauh mana pengurangan tingkat anotasi masih dapat ditoleransi tanpa menimbulkan penurunan performa yang signifikan, dengan batas toleransi maksimal sebesar 10% dari baseline model beranotasi penuh (100%)?
5. Sejauh mana pengurangan tingkat anotasi memengaruhi efisiensi komputasi, khususnya dari sisi durasi pelatihan model pada setiap variasi skenario?
6. Model dengan tingkat anotasi dan skenario apa yang paling direkomendasikan untuk digunakan, dengan mempertimbangkan keseimbangan antara akurasi deteksi dan efisiensi komputasi?

1.3 Batasan Masalah

Untuk menjaga fokus dan ruang lingkup penelitian agar tetap terarah dan tidak melebar, penelitian ini dibatasi oleh beberapa hal berikut:

1. Penelitian ini hanya menggunakan model deteksi objek *You Only Look Once version 11 (YOLOv11)* dan tidak melakukan perbandingan dengan arsitektur deteksi objek lainnya seperti *YOLOv8*, *Faster R-CNN*, atau *SSD*. Dengan demikian, analisis berfokus pada performa *YOLOv11* dengan variasi tingkat anotasi, bukan pada perbandingan lintas arsitektur.
2. Dataset yang digunakan merupakan dataset buatan sendiri yang terdiri dari gambar buah-buahan dengan enam kelas objek, yaitu jeruk, apel, stroberi, anggur, rambutan, dan salak. Seluruh anotasi dilakukan secara manual menggunakan *bounding box*

melalui platform Roboflow dan disimpan dalam format *YOLO*. Pemilihan enam kelas buah ini mempertimbangkan kemudahan ditemui di pasar lokal Indonesia serta kompleksitas visual yang relevan untuk pengujian performa model.

3. Tingkat anotasi dikurangi secara bertahap sebesar 10% dari kondisi penuh (100%), sehingga menghasilkan lima variasi dataset dengan tingkat anotasi 100%, 90%, 80%, 70%, dan 60%. Proses pengurangan dilakukan secara sistematis untuk mensimulasikan kondisi nyata keterbatasan anotasi dalam penelitian maupun praktik industri.
4. Terdapat dua skenario pengurangan anotasi yang digunakan dalam penelitian ini:
    - Skenario A: Menghapus anotasi dari objek yang jelas atau utuh.
    - Skenario B: Menghapus anotasi dari objek yang bertumpuk, saling menutupi, atau tidak jelas.

      Kedua skenario ini dipilih agar dapat memberikan gambaran menyeluruh tentang dampak kualitas anotasi pada performa model.
5. Evaluasi performa model dilakukan secara keseluruhan menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Selain itu, analisis efisiensi komputasi juga diperhatikan dengan memasukkan waktu pelatihan (*training time*) sebagai salah satu indikator. Analisis per kelas ditampilkan secara deskriptif sebagai informasi tambahan, namun bukan menjadi fokus utama penelitian.
6. Proses pelatihan dan pengujian model dilakukan di lingkungan Python dengan dukungan perangkat keras berbasis *cloud* melalui *Google Colab*. GPU yang digunakan adalah *NVIDIA Tesla T4*, dengan keterbatasan sumber daya yang menjadi pertimbangan utama dalam pemilihan arsitektur *YOLOv11s*. Penelitian ini tidak melakukan optimalisasi terhadap perangkat lokal ataupun GPU lanjutan di luar platform *Colab*.
7. Penelitian ini tidak membahas teknik pengurangan anotasi secara otomatis melalui pendekatan *semi-supervised learning* atau *active learning*. Seluruh pengurangan anotasi dilakukan secara manual dan terkontrol untuk menjaga konsistensi serta validitas eksperimen dalam membandingkan setiap variasi dataset.
8. Penelitian ini tidak berfokus pada pengembangan aplikasi sebagai produk akhir. Namun demikian, sebuah antarmuka visual sederhana berbasis *Streamlit* dikembangkan untuk membantu menampilkan, memvisualisasikan, serta membandingkan hasil deteksi dari masing-masing model secara interaktif. Penggunaan *Streamlit* ini dimaksudkan sebagai alat bantu analisis, bukan sebagai kontribusi utama penelitian.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengkaji secara mendalam pengaruh pengurangan tingkat anotasi objek dalam dataset terhadap performa model deteksi objek berbasis *YOLOv11*, serta melakukan perbandingan antara dua skenario pemilihan objek yang dihapus anotasinya. Pendekatan dilakukan secara eksperimental dengan mengurangi anotasi secara bertahap sebesar 10% dari kondisi penuh (100%) untuk mengevaluasi penurunan performa, menilai efisiensi komputasi, serta mengidentifikasi ambang batas (*threshold*) yang masih dapat ditoleransi. Objek penelitian mencakup enam kelas buah, yaitu jeruk, apel, stroberi, anggur, rambutan, dan salak.

Secara lebih rinci, tujuan penelitian ini adalah sebagai berikut:

1. Menganalisis kinerja model *YOLOv11* dalam mendeteksi objek buah pada dataset dengan pengurangan anotasi berdasarkan dua skenario, yaitu penghapusan anotasi pada objek yang jelas (*Skenario A*) dan penghapusan anotasi pada objek yang bertumpuk (*Skenario B*).
2. Mengevaluasi perubahan performa model pada setiap tingkat anotasi berdasarkan metrik evaluasi, meliputi *precision*, *recall*, *mAP@0.5*, *mAP@0.5:0.95*, serta waktu pelatihan (*training time*) pada kedua skenario.
3. Menentukan ambang batas minimum tingkat anotasi yang masih menghasilkan performa stabil dengan toleransi penurunan kurang dari 10% dibanding model baseline (100% anotasi).
4. Memberikan rekomendasi strategi anotasi yang lebih efisien berdasarkan hasil perbandingan kedua skenario, khususnya untuk kasus deteksi objek kompleks seperti buah-buahan yang memiliki kemiripan bentuk dan sering saling menumpuk.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi yang bermakna, baik secara teoritis maupun praktis, dalam bidang deteksi objek berbasis *computer vision*, khususnya terkait efisiensi proses anotasi data pada pengembangan model berbasis YOLOv11. Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. **Manfaat Akademik dan Ilmiah**

   Penelitian ini diharapkan dapat memperkaya khazanah literatur ilmiah terkait dampak pengurangan tingkat anotasi terhadap performa model deteksi objek berbasis *deep learning*. Melalui pendekatan eksperimen bertahap dan sistematis, penelitian ini

memberikan landasan empiris bagi peneliti lain dalam menentukan ambang batas minimal anotasi yang masih dapat diterima. Selain itu, penelitian ini berkontribusi terhadap pengembangan studi di bidang *data-centric AI*, khususnya dalam konteks keterbatasan anotasi manual, efisiensi komputasi, serta integrasi metrik tambahan seperti waktu pelatihan (*training time*) dalam evaluasi performa model.

2. **Manfaat Praktis dalam Efisiensi Anotasi dan Komputasi**

   Hasil evaluasi dari berbagai variasi tingkat anotasi memberikan wawasan praktis mengenai sejauh mana pengurangan anotasi dapat diterapkan tanpa mengorbankan performa model secara signifikan. Informasi ini tidak hanya bermanfaat untuk menghemat waktu, biaya, dan tenaga dalam proses anotasi manual, tetapi juga memberikan pemahaman praktis tentang efisiensi penggunaan sumber daya komputasi. Dengan adanya pengukuran *training time*, penelitian ini turut menawarkan pertimbangan strategis bagi praktisi dalam memilih model dan skenario anotasi yang lebih hemat biaya sekaligus efektif untuk proyek berskala besar maupun terbatas.

3. **Manfaat bagi Pengembangan Teknologi Deteksi Buah**

   Dengan fokus pada enam jenis buah, yaitu jeruk, apel, stroberi, anggur, rambutan, dan salak, penelitian ini memiliki relevansi yang tinggi terhadap sektor pertanian digital. Temuan yang dihasilkan dapat dimanfaatkan untuk mengembangkan sistem deteksi buah otomatis yang lebih efisien, misalnya dalam proses pemilahan hasil panen, klasifikasi mutu, dan pencatatan inventaris berbasis citra digital. Selain itu, pemanfaatan *YOLOv11s* sebagai arsitektur terkini memberikan nilai tambah karena hasil penelitian lebih mudah diadaptasikan pada sistem pertanian cerdas yang membutuhkan solusi deteksi cepat dan akurat dengan keterbatasan perangkat keras.

4. **Manfaat bagi Strategi Penyusunan Dataset dan Anotasi**

   Penelitian ini memberikan panduan strategis bagi praktisi *data science* dan pengembang sistem kecerdasan buatan dalam merancang proses anotasi dataset. Dengan mengetahui tingkat anotasi minimum yang masih mempertahankan performa model, proses pengumpulan dan pelabelan data dapat dilakukan secara lebih terukur dan hemat sumber daya. Selain itu, dengan adanya antarmuka visual berbasis *Streamlit* yang digunakan dalam penelitian ini, hasil perbandingan antar model dapat ditampilkan secara interaktif sehingga mempermudah proses analisis dan validasi. Hal ini menjadi nilai tambah praktis yang memperkuat kontribusi penelitian dalam bidang pengolahan data berbasis citra.

5. **Manfaat sebagai Dasar Implementasi Teknik Pembelajaran Modern**

   Hasil penelitian ini dapat menjadi pijakan awal dalam penerapan pendekatan modern seperti *semi-supervised learning*, *active learning*, maupun *pseudo-labeling*. Teknik-teknik tersebut sangat penting ketika data beranotasi lengkap sulit diperoleh, namun tetap dibutuhkan model yang akurat dan efisien untuk digunakan dalam aplikasi dunia nyata. Dengan adanya hasil evaluasi mengenai ambang batas anotasi dan efisiensi komputasi, penelitian ini memberikan gambaran yang lebih utuh tentang bagaimana model dapat tetap optimal meskipun dilatih dengan keterbatasan anotasi.

1.6 Metodologi Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental yang bertujuan untuk mengevaluasi dampak pengurangan tingkat anotasi terhadap performa model deteksi objek berbasis *YOLOv11*. Seluruh proses dilakukan secara sistematis melalui tahapan pembuatan dataset, pelatihan model, pengujian, dan evaluasi performa.

Dataset yang digunakan terdiri dari enam kelas buah, yaitu apel, jeruk, stroberi, anggur, rambutan, dan salak. Seluruh gambar dikumpulkan dari sumber terbuka dan dianotasi secara manual menggunakan *bounding box* melalui platform *Roboflow*, kemudian dikonversi ke dalam format *YOLO*. Dataset awal dianotasi secara lengkap (100%) dan kemudian dikurangi tingkat anotasinya secara bertahap sebesar 10% untuk membentuk variasi dataset beranotasi 90%, 80%, 70%, dan 60%.

Penelitian ini menggunakan dua skenario berbeda dalam melakukan pengurangan anotasi untuk mendapatkan perspektif yang lebih komprehensif mengenai dampaknya terhadap performa model:

- Skenario A: Pengurangan anotasi dilakukan terhadap objek buah yang tampak jelas, tidak tertutup, dan utuh, sementara objek-objek yang bertumpuk atau sebagian tertutup tetap dianotasi.
- Skenario B: Sebaliknya, pengurangan dilakukan terhadap objek buah yang bertumpuk, saling menutupi, atau tampak sebagian, sedangkan objek-objek yang jelas tetap dipertahankan dalam anotasi.

Kedua skenario ini dirancang untuk mensimulasikan kondisi nyata dalam proses anotasi manual, di mana pengurangan data dapat terjadi baik pada objek yang mudah terlihat maupun objek yang sulit terlihat. Dengan demikian, penelitian ini tidak hanya mengevaluasi penurunan

performa model, tetapi juga menilai bagaimana jenis objek yang dihilangkan dapat memengaruhi akurasi serta stabilitas deteksi.

Proses pelatihan model dilakukan menggunakan pustaka Ultralytics *YOLOv11* pada platform *Google Colab* dengan dukungan akselerasi GPU berbasis CUDA. Seluruh model dilatih dengan arsitektur dan parameter yang seragam agar hasil perbandingan tetap objektif. Evaluasi performa dilakukan menggunakan metrik *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Selain itu, penelitian ini juga mengukur waktu pelatihan (*training time*) pada setiap variasi dataset dan skenario sebagai indikator efisiensi komputasi. Dengan demikian, penelitian tidak hanya menilai ketepatan deteksi, tetapi juga mengamati seberapa besar penghematan waktu komputasi dapat dicapai melalui pengurangan anotasi.

Perbandingan antar model dilakukan untuk menentukan ambang batas penurunan tingkat anotasi yang masih dapat menghasilkan performa deteksi yang stabil dan layak digunakan. Ambang batas ditetapkan berdasarkan toleransi penurunan performa maksimal sebesar 10% dari model baseline (100%) terhadap metrik utama, yaitu *precision* dan *mAP@0.5*.

Pemilihan *YOLOv11* sebagai arsitektur utama dalam penelitian ini didasarkan pada efektivitasnya dalam skenario deteksi objek kompleks dengan latensi rendah (*low-latency detection*) (Rohiman, Kanata, Ahmad, & Akbar, 2025). Strategi eksperimen dalam penelitian ini mengacu pada pendekatan evaluasi model deteksi berbasis *deep learning* dalam domain pengolahan citra geospasial (Pratama, Gunawan, & Gultom, 2020). Sementara itu, metode perbandingan performa lintas model merujuk pada studi yang mengevaluasi *YOLO*, *SSD*, dan *Faster R-CNN* dalam konteks pencahayaan rendah (Achmad, Pramudwiatmoko, Satrio Gumilang, Al Karim, & Wiyono, 2025).

Untuk mendukung analisis hasil, penelitian ini mengembangkan aplikasi berbasis *Streamlit* yang memungkinkan pengguna mengunggah gambar uji dan membandingkan hasil deteksi dari seluruh model yang telah dilatih. Antarmuka visual ini berfungsi bukan sekadar sebagai pelengkap, melainkan sebagai alat bantu analisis visual yang penting untuk menampilkan, memvalidasi, serta mengeksplorasi hasil deteksi secara interaktif dan komprehensif. Kehadiran *Streamlit* membantu memperjelas perbedaan kinerja model yang mungkin sulit diinterpretasikan hanya melalui angka pada tabel evaluasi.

Dengan metodologi ini, penelitian diharapkan mampu memberikan kontribusi baik secara teoritis maupun praktis. Secara teoritis, penelitian memperkaya pemahaman mengenai hubungan antara tingkat anotasi, performa deteksi, dan efisiensi komputasi. Secara praktis, penelitian menawarkan strategi anotasi yang lebih hemat sumber daya, sekaligus menyediakan

bukti visual interaktif melalui *Streamlit* untuk memperkuat temuan, sehingga hasil penelitian relevan baik bagi akademisi maupun praktisi yang menghadapi keterbatasan waktu, biaya, dan tenaga dalam proses anotasi data.

1.7 Sistematika Penulisan

Laporan tugas akhir ini disusun secara sistematis ke dalam lima bab utama sehingga pembahasan tersaji secara terstruktur, logis, dan mudah dipahami. Adapun gambaran isi dari masing-masing bab adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi uraian mengenai latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi yang digunakan secara ringkas, serta sistematika penulisan skripsi.

BAB II TINJAUAN PUSTAKA

Bab ini memuat landasan teori dan kajian pustaka yang relevan dengan topik penelitian, mencakup konsep dasar deteksi objek, arsitektur *YOLOv11*, pendekatan anotasi data, serta penelitian-penelitian terdahulu yang dijadikan referensi dalam perancangan eksperimen.

BAB III METODOLOGI PENELITIAN

Bab ini menguraikan secara rinci tahapan eksperimen, meliputi pengumpulan dataset, strategi pengurangan anotasi berdasarkan dua skenario (objek jelas vs. bertumpuk), konfigurasi pelatihan model, serta prosedur evaluasi performa menggunakan metrik yang relevan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil pelatihan dan pengujian model dari kedua skenario, dilanjutkan dengan analisis perbandingan performa, identifikasi kelas buah yang paling terdampak, serta penentuan ambang batas pengurangan anotasi yang masih efektif. Grafik, tabel, dan visualisasi digunakan sebagai pendukung interpretasi agar analisis lebih komprehensif.

BAB V KESIMPULAN DAN SARAN

Bab terakhir memuat kesimpulan dari keseluruhan rangkaian penelitian yang telah dilaksanakan, termasuk temuan utama mengenai pengaruh pengurangan anotasi terhadap performa model deteksi objek. Selain itu, bab ini juga menyajikan saran untuk penelitian lanjutan serta rekomendasi praktis bagi pengembangan sistem deteksi objek yang efisien.

BAB II
TINJAUAN PUSTAKA

2.1 Definisi dan Evolusi Deteksi Objek: Fokus YOLO dan YOLOv11

Deteksi objek merupakan salah satu cabang utama dalam *computer vision* yang bertujuan mengidentifikasi serta menentukan posisi objek pada citra maupun video melalui pembuatan *bounding box* di sekitar objek yang terdeteksi. Seiring dengan pesatnya perkembangan teknologi *deep learning*, bidang ini telah mengalami kemajuan signifikan dengan hadirnya berbagai algoritma berbasis *Convolutional Neural Networks* (*CNN*). Salah satu algoritma deteksi objek yang paling populer adalah *You Only Look Once* (*YOLO*), yang diperkenalkan dengan konsep deteksi satu tahap (*one-stage detection*). Pada pendekatan ini, proses deteksi dilakukan hanya dengan satu kali inferensi terhadap citra (*single forward pass*), sehingga menghasilkan kecepatan dan efisiensi yang tinggi dalam mendeteksi objek (Aini, Lutfiani, Kusumah, & Zahran, 2021).

Sebelum hadirnya *YOLO*, sebagian besar detektor objek menggunakan pendekatan dua tahap (*two-stage detectors*), seperti *R-CNN*, *Fast R-CNN*, dan *Faster R-CNN*. Pendekatan ini dikenal memiliki akurasi tinggi, tetapi keterbatasannya adalah waktu inferensi yang relatif lambat karena harus melalui tahap *region proposal* terlebih dahulu. *YOLO* muncul sebagai solusi atas keterbatasan tersebut dengan menyatukan proses prediksi kelas dan posisi objek dalam satu tahap, sehingga mampu mencapai kecepatan deteksi *real-time* tanpa mengorbankan akurasi secara signifikan.

Keluarga *YOLO* terus mengalami perkembangan sejak versi awal hingga mencapai generasi terbaru, yakni *YOLOv11*. Setiap versi membawa peningkatan baik pada sisi arsitektur maupun metode optimasi yang digunakan. Sebagai contoh, *YOLOv3* memperkenalkan penggunaan *feature pyramid networks* untuk mendeteksi objek dalam berbagai skala, sementara versi-versi berikutnya, seperti *YOLOv5* dan *YOLOv8*, lebih menekankan pada efisiensi komputasi serta peningkatan akurasi deteksi. *YOLOv11*, yang dirilis pada tahun 2024, memperkenalkan pendekatan *anchor-free detection*, blok arsitektur *C3k2*, serta *Cross-Stage Partial Spatial Attention* (*C2PSA*), yang mampu meningkatkan pemahaman model terhadap fitur spasial objek secara lebih mendalam (Yang et al., 2025).

Dalam berbagai penelitian sebelumnya, *YOLO* telah banyak diterapkan pada beragam bidang, misalnya untuk pengenalan objek manusia menggunakan wahana udara tak berawak (*unmanned aerial vehicle*) (Sugandi & Hartono, 2022) maupun untuk deteksi jumlah kendaraan

di jalan raya (Therino Elevan, Hermanto, & Widiyanto, 2025). Performa *YOLO* yang cepat dan akurat menjadikannya lebih unggul dibandingkan pendekatan lain seperti *SSD* maupun *Faster R-CNN* dalam konteks aplikasi *real-time*. Karakteristik tersebut menjadikan *YOLOv11* sebagai kandidat ideal untuk mendukung pengembangan teknologi berbasis *computer vision* yang membutuhkan kinerja tinggi dengan latensi minimal.

Evolusi *YOLOv11* tidak hanya menitikberatkan pada peningkatan kecepatan, tetapi juga mengedepankan akurasi deteksi terhadap objek berukuran kecil maupun pada kondisi pencahayaan yang sulit. Dengan arsitektur *backbone–neck–head* yang telah diperbarui serta integrasi *enhanced transformer modules*, *YOLOv11* dirancang untuk menghadapi tantangan deteksi objek pada skenario dunia nyata yang kompleks dan bervariasi (Yang et al., 2025). Selain itu, efisiensi komputasi yang dimiliki *YOLOv11* menjadikannya sangat sesuai untuk eksperimen di lingkungan berbasis *cloud computing* dengan keterbatasan GPU, seperti *Google Colab*, sehingga dapat digunakan secara optimal meskipun dengan sumber daya terbatas.

Oleh karena itu, dalam penelitian ini *YOLOv11* dipilih sebagai arsitektur utama untuk mengevaluasi pengaruh pengurangan tingkat anotasi terhadap performa model deteksi objek, dengan mempertimbangkan kecepatan, efisiensi, serta relevansi penerapannya pada dataset buah dengan ukuran kecil hingga menengah.

2.2 Arsitektur YOLOv11

YOLOv11 merupakan pengembangan terbaru dari arsitektur *You Only Look Once* yang mengadopsi prinsip *one-stage detector* dengan peningkatan signifikan pada akurasi, kecepatan, dan efisiensi komputasi. Model ini tetap mempertahankan struktur inti dari generasi sebelumnya (*YOLOv8*), namun dilengkapi dengan optimisasi yang lebih mendalam pada proses *feature extraction* dan *feature aggregation* (Shan et al., 2025). Peningkatan ini menjadikan *YOLOv11* lebih adaptif dalam menghadapi variasi objek, baik dari segi ukuran maupun kompleksitas visual.

Secara garis besar, *YOLOv11* terdiri atas tiga komponen utama, yaitu *backbone*, *neck*, dan *head*:

1. **Backbone**

   Bagian *backbone* berfungsi untuk mengekstraksi fitur dari citra masukan melalui serangkaian lapisan konvolusi. *YOLOv11* menggunakan blok *C3k2* yang telah dioptimalkan untuk meningkatkan efisiensi pelatihan serta memaksimalkan *feature reuse* dibandingkan blok konvensional. Selain itu, modul *SPPF* (*Spatial Pyramid*

Pooling-Fast) digunakan untuk menghasilkan peta fitur multi-skala dengan waktu komputasi yang lebih singkat, sehingga mampu mempercepat proses deteksi tanpa kehilangan representasi spasial yang penting (Li, Wang, Zhao, & Li, 2025).

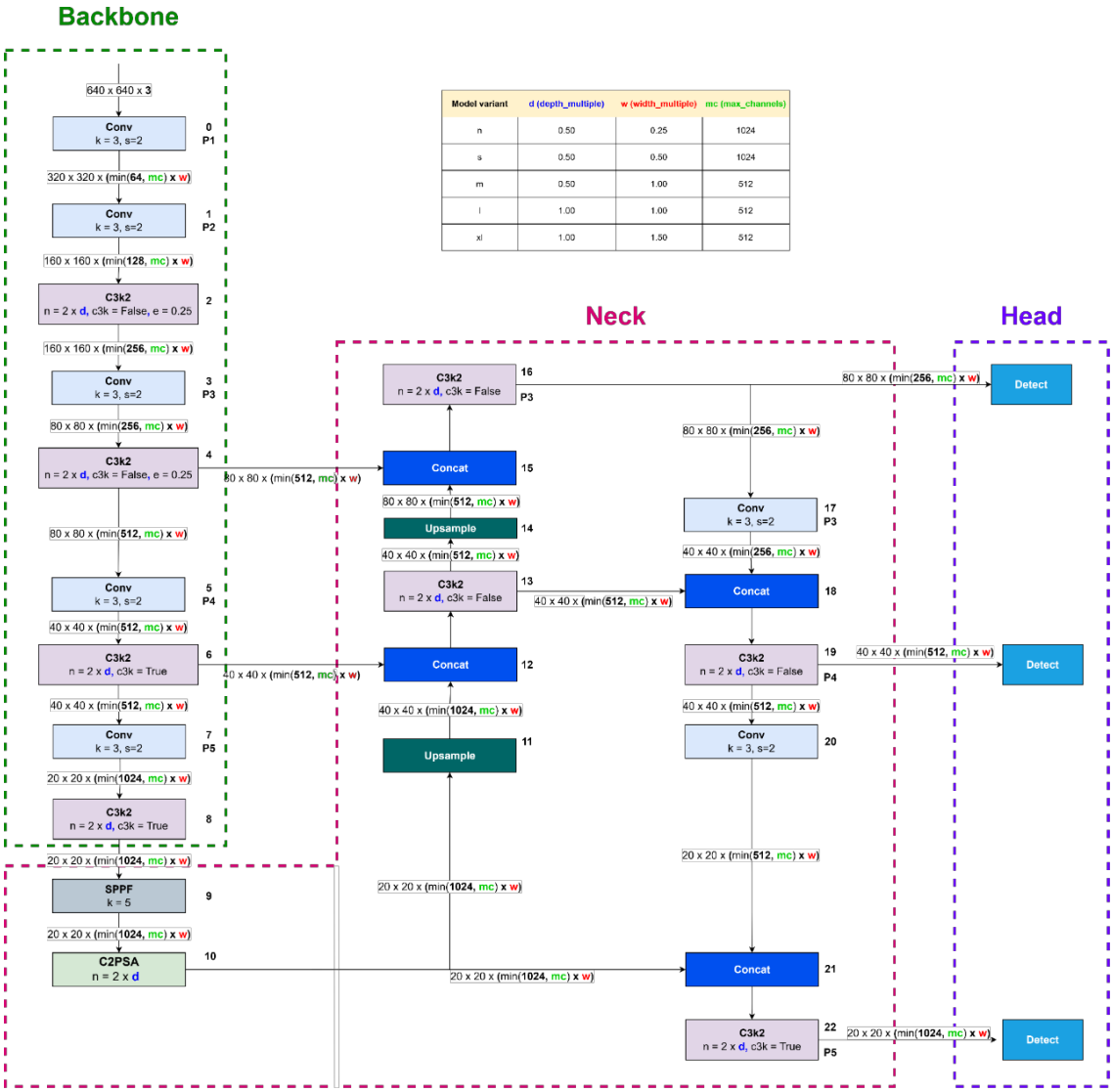
2. **Neck**

   Pada bagian *neck*, *YOLOv11* memanfaatkan blok *C2f* serta *path aggregation networks* untuk menggabungkan fitur dari berbagai skala. Mekanisme penggabungan ini memungkinkan model melakukan deteksi pada objek dengan ukuran bervariasi sekaligus memperkuat informasi konteks, terutama untuk mendeteksi objek berukuran kecil maupun objek yang saling bertumpuk. Dengan demikian, *neck* berperan penting dalam menjaga konsistensi representasi fitur lintas skala dan memperbaiki sensitivitas deteksi pada kondisi kompleks (Ayuningtyas et al., 2025).

3. **Head**

   Bagian *head* pada *YOLOv11* dirancang dengan pendekatan *anchor-free detection*, yang menghilangkan kebutuhan akan *anchor boxes* konvensional. Desain ini membuat prediksi *bounding box* menjadi lebih adaptif terhadap bentuk objek yang tidak beraturan. Selain itu, pendekatan *anchor-free* terbukti mampu mempercepat proses inferensi sekaligus meningkatkan akurasi deteksi, khususnya pada objek dengan bentuk kompleks dan distribusi spasial yang tidak seragam (Elef et al., 2025).

Keunggulan utama *YOLOv11* terletak pada efisiensinya yang memungkinkan optimalisasi pada perangkat dengan sumber daya terbatas, seperti GPU di lingkungan *cloud computing*. Hal ini membuat *YOLOv11* relevan tidak hanya dalam konteks penelitian, tetapi juga pada implementasi dunia nyata, misalnya untuk deteksi buah, *autonomous vehicles*, serta sistem pengawasan industri. Dukungan terhadap *low-latency inference* menjadikan arsitektur ini mampu menghadirkan keseimbangan antara akurasi dan kecepatan, sehingga layak diadopsi untuk aplikasi berskala besar maupun skenario praktis dengan keterbatasan komputasi.



Gambar 2.1 Arsitektur YOLOv11

Sumber : Hidayatullah, Syakrani, Sholahuddin, Gelar, & Tubagus, (2025)

2.3 YOLO untuk Deteksi Buah

Penerapan *You Only Look Once (YOLO)* dalam deteksi buah telah berkembang pesat karena kemampuannya mendeteksi objek secara cepat dan akurat, bahkan dalam kondisi lingkungan yang kompleks seperti kebun maupun perkebunan. Algoritma ini mampu melakukan deteksi *multi-objek* dalam satu kali *forward pass*, sehingga lebih efisien dibandingkan dengan metode deteksi konvensional yang memerlukan tahapan lebih panjang(Saputra, Imran, & Juhartini, 2023).

Berbagai penelitian telah menunjukkan bahwa varian *YOLO*, seperti *YOLOv5* dan *YOLOv8*, memiliki performa unggul dalam mendeteksi buah pada kondisi pencahayaan yang bervariasi, adanya *occlusion*, maupun latar belakang yang kompleks. Misalnya, Tang et al.

(2025) berhasil mengembangkan model *YOLOv5* yang ditingkatkan untuk mendeteksi buah jeruk di kebun alami dengan tingkat akurasi yang tinggi, bahkan ketika buah sebagian tertutup daun atau cabang pohon. Hasil penelitian tersebut menegaskan keandalan *YOLO* dalam mengatasi tantangan deteksi pada kondisi lapangan yang tidak terstruktur.

Selain untuk identifikasi objek, *YOLO* juga diaplikasikan dalam mendeteksi tingkat kematangan buah. Penelitian oleh Aras, Tanra, & Bazhar (2024) mengenai buah tomat menunjukkan bahwa *YOLO* tidak hanya mampu mendeteksi keberadaan buah, tetapi juga mengklasifikasikan tingkat kematangan berdasarkan variasi warna dan tekstur. Hal ini menegaskan potensi *YOLO* sebagai pendekatan komprehensif dalam analisis kualitas buah.

Studi lain oleh Kurniawan, Martadinata, & Cahyo (2023) pada buah sawit menyoroti peran penting *feature extraction* dalam arsitektur *YOLO* untuk membedakan tingkat kematangan, yang berimplikasi pada optimalisasi proses panen di industri perkebunan. Optimalisasi ini menunjukkan bahwa penerapan *YOLO* dapat memberikan dampak langsung pada efisiensi produksi dalam sektor pertanian.

Dalam konteks penelitian ini, *YOLOv11* dipilih sebagai model deteksi utama karena mewarisi keunggulan varian *YOLO* sebelumnya sekaligus menawarkan optimisasi arsitektur modern. Peningkatan ini menjadikannya sangat relevan untuk mendeteksi berbagai jenis buah dengan ukuran, bentuk, dan warna yang bervariasi. Selain itu, *YOLOv11* dinilai lebih sesuai untuk eksperimen dengan dataset yang memiliki keterbatasan anotasi, sehingga sejalan dengan fokus penelitian ini yang mengevaluasi dampak pengurangan anotasi terhadap performa deteksi.

Berdasarkan literatur tersebut, *YOLO* telah terbukti menjadi pendekatan yang efektif dalam deteksi buah, baik untuk identifikasi objek tunggal maupun multi-kelas. Oleh karena itu, penelitian ini memanfaatkan keunggulan *YOLOv11* untuk mendeteksi enam jenis buah, dengan evaluasi menyeluruh terhadap pengaruh variasi tingkat anotasi terhadap performa deteksi yang dihasilkan.

2.4 Teknologi yang Digunakan

Penelitian ini memanfaatkan kombinasi teknologi *software* dan *platform* yang dirancang untuk mendukung keseluruhan proses deteksi objek, mulai dari anotasi data, pelatihan model, hingga visualisasi hasil. Setiap teknologi dipilih berdasarkan kemampuannya dalam memberikan efisiensi, kemudahan integrasi, serta dukungan terhadap pengolahan data berbasis *deep learning*.

1. **Roboflow**

   Roboflow digunakan sebagai *platform* utama dalam pengelolaan dataset, meliputi proses anotasi manual menggunakan *bounding box*, *preprocessing*, serta pembagian data menjadi *training, validation,* dan *testing*. Selain itu, *Roboflow* menyediakan integrasi langsung dengan *Google Colab* melalui *API*, sehingga mempercepat alur kerja penelitian sekaligus mempermudah ekspor dataset dalam format yang kompatibel dengan *YOLOv11* (Brucal, De Jesus, Peruda, Samaniego, & Yong, 2023). Keunggulan ini menjadikan *Roboflow* sebagai solusi praktis dalam mempersiapkan data untuk eksperimen yang melibatkan variasi tingkat anotasi.

2. **Google Colab**

   Proses pelatihan model dilakukan menggunakan *Google Colab* dengan dukungan *GPU* berbasis CUDA. *Colab* dipilih karena menyediakan lingkungan *cloud computing* gratis yang mampu menangani komputasi berat tanpa memerlukan perangkat keras lokal. Selain memberikan fleksibilitas, *Google Colab* juga mendukung eksekusi skrip Python yang kompleks, termasuk eksperimen *deep learning* berbasis visi komputer. Banyak penelitian sebelumnya juga memanfaatkan *Colab* dalam pengembangan sistem berbasis *computer vision*, sehingga platform ini terbukti relevan untuk eksperimen pada penelitian ini (Gunawan et al., 2020).

3. **YOLOv11 (Ultralytics)**

   Model *YOLOv11* digunakan sebagai inti penelitian karena memiliki keunggulan pada kecepatan *inference* serta akurasi deteksi. *YOLOv11* merupakan pengembangan dari *YOLOv8* dengan optimisasi signifikan pada arsitektur *backbone* dan mekanisme ekstraksi fitur, menjadikannya sangat sesuai untuk mendeteksi objek dalam citra dengan variasi visual yang kompleks. Selain itu, *YOLOv11* juga dirancang agar efisien pada perangkat dengan keterbatasan sumber daya, sehingga mendukung penelitian ini yang dijalankan di *Google Colab*.

4. **Streamlit**

   Streamlit digunakan untuk membangun antarmuka interaktif yang menampilkan hasil deteksi objek dari model dengan berbagai variasi tingkat anotasi. Platform ini dipilih karena kemudahannya dalam integrasi dengan Python serta kemampuannya menghasilkan *dashboard* berbasis web tanpa memerlukan pengembangan *frontend* yang rumit. Melalui *Streamlit*, hasil deteksi dapat divisualisasikan secara intuitif,

memungkinkan analisis perbedaan performa antar model dilakukan dengan lebih komprehensif (Akkem, Kumar, & Varanasi, 2023).

5. **Python dan Visual Studio Code**

   Bahasa pemrograman *Python* digunakan untuk berbagai tahap penelitian, termasuk pelatihan model, pengolahan data hasil evaluasi (*results.csv*), serta pembangunan aplikasi berbasis *Streamlit*. Sementara itu, *Visual Studio Code* berfungsi sebagai *Integrated Development Environment (IDE)* utama untuk menulis, menguji, dan mendokumentasikan kode penelitian. Kombinasi keduanya memungkinkan pengelolaan proyek penelitian secara lebih efisien, fleksibel, serta mendukung reproduksibilitas hasil.

2.5 Penelitian Terdahulu dan Gap Penelitian

Berbagai penelitian telah dilakukan untuk mengevaluasi kinerja algoritma *YOLO* dalam mendeteksi objek buah. Irwan Adhi Prasetya, Fadli Sukandiarsyah, Novi Aryani Fitri, & Safri Adam (2024) melakukan klasifikasi kualitas buah jeruk menggunakan arsitektur *YOLOv8*. Penelitian tersebut menunjukkan bahwa *YOLOv8* memiliki kecepatan deteksi yang tinggi dengan akurasi *mAP@0.5* mencapai lebih dari 85%, bahkan pada kondisi pencahayaan yang bervariasi. Fokus penelitian ini adalah mengklasifikasikan tingkat kualitas buah berdasarkan parameter visual seperti warna dan tekstur.

Selain itu, Kuznetsova, Maleva, & Soloviev (2021) membandingkan kinerja *YOLOv3* dengan *YOLOv5* dalam mendeteksi buah apel pada berbagai skenario lapangan. Hasil penelitian mereka menunjukkan bahwa *YOLOv5* unggul dalam hal kecepatan inferensi dan akurasi deteksi, dengan perbedaan *mAP* yang signifikan dibandingkan *YOLOv3*, terutama pada dataset dengan pencahayaan alami di kebun apel.

Namun, kedua penelitian tersebut tidak menyoroti pengaruh tingkat kelengkapan anotasi dataset terhadap performa model. Mayoritas studi terdahulu mengasumsikan bahwa dataset dengan anotasi penuh selalu memberikan hasil terbaik, tanpa mengevaluasi sejauh mana pengurangan anotasi dapat memengaruhi performa model deteksi objek. Selain itu, belum ditemukan pendekatan komparatif yang membedakan antara pengurangan anotasi pada objek yang tampak jelas (*clear objects*) dan objek yang saling menutupi (*occluded objects*).

Sebuah studi terbaru oleh Song, Kim, Jeong, & Park (2025) mengkaji karakteristik dataset optimal untuk deteksi buah dengan membandingkan performa *YOLOv8* dan *YOLOv11*. Hasil penelitian tersebut menunjukkan bahwa *YOLOv11* umumnya menawarkan akurasi deteksi

yang sebanding atau lebih baik meskipun dengan jumlah anotasi yang lebih sedikit, yang mengindikasikan potensi efisiensi dalam pelabelan data. Akan tetapi, *YOLOv11* juga membutuhkan sumber daya komputasi yang lebih besar, sehingga mengungkap adanya *trade-off* antara efisiensi anotasi, akurasi model, dan beban komputasi.

Lebih lanjut, (Sapkota et al., 2025) melakukan evaluasi menyeluruh terhadap berbagai varian *YOLO*, termasuk *YOLOv11*, dalam mendeteksi buah kecil (*fruitlets*) di lingkungan kebun komersial. Penelitian tersebut menyoroti keseimbangan antara kecepatan inferensi dan akurasi, di mana *YOLOv11s* berada pada posisi unggul terkait *mAP@50*, sedangkan *YOLOv11n* menunjukkan kecepatan inferensi tercepat (2,4 ms). Temuan ini menegaskan bahwa pemilihan varian *YOLOv11* sangat dipengaruhi oleh kebutuhan antara akurasi dan efisiensi waktu komputasi.

Dari tinjauan tersebut, dapat disimpulkan adanya *research gap* yang signifikan. Sampai saat ini, belum ada studi yang secara eksplisit mengevaluasi dampak variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) terhadap performa YOLOv11, khususnya dalam konteks perbandingan dua skenario pengurangan anotasi, yaitu pada objek yang tampak jelas dan objek yang saling menutupi. Selain itu, kajian tentang bagaimana efisiensi waktu komputasi dipengaruhi oleh variasi tingkat anotasi juga masih terbatas. Oleh karena itu, penelitian ini diarahkan untuk menutup *gap* tersebut melalui analisis sistematis terhadap pengaruh pengurangan anotasi terhadap performa YOLOv11, dengan mempertimbangkan aspek akurasi, efisiensi waktu pelatihan, *robustness* terhadap variasi kualitas anotasi, serta realisme skenario dalam praktik anotasi. Pada akhirnya, penelitian ini bertujuan untuk memberikan rekomendasi mengenai model YOLOv11 dengan tingkat anotasi serta skenario penghapusan yang paling efektif, berdasarkan kombinasi pertimbangan nilai metrik evaluasi dan efisiensi komputasi.

2.6 Kontribusi Penelitian

Penelitian ini memberikan kontribusi nyata dalam pengembangan metode deteksi objek berbasis *deep learning*, khususnya dalam konteks efisiensi anotasi data, strategi pengurangan anotasi yang realistis, serta evaluasi aspek komputasional. Adapun kontribusi utama dari penelitian ini adalah sebagai berikut:

1. **Analisis Pengaruh Variasi Anotasi**

   Penelitian ini menyajikan pendekatan eksperimen sistematis untuk mengevaluasi dampak pengurangan tingkat anotasi terhadap performa model deteksi objek *YOLOv11*. Dengan membandingkan model hasil pelatihan pada lima tingkat anotasi (100%, 90%,

80%, 70%, dan 60%) dalam dua skenario berbeda, penelitian ini menjelaskan bagaimana performa metrik seperti *precision*, *recall*, dan *mAP* dipengaruhi oleh keterbatasan jumlah label.

2. **Identifikasi Ambang Batas Anotasi Efektif**

   Berdasarkan hasil evaluasi kuantitatif, penelitian ini mengusulkan ambang batas penurunan performa yang masih dapat ditoleransi, yaitu sebesar maksimum 10% dari baseline model beranotasi lengkap. Temuan ini memberikan panduan praktis dalam menentukan sejauh mana pengurangan anotasi masih mampu menghasilkan model yang layak digunakan, sekaligus memperkuat diskursus akademik tentang *trade-off* antara jumlah anotasi dan kualitas deteksi.

3. **Evaluasi Kelas Objek yang Paling Terdampak**

   Penelitian ini tidak hanya menilai performa secara agregat, tetapi juga melakukan analisis mendalam terhadap dampak pengurangan anotasi pada masing-masing kelas buah. Evaluasi ini membantu mengidentifikasi kelas yang paling rentan terhadap anotasi parsial seperti anggur, jeruk, dan stroberi, serta menjelaskan faktor visual yang memengaruhinya, misalnya ukuran kecil, warna yang mirip, atau bentuk yang saling menutupi.

4. **Penerapan Teknik Pengurangan Anotasi yang Representatif**

   Penelitian ini mengimplementasikan pendekatan *sequential per-image pruning*, yaitu teknik pengurangan label yang mensimulasikan perilaku nyata anotator saat hanya menandai sebagian objek dalam gambar. Dua skenario disimulasikan: pengurangan anotasi pada objek yang jelas (*Skenario A*) dan pada objek yang tertumpuk (*Skenario B*). Pendekatan ini dianggap lebih *realistic* dibandingkan penghapusan acak (*random deletion*) karena lebih mencerminkan tantangan dunia nyata dalam proses anotasi manual.

5. **Kontribusi terhadap Efisiensi Proses Anotasi dan Komputasi**

   Penelitian ini juga menekankan aspek efisiensi, tidak hanya dari sisi jumlah anotasi, tetapi juga dari segi waktu pelatihan (*training time*) pada model *YOLOv11*. Analisis ini memberikan wawasan mengenai bagaimana pengurangan anotasi dapat menghemat sumber daya komputasi tanpa menurunkan kualitas secara signifikan. Selain itu, hasil penelitian ini dapat menjadi referensi bagi praktisi yang ingin mengurangi beban kerja anotasi sekaligus mempertahankan efisiensi pelatihan model.

6. **Pemanfaatan Visualisasi Interaktif untuk Evaluasi**

Sebagai kontribusi tambahan, penelitian ini mengembangkan aplikasi berbasis *Streamlit* yang memungkinkan pengguna untuk membandingkan hasil deteksi objek dari model dengan variasi tingkat anotasi. Visualisasi interaktif ini mempermudah interpretasi performa model, sehingga hasil penelitian dapat lebih mudah diakses oleh peneliti, praktisi, maupun pengguna non-teknis.

Dengan kontribusi-kontribusi tersebut, penelitian ini tidak hanya menutup *gap* dalam studi deteksi objek berbasis YOLO, tetapi juga memberikan landasan praktis bagi pengembangan sistem deteksi yang lebih efisien, hemat sumber daya, dan tetap akurat. Selain itu, penelitian ini diarahkan untuk memberikan rekomendasi mengenai model YOLOv11 dan skenario pengurangan anotasi yang paling optimal, berdasarkan kombinasi aspek akurasi, efisiensi *training time*, serta relevansi skenario anotasi terhadap kondisi nyata. Ke depan, penelitian ini juga membuka peluang eksplorasi lanjutan dengan integrasi metode *semi-supervised learning* atau *active learning*, yang berpotensi meningkatkan efisiensi pelabelan *dataset* dalam skala besar.

BAB III
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan pendekatan *kuantitatif eksperimental* yang bertujuan untuk mengevaluasi pengaruh tingkat dan strategi pengurangan anotasi terhadap performa model deteksi objek berbasis *YOLOv11*. Pendekatan ini dipilih karena mampu mengukur hubungan antar variabel secara objektif dan terukur, serta memungkinkan peneliti menguji hipotesis melalui serangkaian eksperimen terkontrol berbasis data.

Jenis penelitian yang digunakan adalah *eksperimen komputasional*, di mana model deteksi objek dilatih dan diuji menggunakan dataset dengan variasi tingkat anotasi yang dikurangi secara bertahap sebesar 10% dari kondisi penuh (*100% annotated dataset*). Selain variasi persentase, penelitian ini juga mengimplementasikan dua skenario penghapusan anotasi:

- **Skenario A**: Menghapus anotasi dari objek yang terlihat jelas (*clear/unoccluded*) terlebih dahulu.
- **Skenario B**: Menghapus anotasi dari objek yang tertumpuk atau mengalami *occlusion* terlebih dahulu.

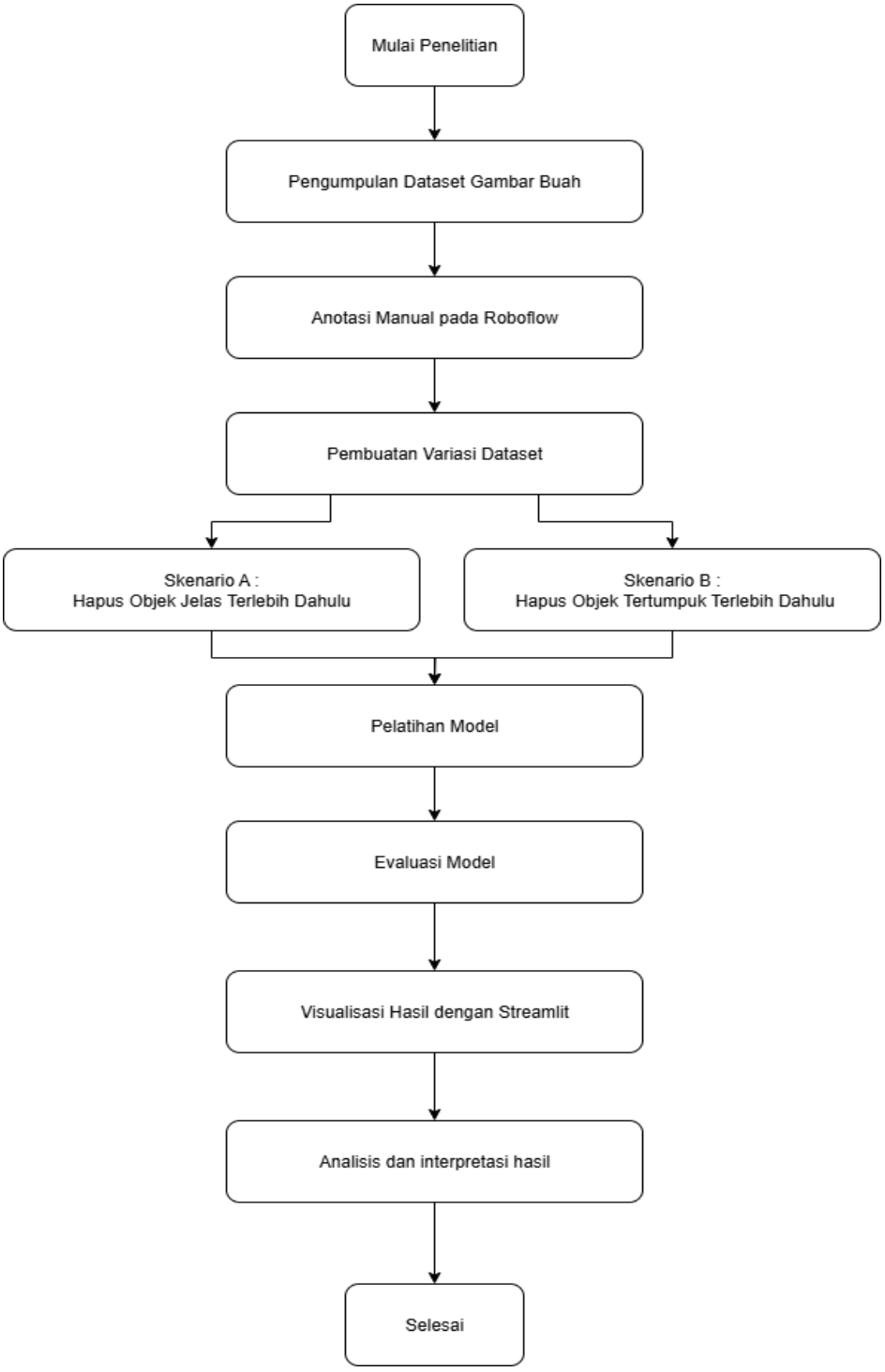
Model hasil pelatihan dari masing-masing skenario dan tingkat anotasi kemudian dievaluasi menggunakan metrik kuantitatif untuk mengukur perubahan performa deteksi secara menyeluruh.

Dengan menggunakan rancangan ini, penelitian secara sistematis mengamati dan membandingkan performa model pada setiap kombinasi tingkat dan skenario pengurangan anotasi, sehingga dapat ditentukan ambang batas (*threshold*) minimal anotasi yang masih mampu menghasilkan performa yang stabil dan layak. Hasil dari pendekatan ini diharapkan memberikan kontribusi teoritis maupun praktis dalam strategi efisiensi anotasi, khususnya dalam konteks pengembangan sistem deteksi objek berbasis *deep learning* yang membutuhkan waktu anotasi yang efisien namun tetap akurat.

3.2 Alur Penelitian

Penelitian ini dirancang menggunakan pendekatan *eksperimen kuantitatif* dengan fokus utama pada analisis pengaruh tingkat dan strategi pengurangan anotasi terhadap performa model deteksi objek berbasis *YOLOv11*. Proses penelitian dilakukan secara sistematis, dimulai dari pengumpulan data, anotasi penuh, pembuatan variasi dataset dengan beberapa skenario

pengurangan anotasi, pelatihan model, evaluasi performa, hingga analisis hasil. Seluruh tahapan ini dirancang untuk mengidentifikasi ambang batas (*threshold*) minimum tingkat anotasi yang masih mampu menghasilkan performa model yang stabil, serta mengetahui jenis objek apa yang paling krusial untuk diprioritaskan dalam proses anotasi.



Gambar 3.1 Diagram Alur Penelitian

Diagram alur pada Gambar 3.1 menggambarkan rangkaian tahapan penelitian, mulai dari pengumpulan citra buah, anotasi penuh 100% di Roboflow, pemisahan dataset menjadi *train/valid/test*, pembuatan variasi dataset berdasarkan skenario pengurangan anotasi, pelatihan model YOLOv11 di Google Colab, evaluasi model dengan metrik performa, hingga visualisasi hasil melalui aplikasi berbasis Streamlit.

Tahapan penelitian dapat dijabarkan secara rinci sebagai berikut:

1. **Pengumpulan Dataset:** Mengumpulkan 240 gambar buah dari berbagai sumber terbuka, terdiri dari enam kelas objek yaitu apel, jeruk, stroberi, anggur, rambutan, dan salak.
2. **Anotasi 100%:** Melakukan anotasi manual seluruh objek pada setiap gambar menggunakan *Roboflow* dengan format *YOLO*.
3. **Pembuatan Variasi Anotasi (Multi-Skenario):**

   Dataset pelatihan (*train set*) dimodifikasi dengan menerapkan dua skenario penghapusan anotasi:

   - **Skenario A**: Menghapus anotasi objek yang terlihat jelas (*clear/unoccluded*) terlebih dahulu.
   - **Skenario B**: Menghapus anotasi objek yang tertumpuk atau mengalami *occlusion* terlebih dahulu.

   Pada masing-masing skenario, tingkat penghapusan dilakukan secara bertahap sebesar 10% dari dataset penuh, menghasilkan variasi anotasi 90%, 80%, 70%, dan 60%.
4. **Pelatihan Model:**

   Melatih masing-masing variasi dataset dengan arsitektur YOLOv11 menggunakan parameter yang diseragamkan (jumlah *epoch*, *learning rate*, ukuran gambar, dan *optimizer*) di Google Colab.
5. **Evaluasi Performa:**

   Menguji performa setiap model pada dataset validasi dan uji yang tetap beranotasi penuh (100%). Evaluasi dilakukan menggunakan metrik:

   - *Precision*
   - *Recall*
   - *mAP@0.5*
   - *mAP@0.5:0.95*
6. **Visualisasi & Analisis:**

   Menyajikan hasil pelatihan dalam bentuk grafik *loss*, *confusion matrix*, tabel performa, serta hasil visual deteksi model. Analisis dilakukan untuk membandingkan performa antar skenario dan tingkat anotasi menggunakan aplikasi Streamlit.
7. **Penentuan Ambang Batas & Strategi Anotasi Efisien:**

   Mengidentifikasi tingkat anotasi minimum dari setiap skenario yang masih mempertahankan penurunan mAP@0.5 di bawah 10% dari baseline, sekaligus menarik

kesimpulan mengenai jenis objek apa yang paling layak untuk diprioritaskan dalam anotasi demi efisiensi waktu dan akurasi model.

3.3 Variabel Penelitian

Penelitian ini melibatkan dua kelompok variabel utama: variabel bebas berupa tingkat anotasi dataset dan variabel terikat berupa performa model deteksi objek. Untuk menjaga validitas perbandingan antar model, sejumlah variabel kendali dipertahankan konstan selama seluruh rangkaian eksperimen. Selain itu, didefinisikan ambang batas (*threshold*) performa sebagai dasar penentuan tingkat anotasi minimum yang masih layak digunakan jika dibandingkan dengan kondisi anotasi penuh (100%).

Penelitian ini juga mengimplementasikan dua skenario penghapusan anotasi untuk mensimulasikan prioritas kerja anotator di dunia nyata, yaitu:

- **Skenario A**: Menghapus objek yang *terlihat utuh/jelas* terlebih dahulu.
- **Skenario B**: Menghapus objek yang *tertumpuk/terocclusion* terlebih dahulu.

Kedua skenario ini dirancang untuk mengevaluasi strategi prioritas anotasi yang lebih efisien dan berdampak minimal terhadap penurunan performa model.

3.3.1 Variabel Bebas (*Independent Variable*)

Variabel bebas terdiri dari dua dimensi utama:

1. **Tingkat anotasi**

   Merupakan proporsi bounding box yang dipertahankan dari kondisi penuh (*fully annotated*) pada subset *train*, dikurangi secara bertahap sebesar 10%. Variasi yang digunakan dalam eksperimen meliputi:

   - 100% (baseline)
   - 90%
   - 80%
   - 70%
   - 60%

   Persentase ini dihitung terhadap jumlah anotasi awal *secara global* pada seluruh citra *train*, bukan per kelas. Subset *validation* dan *test* dipertahankan pada kondisi *100%* untuk menjamin evaluasi antar model dilakukan secara adil.

2. **Skenario penghapusan anotasi**

   Menentukan prioritas objek yang dihapus terlebih dahulu dari gambar:

- **Skenario A**: objek *utuh* dihapus terlebih dahulu, jika kuota habis dilanjutkan dengan objek tertumpuk.
- **Skenario B**: objek *tertumpuk* dihapus terlebih dahulu, jika kuota habis dilanjutkan dengan objek utuh.

Dengan kombinasi dua dimensi ini, penelitian menghasilkan total 10 variasi dataset, yaitu 5 tingkat anotasi × 2 skenario.

3.3.2 Variabel Terikat (*Dependent Variables*)

Variabel terikat adalah performa model deteksi objek *YOLOv11* setelah dilatih pada masing-masing tingkat anotasi. Performa model dievaluasi menggunakan empat metrik kuantitatif standar:

- *Precision*
- *Recall*
- *mAP@0.5*
- *mAP@0.5:0.95*

Seluruh metrik dihitung berdasarkan hasil inferensi terhadap *validation* dan *test set* yang menggunakan anotasi penuh (100%). Fokus analisis adalah pada perubahan performa *secara keseluruhan*, bukan per kelas, meskipun *analisis tambahan per kelas* tetap dilakukan pada Bab 4.

3.3.3 Variabel Kendali (Control Variables)

Untuk memastikan bahwa perubahan performa semata-mata disebabkan oleh variasi tingkat anotasi, faktor-faktor berikut dikendalikan secara konsisten pada seluruh eksperimen:

- **Arsitektur model:** *YOLOv11s* (varian ringan, konsisten di semua percobaan).
- **Parameter pelatihan utama:** jumlah *epoch*, ukuran citra (*imgsz = 640*), *optimizer*, dan *learning rate* seragam untuk semua model.
- **Lingkungan komputasi:** pelatihan dilakukan di *Google Colab* dengan akselerasi GPU berbasis *CUDA*; versi pustaka *Ultralytics* dikunci.
- **Dataset:** kumpulan citra untuk *train*, *validation*, dan *test* tetap sama untuk semua eksperimen; yang berubah hanya proporsi anotasi pada *train*.
- **Preprocessing dataset:** pada saat ekspor dari *Roboflow*, diterapkan *Auto-Orient* dan *Resize* (stretch ke 640×640) secara seragam.

Pengendalian ini penting untuk mengisolasi pengaruh variabel bebas terhadap variabel terikat.

3.3.4 Definisi Operasional *Threshold* Performa

Dalam penelitian ini, *threshold performa* didefinisikan sebagai tingkat anotasi minimum yang masih mampu menghasilkan performa model yang stabil dan layak digunakan jika dibandingkan dengan *baseline* (model beranotasi penuh 100%). Acuan utama penentuan batas ini adalah nilai *mean Average Precision* pada ambang 0.5 (*mAP@0.5*), karena metrik ini paling langsung menggambarkan keberhasilan deteksi objek pada toleransi *Intersection over Union* (IoU) yang umum digunakan.

Model dengan tingkat anotasi yang lebih rendah dikategorikan masih stabil apabila nilai *mAP@0.5* tidak mengalami penurunan lebih dari 10% dibandingkan nilai *baseline* 100%. Rumusnya ada pada persamaan (3.1).

$$mAP_{\text{min }_layak} = (1 - 0.1) \times 100\% \tag{3.1}$$

Jika $\text{mAP}_{\text{model}} \geq \text{mAP}_{\text{min_layak}}$, maka model dianggap masih berada dalam batas toleransi. Sebagai contoh, jika model 100% memiliki *mAP@0.5* = 0.800, maka model pada tingkat anotasi lebih rendah masih dianggap layak jika *mAP@0.5* ≥ 0.720.

Selain *mAP@0.5* sebagai indikator utama, perubahan *precision* turut diamati sebagai indikator sekunder untuk menilai apakah penurunan anotasi meningkatkan kesalahan deteksi (*false positives*) secara signifikan.

Nilai ambang ini digunakan dalam analisis akhir (Bab 4 dan Bab 5) untuk menentukan proporsi anotasi minimum yang dapat diterapkan tanpa menurunkan performa secara signifikan dalam aplikasi praktis, misalnya deteksi buah berbasis citra. Ringkasan jenis variabel, deskripsi, serta perlakuan yang diterapkan dalam setiap eksperimen disajikan pada Tabel 3.1.

Tabel 3.1 Ringkasan Variabel Penelitian

Jenis	Nama / Simbol	Deskripsi	Perlakuan / Nilai / Kondisi
Bebas	Tingkat Anotasi *Train*	Persentase *bounding box* yang dipertahankan dari anotasi penuh	100% (*baseline*), 90%, 80%, 70%, 60%
Bebas	Skenario Anotasi	Urutan prioritas penghapusan objek	Skenario A (hapus utuh dulu), Skenario B (hapus tertumpuk dulu)
Terikat	*Precision*	Proporsi deteksi benar terhadap total deteksi	Diukur pada *val/test* 100% anotasi

Terikat	*Recall*	Proporsi objek terdeteksi terhadap total *ground truth*	Diukur pada *val/test* 100% anotasi
Terikat	*mAP@0.5*	Rata-rata AP antar kelas pada IoU 0.5	Indikator utama *threshold* stabilitas
Terikat	*mAP@0.5:0.95*	AP rata-rata pada IoU 0.5–0.95	Sensitivitas terhadap ketelitian lokalisasi
Tambahan Analisis	Jumlah Deteksi	Total prediksi objek pada *test*	Digunakan untuk melihat kecenderungan over/under-detect
Tambahan Analisis	Rata-rata *Confidence*	Rerata skor keyakinan prediksi	Indikator agresivitas model & kepercayaan deteksi
Kendali	Arsitektur	Varian model yang digunakan	*YOLOv11s* seragam di semua eksperimen
Kendali	Parameter Pelatihan	Epoch, batch, lr, imgsz	Diset sama di seluruh model
Kendali	Lingkungan Komputasi	Platform & runtime	*Google Colab* GPU, versi *Ultralytics* tetap
Kendali	Dataset Val/Test	Data evaluasi	Tetap 100% anotasi; tidak diubah
Kendali	Preprocessing Dataset	Transformasi sebelum pelatihan	*Auto-Orient* + *Resize 640×640* (Roboflow); tanpa augmentasi tambahan
Kendali	Transformasi Pelatihan	Augment pipeline training	Tidak ditambah; hanya transformasi internal minimum YOLO
Indikator Analisis	*Threshold* Stabilitas	Penurunan performa relatif yang dapat ditoleransi	≤10% drop dari *baseline* 100% pada *mAP@0.5* (utama) & *precision* (sekunder)

3.4 Dataset dan Objek Penelitian

Objek penelitian dalam studi ini adalah performa model deteksi objek berbasis *YOLOv11* ketika dilatih pada dataset dengan tingkat dan strategi anotasi yang bervariasi. Fokus utama penelitian bukan sekadar mengukur akurasi model secara absolut, melainkan mengevaluasi sensitivitas performa terhadap kekurangan label anotasi, terutama dalam konteks keterbatasan sumber daya pelabelan.

Untuk itu, digunakan domain uji berupa gambar buah-buahan yang memiliki keragaman visual cukup kompleks, seperti objek bertumpuk, occlusion antar buah, perbedaan ukuran yang signifikan, serta kemiripan warna antara objek dan latar belakang. Karakteristik tersebut menjadikan buah sebagai objek uji yang relevan untuk mengevaluasi strategi efisiensi anotasi dalam pengembangan model deteksi berbasis deep learning.

Selain variasi jumlah anotasi, penelitian ini juga mempertimbangkan dua skenario penghapusan anotasi berdasarkan prioritas visibilitas objek:

- **Skenario A**: menghapus objek yang terlihat utuh dan jelas terlebih dahulu,
- **Skenario B**: menghapus objek yang tertumpuk atau terocclusion terlebih dahulu.

Setiap skenario tersebut diterapkan pada berbagai tingkat anotasi (100% hingga 60%) untuk mengkaji dampak strategi pemilihan objek terhadap performa model.

3.4.1 Kelas Objek

Dataset mencakup enam kelas buah: *jeruk, apel, stroberi, anggur, rambutan*, dan *salak*. Pemilihan kelas ini dilakukan dengan mempertimbangkan dua aspek utama, yaitu ketersediaan secara lokal dan kompleksitas visual yang mewakili tantangan nyata dalam deteksi objek. Seluruh buah tersebut tergolong umum dan mudah ditemukan di pasar-pasar tradisional Indonesia, sehingga relevan secara kontekstual dengan penerapan praktis di lingkungan lokal.

Selain itu, keenam kelas buah ini memiliki variasi signifikan dalam hal warna permukaan (kuning, oranye, merah, hijau, ungu, cokelat), bentuk (bulat, lonjong), dan tekstur (halus, berbulu, bersisik). Karakteristik visual tersebut memengaruhi tingkat kesulitan dalam proses anotasi dan deteksi. Misalnya, anggur dan rambutan sering muncul dalam kondisi bertumpuk dan memiliki ukuran kecil, yang berisiko mengalami *occlusion*. Sementara itu, stroberi dan rambutan memiliki warna dominan merah dan ukuran yang serupa, sehingga mudah tertukar oleh model jika tidak diajarkan dengan baik. Buah seperti salak memiliki tekstur unik yang bisa membingungkan model pada latar serupa seperti kayu atau tanah.

Dengan mempertimbangkan kompleksitas tersebut, enam buah ini dipilih karena mampu mewakili berbagai tantangan visual yang umum ditemui dalam tugas deteksi objek di dunia nyata. Pemilihan yang beragam ini juga memungkinkan evaluasi model dilakukan secara lebih menyeluruh terhadap performa generalisasi dan sensitivitas terhadap kekurangan anotasi.

3.4.2 Sumber dan Jumlah Data

Dataset disusun dari dua sumber utama, yaitu *iStock* (penyedia stok foto berlisensi) dan *Google Images* (pencarian gambar daring). Pemilihan kedua sumber ini bertujuan untuk memperoleh variasi visual yang luas, mencakup kondisi pencahayaan, latar belakang, dan sudut pengambilan gambar yang beragam. Total terkumpul 240 citra, dengan distribusi relatif seimbang untuk setiap kelas buah, yakni sekitar 40 citra per kelas.

Objek yang dilibatkan terdiri dari enam kelas buah: *jeruk, apel, stroberi, anggur, rambutan, dan salak.* Setiap citra dapat berisi satu hingga beberapa objek, sehingga jumlah anotasi (bounding box) melebihi jumlah citra. Proses anotasi penuh dilakukan untuk semua objek yang

terlihat jelas dalam setiap citra sebelum dilakukan pengurangan anotasi secara sistematis pada subset *train*.

3.4.3 Proses Anotasi

Seluruh citra dianotasi secara manual menggunakan *bounding box* di platform *Roboflow*. Anotator menandai setiap buah yang tampak jelas di dalam citra. Anotasi awal ini digunakan sebagai dataset 100% (baseline). Setelah verifikasi awal selesai, dataset diekspor ke format *YOLO* dengan preprocessing seragam: *Auto-Orient* dan *Resize* ke 640×640 piksel (lihat konfigurasi pada Subbab 3.5). Kualitas anotasi sangat menentukan validitas eksperimen karena seluruh variasi anotasi yang lebih rendah berasal dari manipulasi dataset ini.

3.4.4 Variasi Tingkat Anotasi

Untuk mensimulasikan keterbatasan sumber daya anotasi, dilakukan pengurangan label secara global bertahap 10% dari total anotasi awal. Pengurangan hanya diterapkan pada subset *train*, sedangkan pada subset *validation* dan *test* dipertahankan 100% anotasi agar evaluasi antar model adil dan terbandingkan. Dengan strategi ini dihasilkan lima versi dataset *train*:

- 100% (*baseline*)
- 90%
- 80%
- 70%
- 60%

Setiap tingkat anotasi diterapkan pada dua skenario (A dan B), sehingga menghasilkan total 10 kombinasi dataset pelatihan.

3.4.5 Distribusi Dataset Awal

Distribusi jumlah citra per kelas pada dataset awal ditunjukkan pada Tabel 3.2. Jumlah anotasi per kelas dapat bervariasi karena satu citra dapat mengandung banyak objek dan rincian jumlah *bounding box* tersedia pada Lampiran.

Tabel 3.2 Distribusi Dataset Awal per Kelas

Kelas	Jumlah Citra	Karakteristik Visual Singkat	Tantangan Deteksi
Jeruk	~40	Warna oranye/kuning, permukaan halus	Pantulan cahaya, latar plastik
Apel	~40	Merah/hijau, kilap kuat	Spekular highlight, variasi warna

Stroberi	~40	Merah dengan biji kuning, daun hijau	Ukuran kecil, sering tertutup daun
Anggur	~40	Bergerombol, ungu/hijau gelap	Objek kecil padat, occlusion tinggi
Rambutan	~40	Hijau/merah berduri, rambut halus	Tekstur kompleks, bentuk tidak kaku
Salak	~40	Cokelat bersisik, bentuk kerucut	Warna mirip latar tanah/kayu

3.4.6 Integrasi Platform Eksperimen

Seluruh proses dari anotasi hingga pelatihan dan analisis dilakukan lintas platform:

- *Roboflow* → manajemen dataset, anotasi, preprocessing, ekspor format *YOLO*.
- *Google Colab* → pelatihan *YOLOv11* per tingkat anotasi dengan akselerasi GPU.
- *VS Code* / lingkungan lokal → pengolahan hasil, pembuatan tabel, analisis numerik, penulisan skripsi.
- *Streamlit* → visualisasi interaktif perbandingan hasil deteksi antar model.

Alur integrasi platform ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Integrasi Platform Roboflow, Google Colab, dan VSCode dalam Alur Penelitian.

3.4.7 Relevansi Dataset terhadap Tujuan Penelitian

Pemilihan dataset buah-buahan ini sangat mendukung tujuan penelitian, yaitu mengevaluasi dampak pengurangan anotasi terhadap model deteksi objek. Kombinasi objek yang tertumpuk, terpotong, dan terlihat jelas memungkinkan pengujian dua strategi penghapusan anotasi secara komparatif.

Keberadaan berbagai tingkat occlusion dan variasi visibilitas objek memperkuat validitas skenario A dan B dalam konteks nyata, di mana anotator mungkin harus memilih objek mana yang perlu diprioritaskan saat waktu atau sumber daya terbatas. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan rekomendasi yang praktis dan aplikatif dalam strategi anotasi efisien.

3.5 Teknik Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan secara terstruktur melalui beberapa tahap, mulai dari pemilihan gambar hingga persiapan dataset untuk pelatihan *YOLOv11*. Data utama

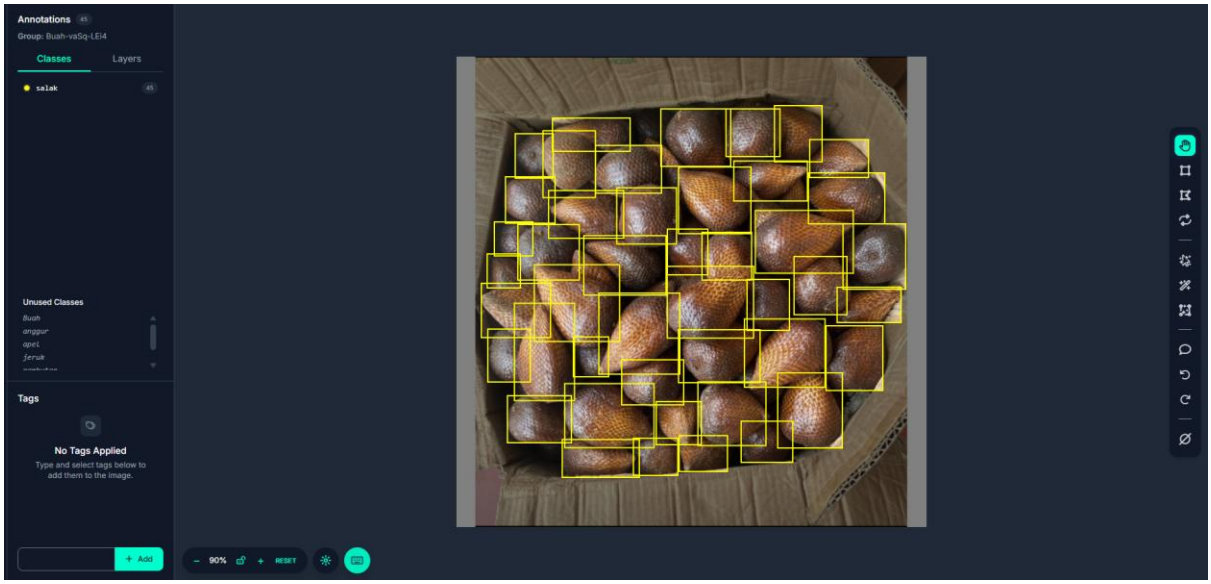
yang digunakan adalah citra buah dengan enam kelas objek: *jeruk, apel, stroberi, anggur, rambutan,* dan *salak.* Proses pengumpulan dan anotasi dilakukan secara manual untuk memastikan kualitas, keragaman visual, serta representasi kelas yang seimbang.

3.5.1 Pengumpulan Gambar

Citra dikumpulkan dari dua sumber utama, yaitu iStock dan Google Images, untuk memperoleh variasi visual yang kaya. Pemilihan gambar mempertimbangkan faktor keragaman latar belakang, kondisi pencahayaan, dan sudut pengambilan gambar. Total diperoleh 240 citra, dengan distribusi ±40 citra per kelas, sehingga setiap kelas objek memiliki representasi visual yang cukup seimbang.

3.5.2 Proses Anotasi dan Pembuatan Skenario

Setelah citra terkumpul, seluruh gambar dianotasi secara manual menggunakan platform *Roboflow*. Setiap objek buah yang tampak jelas pada citra diberi *bounding box* sesuai kelasnya dalam format *YOLO*. Anotasi ini menghasilkan dataset 100% (*baseline*), di mana semua objek pada citra dilabeli secara penuh. Contoh tampilan anotasi lengkap ditunjukkan pada Gambar 3.3.



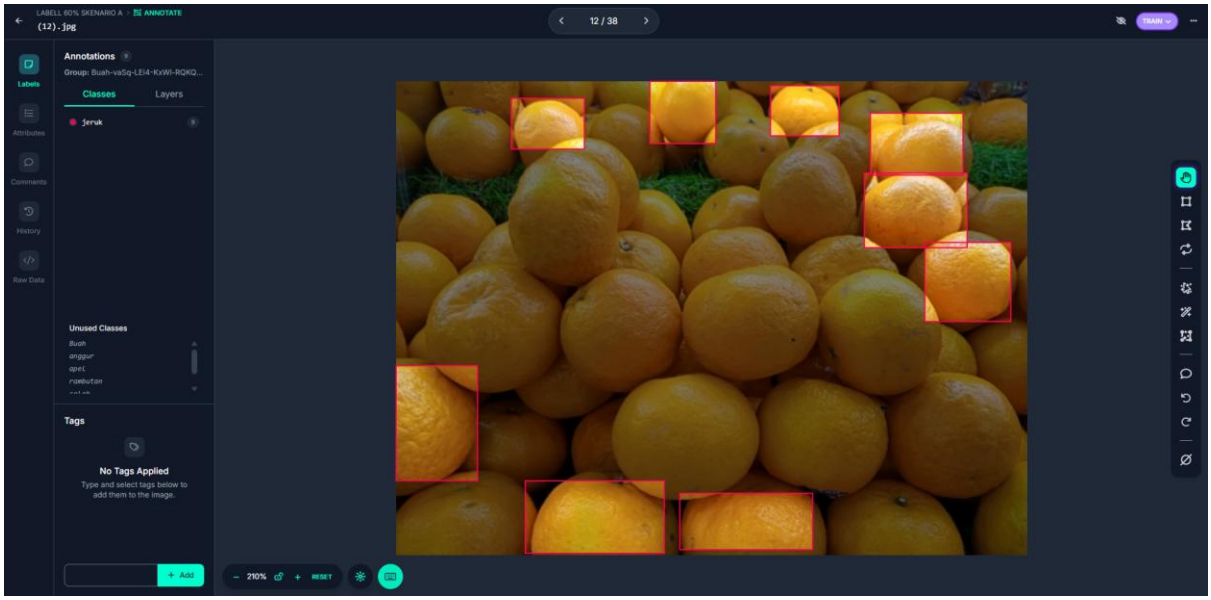
Gambar 3.3 Anotasi untuk Dataset 100%

Untuk mensimulasikan keterbatasan sumber daya anotasi, dilakukan pengurangan jumlah bounding box hanya pada subset *train*. Pengurangan dilakukan bertahap sebesar 10%, sehingga terbentuk variasi tingkat anotasi: 100%, 90%, 80%, 70%, dan 60%.

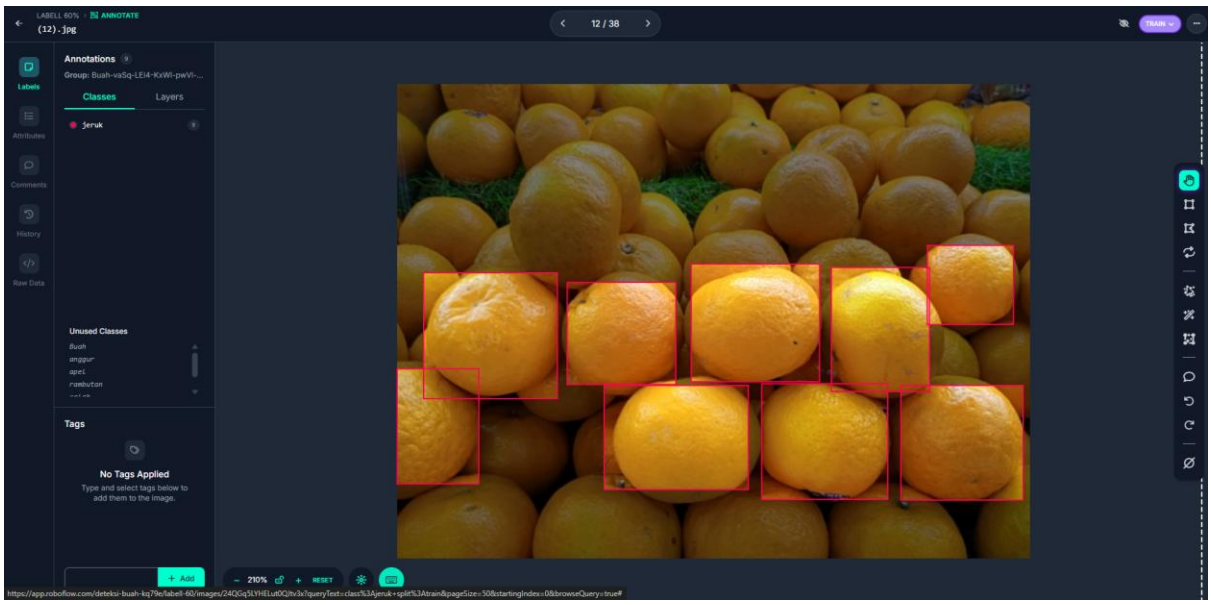
Penghapusan bounding box dilakukan berdasarkan dua strategi berbeda, sehingga menghasilkan dua skenario eksperimen:

- **Skenario A**: Bounding box pada objek yang terlihat jelas, utuh, dan berada di tengah gambar dihapus terlebih dahulu. Tujuannya adalah untuk mensimulasikan kondisi pelabelan terburu-buru yang hanya menangani objek-objek samar atau pinggir gambar.
- **Skenario B**: Bounding box pada objek yang terhalang (occlusion), bertumpuk, atau berada di bagian sudut dihapus terlebih dahulu. Tujuannya adalah mensimulasikan kondisi pelabelan yang hanya memprioritaskan objek yang mudah diakses atau dikenali.

Masing-masing skenario menghasilkan satu set dataset terpisah dengan susunan bounding box yang berbeda walaupun tingkat persentase anotasinya sama. Perbedaan hasil anotasi antar skenario dapat dilihat pada Gambar 3.4 dan Gambar 3.5 berikut.



Gambar 3.4 Anotasi dataset 60 % Skenario A



Gambar 3.5 Anotasi dataset 60 % Skenario B

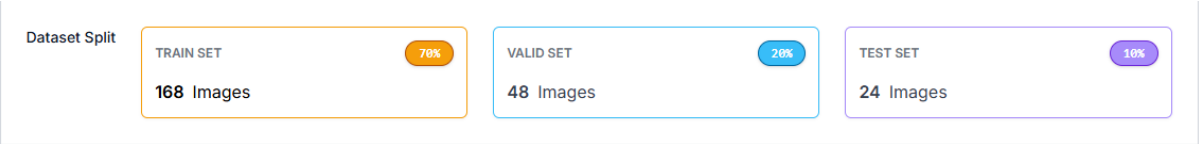
3.5.3 Split Dataset

Dataset lengkap dari Roboflow secara otomatis dibagi menjadi tiga subset:

- 70% untuk pelatihan (*train*),
- 20% untuk validasi (*validation*),
- 10% untuk pengujian (*test*).

Hanya subset *train* yang mengalami pengurangan anotasi bertahap (100%, 90%, 80%, 70%, 60%), sedangkan subset *validation* dan *test* tetap beranotasi penuh untuk evaluasi model. Pembagian ini diatur otomatis pada platform Roboflow, seperti terlihat pada Gambar 3.6 yang

menghasilkan 168 gambar untuk pelatihan, 48 gambar untuk validasi, dan 24 gambar untuk pengujian.



Gambar 3.6 Pembagian Dataset pada Platform Roboflow (Train/Valid/Test).

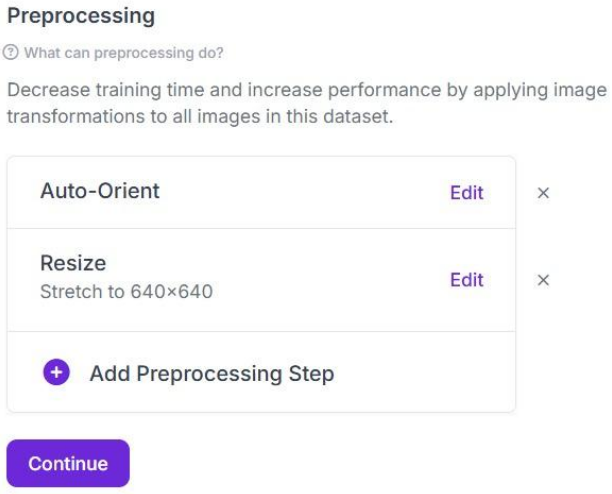
3.5.4 Preprocessing

Sebelum digunakan dalam pelatihan, data citra yang telah dianotasi diproses lebih lanjut melalui tahap *preprocessing* pada platform *Roboflow*. *Preprocessing* bertujuan untuk memastikan bahwa semua citra memiliki ukuran dan orientasi seragam sehingga model *YOLOv11* dapat memproses data secara efisien. Selain itu, *preprocessing* dapat membantu mempercepat waktu pelatihan serta mengoptimalkan performa model.

Dalam penelitian ini, metode preprocessing yang digunakan meliputi:

1. **Auto-Orient** – Menyesuaikan orientasi gambar berdasarkan metadata EXIF, sehingga tidak ada gambar yang terbalik atau miring yang berpotensi mengganggu deteksi objek.
2. **Resize (640 × 640 piksel)** – Mengubah resolusi seluruh citra ke ukuran 640 × 640 piksel agar sesuai dengan standar input model *YOLOv11*. Hal ini memastikan konsistensi dimensi input dan meminimalkan risiko distorsi skala saat pelatihan.

Contoh tampilan antarmuka pemilihan metode preprocessing pada Roboflow ditunjukkan pada Gambar 3.7.



Gambar 3.7 Proses Pemilihan Metode *Preprocessing* di *Roboflow*

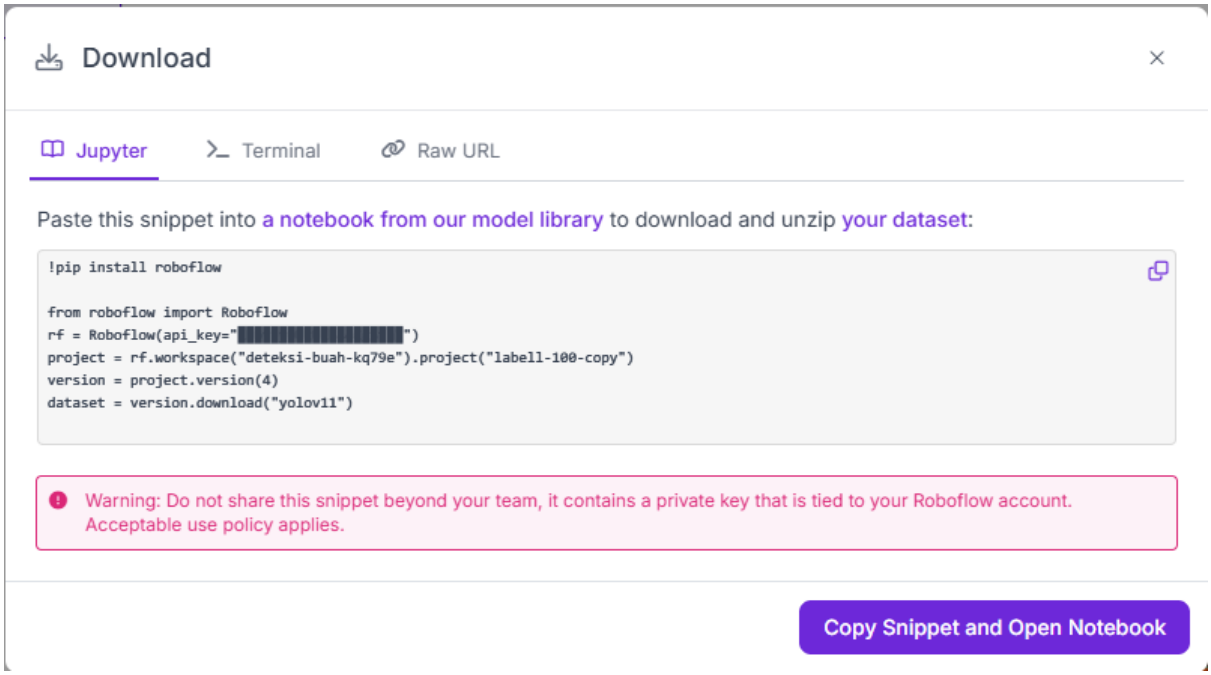
3.5.5 Ekspor Dataset

Tahap ekspor dataset merupakan proses penting untuk memindahkan dataset yang telah dianotasi di platform Roboflow ke lingkungan pelatihan model, dalam hal ini *Google Colab*. Roboflow menyediakan fitur *API-based dataset export*, yang memungkinkan pengguna mengunduh dataset secara otomatis ke dalam format yang sesuai dengan arsitektur model yang digunakan. Dalam penelitian ini, format *YOLOv11* dipilih untuk kompatibilitas langsung dengan pustaka *Ultralytics YOLO*.

Proses ekspor ini dimulai dengan memilih format dataset pada menu ekspor di Roboflow, di mana pengguna dapat menentukan format output (misalnya YOLOv11, YOLOv8, COCO JSON, atau Pascal VOC). Setelah memilih format, Roboflow secara otomatis menghasilkan *download snippet* (cuplikan kode Python) yang dapat dijalankan di notebook Google Colab. *Snippet* ini mencakup perintah untuk:

- **Menginstal pustaka Roboflow** (!pip install roboflow).
- **Melakukan autentikasi** dengan API key yang terkait dengan akun Roboflow pengguna.
- **Memilih workspace dan proyek** dataset.
- **Mengunduh dataset** sesuai versi yang ditentukan (misalnya, versi 4 untuk dataset 100% anotasi).

Cuplikan antarmuka *download snippet* dari Roboflow ditunjukkan pada Gambar 3.8. Antarmuka ini menampilkan peringatan keamanan untuk tidak membagikan *API key* kepada pihak lain, karena dapat memberikan akses penuh ke dataset pengguna.



Gambar 3.8 Cuplikan kode dari Roboflow untuk mengakses dataset dalam format YOLOv11, yang digunakan untuk integrasi langsung ke Google Colab.

Selain itu, Roboflow juga memberikan opsi *Copy Snippet and Open Notebook*, yang memungkinkan pengguna menyalin kode langsung ke notebook Colab. Ini sangat membantu untuk mempercepat integrasi dataset dengan *pipeline* pelatihan model. Tampilan kode hasil ekspor ditunjukkan pada Gambar 3.9.

```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

from roboflow import Roboflow
rf = Roboflow(api_key=" --------------------")
project = rf.workspace("deteksi-buah-kq79e").project("labell-100-copy")
version = project.version(4)
dataset = version.download("yolov11")
```

Gambar 3.9 Kode Python yang digunakan untuk mengunduh dataset dari *Roboflow* ke *Google Colab* menggunakan API.

3.6 Teknik Pengurangan Anotasi

Teknik *pengurangan anotasi* digunakan untuk mensimulasikan kondisi kerja anotasi yang tidak diselesaikan secara menyeluruh, seperti ketika *annotator* berhenti memberi label karena keterbatasan waktu, kelelahan, atau fokus terhadap objek-objek mayoritas. Strategi ini tidak hanya relevan dalam konteks akurasi, tetapi juga menyangkut efisiensi waktu. Dengan

menerapkan pendekatan ini, proses anotasi dapat dipercepat secara signifikan, yang pada penelitian ini ditunjukkan dengan penurunan beban anotasi hingga 40%, sehingga berpotensi menghemat waktu kerja manual hingga 2–3 jam per 100 citra, tergantung pada kompleksitas objek di dalam gambar. Dalam implementasinya, pengurangan hanya diterapkan pada *subset* dari *train set*, sementara *validation set* dan *test set* tetap menggunakan data dengan anotasi penuh (*fully annotated*) sebesar 100%. Pendekatan ini bertujuan menjaga validitas evaluasi antarvarian model secara adil dan terstandarisasi.

Dalam penelitian ini, digunakan dua skenario eksperimen berbeda untuk menghapus *bounding box* (label) dalam dataset. Kedua skenario tersebut dirancang untuk merepresentasikan kebiasaan pelabelan dalam situasi nyata.

Pengurangan dilakukan secara bertahap dalam interval 10%, dimulai dari kondisi penuh (100%), sehingga diperoleh variasi *train set* dengan tingkat anotasi 100% (baseline), 90%, 80%, 70%, dan 60%.

3.6.1 Skenario Pengurangan Anotasi

Dalam penelitian ini, digunakan dua skenario penghapusan anotasi *bounding box* pada dataset. Rancangan ini bertujuan untuk mensimulasikan praktik pelabelan dalam kondisi nyata (*real-world annotation practices*), di mana keterbatasan waktu, tingkat kesulitan objek, serta persepsi annotator terhadap prioritas objek dapat memengaruhi hasil akhir dataset. Strategi pengurangan anotasi dipandang penting karena dapat menunjukkan bagaimana variasi kualitas *label* berdampak langsung terhadap performa model deteksi objek.

Pengurangan anotasi dilakukan secara bertahap dengan interval 10% dari kondisi penuh (100%), sehingga dihasilkan lima variasi tingkat anotasi: 100% (*baseline*), 90%, 80%, 70%, dan 60%. Melalui pendekatan ini, penelitian mampu mengamati secara sistematis perubahan performa model YOLOv11 terhadap tingkat penghapusan anotasi pada objek dengan karakteristik yang berbeda.

- **Skenario A: Prioritaskan Pengurangan Anotasi Objek Jelas**

Pada skenario ini, penghapusan dilakukan terlebih dahulu pada objek yang berukuran besar, berada di pusat citra, terlihat *clear*, dan tidak tertutup objek lain. Pendekatan ini mensimulasikan situasi ketika annotator hanya sempat melabeli objek-objek kecil atau kompleks, sedangkan objek utama yang jelas justru diabaikan. Strategi ini relevan karena objek yang *fully visible* terbukti memberikan informasi visual yang lebih stabil dan konsisten dalam mendukung performa model deteksi. Penelitian lokal menunjukkan bahwa deteksi objek

berbasis YOLO menghasilkan akurasi lebih tinggi ketika target berada pada kondisi visual yang jelas dan utuh, misalnya pada kasus deteksi manusia dan kendaraan dalam citra nyata (Setiyadi, Utami, & Ariatmanto, 2023).

- Aturan (Rule):
  - *Bounding box* objek dominan (besar, utuh, pusat gambar) dihapus terlebih dahulu.
  - Jika kuota atau jumlah anotasi untuk objek jelas/utuh sudah terpenuhi, maka penghapusan dilanjutkan pada objek yang tertutup atau mengalami *occlusion*.

- **Skenario B: Prioritaskan Pengurangan Anotasi Objek Tertumpuk**

Sebaliknya, pada skenario ini penghapusan dilakukan terlebih dahulu pada objek yang relatif sulit dikenali, seperti objek kecil, terletak di tepi citra, atau mengalami *occlusion*. Pendekatan ini merepresentasikan kecenderungan annotator yang lebih dulu memberi *label* pada objek mudah dikenali, sementara objek kompleks atau tidak terlihat sepenuhnya diabaikan. Fenomena ini konsisten dengan literatur yang menegaskan bahwa *occluded objects* menjadi salah satu tantangan terbesar dalam meningkatkan akurasi deteksi (Ning, Menglu, Hao, Xueping, & Yunhong, 2021).

- Aturan (Rule):
  - *Bounding box* objek minor (*occluded*, bertumpuk, kecil) dihapus terlebih dahulu.
  - Jika kuota atau jumlah anotasi untuk objek tertutup sudah terpenuhi, maka penghapusan dilanjutkan pada objek yang terlihat jelas/utuh.

Kedua skenario ini diterapkan pada seluruh variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%). Tujuannya adalah memperoleh gambaran yang lebih komprehensif mengenai dampak penghapusan anotasi berdasarkan visibilitas objek. Dengan cara ini, penelitian tidak hanya menilai penurunan akurasi deteksi, tetapi juga mengkaji *trade-off* antara efisiensi anotasi (*annotation efficiency*) dan kualitas data. Hasil dari pendekatan ini diharapkan dapat menjadi dasar dalam menentukan strategi anotasi yang lebih efisien dan realistis, khususnya pada aplikasi deteksi multi-objek buah menggunakan YOLOv11.

Penting untuk dicatat bahwa pengurangan anotasi tidak dilakukan dengan mengurangi jumlah citra dalam dataset, melainkan hanya dengan menghapus sebagian *bounding box* pada subset *train*. Dengan demikian, jumlah gambar yang digunakan pada setiap tingkat anotasi tetap sama, sedangkan variasi terletak pada jumlah objek yang dianotasi. Selain itu, subset

validation dan *test* dipertahankan dalam kondisi 100% beranotasi, sehingga proses evaluasi model tetap konsisten, objektif, dan dapat merefleksikan performa secara adil.

3.6.2 Prinsip Pengurangan Proporsional

Untuk setiap citra dalam *train set*, diasumsikan terdapat *N bounding box* pada kondisi awal (100%). Untuk menghasilkan varian dengan tingkat anotasi *p* (misalnya *p = 0.90*), jumlah label yang dipertahankan dihitung menggunakan Persamaan (3.2):

$$N_{\mathrm{p}} = [\mathrm{p} \times N] \tag{3.2}$$

Artinya, hanya N_{p} entri pertama pada file label yang dipertahankan, sementara sisanya dihapus. Skema ini dapat dirangkum sebagai berikut:

- 100% → simpan semua N label.
- 90% → simpan 0.9N label pertama.
- 80% → simpan 0.8N label pertama.
- 70% → simpan 0.7N label pertama.
- 60% → simpan 0.6N label pertama.

Jika nilai *N* sangat kecil dan pembulatan menyebabkan kehilangan terlalu besar, maka diterapkan aturan minimum penyimpanan satu label per citra.

3.6.3 Contoh Ilustratif

Sebagai ilustrasi, misalkan terdapat sebuah citra dalam *train set* pada versi 100% yang memiliki 75 *bounding box* untuk kelas salak. Untuk membuat versi 90%, hanya 90% dari entri awal yang dipertahankan, yaitu: $0.90 \times 75 = 67.5$ → disimpan 67 label. Untuk 80% → simpan 60 label, 70% → 52 label, 60% → 45 label. Label yang tidak disertakan diasumsikan tidak sempat dianotasi oleh *annotator* dalam skenario dunia nyata yang dihadapkan pada tekanan waktu atau kelelahan.

3.6.4 Validasi Proporsi

Setelah proses *pruning* dilakukan, dilakukan beberapa langkah verifikasi sebagai berikut:

- Pemeriksaan total jumlah *bounding box* pada setiap varian dataset untuk memastikan proporsinya mendekati target global.
- Verifikasi bahwa semua file gambar tetap sama, yang dimana tidak ada citra yang dihapus dari *train set*.

- Inspeksi visual terhadap beberapa sampel citra melalui *Roboflow*, guna memastikan bahwa pengurangan label benar-benar tercermin (ditunjukkan pada Gambar 3.3**,** Error! Reference source not found.**,** Error! Reference source not found. **,**Error! Reference source not found. dan **Error! Reference source not found.**).

3.6.5 Catatan Bias dan Relevansi Metodologis

Metode pengurangan anotasi ini membawa potensi bias:

- **Skenario A** cenderung menghasilkan model yang mengenali objek kecil dan sulit, namun mungkin kehilangan konteks pada objek dominan.
- **Skenario B** menghasilkan model yang kuat pada objek utama, namun cenderung mengabaikan objek pinggir atau kecil, menyebabkan potensi *false negatives*.

Justru karena bias ini mencerminkan kondisi nyata proses anotasi di dunia praktis (misal proyek *crowdsourcing* atau *labeling* manual terbatas), maka kedua skenario ini sangat relevan untuk menunjukkan bagaimana efisiensi anotasi dapat diatur tanpa terlalu mengorbankan performa deteksi.

3.7 Pemilihan Model

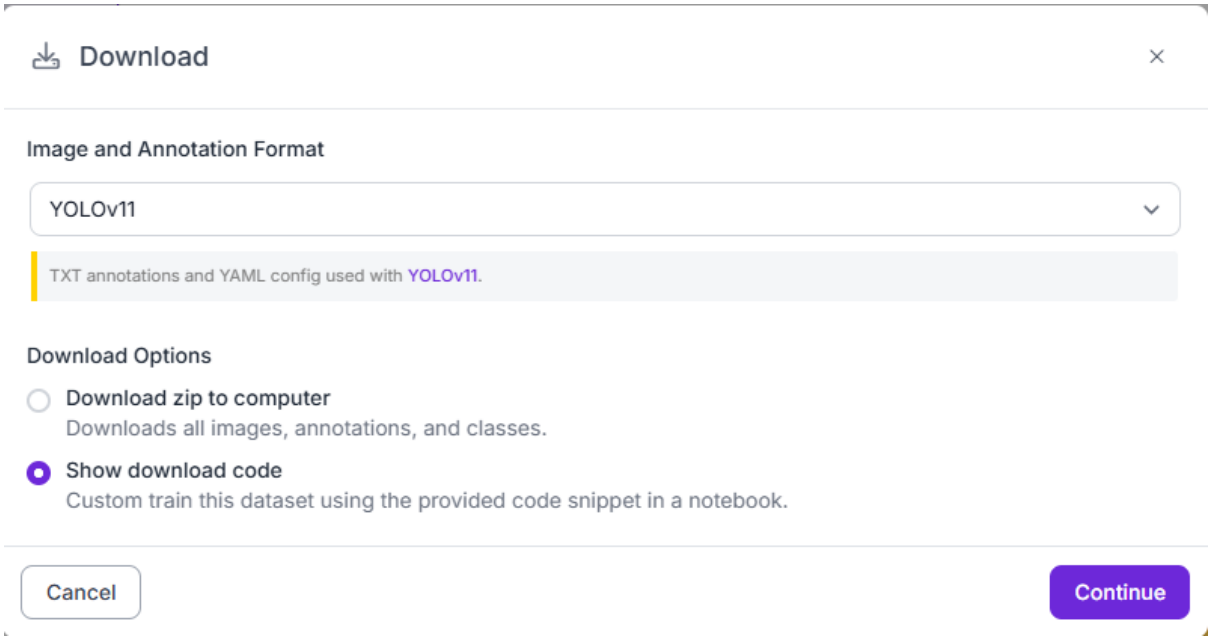
Pemilihan model deteksi objek dalam penelitian ini merupakan tahap yang krusial karena menentukan kualitas hasil yang diperoleh. Berbagai arsitektur telah dikembangkan dalam bidang deteksi objek, mulai dari algoritma dua tahap seperti *Faster R-CNN* yang dikenal akurat namun memerlukan waktu komputasi relatif lama, hingga algoritma satu tahap seperti *SSD* dan keluarga *YOLO* yang menekankan efisiensi serta kecepatan inferensi. Dibandingkan dengan *Faster R-CNN*, algoritma *YOLO* secara umum lebih sesuai untuk aplikasi *real-time* karena mampu memberikan kecepatan proses yang jauh lebih tinggi (Sharma, Kumar, & Longchamps, 2024). Generasi *YOLOv5* dan *YOLOv8* memang populer, tetapi arsitekturnya masih memiliki keterbatasan dalam efisiensi dan akurasi, sehingga perkembangan terbaru yaitu *YOLOv11* dinilai lebih unggul.

YOLOv11 merupakan pengembangan terbaru dari keluarga *YOLO* yang dirilis pada tahun 2024. Algoritma ini menghadirkan sejumlah penyempurnaan arsitektural dibandingkan pendahulunya, termasuk penggunaan pendekatan *anchor-free detection*, blok *C3k2*, serta penerapan *C2PSA (Cross Stage Partial Spatial Attention)*. Peningkatan ini membuat *YOLOv11* mampu menggabungkan efisiensi dengan ketelitian dalam mendeteksi objek, baik pada citra sederhana maupun citra dengan tingkat kompleksitas tinggi (Bochkovskiy et al., 2020). Studi

perbandingan terbaru juga menunjukkan bahwa *YOLOv11* memiliki keunggulan signifikan dibanding *YOLOv8*, *YOLOv9*, dan *YOLOv10* dalam hal keseimbangan akurasi dan kecepatan pada berbagai tugas deteksi (Sharma et al., 2024;Sapkota et al., 2025).

Di antara empat varian utama *YOLOv11*, yaitu *YOLOv11n (nano)*, *YOLOv11s (small)*, *YOLOv11m (medium)*, dan *YOLOv11x (extra-large)*, penelitian ini memilih *YOLOv11s*. Varian ini dipandang paling seimbang karena memiliki jumlah parameter yang relatif kecil sehingga dapat diproses lebih cepat di lingkungan *Google Colab* dengan GPU terbatas, tanpa mengorbankan akurasi secara signifikan. *YOLOv11s* memiliki ukuran file yang ringan (~9.4 MB), *inference time* yang cepat, serta nilai *mAP* yang lebih tinggi dibandingkan *YOLOv11n*. Berdasarkan laporan resmi Ultralytics serta penelitian terkini, *YOLOv11s* mampu mencapai *mAP* sekitar 47% pada dataset COCO dengan waktu inferensi singkat, serta menunjukkan kinerja yang kompetitif dalam mendeteksi objek di lingkungan pertanian kompleks (Sapkota et al., 2025). Dengan karakteristik ini, *YOLOv11s* menjadi kandidat ideal untuk penelitian yang membutuhkan pelatihan berulang pada dataset dengan ukuran kecil hingga menengah.

Pada platform *Roboflow*, proses pemilihan model dilakukan dengan memilih format ekspor ke *YOLOv11*. Gambar 3.10 menunjukkan antarmuka *Roboflow* pada saat pemilihan model *YOLOv11* yang menjadi dasar pelatihan.



Gambar 3.10 Antarmuka pemilihan model YOLOv11 di Roboflow sebagai bagian dari konfigurasi ekspor dataset.

Untuk memperkuat justifikasi pemilihan model, berikut ditampilkan perbandingan performa empat varian *YOLOv11* (*n*, *s*, *m*, *x*) dalam bentuk tabel berdasarkan dokumentasi resmi *Ultralytics*:

Tabel 3.3 Tabel perbandingan performa dan spesifikasi empat varian YOLOv11

Varian Model	Ukuran Model (MB)	mAP val 50-95	Waktu Inferensi (ms)	Keunggulan
YOLOv11n	~4.1 MB	39.5	Sangat cepat	Cocok untuk perangkat edge, kecepatan tinggi namun akurasi lebih rendah.
YOLOv11s	~9.4 MB	47.0	Cepat	Kompromi terbaik antara kecepatan dan akurasi, cocok untuk cloud training.
YOLOv11m	~22 MB	51.5	Sedang	Akurasi lebih tinggi, tapi membutuhkan komputasi lebih besar.
YOLOv11x	~54 MB	54.7	Lambat	Akurasi terbaik, namun memerlukan resource sangat besar.

Seperti yang ditampilkan pada Tabel 3.3, *YOLOv11s* tetap menjadi pilihan yang rasional dalam konteks penelitian ini karena mampu memberikan keseimbangan yang baik antara akurasi dan efisiensi proses pelatihan. Selain pertimbangan teoretis dan hasil studi literatur, pemilihan *YOLOv11s* dalam penelitian ini juga didasarkan pada alasan pribadi peneliti yang berkaitan dengan konteks penelitian. Pertama, penggunaan *Google Colab* dengan GPU *T4* memiliki keterbatasan dalam daya komputasi, sehingga model ringan seperti *YOLOv11s* lebih

memungkinkan untuk dilatih berulang kali tanpa menghabiskan waktu terlalu lama. Kedua, penelitian ini mengharuskan pelatihan model pada lima variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) serta dua skenario berbeda, sehingga model yang efisien secara komputasi lebih tepat dibanding algoritma berat seperti *Faster R-CNN* atau varian besar *YOLOv11x*. Ketiga, dataset yang digunakan berjumlah 240 citra dengan ribuan anotasi, yang tergolong kecil–menengah, sehingga varian ringan seperti *YOLOv11s* lebih sesuai dibanding *YOLOv11m* atau *YOLOv11x* yang membutuhkan dataset jauh lebih besar agar optimal. Keempat, penggunaan *YOLOv11* sebagai arsitektur terbaru memastikan penelitian ini tetap relevan dengan perkembangan terkini dalam bidang deteksi objek, sehingga hasilnya memiliki nilai kebaruan baik secara akademik maupun praktis.

Dengan demikian, pemilihan *YOLOv11s* dalam penelitian ini dapat dipandang sebagai langkah strategis. Dari sisi teori, *YOLOv11* menghadirkan penyempurnaan arsitektural signifikan yang membuatnya unggul dibandingkan generasi sebelumnya dan lebih efisien dibandingkan algoritma dua tahap seperti *Faster R-CNN*. Dari sisi teknis, *YOLOv11s* memungkinkan penelitian tetap dapat dijalankan dengan sumber daya terbatas, namun tetap menghasilkan keluaran yang representatif. Dari sisi praktis, penggunaan arsitektur terkini seperti *YOLOv11* memastikan penelitian ini relevan, mutakhir, dan memiliki kontribusi akademik yang signifikan dalam kajian pengaruh variasi tingkat anotasi terhadap performa model deteksi objek.

3.8 Teknik Pelatihan Model

Proses pelatihan model YOLOv11 dilakukan untuk mempelajari pola visual dari dataset buah yang telah disiapkan. Model dilatih menggunakan arsitektur *YOLOv11s*, yang merupakan varian ringan (*small*) dari keluarga YOLOv11, dipilih karena keseimbangan antara kecepatan pelatihan, kebutuhan komputasi, dan tingkat akurasi yang memadai untuk penelitian ini.

Pelatihan dilakukan pada platform *Google Colab* dengan dukungan *GPU (Graphics Processing Unit) Nvidia TESLA T4*. Pelatihan menggunakan *Ultralytics YOLO framework* dengan skrip utama dalam format *notebook* (.ipynb). Model dilatih pada setiap variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) untuk dibandingkan performanya.

3.8.1 Hyperparameter Pelatihan

Hyperparameter yang digunakan dirancang agar konsisten pada semua eksperimen untuk memastikan perbandingan antar model bersifat adil (*fair comparison*). Parameter yang digunakan adalah:

- **Epoch:** 100 (jumlah iterasi penuh terhadap seluruh dataset pelatihan).
- **Image Size (imgsz):** 640×640 piksel.
- **Learning Rate (lr0):** 0.001.
- **Optimizer:** *Stochastic Gradient Descent (SGD)*.
- **Plots:** True untuk menghasilkan grafik hasil pelatihan secara otomatis (*results.png*).

Perintah pelatihan ditunjukkan pada Gambar 3.11 berikut:

```
%cd {HOME}

!yolo task=detect mode=train model=yolo11s.pt data={dataset.location}/data.yaml
epochs=50 imgsz=640 plots=True
```

Gambar 3.11 Perintah pelatihan YOLOv11 dengan *hyperparameter* penelitian.

3.8.2 Output Pelatihan

Proses pelatihan menghasilkan beberapa keluaran utama yang digunakan sebagai dasar analisis pada tahap evaluasi, yaitu:

1. **Model hasil pelatihan**: berkas *best.pt* (model terbaik berdasarkan performa pada dataset validasi) dan *last.pt* (model terakhir pada akhir *epoch*).
2. **Grafik hasil pelatihan (*results.png*)**: menampilkan tren *loss*, *precision*, *recall*, dan *mAP* pada setiap *epoch*, sehingga dapat menggambarkan stabilitas dan pola konvergensi model selama pelatihan.
3. **Confusion matrix**: menyajikan distribusi prediksi model terhadap kelas buah pada dataset validasi, yang berguna untuk mengevaluasi tingkat kesalahan klasifikasi antar kelas.
4. **File CSV (*results.csv*)**: berisi catatan log pada setiap *epoch* yang mencakup metrik evaluasi seperti *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*. Selain itu, berkas ini juga memuat informasi total durasi waktu pelatihan (*training time*) untuk setiap model.

Seluruh keluaran ini dimanfaatkan dalam Bab IV untuk melakukan analisis performa model, perbandingan antar tingkat anotasi, serta visualisasi hasil pengujian. Secara khusus, data *training time* dari *results.csv* digunakan untuk mengevaluasi aspek efisiensi komputasi.

Pengukuran dilakukan berdasarkan total durasi pelatihan pada 100 *epoch* untuk masing-masing model, dengan lingkungan komputasi yang seragam menggunakan Google Colab dan GPU *NVIDIA Tesla T4*. Dengan pendekatan ini, penelitian dapat menilai seberapa besar pengurangan tingkat anotasi berpengaruh terhadap percepatan proses pelatihan, sekaligus menyeimbangkan antara akurasi deteksi dan efisiensi komputasi.

3.9 Teknik Analisis Data

Teknik analisis data dalam penelitian ini bertujuan untuk mengevaluasi pengaruh pengurangan tingkat anotasi terhadap performa model deteksi objek berbasis YOLOv11. Analisis dilakukan secara menyeluruh dengan pendekatan kuantitatif dan kualitatif terhadap lima varian model berdasarkan tingkat anotasi yang berbeda (100%, 90%, 80%, 70%, dan 60%). Tujuan utama dari analisis ini adalah untuk mengidentifikasi sejauh mana pengurangan jumlah anotasi memengaruhi kualitas prediksi model, menilai efisiensi komputasi, serta menentukan ambang batas minimum anotasi yang masih dapat menghasilkan performa yang dapat diterima.

Evaluasi kuantitatif dilakukan menggunakan empat metrik utama yang umum dalam sistem deteksi objek, yaitu *precision*, *recall*, mean Average Precision (mAP) pada ambang Intersection over Union (IoU) 0.5 (ditulis mAP@0.5), serta mAP pada rentang ambang 0.5–0.95 (mAP@0.5:0.95). Metrik-metrik ini didefinisikan sebagai berikut:

- **Precision** mengukur ketepatan deteksi model, yaitu rasio antara prediksi benar (*true positive*) dibandingkan dengan seluruh prediksi objek.
- **Recall** mengukur sensitivitas model dalam menemukan semua objek yang ada, yaitu rasio antara prediksi benar dibandingkan dengan seluruh *ground truth*.
- **mAP@0.5** menjadi indikator utama untuk mengukur akurasi spasial deteksi objek berdasarkan ambang IoU sebesar 0.5.
- **mAP@0.5:0.95** memberikan evaluasi yang lebih ketat karena menghitung rata-rata akurasi pada ambang IoU bertingkat dari 0.5 hingga 0.95.

Selain keempat metrik utama tersebut, dilakukan juga evaluasi terhadap **confusion matrix**, yaitu matriks yang menggambarkan kesalahan dan keberhasilan prediksi pada setiap kelas buah. Matriks ini membantu mengidentifikasi jenis kesalahan (*false positive*, *false negative*) dan mengukur distribusi performa model pada masing-masing kelas (apel, jeruk, stroberi, anggur, rambutan, dan salak). Dengan ini, dapat dianalisis kelas buah mana yang paling terdampak oleh pengurangan anotasi.

Penelitian ini juga menganalisis aspek efisiensi komputasi melalui pencatatan waktu pelatihan (*training time*). Informasi ini diperoleh dari berkas *results.csv* yang dihasilkan framework Ultralytics, yang mencatat total durasi pelatihan untuk 100 *epoch* pada setiap model. Perbandingan dilakukan antar tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) pada kedua skenario (A dan B) untuk mengetahui seberapa besar pengurangan anotasi berdampak pada percepatan waktu pelatihan. Dengan demikian, penelitian ini dapat menilai trade-off antara akurasi deteksi dan efisiensi komputasi secara lebih komprehensif.

Data analisis diperoleh dari dua keluaran utama model YOLOv11, yaitu *results.csv* dan *results.png* yang dihasilkan otomatis oleh framework Ultralytics. File *results.csv* berisi nilai numerik setiap *epoch* pelatihan seperti *box loss*, *class loss*, *DFL loss*, *precision*, *recall*, *mAP*, serta total waktu pelatihan. Sementara itu, *results.png* menampilkan grafik visualisasi tren pelatihan model untuk dataset pelatihan dan validasi.

Untuk melengkapi evaluasi numerik, analisis kualitatif dilakukan melalui visualisasi hasil deteksi model terhadap dataset uji menggunakan aplikasi *Streamlit* yang dikembangkan khusus. Aplikasi ini memuat informasi berupa jumlah objek yang berhasil dideteksi, *confidence score* dari setiap *bounding box*, serta rata-rata tingkat kepercayaan model terhadap deteksinya. Analisis ini memberikan gambaran nyata mengenai perilaku model saat mendeteksi objek dalam skenario nyata.

Penelitian ini juga membandingkan dua strategi pengurangan anotasi (Skenario A dan B), masing-masing mewakili pendekatan *annotator* yang berbeda: Skenario A menghapus label objek dominan terlebih dahulu, sedangkan Skenario B menghapus label objek tersembunyi atau tertumpuk terlebih dahulu. Dengan demikian, teknik analisis data tidak hanya membandingkan performa antar tingkat anotasi, tetapi juga mengukur bagaimana strategi anotasi memengaruhi akurasi, generalisasi model, serta efisiensi waktu komputasi.

Seluruh hasil analisis kemudian direkap dalam bentuk tabel komparatif, grafik performa tiap metrik, grafik waktu pelatihan, serta visualisasi confusion matrix antar model. Kombinasi pendekatan numerik dan visual ini diharapkan mampu memberikan gambaran menyeluruh mengenai dampak pengurangan anotasi terhadap kualitas deteksi objek dan efisiensi pelabelan, serta menjadi dasar untuk penentuan ambang batas anotasi yang paling efektif dan efisien.

3.10 Penyimpanan Model

Setelah proses pelatihan selesai, model yang sudah terlatih disimpan agar dapat digunakan pada tahap evaluasi, pengujian, dan implementasi. Penyimpanan ini memungkinkan peneliti

untuk menghindari proses pelatihan ulang, yang memerlukan waktu dan sumber daya komputasi yang besar. Model yang tersimpan biasanya berupa file dengan ekstensi .pt, seperti *best.pt* (model dengan performa terbaik berdasarkan validasi) atau *last.pt* (model pada epoch terakhir). File ini berisi bobot (*weights*), arsitektur, serta parameter yang telah dipelajari selama pelatihan, sehingga dapat langsung digunakan untuk proses inferensi. Penyimpanan model ini bertujuan untuk memungkinkan penggunaan model yang sudah dilatih untuk evaluasi lebih lanjut dan pengujian menggunakan data yang belum pernah dilihat sebelumnya.

```
!ls {HOME}/runs/detect/train/
```

Gambar 3.12 Kode program untuk penyimpanan model

Perintah ini akan menampilkan daftar file dan direktori di dalam folder {HOME}/runs/detect/train/. Folder tersebut biasanya berisi file bobot model, *log*, hasil evaluasi (*results.png*), serta *confusion_matrix.png*. Dengan cara ini, peneliti dapat dengan mudah memverifikasi file hasil pelatihan.

3.11 Evaluasi Model

Evaluasi model dilakukan untuk menilai performa YOLOv11 dalam mendeteksi objek buah pada setiap variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%). Tujuan utama dari evaluasi ini adalah untuk mengukur akurasi, ketepatan, efisiensi komputasi, serta kemampuan generalisasi model terhadap data uji, sekaligus mengevaluasi dampak dari strategi pengurangan anotasi pada dua skenario berbeda (Skenario A dan Skenario B). Evaluasi dilakukan dengan dua pendekatan utama, yaitu kuantitatif numerik melalui metrik evaluasi dan *training time*, serta visualisasi performa pelatihan melalui grafik, matriks, dan hasil deteksi.

3.11.1 Metrik Evaluasi Kuantitatif

Beberapa metrik evaluasi yang digunakan meliputi:

1. **Confusion Matrix**: *Confusion matrix* digunakan untuk mengevaluasi kinerja klasifikasi model dengan membandingkan hasil prediksi terhadap label sebenarnya. Matriks ini menampilkan jumlah prediksi yang benar (*true positive*) maupun salah (*false positive* dan *false negative*) pada setiap kelas buah (jeruk, apel, stroberi, anggur, rambutan, dan salak). Dengan melihat confusion matrix, dapat diketahui kelas mana yang sering terdeteksi dengan benar atau yang mengalami kesalahan prediksi.

2. **Training Time**: selain metrik akurasi, penelitian ini juga memasukkan *training time* sebagai salah satu indikator evaluasi. Pengukuran dilakukan berdasarkan total durasi pelatihan untuk 100 *epoch* dari setiap model, sebagaimana dicatat pada *results.csv*. Dengan memasukkan *training time*, evaluasi model tidak hanya menilai kualitas deteksi, tetapi juga menilai efisiensi komputasi yang dicapai melalui pengurangan anotasi.
3. **Combination Chart**: Grafik ini menggambarkan perkembangan metrik pelatihan secara simultan, meliputi:
    a. **Box Loss, CLS Loss, dan DFL Loss** untuk data *training* dan *validation*, yang menunjukkan konvergensi model selama proses pelatihan.
    b. **Precision, Recall, mAP@0.5, dan mAP@0.5–0.95**, yang menjadi indikator utama performa deteksi objek. Precision menunjukkan ketepatan deteksi, recall menggambarkan kelengkapan deteksi, sedangkan mAP mengukur akurasi rata-rata model dalam mendeteksi objek pada berbagai nilai *IoU threshold*.

```
from IPython.display import Image as IPyImage
IPyImage(filename=f'{HOME}/runs/detect/train/confusion_matrix.png',
width=600)
```

Gambar 3.13 Kode menampilkan *confusion matrix* pada Google Colab

```
from IPython.display import Image as IPyImage
IPyImage(filename=f'{HOME}/runs/detect/train/results.png', width=600)
```

Gambar 3.14 Kode menampilkan *chart* hasil *training model*

Kode pada Gambar 3.13 dan Gambar 3.14 ini hampir sama, digunakan untuk menampilkan gambar *confusion matrix* dan *combination chart* dalam *IPython*. Fungsi `IPyImage` diimpor dari `IPython.display` untuk menampilkan gambar. Gambar yang ditampilkan berasal dari file `confusion_matrix.png` dan `results.png` yang terletak di `{HOME}/runs/detect/train/` dan `{HOME}/runs/detect/train/`, dengan lebar gambar yang disesuaikan menjadi 600 piksel. `{HOME}` merujuk ke direktori home pengguna, sehingga path ini dapat dijalankan di berbagai sistem.

3.11.2 Prinsip Penentuan Ambang Batas (Threshold)

Dalam penelitian ini, ditetapkan ambang batas toleransi penurunan performa sebesar 10% dari nilai *baseline mAP@0.5* model dengan tingkat anotasi 100%. Dengan kata lain, model

dengan anotasi terbatas masih dianggap layak apabila nilai *mAP* tidak mengalami penurunan lebih dari 10% dibandingkan *baseline*.

Secara matematis, prinsip ini dirumuskan sebagai:

$mAP_p \geq (1 - 0{,}1) \times mAP_{100\%}$	(3.3)

Dengan:

- mAP_p : nilai mAP@0.5 dari model dengan tingkat anotasi $_p$ %
- $mAP_{100\%}$: nilai mAP@0.5 dari model baseline dengan anotasi lengkap (100%)

Ambang batas ini ditetapkan untuk menyeimbangkan kebutuhan akurasi dengan efisiensi anotasi, serta didasarkan pada pertimbangan praktis bahwa pengurangan anotasi sering kali dilakukan untuk menghemat waktu dan biaya dalam proses *data labeling*. Evaluasi ini membantu dalam menentukan titik optimal efisiensi pelabelan, yakni tingkat anotasi minimum yang tidak mengorbankan performa model secara signifikan.

3.11.3 Relevansi Evaluasi terhadap Strategi Pengurangan Anotasi

Evaluasi ini juga digunakan untuk membandingkan pengaruh dari dua strategi pengurangan anotasi:

- **Skenario A**, yaitu penghapusan label pada objek jelas dan dominan terlebih dahulu. Strategi ini cenderung menurunkan performa karena menghilangkan objek-objek yang paling mudah dikenali model. Dengan demikian, meskipun jumlah anotasi berkurang, informasi yang hilang berdampak signifikan terhadap akurasi deteksi.
- **Skenario B**, yaitu penghapusan label pada objek tersembunyi, bertumpuk, atau tertutup sebagian. Strategi ini lebih realistis karena dalam praktik anotasi manual, anotator sering melewatkan objek-objek yang sulit terlihat. Dampaknya, performa model masih relatif terjaga untuk objek mayoritas, meskipun *recall* dapat menurun akibat hilangnya anotasi pada objek-objek minor.

Perbandingan kedua skenario ini merepresentasikan dua pendekatan annotator yang berbeda: annotator yang fokus hanya pada objek dominan (Skenario A) dan annotator yang lebih ketat menandai objek yang terlihat jelas tetapi mengabaikan objek tumpang tindih (Skenario B). Dengan demikian, hasil evaluasi diharapkan memberikan wawasan praktis mengenai strategi anotasi yang lebih efisien tanpa mengorbankan performa secara signifikan.

3.11.4 Rangkuman Evaluasi

Seluruh hasil evaluasi, baik berupa angka (*precision*, *recall*, *mAP*, dan *training time*), matriks (*confusion matrix*), maupun deteksi visual (jumlah objek terdeteksi dan *confidence score*), disajikan secara lengkap pada Bab IV Hasil dan Pembahasan. Data ini kemudian dianalisis lebih lanjut pada Subbab 4.3 untuk mengidentifikasi pola penurunan performa, perbedaan antar kelas, serta ambang batas efektif anotasi.

Selain hasil numerik dan matriks, evaluasi ini juga memanfaatkan visualisasi berbasis *Streamlit* untuk memperlihatkan secara intuitif perbedaan hasil deteksi antar model. Aplikasi ini berfungsi sebagai pendukung analisis kualitatif, sehingga pembaca dapat melihat secara langsung implikasi dari pengurangan anotasi terhadap keluaran model, tanpa menggantikan analisis berbasis metrik kuantitatif.

3.12 Pengujian Model

Pengujian model dilakukan untuk mengevaluasi kemampuan generalisasi model deteksi objek YOLOv11 terhadap data baru yang tidak pernah dilihat selama pelatihan. Proses ini menggunakan *test set* yang telah dipisahkan sejak tahap *dataset splitting*, sehingga hasil evaluasi benar-benar mencerminkan performa model di dunia nyata. Tujuan pengujian ini adalah untuk mengukur seberapa baik model dapat mendeteksi enam kelas buah (apel, jeruk, stroberi, anggur, rambutan, dan salak) pada data uji, sekaligus memastikan tidak terjadi *overfitting* terhadap data pelatihan.

Evaluasi model dilakukan dengan menjalankan perintah *YOLO* pada mode *validation* (val), yang menggunakan bobot model terbaik (*best.pt*) dari hasil pelatihan. Perintah tersebut dijalankan pada file konfigurasi *data.yaml*, yang memuat informasi tentang kelas objek, struktur dataset, dan path *test set*.

```
from IPython.display import Image as IPyImage
IPyImage(filename=f'{HOME}/runs/detect/train/results.png', !yolo
task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt
data={dataset.location}/data.yaml
```

Gambar 3.15 Kode untuk melakukan *testing* model

Penjelasan perintah pada Gambar 3.15 adalah:

1. **task=detect**: menentukan jenis tugas yaitu deteksi objek.
2. **mode=val**: mengatur mode evaluasi menggunakan data validasi atau pengujian.

3. **model=.../best.pt**: memilih bobot model terbaik.
4. **data=.../data.yaml**: memuat konfigurasi dataset.

Hasil pengujian menghasilkan metrik evaluasi seperti *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*, yang akan dianalisis di Bab 4.

3.13 Analisis Hasil

Analisis hasil merupakan tahap krusial dalam penelitian ini untuk mengevaluasi dampak pengurangan tingkat anotasi terhadap performa model deteksi objek YOLOv11. Setelah seluruh model dilatih dan diuji, dilakukan analisis mendalam terhadap data evaluasi guna memahami pola perubahan performa, baik dari aspek kuantitatif maupun kualitatif. Proses ini bertujuan untuk mengidentifikasi tren penurunan atau stabilitas performa model sebagai akibat dari berkurangnya jumlah anotasi, sekaligus memberikan justifikasi terhadap ambang batas anotasi yang masih tergolong layak secara performa dan efisiensi komputasi.

3.13.1 Fokus Analisis

Fokus utama analisis ini mencakup empat aspek:

1. Evaluasi numerik terhadap metrik performa seperti *precision*, *recall*, *mAP@0.5*, dan *mAP@0.5–0.95* pada setiap variasi tingkat anotasi.
2. Evaluasi visual, yang memanfaatkan *confusion matrix* dan grafik performa untuk melihat perubahan kualitas deteksi antar kelas buah.
3. Analisis hasil inferensi, berupa jumlah deteksi dan rata-rata *confidence* dari prediksi model terhadap dataset uji, yang divisualisasikan melalui aplikasi *Streamlit*.
4. Evaluasi efisiensi komputasi, yang dilakukan dengan membandingkan waktu pelatihan (*training time*) antar model pada berbagai tingkat anotasi dan kedua skenario, untuk melihat seberapa besar pengurangan anotasi dapat mempercepat proses pelatihan.

3.13.2 Metrik Evaluasi yang Digunakan

Analisis dilakukan menggunakan beberapa metrik kunci:

- **Precision dan Recall**: digunakan untuk mengevaluasi ketepatan dan kelengkapan prediksi model.
- **mAP@0.5 dan mAP@0.5–0.95**: digunakan untuk mengukur akurasi deteksi berdasarkan kesesuaian *bounding box* model terhadap *ground truth*.

- **Confusion Matrix**: menyajikan distribusi prediksi benar (*True Positives*) dan salah (*False Positives / False Negatives*) pada masing-masing kelas buah. Matriks ini membantu mengidentifikasi kelas yang paling terdampak akibat pengurangan anotasi.
- **Training Time**: yaitu total durasi pelatihan untuk 100 *epoch* yang dicatat pada *results.csv*. Nilai ini digunakan sebagai indikator efisiensi komputasi dan dibandingkan antar model untuk menilai pengaruh pengurangan anotasi terhadap kecepatan pelatihan.

Nilai-nilai metrik tersebut diperoleh dari hasil pelatihan dan pengujian model, lalu dianalisis untuk setiap variasi tingkat anotasi: 100% (baseline), 90%, 80%, 70%, dan 60%.

3.13.3 Visualisasi Hasil Evaluasi

Untuk mempermudah interpretasi dan penyajian data, hasil evaluasi divisualisasikan dalam bentuk:

- **Confusion Matrix**: menunjukkan kelas buah mana yang mengalami peningkatan atau penurunan akurasi seiring dengan pengurangan anotasi, serta membantu mengidentifikasi kelas yang paling terdampak (misalnya objek kecil atau bertumpuk).
- **Grafik Tren Performa**: menampilkan perubahan nilai *precision*, *recall*, dan *mAP* pada setiap model. Grafik ini memperlihatkan tren penurunan performa dan mendukung penentuan threshold anotasi.
- **Grafik Waktu Pelatihan**: memperlihatkan perbandingan total durasi pelatihan antar model, sehingga dapat mengungkap trade-off antara kualitas deteksi dan efisiensi komputasi.
- **Analisis Jumlah Deteksi dan Confidence Rate**: divisualisasikan melalui aplikasi *Streamlit*, yang menunjukkan seberapa banyak objek yang berhasil dideteksi serta seberapa besar tingkat kepercayaan model terhadap hasil prediksinya.

Visualisasi ini tidak hanya mendukung pemahaman pembaca, tetapi juga berfungsi sebagai bukti visual terhadap validitas temuan penelitian.

3.13.4 Integrasi Temuan ke Pembahasan

Temuan-temuan dari analisis ini akan dibahas secara menyeluruh pada Bab IV. Subbab khusus yang disiapkan meliputi:

- Analisis penurunan performa model,
- Identifikasi kelas buah yang paling terdampak,

- Penentuan ambang batas anotasi efektif,
- Analisis efisiensi komputasi melalui *training time*.

Integrasi ini memastikan bahwa pembahasan tidak hanya berfokus pada aspek akurasi, tetapi juga mencakup aspek efisiensi pelatihan, sehingga hasil penelitian dapat memberikan rekomendasi yang lebih relevan bagi pengembangan model deteksi objek dalam kondisi keterbatasan sumber daya anotasi maupun komputasi.

3.14 Proof of Concept

Proof of Concept (PoC) dalam penelitian ini direalisasikan melalui aplikasi berbasis *web interface* menggunakan Streamlit. Aplikasi ini digunakan untuk membandingkan kinerja lima model YOLOv11 (dengan tingkat anotasi 60%, 70%, 80%, 90%, dan 100%) pada gambar uji yang sama. PoC bertujuan untuk memberikan demonstrasi nyata (*real-world testing*) terhadap performa model, sekaligus membantu analisis visual hasil deteksi.

Aplikasi ini menampilkan hasil deteksi dalam bentuk *bounding box*, jumlah objek yang terdeteksi, serta rata-rata nilai *confidence* dari setiap model. Berikut ini adalah uraian detail mengenai komponen utama dari kode aplikasi.

3.14.1 Inisialisasi dan Import Library

Bagian awal kode memuat *import* library penting, yaitu:

```
import streamlit as st
from ultralytics import YOLO
from PIL import Image
import numpy as np
import cv2
```

Gambar 3.16 Import *Library* yang Diperlukan Sistem

- **Streamlit** digunakan untuk membangun antarmuka web interaktif.
- **Ultralytics YOLO** menyediakan *API* untuk memuat model YOLOv11 dan melakukan deteksi objek.
- **PIL (Python Imaging Library)** digunakan untuk memuat dan memproses gambar input.
- **NumPy dan OpenCV (cv2)** dipakai untuk manipulasi array gambar dan konversi format warna (BGR ke RGB).

3.14.2 Mapping Model YOLOv11

Kode berikut mendefinisikan *dictionary* model_files, yang memetakan nama model dengan tingkat anotasi berbeda ke lokasi file *weights* masing-masing:

```
model_files = {
    "YOLOv11_60": "models/Skripsi/best_60.pt",
    "YOLOv11_70": "models/Skripsi/best_70.pt",
    "YOLOv11_80": "models/Skripsi/best_80.pt",
    "YOLOv11_90": "models/Skripsi/best_90.pt",
    "YOLOv11_100": "models/Skripsi/best_100.pt"
}
```

Gambar 3.17 Mapping Model yang Digunakan pada Pengujian di Aplikasi.

Hal ini memudahkan pemanggilan model dalam satu struktur, serta memastikan setiap model dapat diuji secara paralel.

3.14.3 Fungsi *Load Model*

Fungsi berikut bertugas memuat model secara efisien dengan *cache*:

```
@st.cache_resource
def load_model(model_path):
    try:
        model = YOLO(model_path)
        return model
    except Exception as e:
        st.error(f"Error loading model: {e}")
        return None
```

Gambar 3.18 Fungsi untuk *Load Model* YOLOv11.

Fungsi ini memanfaatkan @st.cache_resource agar model hanya dimuat sekali, sehingga mempercepat proses deteksi saat tombol ditekan berulang.

3.14.4 Upload Gambar dan Tampilan Awal

Aplikasi menyediakan antarmuka *file uploader*:

```
st.set_page_config(page_title="Perbandingan 4 Model YOLOv11", layout="wide")
st.title("Perbandingan 4 Model YOLOv11 Pada Gambar yang Sama")

uploaded_file = st.file_uploader("Upload gambar", type=["jpg", "jpeg", "png"])
if uploaded_file:
    image = Image.open(uploaded_file).convert("RGB")
    st.image(image, caption="Gambar yang Diupload", use_container_width=True)
```

Gambar 3.19 Tampilan Awal dan *Upload* Gambar oleh Pengguna

- Setelah gambar diunggah, gambar ditampilkan menggunakan st.image().
- Jika tidak ada gambar yang diunggah, pesan informasi st.info("Silakan upload gambar terlebih dahulu.") akan muncul.

3.14.5 Proses Deteksi dan Visualisasi

Saat pengguna menekan tombol "Deteksi dengan Semua Model", aplikasi akan:

1. Membuat kolom dinamis (st.columns) sebanyak jumlah model.
2. Melakukan deteksi untuk setiap model dengan model(image).
3. Memvisualisasikan hasil deteksi menggunakan results[0].plot().
4. Menampilkan jumlah objek yang terdeteksi dan confidence rata-rata dengan:

```
st.set_page_config(page_title="Perbandingan 4 Model YOLOv11", layout="wide")
st.title("Perbandingan 4 Model YOLOv11 Pada Gambar yang Sama")

uploaded_file = st.file_uploader("Upload gambar", type=["jpg", "jpeg", "png"])
if uploaded_file:
    image = Image.open(uploaded_file).convert("RGB")
    st.image(image, caption="Gambar yang Diupload", use_container_width=True)

results = model(image)
# Plot hasil deteksi (output BGR)
res_plotted = results[0].plot()
# Konversi BGR ke RGB sebelum ditampilkan di Streamlit
if isinstance(res_plotted, np.ndarray):
    res_plotted = cv2.cvtColor(res_plotted, cv2.COLOR_BGR2RGB)
st.image(res_plotted, caption=f"Hasil Deteksi {model_name}",
use_container_width=True)

# Analisis jumlah deteksi dan rata-rata confidence
if hasattr(results[0], "boxes") and results[0].boxes is not None and
len(results[0].boxes) > 0:
    confs = [box.conf.item() for box in results[0].boxes]
    avg_conf = float(np.mean(confs))
    st.success(f"Jumlah deteksi: {len(confs)}")
    st.success(f"Confidence rata-rata: {avg_conf:.2%}")
else:
    st.warning("Tidak ada deteksi pada gambar ini.")
```

Gambar 3.20 Proses Deteksi dengan Semua Model

3.14.6 Keunggulan PoC

- Mempermudah analisis perbandingan antar model dengan tampilan visual yang intuitif.
- Memberikan *feedback* langsung tentang jumlah deteksi dan tingkat kepercayaan.
- Bersifat *real-time* dan dapat dijalankan di server lokal maupun *cloud deployment*.

BAB IV
HASIL DAN PEMBAHASAN

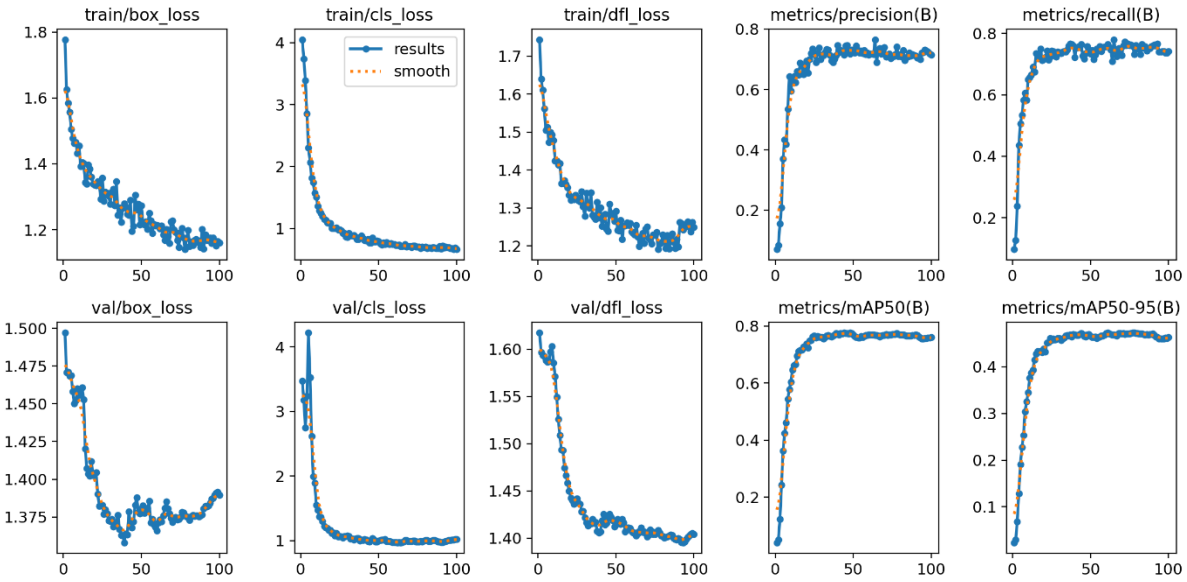
4.1 Hasil Pelatihan Model

Pelatihan model dilakukan terhadap sembilan variasi dataset berdasarkan dua skenario utama, yaitu Skenario A dan Skenario B, serta baseline 100% anotasi. Setiap variasi mewakili tingkat kelengkapan anotasi yang berbeda: 100%, 90%, 80%, 70%, dan 60%. Pelatihan dilakukan menggunakan arsitektur YOLOv11s selama 100 epoch dengan parameter yang dikontrol secara konstan untuk menjaga konsistensi eksperimen.

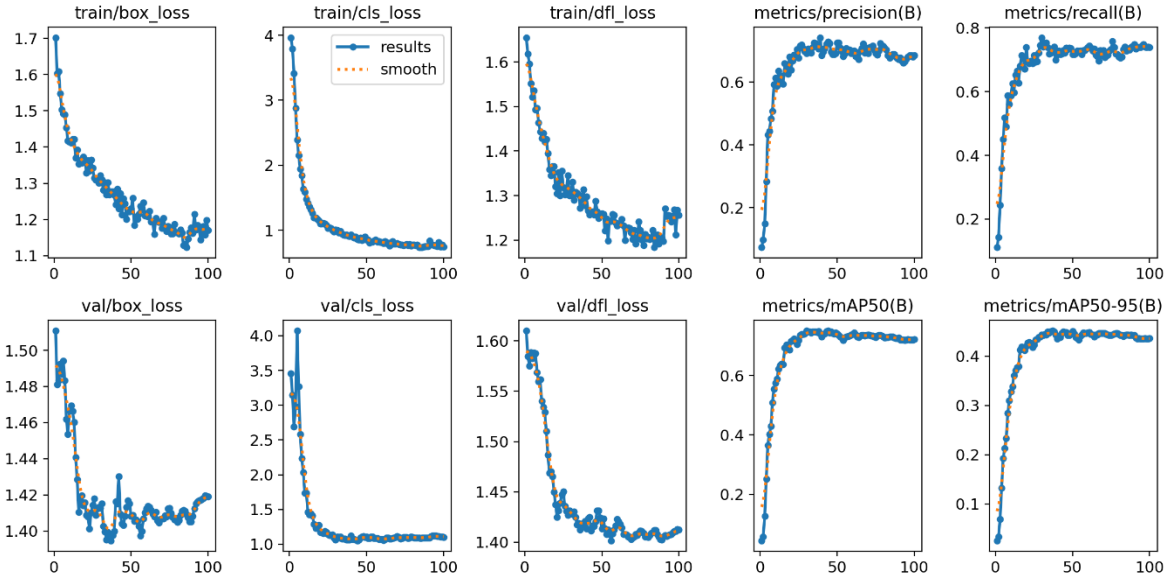
Seluruh proses pelatihan menghasilkan berkas log dalam format .csv serta visualisasi grafik *results.png* yang memuat tren box loss, classification loss, objectness loss, dan performa metrik evaluasi seperti *precision, recall, mAP@0.5,* dan *mAP@0.5:0.95*. Visualisasi tersebut digunakan sebagai dasar analisis performa tiap model.

4.1.1 Visualisasi Hasil Pelatihan

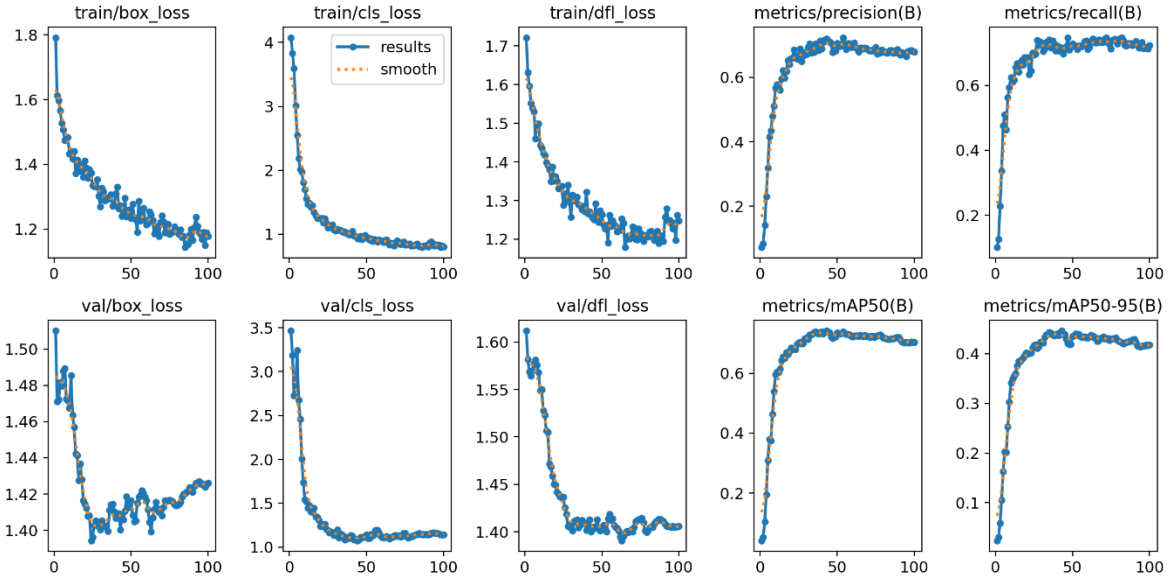
Berikut adalah hasil visualisasi pelatihan masing-masing model:



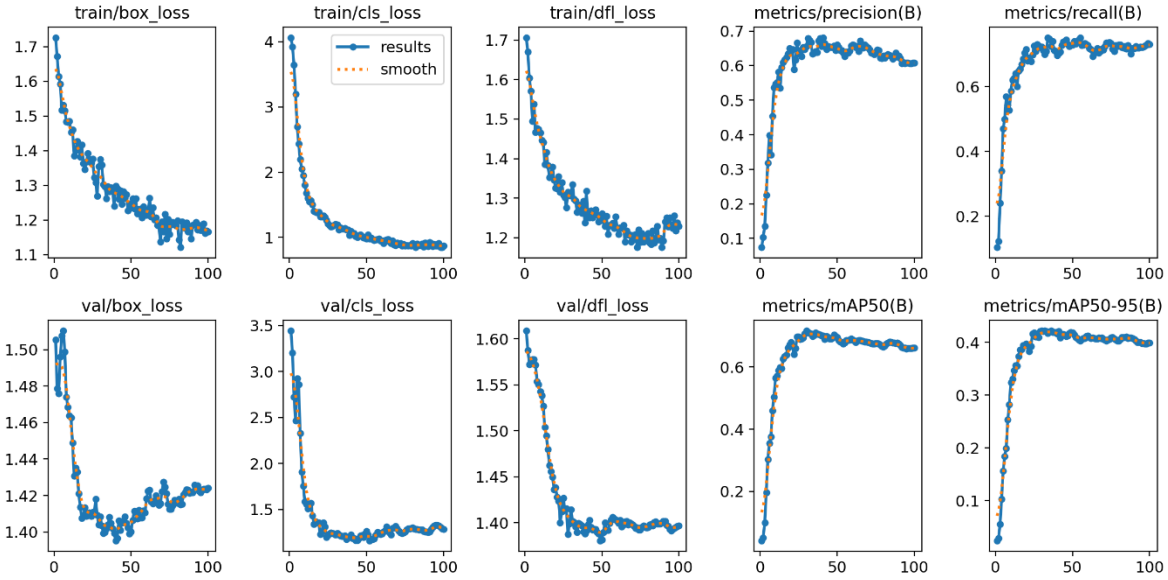
Gambar 4.1 Grafik hasil pelatihan model baseline 100% anotasi. Menunjukkan konvergensi loss yang stabil dan capaian metrik tertinggi.



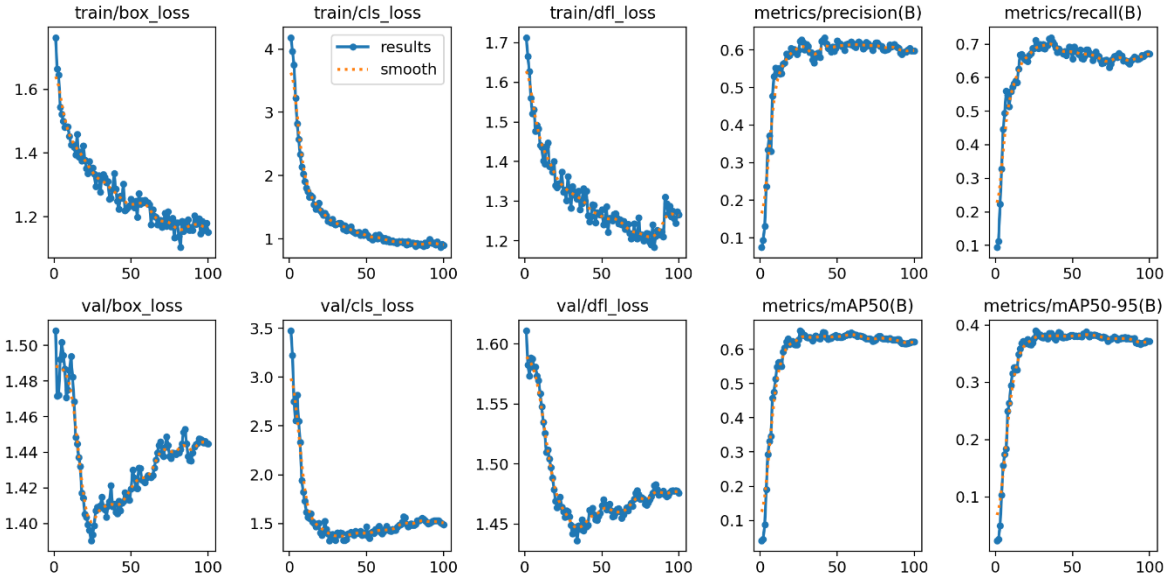
Gambar 4.2 Grafik hasil pelatihan model Skenario A - 90%. Masih menunjukkan performa mendekati baseline.



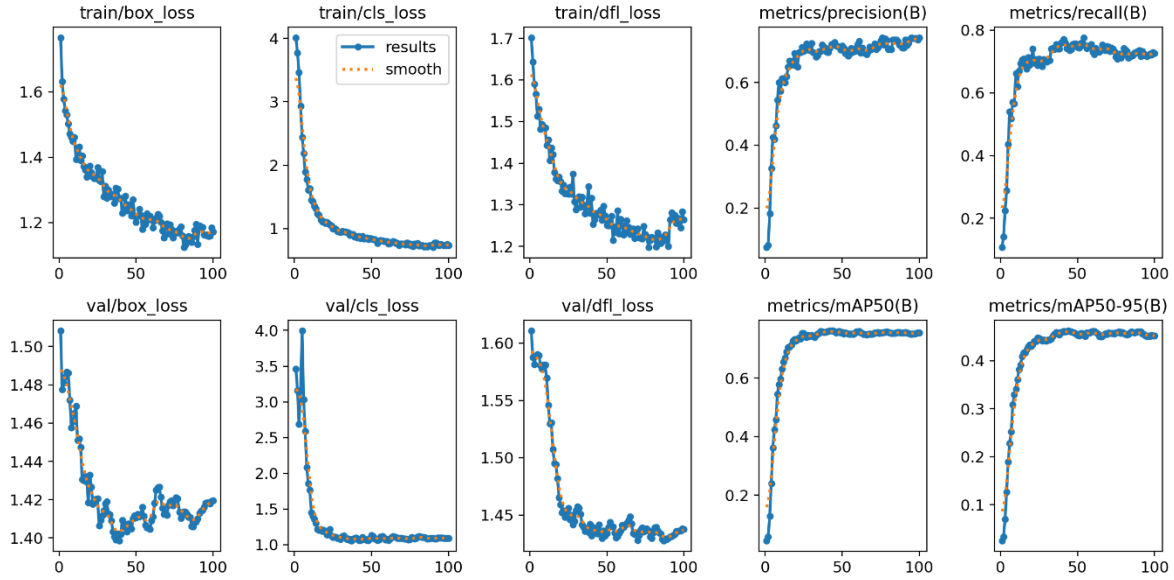
Gambar 4.3 Grafik hasil pelatihan model Skenario A - 80%. Terlihat penurunan metrik dan fluktuasi nilai loss mulai meningkat.



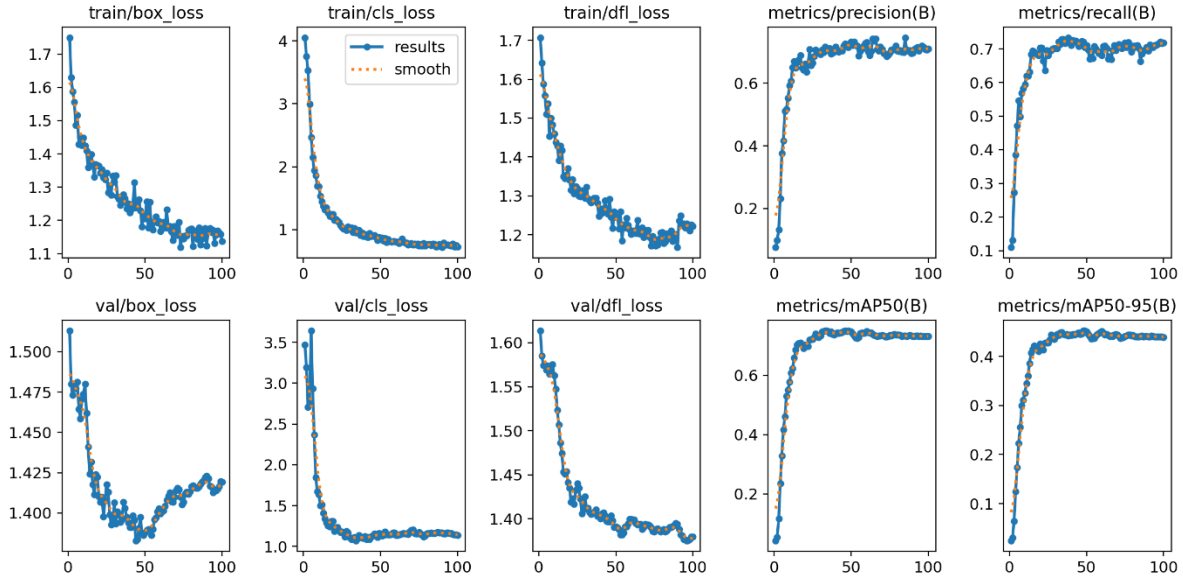
Gambar 4.4 Grafik hasil pelatihan model Skenario A - 70%. Penurunan performa mulai signifikan.



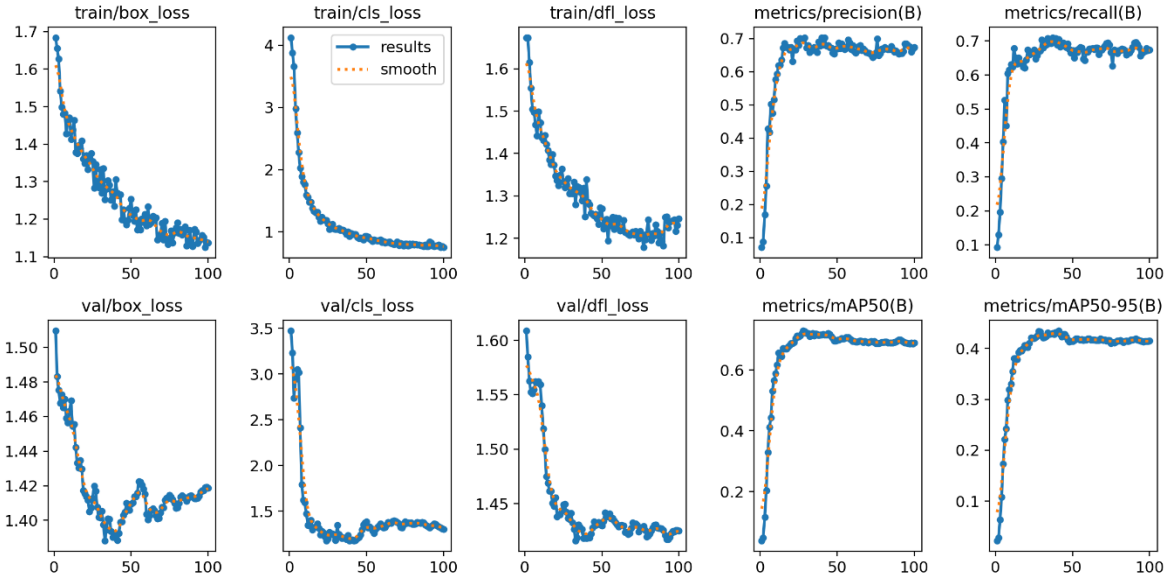
Gambar 4.5 Grafik hasil pelatihan model Skenario A - 60%. Terdapat penurunan tajam pada seluruh metrik evaluasi.



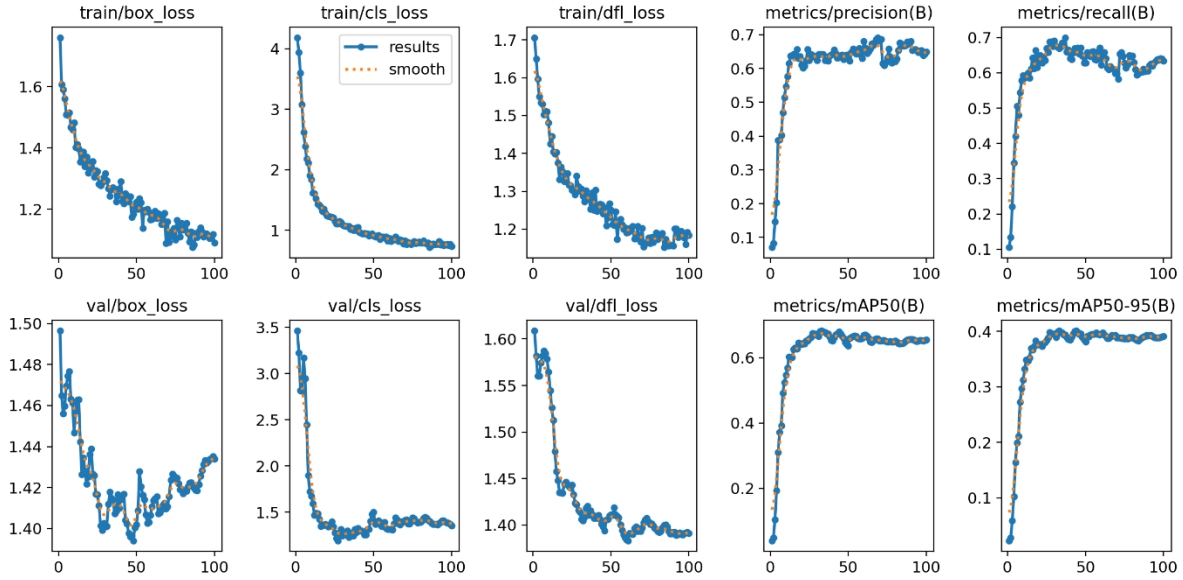
Gambar 4.6 Grafik hasil pelatihan model Skenario B - 90%. Menunjukkan metrik precision tertinggi dibandingkan seluruh model.



Gambar 4.7 Grafik hasil pelatihan model Skenario B - 80%. Performa masih stabil dan berada dalam ambang toleransi.



Gambar 4.8 Grafik hasil pelatihan model Skenario B - 70%. Metrik menurun secara bertahap tetapi belum drastis.



Gambar 4.9 Grafik hasil pelatihan model Skenario B - 60%. Penurunan mulai menyentuh batas toleransi 10%.

4.1.2 Tren Penurunan Performansi

Penurunan performa model deteksi objek sebagai akibat dari pengurangan tingkat anotasi tampak jelas pada tren metrik evaluasi yang dihasilkan. Empat metrik utama *Precision, Recall, mAP@0.5,* dan *mAP@0.5:0.95* digunakan untuk menggambarkan performa model secara kuantitatif. Namun, untuk menggali lebih dalam, penting untuk tidak hanya melihat perubahan

angka secara absolut, tetapi juga pola dan kestabilan tren antar skenario, serta interaksi antar metrik yang saling memengaruhi.

a) Analisis Precision dan Recall

Precision menunjukkan proporsi deteksi yang benar dari seluruh prediksi, sedangkan recall merefleksikan keberhasilan model dalam menemukan objek sebenarnya (*ground truth*). Pada Skenario A, *precision* menurun dari 0.685 (90%) ke 0.597 (60%), yang menandakan bahwa model semakin sering membuat prediksi yang salah (*false positive*) seiring pengurangan anotasi. Sebaliknya, *recall* tidak menurun secara linier, bahkan sempat naik pada 70% sebelum turun tajam di 60%. Hal ini dapat diartikan bahwa meskipun model masih mendeteksi sebagian besar objek, banyak di antaranya tidak akurat secara spasial atau keliru dalam klasifikasi.

Skenario B menunjukkan pola yang berbeda. *Precision* tetap tinggi pada 90% (0.743) dan hanya turun ke 0.649 di 60%, sementara *recall* juga menurun lebih halus. Ini menunjukkan bahwa pemilihan objek tertumpuk untuk dipertahankan (strategi Skenario B) memberikan informasi penting bagi model, karena objek-objek tersebut justru mewakili kompleksitas visual yang diperlukan untuk generalisasi.

b) mAP@0.5 dan mAP@0.5:0.95 sebagai Indikator Presisi Lokasi

mAP@0.5 dan *mAP@0.5:0.95* adalah metrik yang mempertimbangkan ketepatan posisi dan ukuran *bounding box* terhadap *ground truth*. *mAP@0.5* cenderung lebih toleran, sementara *mAP@0.5:0.95* jauh lebih ketat karena mencakup berbagai ambang IoU. Maka, ketika penurunan *mAP@0.5:0.95* lebih tajam dibandingkan *mAP@0.5*, hal ini menunjukkan model mengalami kesulitan dalam menempatkan *bounding box* secara presisi.

Hal ini terlihat jelas pada model Skenario A: *mAP@0.5* turun 13.86% dari 0.722 (90%) ke 0.622 (60%), sementara *mAP@0.5:0.95* turun lebih tajam, yaitu 14.65%. Sebaliknya, pada Skenario B, *mAP@0.5* hanya menurun 13.28% dari 0.754 ke 0.654 dan *mAP@0.5:0.95* hanya turun 13.68%, menunjukkan penurunan yang lebih terkontrol dan konsisten.

c) Dampak Strategi Anotasi terhadap Pembelajaran Model

Dari pola tren tersebut, dapat disimpulkan bahwa strategi penghapusan anotasi sangat berpengaruh terhadap pola pembelajaran model. Skenario A yang menghapus objek yang paling informatif terlebih dahulu (jelas dan utuh) menyebabkan model kehilangan representasi kontekstual yang penting, sehingga lebih cepat mengalami degradasi performa. Sebaliknya, Skenario B menjaga objek kompleks dalam pelatihan, yang ternyata memberi efek stabilisasi terhadap pembelajaran, meskipun jumlah anotasi juga dikurangi.

Temuan ini menjadi penting, karena menunjukkan bahwa bukan hanya jumlah anotasi, tetapi juga kualitas informasi spasial dan visual dari objek yang dipertahankan yang menentukan keberhasilan pelatihan model.

d) Identifikasi Ambang Efektif Penurunan

Dari tren yang diamati, performa model mulai melewati ambang toleransi penurunan 10% dibanding *baseline* (100%) ketika anotasi berkurang menjadi di bawah 70%. Khususnya pada Skenario A, penurunan ini terjadi lebih cepat dibandingkan Skenario B. Oleh karena itu, dapat disimpulkan bahwa ambang efektif pengurangan anotasi (*threshold* yang masih layak untuk operasional) berbeda antar strategi:

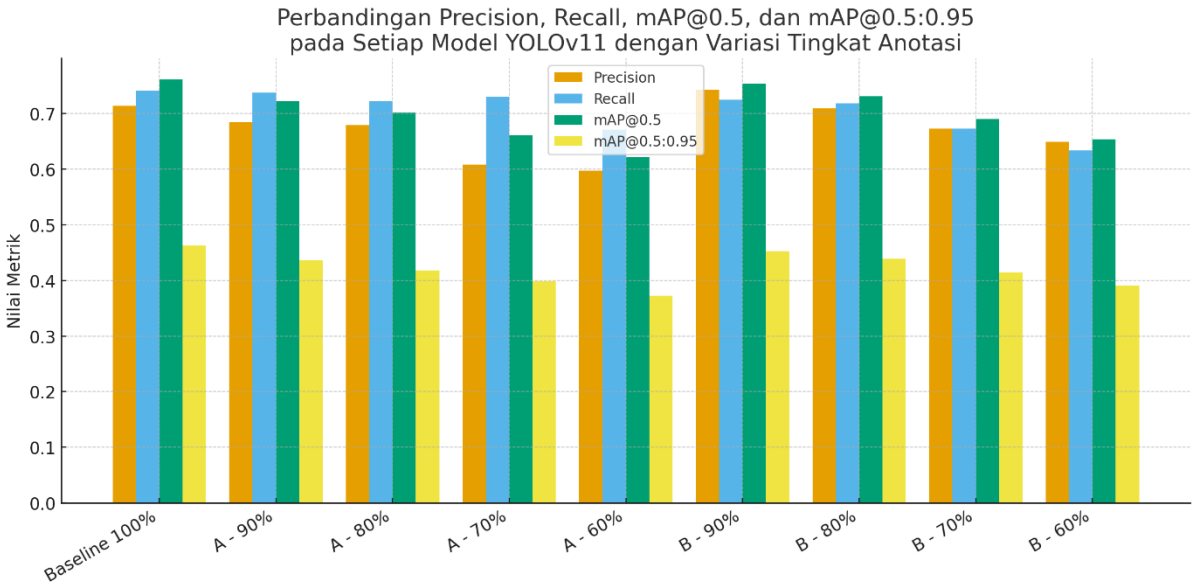
- Pada **Skenario A**, batas aman ada di 80%, karena pada 70% performa mAP@0.5 turun 13.15%.
- Pada **Skenario B**, batas aman bisa diperluas hingga 70%, karena penurunan mAP@0.5 masih dalam toleransi (9.3%).

4.1.3 Rekapitulasi Nilai Metrik Pelatihan

Tabel 4.1 merangkum nilai akhir dari empat metrik utama—*precision*, *recall*, *mAP@0.5*, dan *mAP@0.5:0.95*—untuk seluruh model pada berbagai tingkat anotasi dan skenario.

Tabel 4.1 Ringkasan Nilai *Precision, Recall, mAP@0.5,* dan *mAP@0.5:0.95* pada Setiap Tingkat Anotasi

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Baseline 100%	0.714	0.741	0.761	0.463
A - 90%	0.685	0.738	0.722	0.436
A - 80%	0.679	0.722	0.702	0.418
A - 70%	0.608	0.730	0.661	0.399
A - 60%	0.597	0.671	0.622	0.372
B - 90%	0.743	0.725	0.754	0.453
B - 80%	0.709	0.718	0.731	0.439
B - 70%	0.673	0.673	0.690	0.414
B - 60%	0.649	0.634	0.654	0.391



Gambar 4.10 Grafik Perbandingan Nilai *Precision*, *Recall*, *mAP@0.5*, dan *mAP@0.5:0.95* pada Setiap Model YOLOv11 dengan Variasi Tingkat Anotasi

Dari Tabel 4.1 dan Gambar 4.10 terlihat bahwa model baseline (100%) memperoleh nilai tertinggi pada seluruh metrik, dengan *precision* sebesar 0,714, *recall* sebesar 0,741, *mAP@0.5* sebesar 0,761, dan *mAP@0.5:0.95* sebesar 0,463. Nilai ini menjadi acuan utama dalam menilai penurunan performa pada model dengan tingkat anotasi yang lebih rendah.

Pada Skenario A, penurunan kualitas model terlihat signifikan seiring berkurangnya jumlah anotasi. Model dengan 90% anotasi masih mendekati baseline dengan *precision* 0,685 dan *mAP@0.5* 0,722. Namun, penurunan semakin tajam pada 80% dan 70%, dengan *precision* hanya 0,679 dan 0,608, serta *mAP@0.5* sebesar 0,702 dan 0,661. Pada tingkat anotasi 60%, performa semakin melemah dengan *precision* 0,597 dan *mAP@0.5* 0,622, menandakan hilangnya stabilitas pembelajaran akibat keterbatasan anotasi.

Berbeda dengan Skenario A, Skenario B memperlihatkan performa yang lebih stabil, terutama pada tingkat anotasi 90% dan 80%. Bahkan pada 90% anotasi, model menunjukkan *precision* yang lebih tinggi (0,743) dibanding baseline, sementara *mAP@0.5* hanya turun tipis ke 0,754. Model dengan 80% anotasi juga kompetitif dengan *precision* 0,709 dan *mAP@0.5* 0,731. Penurunan performa mulai signifikan pada tingkat 70% dan 60%, namun tetap lebih baik dibandingkan dengan model pada Skenario A dengan tingkat anotasi yang sama.

Visualisasi grafik pada Gambar 4.10 memperkuat temuan ini. Bar chart memperlihatkan bahwa kurva penurunan performa pada Skenario A lebih curam dibandingkan Skenario B, terutama pada metrik *precision* dan *mAP@0.5*. Skenario B memperlihatkan stabilitas relatif

lebih baik hingga tingkat anotasi 80%, sedangkan Skenario A menunjukkan degradasi performa lebih cepat bahkan sejak 80% anotasi.

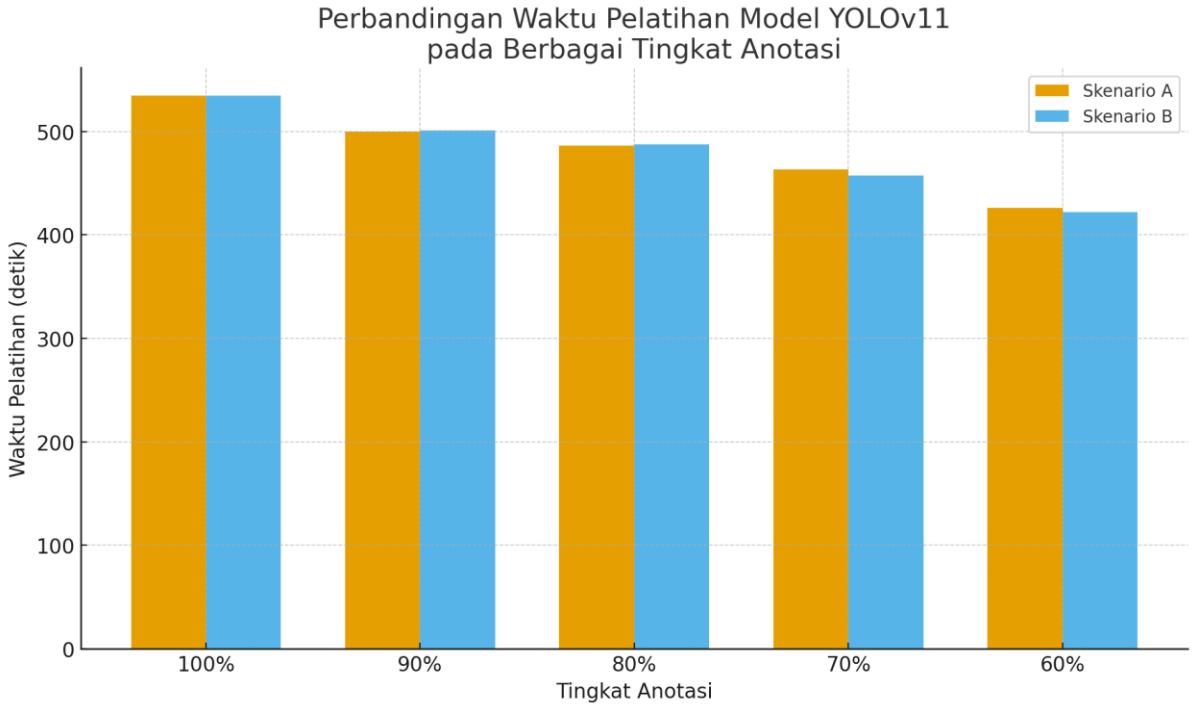
Secara keseluruhan, hasil ini menunjukkan bahwa pengurangan anotasi memang berdampak pada penurunan performa model, namun Skenario B lebih tahan terhadap pengurangan anotasi dibandingkan Skenario A, karena strategi penghapusan anotasi pada objek yang tumpang tindih atau tertutup sebagian menjaga informasi inti yang relevan untuk proses pembelajaran model.

4.1.4 Waktu Pelatihan Model

Selain akurasi, aspek efisiensi komputasi juga menjadi perhatian dalam penelitian ini. Total waktu pelatihan (*training time*) untuk setiap model dicatat berdasarkan keluaran *results.csv* yang dihasilkan selama proses pelatihan menggunakan GPU *NVIDIA Tesla T4* di Google Colab. Analisis ini bertujuan untuk menilai sejauh mana pengurangan anotasi dapat menghemat waktu pelatihan dan meningkatkan efisiensi komputasi.

Tabel 4.2 Waktu Pelatihan Model YOLOv11 pada Berbagai Tingkat Anotasi

Tingkat Anotasi	Jumlah BBox	Waktu Pelatihan Skenario A (detik)	Waktu Pelatihan Skenario B (detik)
100%	4833	534.764	534.764
90%	4501	499.861	501.287
80%	4169	486.156	487.268
70%	3833	463.337	457.310
60%	3500	426.285	422.010



Gambar 4.11 Grafik Tren Waktu Pelatihan Model YOLOv11 pada Berbagai Tingkat Anotasi

Dari Tabel 4.2 terlihat bahwa waktu pelatihan berkurang seiring dengan penurunan jumlah anotasi. Model baseline dengan anotasi penuh (100%) membutuhkan waktu terlama, yaitu sekitar 534,764 detik. Sementara itu, model dengan 60% anotasi menunjukkan waktu pelatihan yang lebih singkat, yakni 426,285 detik pada Skenario A dan 422,010 detik pada Skenario B. Penghematan waktu pelatihan mencapai sekitar 20% ketika anotasi dikurangi dari 100% menjadi 60%. Hal ini memperlihatkan bahwa selain berdampak pada performa model (dibahas pada Subbab 4.1.3), pengurangan anotasi juga memberikan keuntungan nyata dalam hal efisiensi komputasi.

Perbandingan antar skenario menunjukkan tren yang relatif serupa, dengan sedikit perbedaan pada tingkat anotasi rendah. Skenario B cenderung menghasilkan waktu pelatihan yang sedikit lebih singkat dibandingkan Skenario A, terutama pada tingkat 70% dan 60%. Perbedaan ini dapat dikaitkan dengan variasi kompleksitas objek yang tetap dipertahankan dalam anotasi pada masing-masing skenario.

Visualisasi pada Gambar 4.11 memperjelas tren ini. Grafik menunjukkan penurunan yang konsisten pada kedua skenario, di mana garis tren Skenario B sedikit berada di bawah Skenario A. Hal ini menegaskan bahwa strategi penghapusan anotasi pada objek yang saling menutupi (Skenario B) tidak hanya menjaga kualitas deteksi lebih baik (seperti pada Subbab 4.1.3), tetapi juga lebih efisien dari segi komputasi dibandingkan penghapusan anotasi pada objek yang jelas (Skenario A).

Secara keseluruhan, analisis ini memperkuat argumen bahwa pengurangan anotasi tidak hanya menimbulkan *trade-off* pada aspek akurasi, tetapi juga membuka peluang untuk optimasi komputasi. Dengan demikian, aspek waktu pelatihan menjadi dimensi penting dalam menilai efektivitas pengurangan anotasi pada pengembangan model deteksi objek berbasis *deep learning*.

4.2 Hasil Pengujian Model

Setelah seluruh proses pelatihan selesai, setiap model hasil pelatihan dengan variasi tingkat anotasi (100%, 90%, 80%, 70%, dan 60%) diuji menggunakan data uji yang telah disiapkan dalam format YOLO melalui platform Roboflow. Proses pengujian dilakukan secara seragam pada seluruh model untuk menjamin kesetaraan evaluasi.

Evaluasi performa model dilakukan melalui dua pendekatan utama. Pertama, analisis *confusion matrix* digunakan untuk mengukur akurasi klasifikasi per kelas serta mengidentifikasi jenis kesalahan yang paling sering terjadi, seperti kesalahan klasifikasi antar kelas maupun deteksi salah terhadap latar belakang (*false positive detection*). Pendekatan ini memungkinkan identifikasi kelas buah yang paling terdampak akibat pengurangan tingkat anotasi.

Kedua, dilakukan analisis visual terhadap hasil deteksi objek menggunakan aplikasi berbasis *Streamlit*. Aplikasi ini menguji seluruh model pada gambar uji yang sama, sehingga hasil deteksi dari setiap variasi anotasi dapat dibandingkan secara langsung, baik dari segi jumlah objek yang terdeteksi, keberadaan prediksi salah, maupun tingkat *confidence* pada *bounding box* yang dihasilkan.

Pendekatan evaluatif ini tidak hanya memberikan metrik numerik, tetapi juga menawarkan perspektif visual terhadap kemampuan model dalam mengenali objek secara konsisten pada skenario dunia nyata, khususnya saat menghadapi keterbatasan anotasi. Analisis ringkas ini akan dijabarkan lebih lanjut pada Subbab 4.2.1 yang membahas hasil *confusion matrix* dan Subbab 4.2.2 yang memaparkan hasil visualisasi deteksi menggunakan *Streamlit*.

4.2.1 Analisis Confusion Matrix

Confusion matrix digunakan sebagai alat evaluasi utama untuk mengukur performa klasifikasi model YOLOv11 terhadap enam kelas buah: anggur, apel, jeruk, rambutan, salak, dan stroberi. Setiap model diuji menggunakan dataset pengujian yang konsisten, dan perbandingan dilakukan berdasarkan jumlah absolut hasil klasifikasi, tanpa normalisasi.

Pendekatan ini dipilih untuk menampilkan distribusi prediksi secara nyata dan menyoroti kecenderungan kesalahan klasifikasi secara eksplisit.

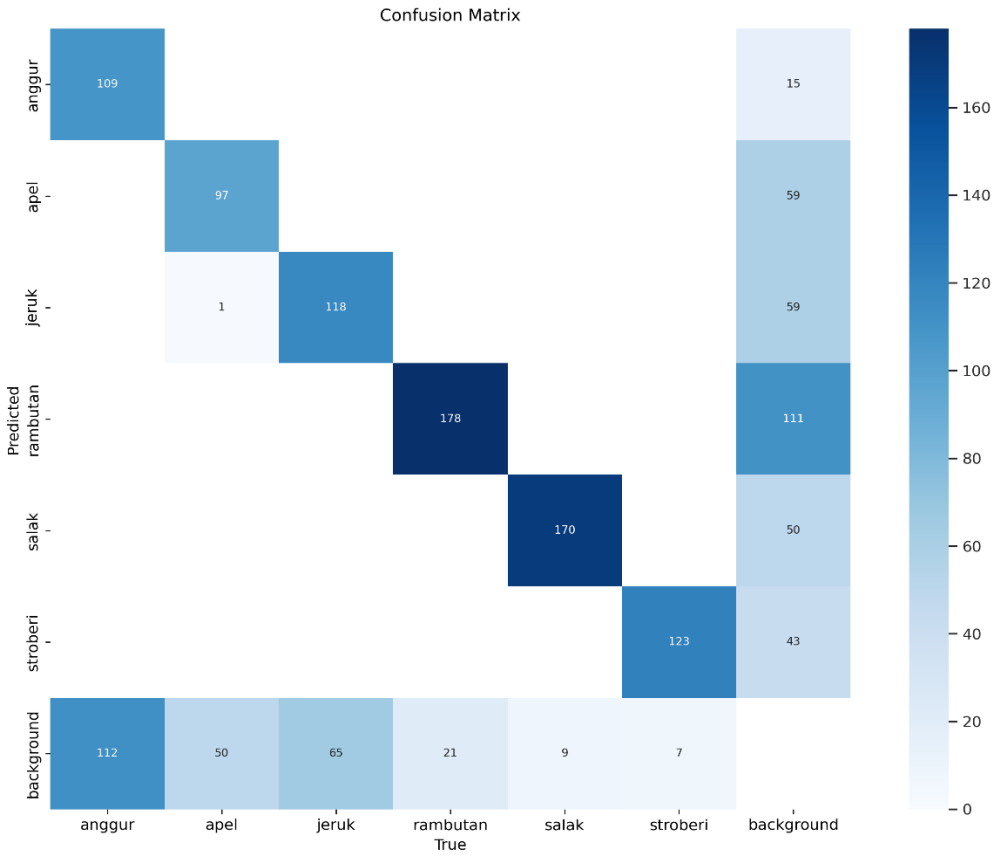
Analisis difokuskan pada:

- **True Positive (nilai diagonal)**: prediksi yang benar terhadap masing-masing kelas.
- **False Negative**: objek yang tidak dikenali dan terklasifikasi sebagai latar belakang.
- **Kesalahan antar kelas (off-diagonal)**: objek yang diklasifikasikan ke kelas yang salah.

Dua skenario digunakan untuk mengevaluasi dampak pengurangan anotasi:

- **Skenario A**: penghapusan dilakukan terhadap objek yang terlihat jelas.
- **Skenario B**: penghapusan dilakukan terhadap objek yang tumpang tindih atau sebagian tersembunyi.

a) Model Baseline (100% Anotasi)



Gambar 4.12 Confusion Matrix Model YOLOv11 dengan Anotasi 100%.

Model baseline menunjukkan performa klasifikasi yang sangat baik, yang tercermin dari dominasi nilai diagonal pada confusion matrix. Hal ini mengindikasikan bahwa sebagian besar objek berhasil diklasifikasikan dengan benar oleh model.

Temuan Utama per Kelas:

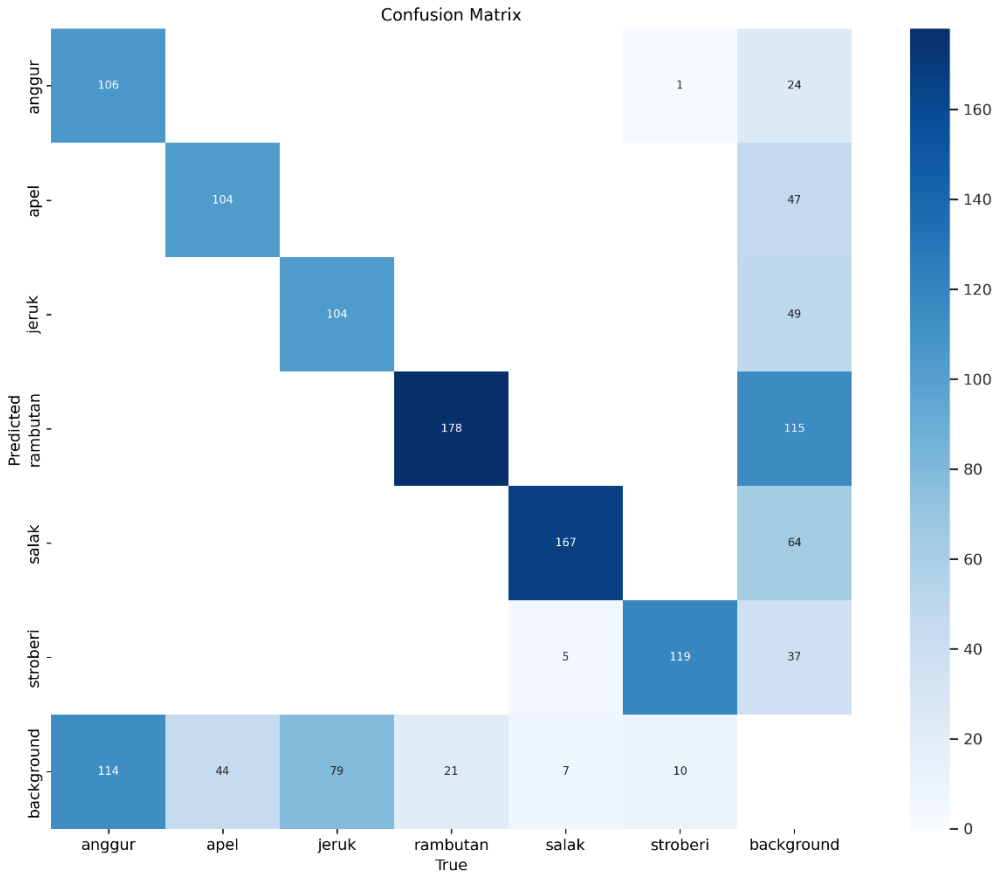
- Rambutan mencatat performa terbaik dengan 178 prediksi benar dan hanya 21 kesalahan ke background, menunjukkan bahwa fitur visual rambutan (seperti tekstur unik) mudah dikenali oleh model.
- Salak juga tampil kuat dengan 170 prediksi benar. Namun, terdapat 50 kesalahan klasifikasi ke kelas buah lain, yang mengindikasikan adanya kemiripan visual (misalnya warna gelap atau bentuk lonjong) antara kedua kelas tersebut.
- Stroberi mencapai 123 prediksi benar, namun masih terjadi 43 kesalahan prediksi ke background, yang menunjukkan adanya kasus *miss detection*, kemungkinan disebabkan oleh objek yang sebagian tersembunyi atau pencahayaan tidak ideal.
- Apel memiliki 97 prediksi benar, tetapi mengalami 59 kesalahan klasifikasi ke kelas buah lain dan 50 kesalahan ke background. Hal ini mengindikasikan bahwa bentuk dan

warna apel dalam dataset mungkin tidak cukup konsisten, atau terlalu mirip dengan stroberi dalam beberapa kasus.

- Jeruk diklasifikasikan benar sebanyak 118 kali, dengan 59 kesalahan ke kelas buah lain dan 65 kesalahan ke background. Tingginya kesalahan ke *background* menunjukkan bahwa objek jeruk kerap tidak terdeteksi jika sebagian tertutup atau tampil kurang kontras.
- Anggur merupakan kelas dengan performa terendah: hanya 109 prediksi benar dengan 112 kesalahan ke background. Ini menandakan bahwa anggur, sebagai objek kecil dengan warna gelap, menjadi tantangan tersendiri bagi model dalam proses deteksi, terutama dalam kondisi visual yang kompleks.

Secara keseluruhan, model dengan anotasi penuh memperlihatkan kapabilitas klasifikasi yang kuat terhadap sebagian besar kelas buah. Namun, miss detection ke background tetap menjadi isu utama, khususnya pada kelas anggur, jeruk, dan stroberi, yang sering kali tertukar dengan latar atau buah lain akibat ukuran kecil, warna gelap, atau kemiripan bentuk antar kelas.

b) Skenario A - 90% Anotasi



Gambar 4.13 Confusion Matrix Model YOLOv11 dengan Anotasi 90% (Skenario A).

Model dengan tingkat anotasi 90% menunjukkan penurunan performa yang tidak terlalu signifikan dibandingkan baseline, namun mulai terlihat adanya peningkatan jumlah kesalahan klasifikasi ke latar belakang maupun ke kelas lain.

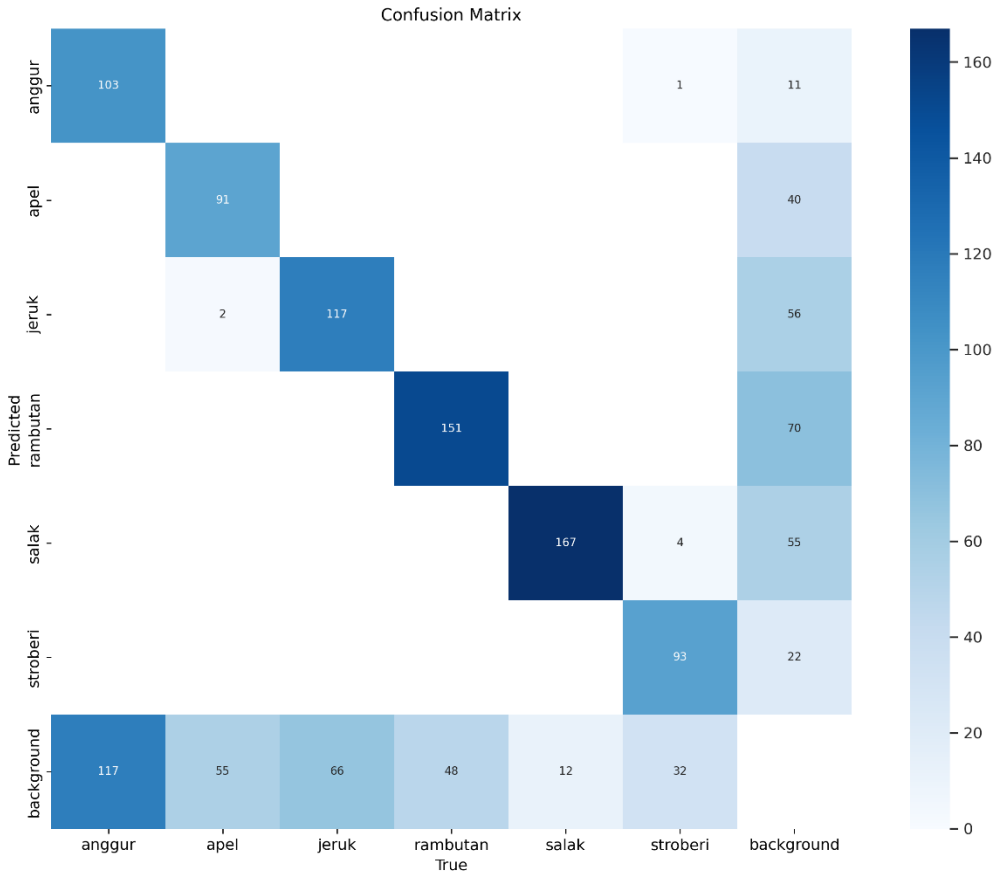
Temuan Utama Per Kelas:

- Rambutan tetap menunjukkan performa paling stabil dengan 178 prediksi benar dan 21 kesalahan ke background. Hal ini menunjukkan bahwa karakteristik visual rambutan tetap mudah dikenali meskipun sebagian anotasi dihapus.
- Salak mengalami sedikit penurunan dengan 167 prediksi benar dan 64 salah ke background, yang menunjukkan peningkatan jumlah miss detection dibandingkan baseline.
- Stroberi berhasil diklasifikasikan dengan benar sebanyak 119 kali, namun masih terdapat 37 kesalahan ke background dan 10 salah ke kelas buah lain. Jumlah kesalahan ini mengindikasikan bahwa model mulai kesulitan membedakan stroberi dari latar belakang dan kelas buah lain yang memiliki warna serupa.

- Jeruk memperoleh 104 prediksi benar, namun mengalami 79 kesalahan ke background dan 49 ke kelas buah lain, menunjukkan penurunan ketepatan yang cukup besar dan tingginya ambiguitas antara jeruk dan stroberi saat tingkat anotasi berkurang.
- Apel diklasifikasikan dengan benar sebanyak 104 kali, dan mengalami 47 kesalahan ke kelas buah lain serta 44 ke background. Kemiripan warna dan bentuk antara apel dan stroberi kemungkinan menjadi penyebab utama kesalahan ini.
- Anggur menunjukkan 106 prediksi benar namun memiliki 114 salah ke background, menegaskan kembali bahwa anggur adalah kelas dengan kesulitan tertinggi untuk dikenali, terutama ketika sebagian anotasi dihapus. Objek kecil dan gelap seperti anggur sangat rentan terhadap miss detection.

Model 90% anotasi pada Skenario A masih mempertahankan performa tinggi pada kelas rambutan dan salak. Namun, kesalahan klasifikasi meningkat, khususnya berupa false negative ke background, terutama pada kelas anggur, jeruk, dan stroberi. Kelas-kelas dengan ukuran kecil, warna gelap, atau objek yang tertumpuk menjadi lebih rentan terhadap penurunan performa ketika tingkat anotasi dikurangi.

c) Skenario A - 80% Anotasi



Gambar 4.14 Confusion Matrix Model YOLOv11 dengan Anotasi 80% (Skenario A).

Model dengan tingkat anotasi 80% mulai menunjukkan penurunan performa yang lebih nyata dibandingkan baseline dan model 90%, terutama dalam bentuk peningkatan jumlah kesalahan ke latar belakang. Penghapusan sebagian objek yang paling jelas dari dataset pelatihan pada skenario A terbukti mulai memengaruhi ketepatan deteksi untuk sebagian besar kelas.

Temuan Utama Per Kelas:

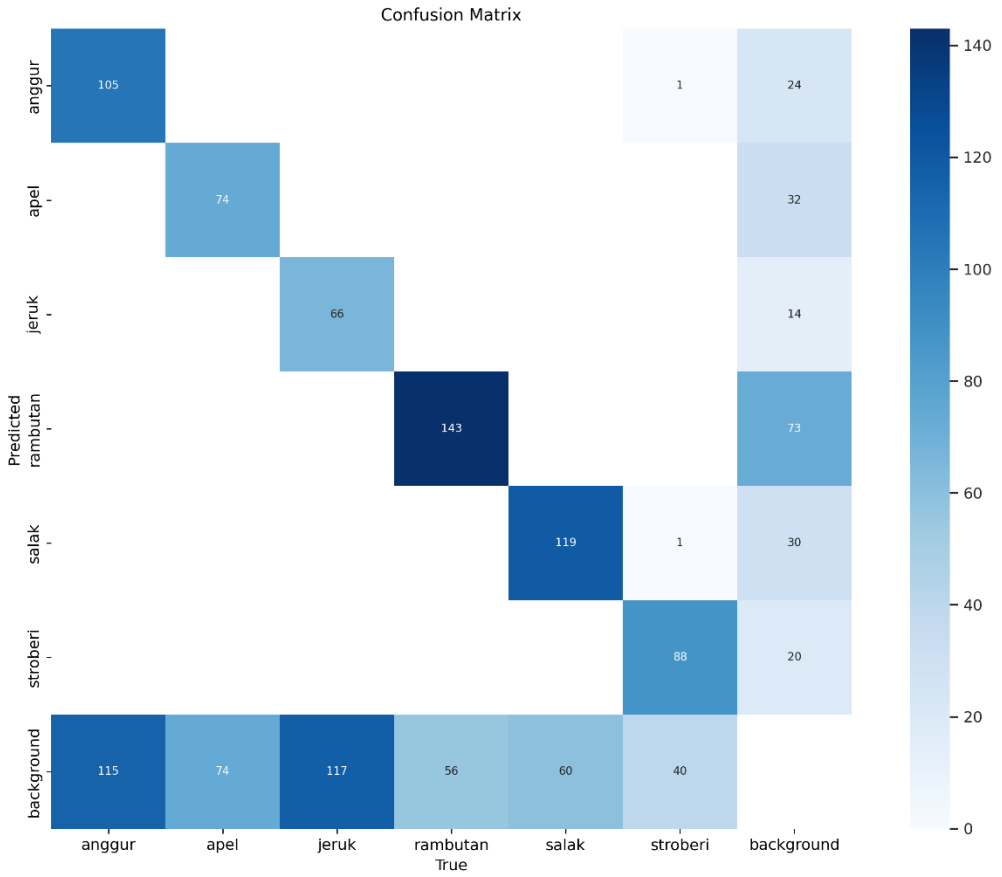
- Rambutan tetap mencatat performa yang cukup baik meskipun mulai menurun, dengan 151 prediksi benar dan 48 kesalahan ke background. Jumlah ini menandakan bahwa model masih mampu mengenali ciri khas visual rambutan meskipun sebagian anotasi dihilangkan, namun sensitivitasnya terhadap objek yang tidak sepenuhnya terlihat mulai menurun.
- Salak berhasil diklasifikasikan dengan benar sebanyak 167 kali, namun mengalami 70 kesalahan ke kelas stroberi. Jumlah kesalahan ini meningkat cukup tajam dibandingkan pada 90%, mengindikasikan bahwa model semakin kesulitan membedakan dua buah

dengan bentuk dan warna gelap yang serupa ketika data pelatihan tidak mencakup objek paling jelas.

- Stroberi menunjukkan 93 prediksi benar, dengan 55 kesalahan ke kelas buah lain dan 32 kesalahan ke background. Penurunan akurasi ini memperkuat bahwa stroberi menjadi lebih sulit dikenali ketika anotasi tidak mencakup contoh yang lengkap dan jelas, terutama jika objek tertutup sebagian atau tumpang tindih.
- Jeruk mengalami penurunan performa dengan 117 prediksi benar, namun memiliki 66 kesalahan ke background dan 56 ke kelas buah lain. Tingginya jumlah kesalahan ini menunjukkan bahwa model semakin kehilangan kemampuan membedakan jeruk dari latar belakang dan dari buah yang serupa secara visual.
- Apel memperoleh 91 prediksi benar, namun kesalahan ke background meningkat menjadi 55 kasus, ditambah 40 kesalahan ke kelas buah lain. Apel tampaknya terdampak oleh variasi warna dan bentuk dalam dataset, dan mulai sulit dibedakan ketika anotasi objek utuh dikurangi.
- Anggur kembali menjadi kelas dengan performa paling rendah. Hanya terdapat 103 prediksi benar, dan 117 kesalahan ke background, menjadikannya kelas dengan tingkat kesalahan ke background tertinggi. Hal ini menandakan bahwa model gagal mendeteksi anggur dengan baik saat kehilangan referensi visual dari objek anggur yang paling utuh dan terang.

Model YOLOv11 dengan tingkat anotasi 80% pada Skenario A mulai menunjukkan penurunan performa yang lebih signifikan, terutama untuk kelas-kelas yang memiliki ukuran kecil, warna gelap, atau bentuk tumpang tindih seperti anggur, apel, dan stroberi. Kesalahan klasifikasi paling umum terjadi sebagai false negative ke background, mengindikasikan penurunan kepercayaan model dalam mengenali objek yang tidak sepenuhnya terlihat. Sementara itu, kelas rambutan dan salak masih menunjukkan stabilitas relatif lebih baik, meskipun tetap terdampak oleh pengurangan kualitas informasi dalam dataset pelatihan.

d) Skenario A - 70% Anotasi



Gambar 4.15 Confusion Matrix Model YOLOv11 dengan Anotasi 70% (Skenario A).

Model dengan tingkat anotasi 70% memperlihatkan penurunan performa yang lebih mencolok dibandingkan tingkat anotasi sebelumnya. Jumlah prediksi benar pada setiap kelas mulai menurun signifikan, diiringi dengan meningkatnya kesalahan klasifikasi ke latar belakang dan antar kelas buah.

Temuan Utama Per Kelas:

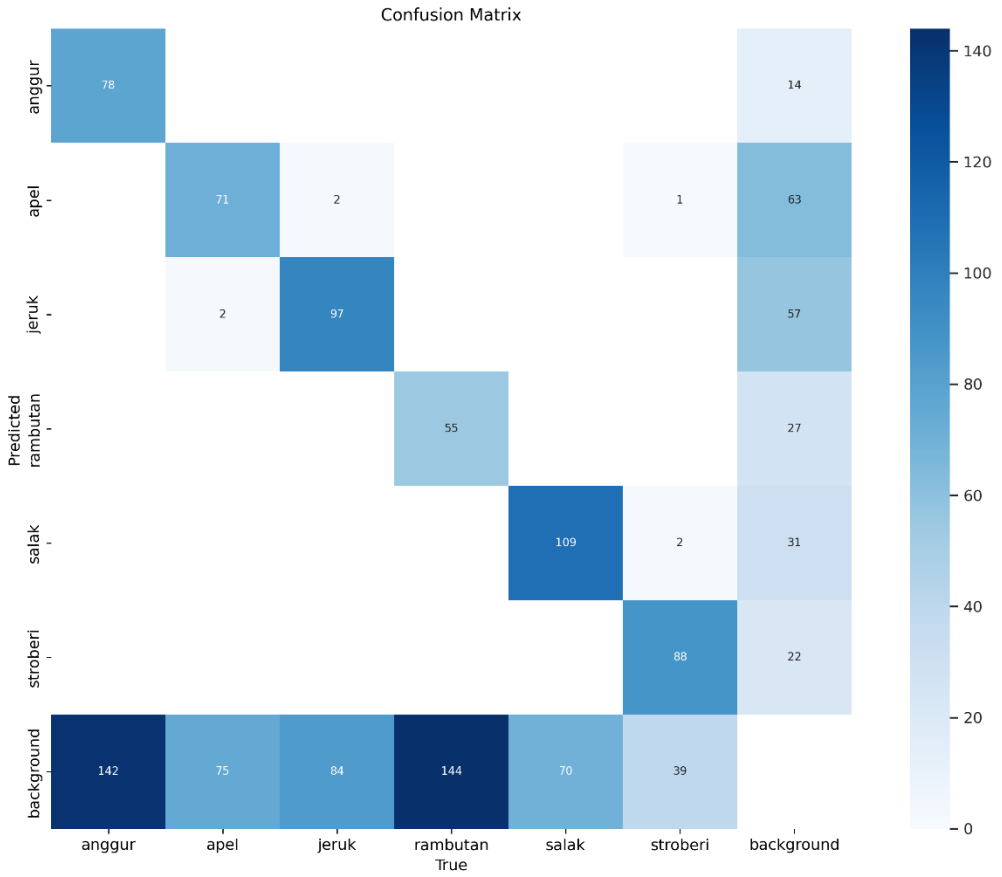
- Rambutan mencatat 143 prediksi benar dengan 56 kesalahan ke background. Meskipun mengalami penurunan dibanding model 90%, model masih cukup mampu mengenali rambutan berkat fitur visual khas seperti tekstur permukaan yang berduri.
- Salak mengalami degradasi performa dengan hanya 119 prediksi benar dan 60 salah ke background. Ini menunjukkan bahwa kemampuan model membedakan salak mulai terganggu, kemungkinan karena pengurangan anotasi membuat variasi posisi dan pencahayaan kurang tercakup.
- Stroberi hanya berhasil dikenali dengan 88 prediksi benar, dan mengalami 40 kesalahan ke background serta 30 salah ke kelas buah lain. Angka ini menunjukkan ambiguitas

antara stroberi dan salak mulai meningkat, kemungkinan akibat kemiripan bentuk dan warna yang tidak tertangkap optimal dalam pelatihan.

- Jeruk hanya terklasifikasi benar sebanyak 66 kali, sementara 117 prediksi salah ke background. Angka ini menunjukkan penurunan drastis dalam kemampuan deteksi jeruk, mengindikasikan bahwa model sangat terganggu oleh objek tertutup atau kurang terang.
- Apel hanya mendapatkan 74 prediksi benar dengan 74 kesalahan ke background dan 32 salah ke kelas buah lain. Kesalahan ke stroberi makin dominan, yang memperkuat dugaan bahwa warna atau bentuk apel dalam dataset terlalu mirip dengan stroberi setelah banyak anotasi dihapus.
- Anggur mencatat 105 prediksi benar dan 115 salah ke background. Anggur tetap menjadi kelas dengan tantangan terbesar bagi model. Ukuran kecil, warna gelap, dan posisi bertumpuk membuat objek ini semakin sulit dikenali seiring menurunnya jumlah data anotasi yang tersedia.

Model dengan 70% tingkat anotasi pada Skenario A mengalami penurunan performa yang signifikan dibandingkan model dengan anotasi lebih lengkap. Kesalahan klasifikasi ke background meningkat tajam, terutama pada kelas jeruk dan anggur. Kelas rambutan masih relatif stabil, sementara kelas stroberi dan apel semakin sering tertukar. Penurunan ini menunjukkan bahwa pengurangan anotasi mulai mengganggu kemampuan model dalam membedakan karakteristik visual yang mirip antar kelas serta mengidentifikasi objek dalam kondisi visual yang tidak ideal.

e) Skenario A - 60% Anotasi



Gambar 4.16 Confusion Matrix Model YOLOv11 dengan Anotasi 60% (Skenario A).

Pada tingkat anotasi 60%, penurunan performa model semakin terlihat signifikan dibandingkan baseline. Dominasi nilai diagonal mulai melemah, dengan peningkatan besar pada kesalahan klasifikasi ke latar belakang. Ini menunjukkan bahwa pengurangan anotasi sebesar 40% mulai mengganggu pembelajaran fitur visual yang stabil.

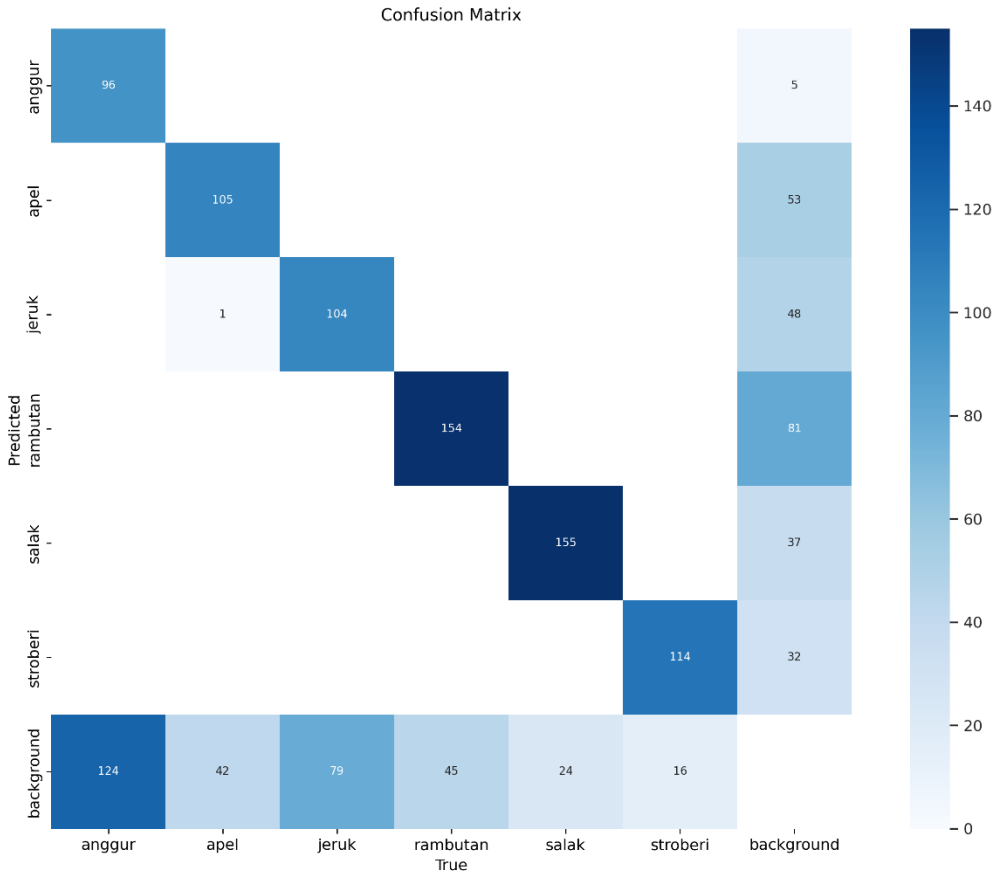
Temuan Utama Per Kelas:

- Rambutan hanya berhasil diklasifikasikan dengan benar sebanyak 55 kali. Sebaliknya, sebanyak 144 prediksi justru terdeteksi sebagai latar belakang. Ini menjadi indikasi kuat bahwa objek rambutan mulai tidak dikenali ketika proporsi anotasinya dikurangi secara drastis.
- Salak memperoleh 109 prediksi benar, tetapi masih terdapat 70 kesalahan ke background dan 31 ke kelas buah lain. Model tampak kesulitan membedakan salak dengan objek serupa, sekaligus menandakan mulai tingginya false negative akibat kekurangan data beranotasi.

- Stroberi memiliki 88 prediksi benar, namun disertai dengan 39 kesalahan ke background dan 22 salah ke kelas buah lain. Ini menunjukkan bahwa meskipun masih cukup dikenali, model semakin kesulitan membedakan stroberi dari latar dan objek mirip seperti salak.
- Jeruk mengalami penurunan akurasi signifikan dengan hanya 97 prediksi benar. Sebanyak 84 kesalahan terjadi ke background, menegaskan bahwa jeruk menjadi salah satu kelas yang paling terdampak akibat hilangnya sebagian besar anotasi, kemungkinan karena faktor bentuk dan warna yang tidak mencolok.
- Apel berhasil dikenali dengan benar sebanyak 71 kali, tetapi mengalami 75 kesalahan ke background dan 63 ke kelas buah lain. Ketidakkonsistenan visual apel dalam data tampaknya semakin sulit ditangani oleh model dengan pengurangan anotasi tinggi.
- Anggur mengalami penurunan besar dengan hanya 78 prediksi benar, dan 142 kesalahan ke background. Objek kecil dan gelap seperti anggur tampaknya menjadi yang paling sensitif terhadap berkurangnya data anotasi, membuat performanya turun drastis.

Model YOLOv11 dengan 60% anotasi dalam Skenario A menunjukkan penurunan performa yang nyata. Terutama terjadi pada kelas-kelas yang secara visual lebih sulit dibedakan, seperti anggur dan jeruk. Jumlah kesalahan klasifikasi ke latar belakang meningkat secara signifikan, mengindikasikan tingginya tingkat *miss detection*. Dapat disimpulkan bahwa pada titik ini, pengurangan anotasi mulai menyebabkan model kehilangan kemampuan representasi visual yang baik, terutama untuk objek kecil, tumpang tindih, atau memiliki kontras rendah.

f) Skenario B - 90% Anotasi



Gambar 4.17 Confusion Matrix Model YOLOv11 dengan Anotasi 90% (Skenario B).

Model pada Skenario B dengan tingkat anotasi 90% masih menunjukkan performa deteksi yang cukup baik untuk beberapa kelas buah, namun terlihat penurunan performa pada kelas-kelas tertentu, terutama akibat peningkatan kesalahan klasifikasi ke latar belakang dan ke kelas yang memiliki karakteristik visual serupa.

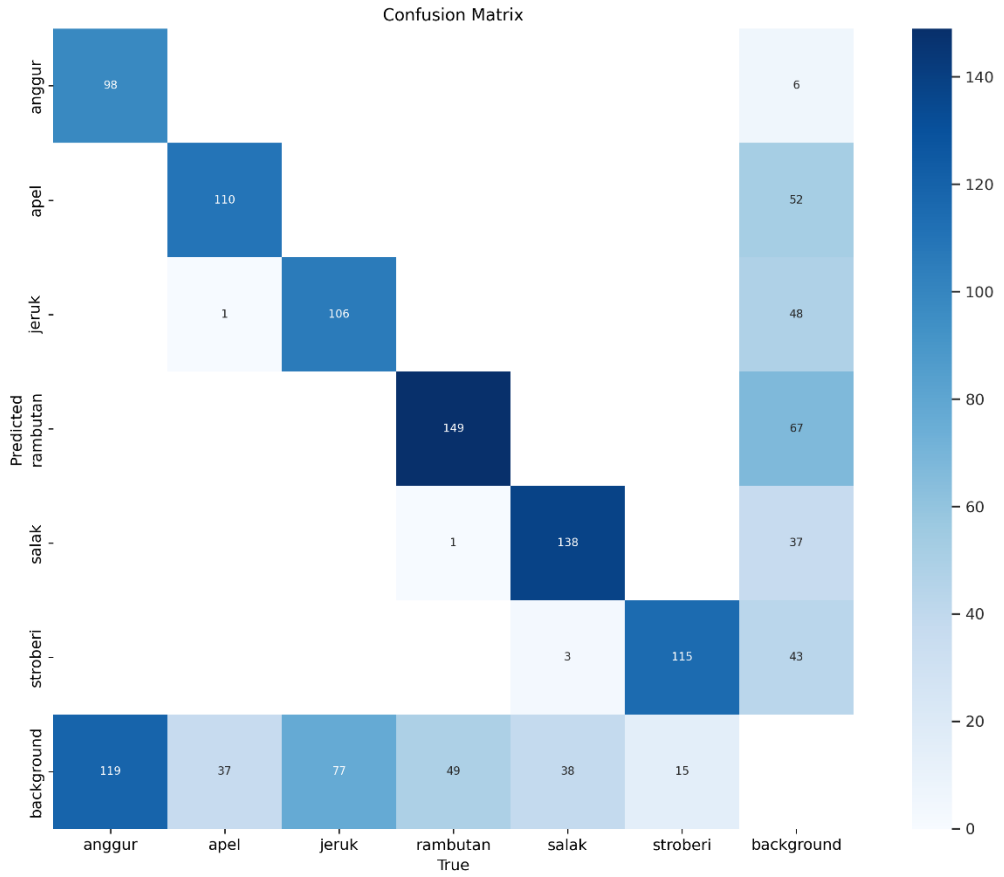
Temuan Utama Per Kelas:

- Rambutan berhasil diklasifikasikan dengan benar sebanyak 154 kali dan mengalami 45 kesalahan ke background. Penurunan jumlah prediksi benar menunjukkan bahwa model mulai kesulitan mengenali ciri khas rambutan saat sebagian anotasi tidak tersedia, terutama pada kondisi pencahayaan atau sudut pengambilan gambar yang tidak ideal.
- Salak mempertahankan performa cukup baik dengan 155 prediksi benar, meskipun terjadi 81 kesalahan ke background. Hal ini menunjukkan bahwa model masih bisa mengenali bentuk dan tekstur salak, tetapi kehilangan kepekaan dalam beberapa kasus yang mengarah pada false negative.

- Stroberi diklasifikasikan benar sebanyak 114 kali, tetapi mengalami 32 kesalahan ke background dan 16 kesalahan ke kelas buah lain. Ini menandakan adanya tantangan dalam membedakan stroberi dari jeruk, terutama jika warna dan ukuran antar kelas saling mendekati.
- Jeruk memperoleh 104 prediksi benar, namun mengalami 79 kesalahan ke background dan 48 ke kelas buah lain. Jumlah kesalahan ini menunjukkan bahwa model kesulitan membedakan jeruk dari latar belakang maupun dari buah berwarna serupa seperti stroberi.
- Apel menunjukkan performa cukup baik dengan 105 prediksi benar, tetapi terdapat 42 kesalahan ke background dan 53 ke kelas buah lain. Mirip dengan jeruk, apel cenderung salah diklasifikasikan akibat kemiripan warna atau bentuk dengan stroberi.
- Anggur berhasil dikenali sebanyak 96 kali, namun merupakan kelas dengan kesalahan ke background tertinggi, yaitu sebanyak 124. Ini mengonfirmasi bahwa anggur sebagai objek kecil dan gelap sangat rentan terhadap miss detection saat tingkat anotasi berkurang.

Pada skenario B dengan 90% anotasi, performa model masih terjaga baik untuk kelas salak dan stroberi, namun kelas anggur dan jeruk mengalami peningkatan jumlah kesalahan cukup besar. False negative ke background mendominasi sebagian besar kesalahan, menandakan bahwa objek buah mulai tidak terdeteksi ketika anotasi dikurangi. Hal ini terutama berdampak pada buah berukuran kecil, berwarna gelap, atau memiliki kemiripan visual dengan kelas lain.

g) Skenario B - 80% Anotasi



Gambar 4.18 Confusion Matrix Model YOLOv11 dengan Anotasi 80% (Skenario B).

Model dengan tingkat anotasi sebesar 80% pada Skenario B menunjukkan kecenderungan penurunan performa yang semakin nyata dibandingkan versi sebelumnya. Penurunan terlihat pada meningkatnya jumlah kesalahan ke latar belakang serta perpindahan klasifikasi antar kelas buah. Meskipun beberapa kelas tetap menunjukkan performa yang cukup baik, model mulai menunjukkan kesulitan dalam membedakan objek dengan fitur visual yang mirip.

Temuan Utama Per Kelas:

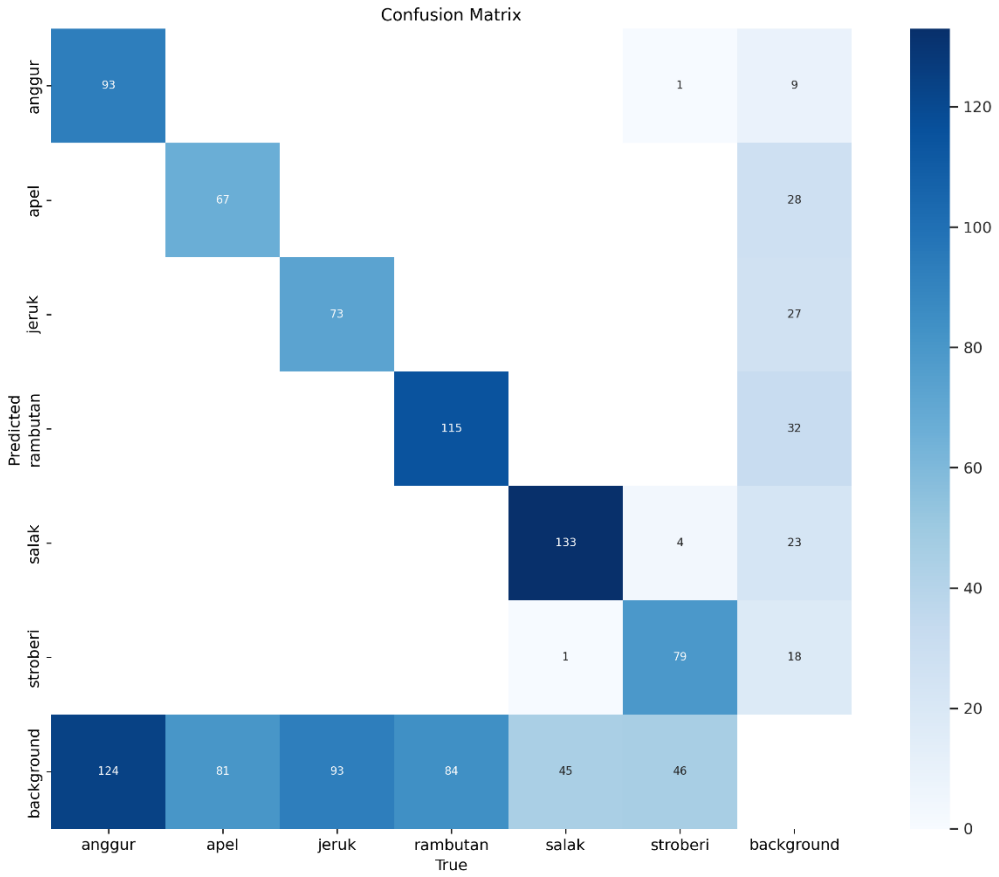
- Rambutan memperoleh 149 prediksi benar. Namun, terjadi peningkatan kesalahan klasifikasi ke kelas buah lain sebanyak 67 dan ke background sebanyak 49. Hal ini menunjukkan bahwa bentuk dan tekstur rambutan mulai sulit dikenali model ketika sebagian anotasi hilang, dan cenderung dikira sebagai salak yang memiliki permukaan mirip.
- Salak diklasifikasikan dengan benar sebanyak 138 kali, dengan 37 kesalahan ke kelas buah lain dan 38 ke background. Model masih cukup mampu mengenali salak, tetapi

mulai menunjukkan ambiguitas terhadap stroberi, kemungkinan karena warna kulit yang serupa dan pencahayaan yang tidak seragam.

- Stroberi hanya memperoleh 115 prediksi benar, dengan kesalahan cukup besar ke background sebanyak 15 dan ke kelas buah lain sebanyak 43. Hal ini mengindikasikan bahwa model kesulitan mendeteksi stroberi saat anotasi berkurang, apalagi jika objek tertutup sebagian atau tumpang tindih.
- Jeruk hanya diklasifikasikan benar sebanyak 106 kali, dengan 117 salah ke background dan 14 ke kelas buah lain. Tingginya nilai false negative ke background menandakan bahwa objek jeruk kerap luput terdeteksi saat jumlah anotasi menurun.
- Apel menunjukkan penurunan signifikan dengan hanya 110 prediksi benar. Sebanyak 37 kasus salah ke background dan 52 ke kelas buah lain, menunjukkan bahwa kemiripan warna dan bentuk antara apel dan stroberi menyulitkan model dalam mengenali perbedaan di antara keduanya.
- Anggur berhasil diklasifikasikan dengan benar sebanyak 98 kali, tetapi tetap menjadi salah satu kelas yang rentan dengan 119 salah ke background dan 6 ke kelas buah lain. Hal ini memperkuat bahwa objek kecil dan gelap seperti anggur tetap menjadi tantangan meski performanya sedikit lebih stabil dibanding beberapa kelas lain.

Model dengan tingkat anotasi 80% pada Skenario B mulai menunjukkan degradasi performa yang lebih jelas, terutama pada kelas jeruk, apel, dan stroberi. Kelas rambutan dan salak masih relatif stabil, meskipun mulai terpengaruh oleh ambiguitas antar kelas. Kesalahan klasifikasi ke background mendominasi, mengindikasikan bahwa berkurangnya anotasi mulai berdampak pada kemampuan model dalam mendeteksi objek secara akurat, terutama saat objek memiliki fitur visual yang mirip atau dalam kondisi kompleks seperti tumpang tindih dan pencahayaan tidak ideal.

h) Skenario B - 70% Anotasi



Gambar 4.19 Confusion Matrix Model YOLOv11 dengan Anotasi 70% (Skenario B).

Model dengan tingkat anotasi 70% pada Skenario B menunjukkan penurunan performa yang lebih nyata dibandingkan level sebelumnya. Kesalahan klasifikasi ke background meningkat secara signifikan, terutama pada kelas yang sebelumnya sudah memiliki tingkat kesalahan tinggi.

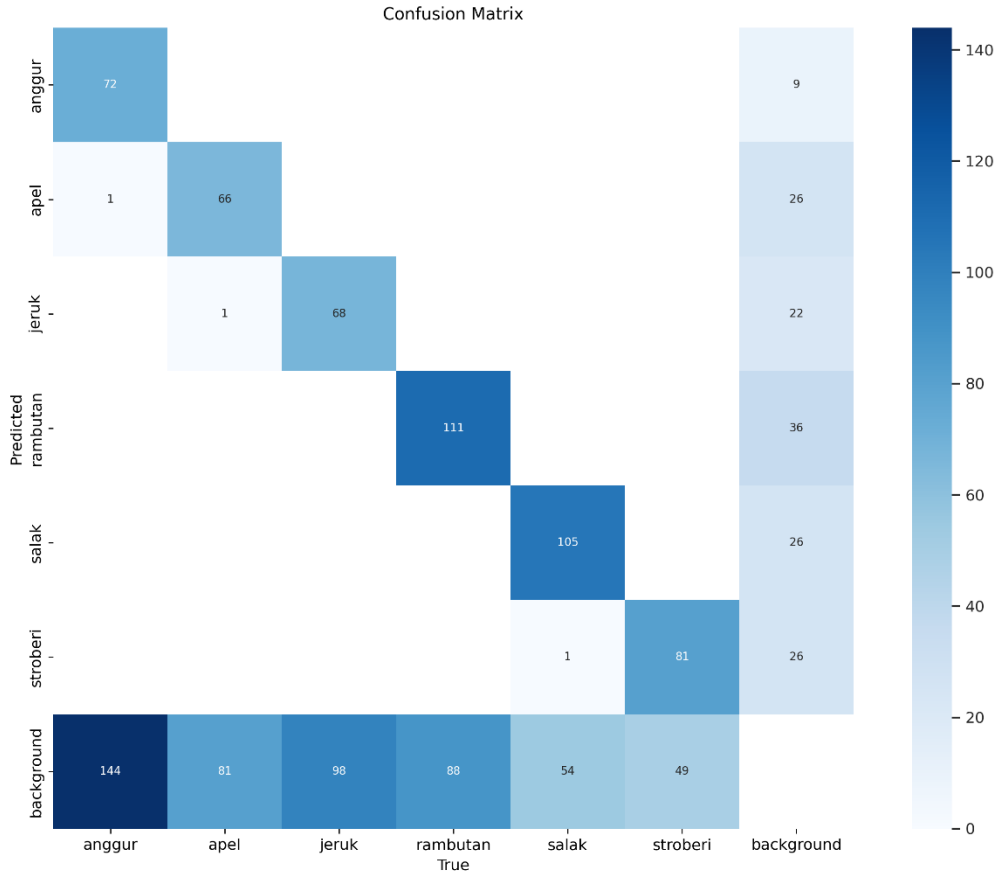
Temuan Utama Per Kelas:

- Rambutan hanya mampu dikenali dengan benar sebanyak 115 kali dan mengalami 84 kesalahan ke background. Ini merupakan penurunan drastis dari skenario sebelumnya, yang menunjukkan bahwa rambutan sangat tergantung pada kelengkapan anotasi.
- Salak tetap menjadi salah satu kelas paling stabil dengan 133 prediksi benar, namun kesalahan ke background meningkat menjadi 45. Ini menandakan mulai adanya tantangan dalam membedakan salak dari latar belakang saat anotasi berkurang.
- Stroberi dikenali benar sebanyak 79 kali, dan mengalami 46 salah ke background serta 18 salah ke kelas buah lain. Meskipun jumlah benar masih tinggi, akurasi menurun dibanding 80%, dengan peningkatan kesalahan klasifikasi ke kelas serupa.

- Jeruk mengalami penurunan performa yang cukup jelas, dengan hanya 73 prediksi benar dan 93 kesalahan ke background. Angka ini menegaskan bahwa model mulai kesulitan dalam mendeteksi jeruk secara konsisten.
- Apel menunjukkan 67 prediksi benar, dengan 81 salah ke background dan 63 ke kelas buah lain. Kesalahan klasifikasi yang tinggi ke stroberi menunjukkan ambiguitas visual yang semakin meningkat ketika data anotasi semakin sedikit.
- Anggur terdeteksi dengan benar sebanyak 93 kali, dan salah ke background sebanyak 124. Angka ini merupakan sinyal kuat bahwa anggur adalah kelas paling terdampak oleh pengurangan anotasi, sejalan dengan karakteristik visualnya yang kecil dan bertumpuk.

Pada tingkat anotasi 70%, model mulai menunjukkan degradasi performa yang cukup signifikan di hampir seluruh kelas, terutama rambutan dan anggur. Sebagian besar kesalahan berupa miss detection ke background, yang menandakan penurunan kemampuan model dalam mengidentifikasi keberadaan objek. Kombinasi anotasi terbatas dan karakteristik visual tertentu seperti warna gelap, bentuk kecil, atau objek tumpuk sangat memengaruhi keberhasilan deteksi.

i) Skenario B - 60% Anotasi



Gambar 4.20 Confusion Matrix Model YOLOv11 dengan Anotasi 60% (Skenario B).

Model YOLOv11 pada skenario anotasi 60% menunjukkan penurunan performa yang cukup signifikan dibandingkan skenario-skenario sebelumnya. Jumlah prediksi benar menurun pada sebagian besar kelas, dan kesalahan prediksi terutama ke background meningkat drastis. Hal ini mengindikasikan bahwa model semakin kesulitan mengenali objek ketika proporsi anotasi yang tersedia terlalu rendah.

Temuan Utama Per Kelas:

- Anggur hanya berhasil dikenali dengan benar sebanyak 73 kali, dengan 144 kesalahan prediksi ke background. Jumlah kesalahan ini menunjukkan bahwa objek anggur semakin sulit dideteksi, terutama karena bentuknya kecil dan sering saling bertumpuk.
- Apel memiliki 66 prediksi benar dan 81 kesalahan ke background. Selain itu, terdapat satu kesalahan klasifikasi ke anggur, yang dapat disebabkan oleh kemiripan warna antara kedua buah tersebut pada beberapa kondisi cahaya.

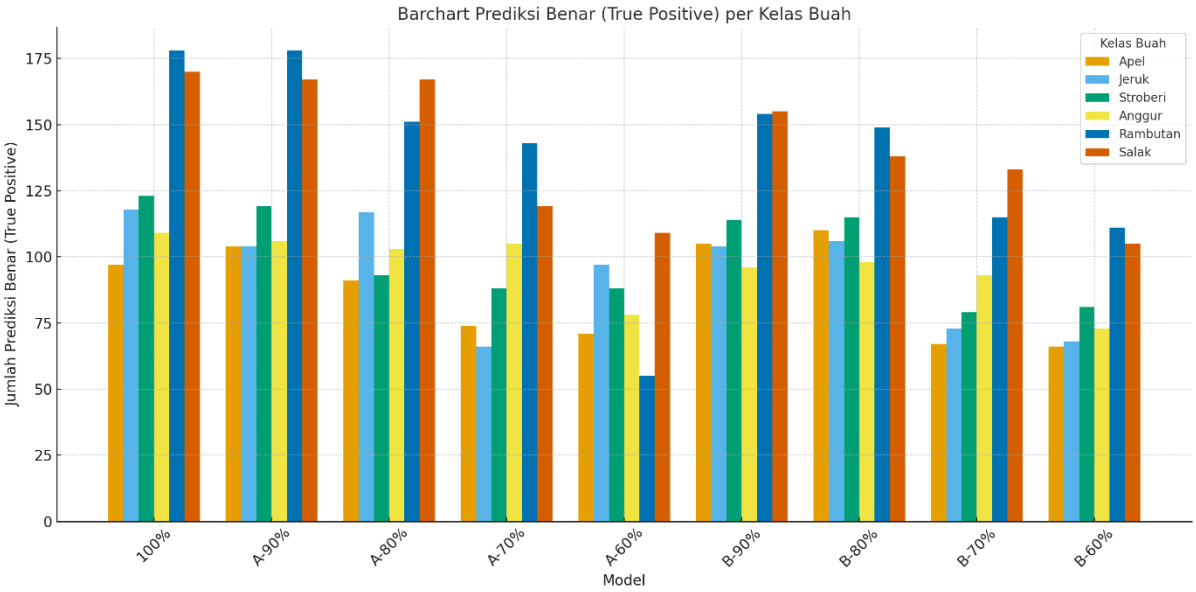
- Jeruk memperoleh 68 prediksi benar, namun salah diprediksi ke background sebanyak 98 kali dan 22 kali ke kelas buah lain. Ini menunjukkan ambiguitas semakin tinggi karena penurunan jumlah anotasi yang memperlemah pemahaman spasial model.
- Rambutan dikenali dengan benar sebanyak 111 kali dan mengalami 88 kesalahan ke background. Penurunan performa ini cukup drastis jika dibandingkan dengan skenario 100% dan 90%, menunjukkan bahwa jumlah anotasi yang terbatas membuat ciri khas rambutan seperti rambut halus menjadi tidak cukup untuk pembelajaran model.
- Salak mendapatkan 105 prediksi benar namun salah ke background sebanyak 54 kali dan satu kali ke stroberi. Ini menunjukkan stabilitasnya masih cukup baik meski ada penurunan, kemungkinan karena tekstur dan bentuk salak yang cukup khas.
- Stroberi berhasil dideteksi dengan benar sebanyak 81 kali, dengan 49 kesalahan ke background dan satu kesalahan ke salak. Jumlah ini mengindikasikan bahwa model mulai kesulitan membedakan stroberi dari latar dan buah lain yang warnanya mirip.

Analisis *confusion matrix* digunakan untuk mengevaluasi distribusi prediksi benar (*true positive*) dan kesalahan prediksi (*false positive* maupun *false negative*) pada setiap kelas buah. Dengan pendekatan ini, dapat diidentifikasi kelas buah yang stabil maupun yang paling terdampak akibat pengurangan tingkat anotasi. Nilai diagonal utama pada *confusion matrix* menjadi indikator penting karena menunjukkan jumlah prediksi benar pada masing-masing kelas.

Tabel 4.3 berikut merangkum jumlah prediksi benar pada enam kelas buah (apel, jeruk, stroberi, anggur, rambutan, dan salak) untuk model baseline 100% hingga model dengan anotasi terbatas (90%, 80%, 70%, dan 60%) pada kedua skenario.

Tabel 4.3 Rekapitulasi Prediksi Benar (True Positive) pada Confusion Matrix

Model	Apel	Jeruk	Stroberi	Anggur	Rambutan	Salak
100% (Baseline)	97	118	123	109	178	170
A - 90%	104	104	119	106	178	167
A - 80%	91	117	93	103	151	167
A - 70%	74	66	88	105	143	119
A - 60%	71	97	88	78	55	109
B - 90%	105	104	114	96	154	155
B - 80%	110	106	115	98	149	138
B - 70%	67	73	79	93	115	133
B - 60%	66	68	81	73	111	105



Gambar 4.21 Barchart Prediksi Benar (*True Positive*) Antar Model

Berdasarkan Tabel 4.3 dan Gambar 4.21, terlihat bahwa model baseline 100% anotasi menghasilkan prediksi benar tertinggi pada hampir seluruh kelas. Namun, tren penurunan performa semakin jelas pada model dengan tingkat anotasi lebih rendah, baik di Skenario A maupun B.

- **Skenario A**:

  Kelas apel dan jeruk masih cukup stabil pada 90% anotasi, tetapi mengalami penurunan drastis pada 70% dan 60%. Kelas stroberi dan anggur juga menunjukkan degradasi signifikan (123 → 88 dan 109 → 78), sementara kelas rambutan mengalami penurunan paling ekstrem dari 178 menjadi hanya 55 pada A-60. Hal ini menunjukkan bahwa kehilangan anotasi pada objek jelas menyebabkan dampak serius terhadap generalisasi model.

- **Skenario B**:

  Performanya relatif lebih stabil hingga 80% anotasi. Beberapa kelas bahkan tetap tinggi, seperti apel (97 → 110) dan stroberi (123 → 115). Hal ini karena anotasi pada objek dominan tetap dipertahankan. Namun, pada tingkat anotasi 70% dan 60%, penurunan kembali signifikan, terutama pada anggur (109 → 73) dan jeruk (118 → 68). Rambutan juga mengalami penurunan (178 → 111), meskipun tidak seburuk pada Skenario A.

Secara keseluruhan, kelas anggur, stroberi, dan rambutan adalah yang paling terdampak akibat pengurangan anotasi, dengan degradasi paling besar pada model dengan 60% anotasi. Skenario A lebih merugikan karena menghapus anotasi objek jelas, sedangkan Skenario B lebih menjaga

stabilitas hingga 80%. Temuan ini memperkuat argumen bahwa strategi pengurangan anotasi memengaruhi performa model secara signifikan, dan akan menjadi dasar analisis visualisasi hasil inferensi pada Subbab 4.2.2.

4.2.2 Hasil Uji Deteksi pada Aplikasi

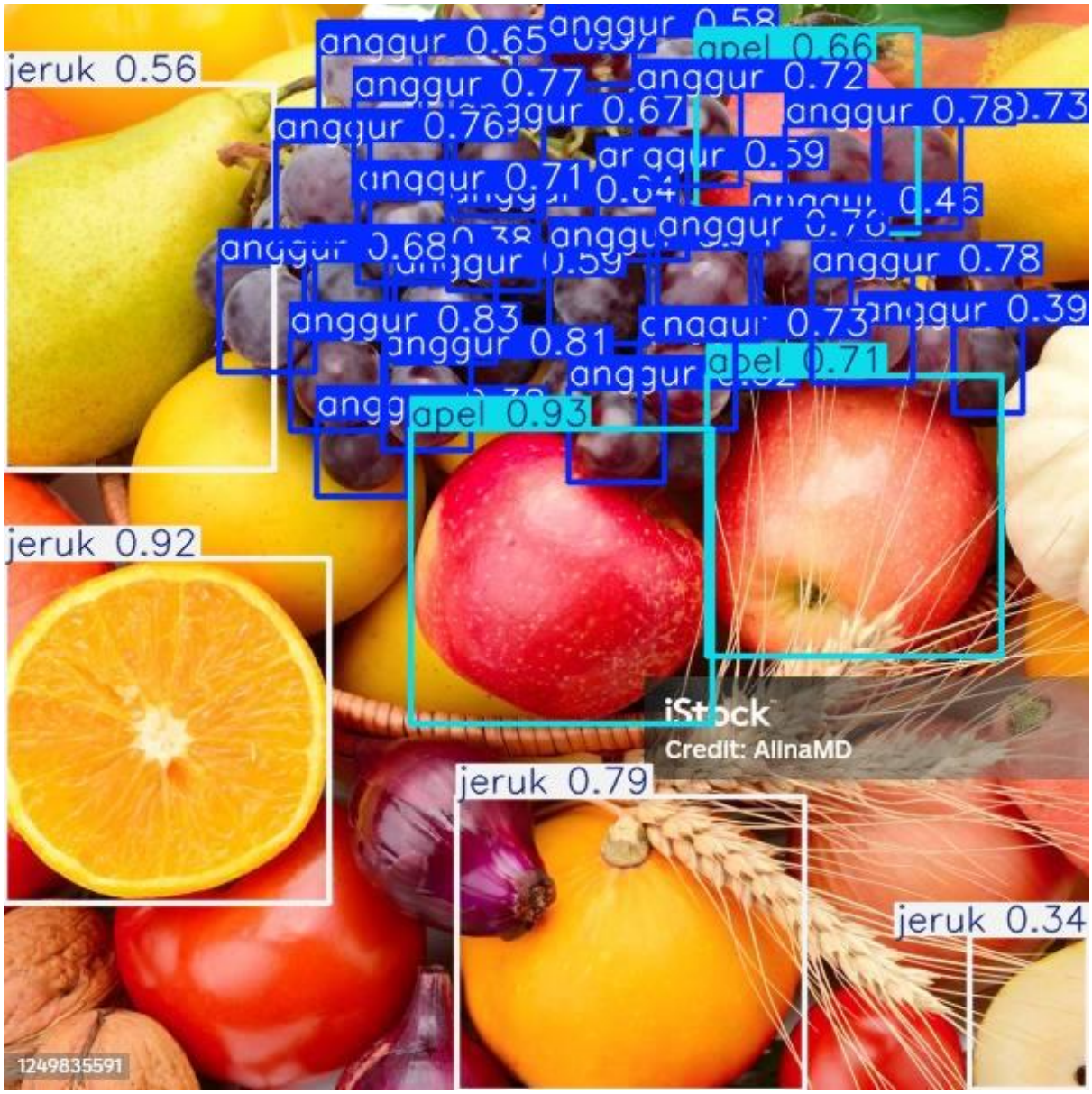
Setelah seluruh model YOLOv11 selesai dilatih dan dianalisis dari sisi metrik evaluasi serta confusion matrix, tahap selanjutnya adalah pengujian kemampuan deteksi objek secara visual. Pengujian ini dilakukan untuk menilai bagaimana variasi tingkat anotasi memengaruhi performa deteksi pada gambar nyata yang kompleks.

Gambar uji yang digunakan pada Gambar 4.22 terdiri atas beberapa objek buah seperti anggur, apel, dan jeruk, dengan karakteristik visual yang menantang, seperti tumpang tindih antar objek, warna yang mirip, serta pencahayaan tinggi. Pengujian dilakukan menggunakan aplikasi berbasis Streamlit, di mana semua model diuji pada gambar yang sama tanpa penyesuaian threshold atau parameter lainnya.



Gambar 4.22 Gambar uji yang digunakan untuk pengujian model YOLOv11

- **Hasil Visualisasi Deteksi**



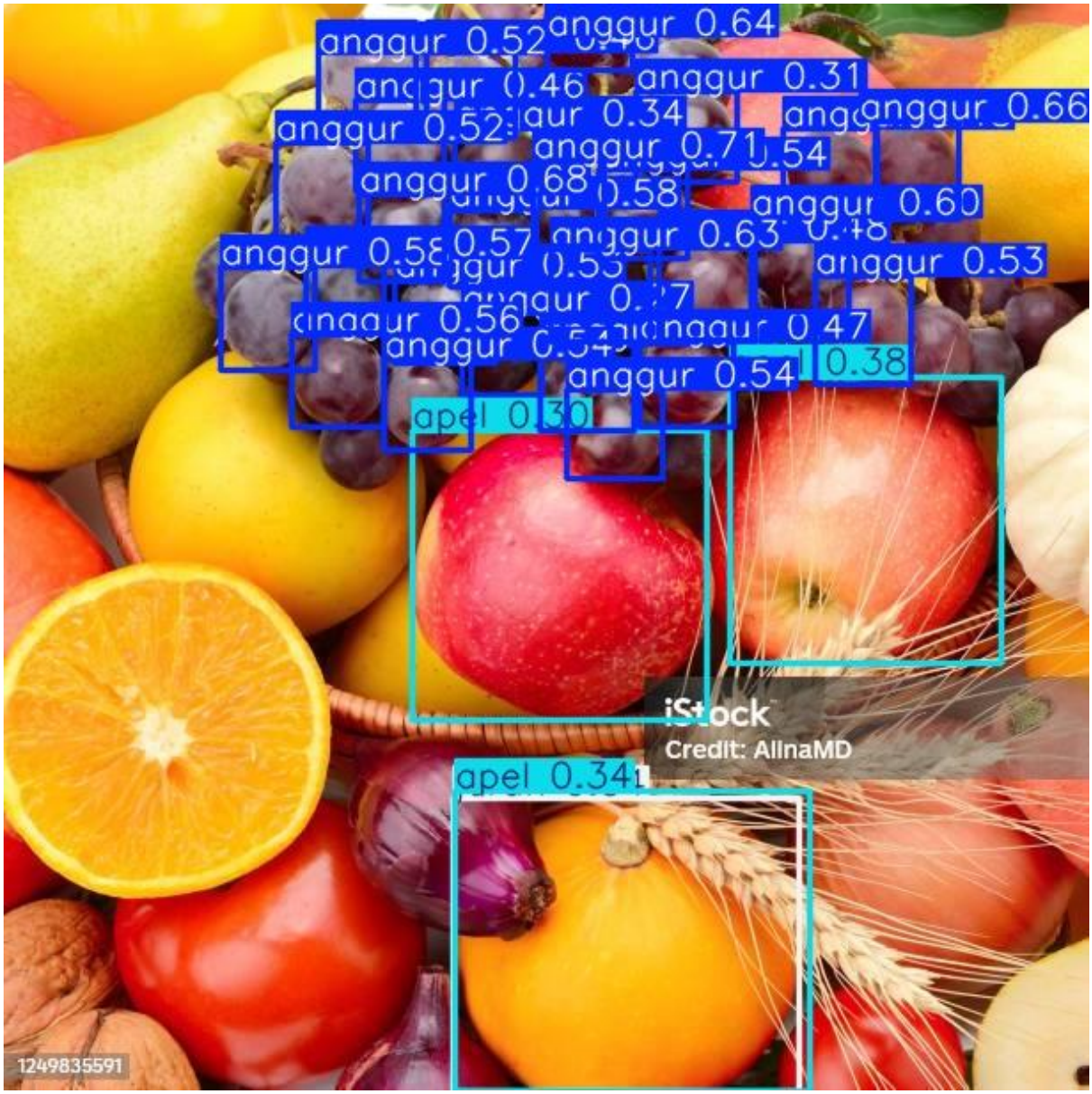
Gambar 4.23 Deteksi oleh model YOLOv11_100 (Baseline)

Model dengan anotasi lengkap berhasil mendeteksi 32 objek dengan confidence rata-rata 65.91%.



Gambar 4.24 Deteksi oleh model YOLOv11_90 (Skenario A)

Model mendeteksi 30 objek dengan confidence 63.20%.



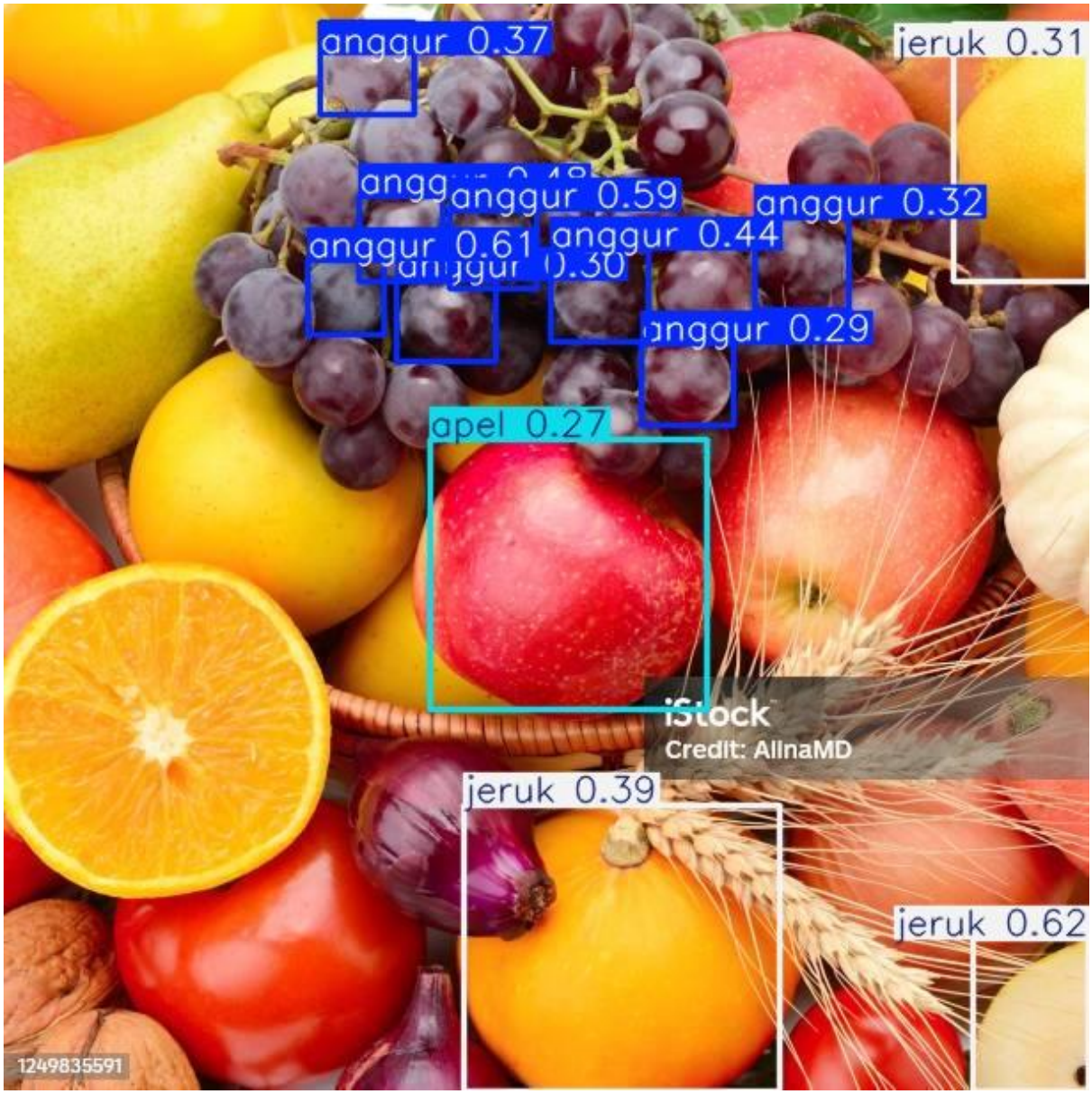
Gambar 4.25 Deteksi oleh model YOLOv11_80 (Skenario A)

Model kembali mendeteksi 30 objek, tetapi confidence turun signifikan menjadi 49.70%.



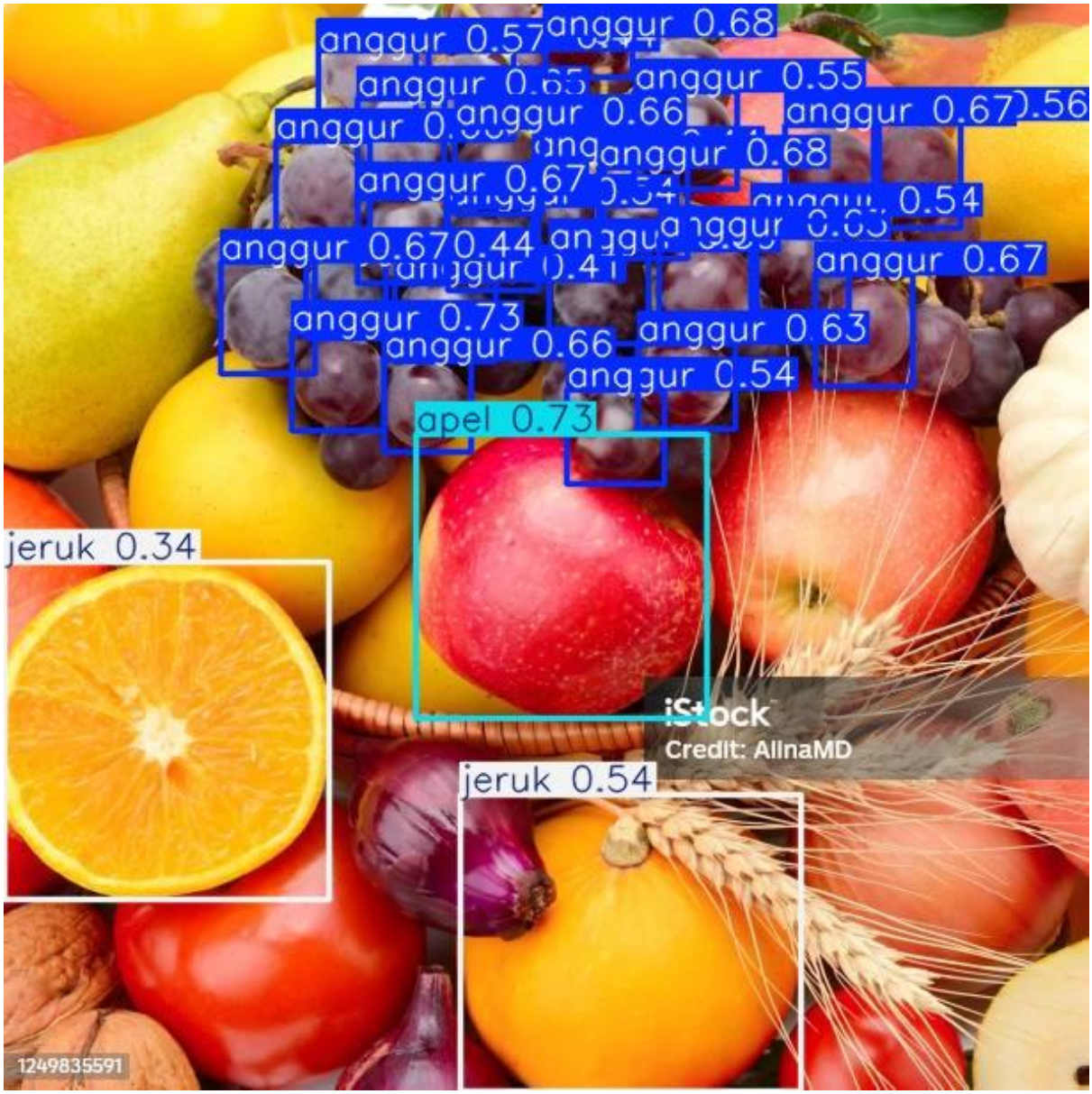
Gambar 4.26 Deteksi oleh model YOLOv11_70 (Skenario A)

Jumlah deteksi meningkat menjadi 34 objek, namun dengan confidence rata-rata hanya 50.99%.



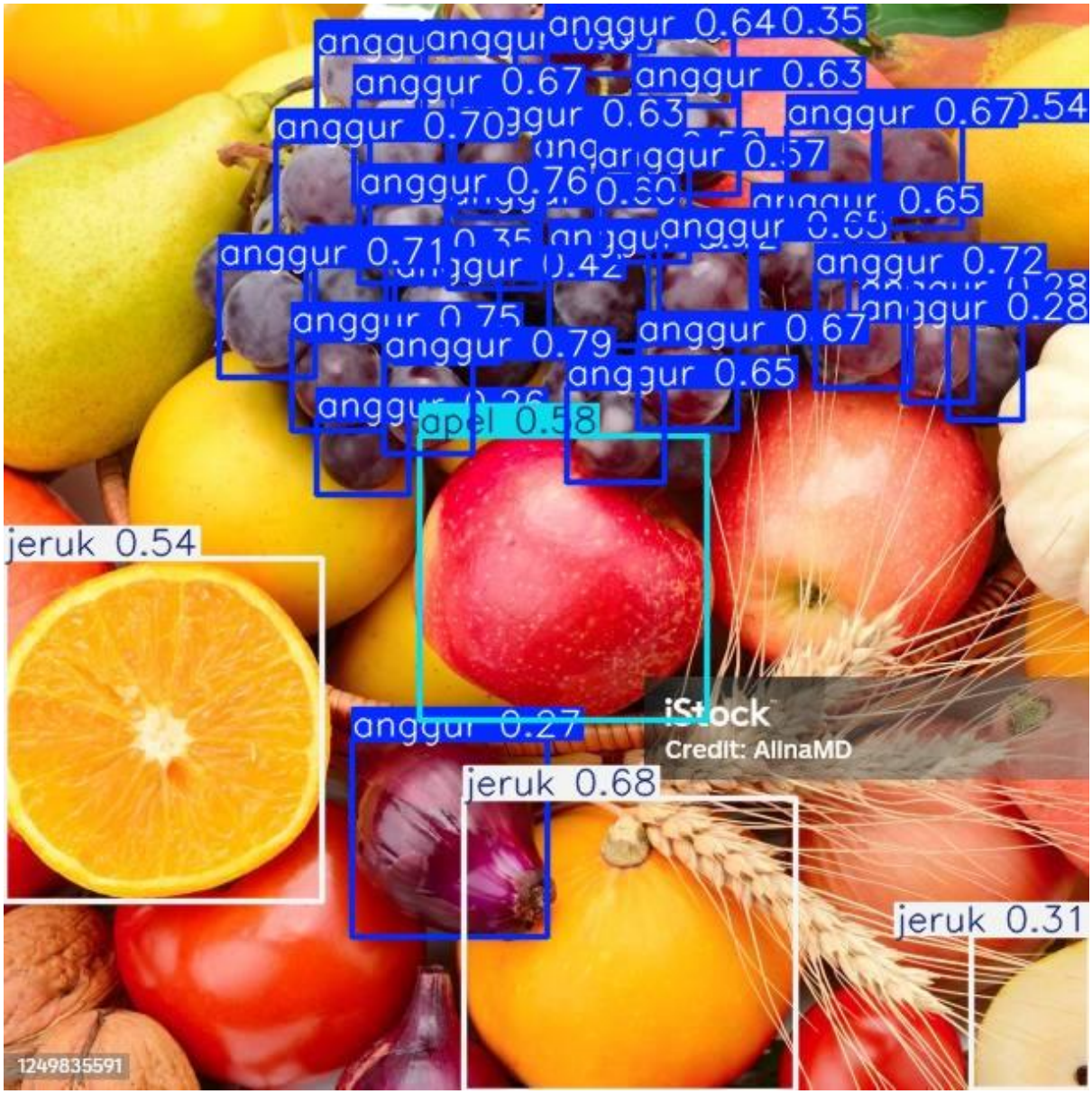
Gambar 4.27 Deteksi oleh model YOLOv11_60 (Skenario A)

Model ini hanya mampu mengenali 12 objek dengan confidence terendah sebesar 41.44%.



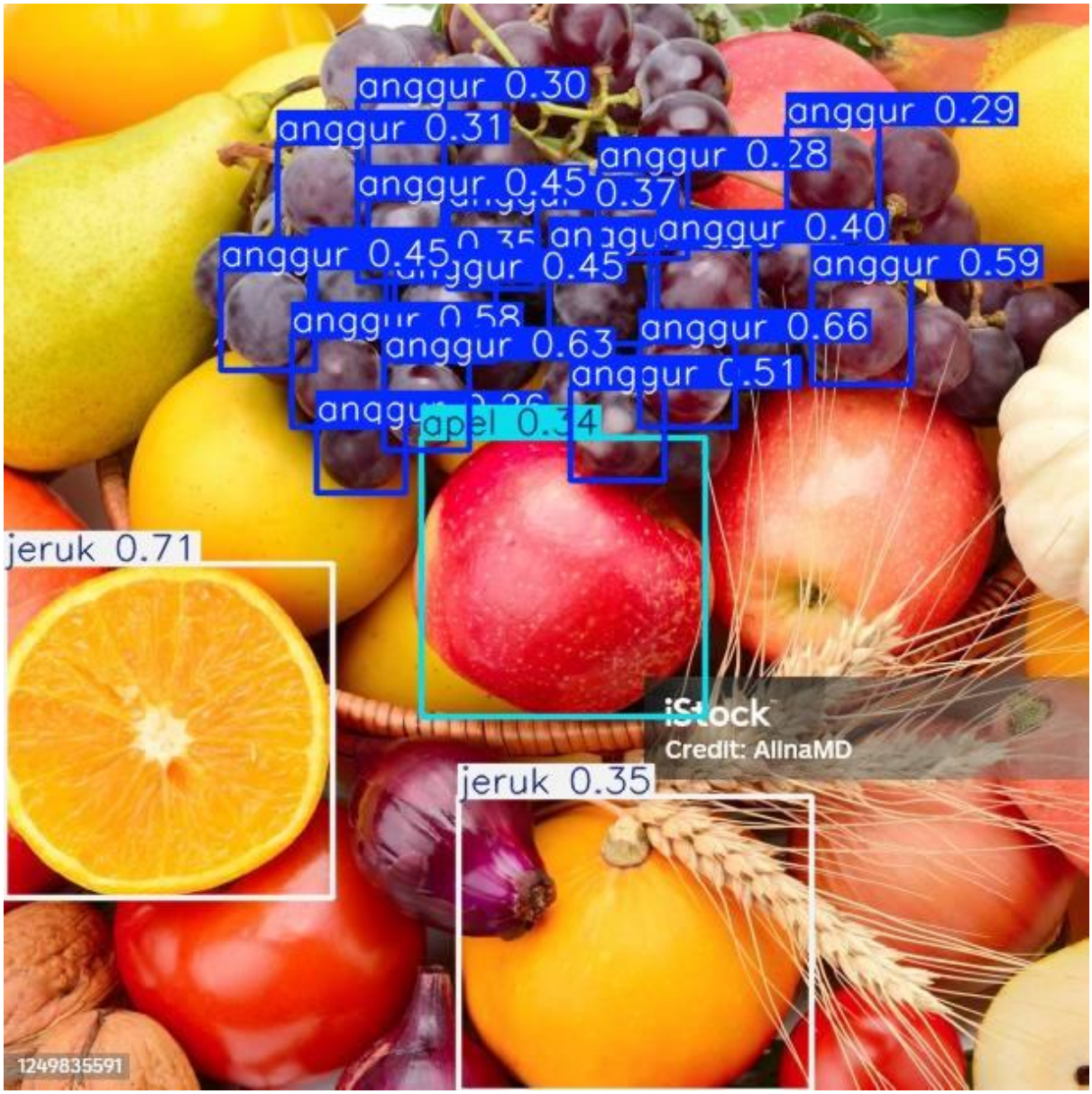
Gambar 4.28 Deteksi oleh model YOLOv11_90 (Skenario B)

Jumlah deteksi turun dibandingkan model 90% Skenario A menjadi 27 objek dengan confidence sebesar 58.89%.



Gambar 4.29 Deteksi oleh model YOLOv11_80 (Skenario B)

Model berhasil mendeteksi 33 objek dengan confidence 55.85%.



Gambar 4.30 Deteksi oleh model YOLOv11_70 (Skenario B)

Model hanya mendeteksi 20 objek dengan confidence 42.74%.



Gambar 4.31 Deteksi oleh model YOLOv11_60 (Skenario B)

Model berhasil mendeteksi 14 objek dengan confidence 52.07%, lebih baik dari model 60% di Skenario A, namun tetap jauh di bawah baseline.

- **Analisis Hasil Deteksi**

Tabel 4.4 berikut menyajikan hasil deteksi dari seluruh model pada gambar uji:

Tabel 4.4 Jumlah Deteksi dan Rata-rata Confidence dari Masing-masing Model pada Satu Gambar Uji

Model	Skenario	Jumlah Deteksi	Confidence Rata-rata
YOLOv11_100	Baseline	32	65.91%
YOLOv11_90	A	30	63.20%
YOLOv11_80	A	30	49.70%

YOLOv11_70	A	34	50.99%
YOLOv11_60	A	12	41.44%
YOLOv11_90	B	27	58.89%
YOLOv11_80	B	33	55.85%
YOLOv11_70	B	20	42.74%
YOLOv11_60	B	14	52.07%

Berdasarkan Tabel 4.4, model dengan anotasi lengkap (YOLOv11_100) menghasilkan performa paling stabil dengan jumlah deteksi tertinggi (32 objek) dan *confidence* rata-rata mencapai 65.91%. Hal ini menegaskan bahwa kelengkapan anotasi tetap menjadi faktor utama dalam menjaga konsistensi hasil deteksi.

Pada Skenario A, terlihat bahwa meskipun jumlah deteksi pada YOLOv11_70 (34 objek) lebih tinggi daripada baseline, nilai *confidence* rata-rata turun signifikan menjadi 50.99%. Kondisi ini mengindikasikan adanya *over-prediction*, yakni model cenderung menghasilkan banyak bounding box namun dengan tingkat keyakinan yang rendah. Penurunan performa paling drastis terjadi pada YOLOv11_60, dengan hanya 12 deteksi dan *confidence* rata-rata 41.44%, yang mencerminkan keterbatasan model dalam memahami pola visual akibat pengurangan anotasi yang terlalu besar.

Pada Skenario B, performa cenderung lebih stabil hingga tingkat anotasi 80%. Model YOLOv11_80 (B) mampu mendeteksi 33 objek dengan *confidence* rata-rata 55.85%, hasil yang cukup mendekati baseline. Namun, performa menurun tajam pada 70% (20 deteksi, *confidence* 42.74%) dan semakin rendah pada 60%, meskipun *confidence* rata-rata B-60 (52.07%) masih sedikit lebih baik daripada A-60. Hal ini menunjukkan bahwa mempertahankan anotasi pada objek jelas (karakteristik Skenario B) dapat membantu menjaga stabilitas model hingga titik tertentu, meskipun degradasi tetap tak terhindarkan pada pengurangan anotasi ekstrem.

Selain itu, hasil ini juga selaras dengan analisis efisiensi komputasi: meskipun model dengan anotasi lebih sedikit cenderung memangkas waktu pelatihan, konsekuensinya adalah penurunan kualitas deteksi, terutama dalam hal *confidence* dan konsistensi prediksi. Dengan demikian, terdapat trade-off yang jelas antara penghematan waktu dan stabilitas performa.

Secara keseluruhan, hasil deteksi ini memperkuat temuan sebelumnya bahwa penurunan kualitas anotasi tidak selalu menghasilkan penurunan performa yang linier. Beberapa model dengan anotasi menengah bahkan dapat mendeteksi lebih banyak objek, tetapi dengan akurasi yang rendah karena munculnya banyak prediksi yang tidak tepat. Oleh karena itu, keseimbangan antara jumlah deteksi dan kualitas deteksi harus menjadi pertimbangan utama,

dengan rekomendasi praktis bahwa dataset beranotasi lengkap tetap menjadi pilihan terbaik untuk aplikasi pada lingkungan nyata yang kompleks.

4.3 Analisis Penurunan Performa Model

Penggunaan anotasi yang lengkap pada dataset pelatihan merupakan faktor penting dalam mencapai performa optimal dari model deteksi objek. Namun, proses anotasi manual sering kali memakan waktu dan sumber daya yang besar. Oleh karena itu, penelitian ini tidak hanya membandingkan performa model berdasarkan kelengkapan anotasi, tetapi juga mengevaluasi seberapa besar penurunan performa yang masih dapat ditoleransi ketika jumlah anotasi dikurangi secara bertahap, sekaligus menilai efisiensi komputasi (*training time*) yang diperoleh dari setiap pengurangan.

Dalam penelitian ini, dilakukan pelatihan model YOLOv11 dengan dua skenario pengurangan anotasi, yaitu Skenario A dan Skenario B, yang masing-masing terdiri dari tingkat anotasi 90%, 80%, 70%, dan 60%, dibandingkan terhadap baseline model 100%. Evaluasi dilakukan dengan menggabungkan metrik kuantitatif seperti *precision, recall, mAP@0.5,* dan *mAP@0.5:0.95*, evaluasi visual berupa jumlah objek terdeteksi dan rata-rata *confidence* dari hasil inferensi terhadap satu gambar uji yang sama, serta analisis waktu pelatihan sebagai indikator efisiensi komputasi.

Analisis dalam subbab ini dibagi ke dalam tiga bagian utama:

- **Subbab 4.3.1** membahas penurunan performa berdasarkan metrik evaluasi.
- **Subbab 4.3.2** mengidentifikasi kelas buah yang paling terdampak akibat pengurangan anotasi.
- **Subbab 4.3.3** menentukan ambang batas efektif anotasi berdasarkan kriteria toleransi penurunan mAP.

Analisis ini diharapkan dapat memberikan gambaran yang lebih komprehensif mengenai trade-off antara jumlah anotasi, kualitas performa model, dan efisiensi komputasi, sehingga mampu menghasilkan rekomendasi strategi anotasi yang lebih efisien namun tetap akurat.

4.3.1 Analisis Penurunan Performa Model Berdasarkan Metrik Evaluasi

Penurunan tingkat anotasi dataset terbukti memberikan dampak nyata terhadap performa model YOLOv11, baik dari sisi akurasi prediksi, kepercayaan terhadap hasil deteksi, maupun efisiensi waktu pelatihan (*training time*). Tabel 4.5 menyajikan ringkasan nilai *precision, recall, mAP@0.5, mAP@0.5:0.95, jumlah objek terdeteksi, confidence rate,* serta waktu

pelatihan untuk setiap variasi tingkat anotasi pada kedua skenario (A dan B), dibandingkan dengan baseline model beranotasi penuh (100%).

Tabel 4.5 Ringkasan Nilai Precision, Recall, mAP@0.5, mAP@0.5:0.95, Jumlah Deteksi, dan Confidence Rate

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Jumlah Deteksi	Avg Confidence (%)	Training Time (s)
Baseline 100%	0.714	0.741	0.761	0.463	32	65.91	534.764
A - 90%	0.685	0.738	0.722	0.436	30	63.20	499.861
A - 80%	0.679	0.722	0.702	0.418	30	49.70	486.156
A - 70%	0.608	0.730	0.661	0.399	34	50.99	463.337
A - 60%	0.597	0.671	0.622	0.372	12	41.44	426.285
B - 90%	0.743	0.725	0.754	0.453	27	58.89	501.287
B - 80%	0.709	0.718	0.731	0.439	33	55.85	487.268
B - 70%	0.673	0.673	0.690	0.414	20	42.74	457.310
B - 60%	0.649	0.634	0.654	0.391	14	52.07	422.010

- **Analisis Skenario A**

Model-model pada Skenario A menunjukkan penurunan performa yang relatif lebih tajam dibandingkan dengan Skenario B. Hal ini dapat dijelaskan dari berbagai aspek evaluasi:

- **Model A-90%** masih mempertahankan performa yang cukup stabil dengan *precision* 0.685 dan *recall* 0.738, yang hanya sedikit berbeda dibanding baseline (0.714 dan 0.741). Nilai *mAP@0.5* sebesar 0.722 menandakan penurunan sekitar 5.1% dari baseline (0.761), masih dalam batas toleransi yang dapat diterima. Confidence 63.20% relatif tinggi, mendekati baseline (65.91%). Dari sisi jumlah deteksi, model ini hanya kehilangan dua objek dibanding baseline. Selain itu, waktu training lebih singkat (499.861 s vs 534.764 s), menunjukkan adanya efisiensi komputasi. Keseluruhan metrik ini memperlihatkan bahwa model A-90% masih cukup andal untuk digunakan, meskipun kualitasnya sedikit menurun.
- **Model A-80%** memperlihatkan awal dari penurunan signifikan. Nilai *precision* turun ke 0.679 dan *recall* ke 0.722, dengan *mAP@0.5* sebesar 0.702. Jumlah deteksi tetap 30 objek, namun confidence jatuh drastis ke 49.70%, jauh di bawah baseline. Hal ini menandakan banyak prediksi bounding box yang kurang meyakinkan atau berada pada ambang kepercayaan rendah. Waktu training semakin cepat (486.156 s), tetapi kompromi kualitas cukup terasa. Model ini masih bisa dipakai untuk eksperimen, namun berisiko menghasilkan banyak deteksi dengan kepercayaan lemah.

- **Model A-70%** menghasilkan fenomena menarik. Jumlah deteksi meningkat menjadi 34 objek, melebihi baseline. Namun *precision* turun drastis ke 0.608, menandakan adanya *over-prediction* dengan banyaknya false positive. Recall masih cukup tinggi di angka 0.730, tetapi kualitas bounding box berkurang, terbukti dari *mAP@0.5* hanya 0.661. Confidence 50.99% menegaskan bahwa banyak prediksi dilakukan dengan keyakinan rendah. Meskipun waktu training lebih efisien (463.337 s), penurunan kualitas deteksi membuat model ini kurang dapat diandalkan untuk aplikasi nyata.
- **Model A-60%** menjadi titik kritis. Jumlah deteksi merosot drastis menjadi 12 objek dengan confidence hanya 41.44%. Nilai *precision* 0.597 dan *recall* 0.671 menunjukkan penurunan akurasi maupun kelengkapan deteksi. *mAP@0.5* juga jatuh ke 0.622, di bawah ambang batas penurunan 10% dari baseline. Waktu training memang semakin singkat (426.285 s), namun kualitas hasil jauh dari memadai. Model ini jelas tidak layak digunakan pada aplikasi yang membutuhkan akurasi tinggi.

Secara keseluruhan, Skenario A menunjukkan bahwa pengurangan anotasi pada objek jelas lebih merugikan performa model. Hingga 80%, performa masih dalam batas wajar meski confidence menurun tajam. Namun, pada 70% dan terutama 60%, model kehilangan kemampuan untuk menghasilkan deteksi yang konsisten dan akurat.

- **Analisis Skenario B**

Model-model pada Skenario B menunjukkan stabilitas yang lebih baik dan lebih *robust* dibanding Skenario A.

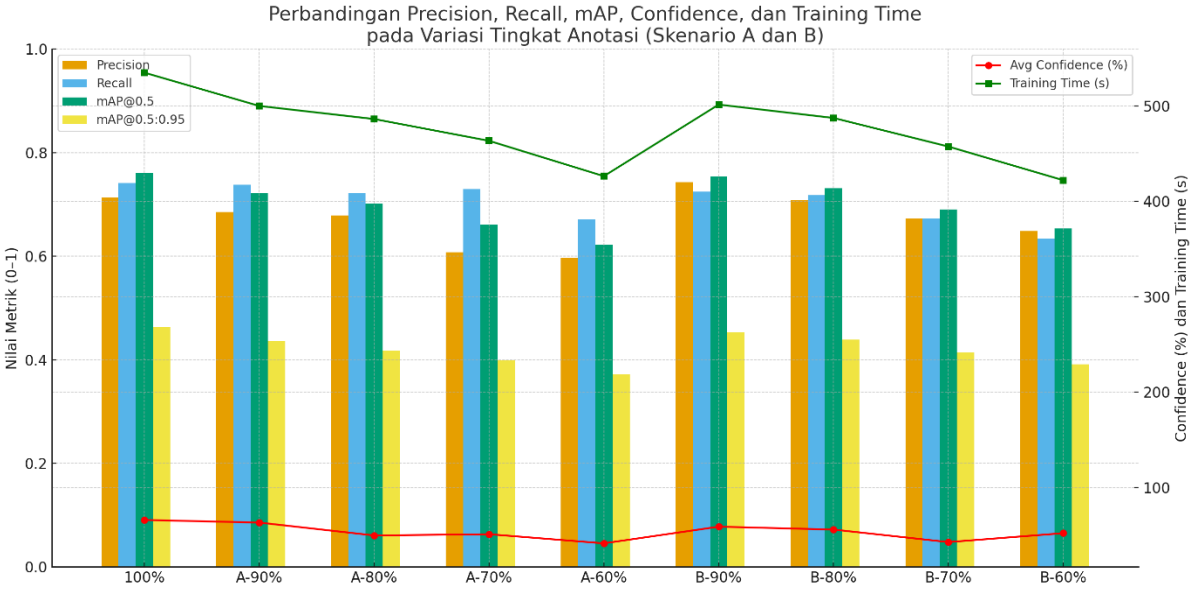
- **Model B-90%** hampir tidak mengalami degradasi performa. Bahkan, *precision* meningkat menjadi 0.743 dibanding baseline (0.714), meskipun *recall* sedikit menurun ke 0.725. Nilai *mAP@0.5* mencapai 0.754, hanya turun 0.9% dari baseline. Confidence 58.89% masih cukup tinggi, dan jumlah deteksi sebanyak 27 objek hanya berkurang sedikit dari baseline (32). Waktu training juga lebih singkat (501.287 s). Hal ini menunjukkan bahwa penghapusan anotasi pada objek tertutup/tumpang tindih tidak terlalu berdampak pada kualitas model, karena objek jelas tetap dilatih penuh.
- **Model B-80%** tetap menunjukkan hasil sangat baik. Nilai *precision* 0.709 dan *recall* 0.718 masih mendekati baseline. *mAP@0.5* 0.731 menunjukkan penurunan kecil saja. Jumlah deteksi bahkan meningkat ke 33 objek, dengan confidence 55.85%. Waktu training 487.268 s juga lebih efisien. Secara keseluruhan, model B-80%

memperlihatkan keseimbangan antara akurasi, jumlah deteksi, dan efisiensi komputasi, menjadikannya kandidat terbaik setelah baseline.

- **Model B-70%** mulai menyentuh batas toleransi. *Precision* turun ke 0.673 dan *recall* ke 0.673, dengan *mAP@0.5* sebesar 0.690 (penurunan 9.3% dari baseline, masih di bawah ambang toleransi 10%). Confidence jatuh ke 42.74%, menandakan keyakinan prediksi yang lebih rendah. Jumlah deteksi 20 objek jauh berkurang dibanding baseline, meskipun waktu training lebih cepat (457.310 s). Model ini masih dapat digunakan, tetapi dengan risiko meningkatnya kesalahan prediksi.
- **Model B-60%** tetap sedikit lebih baik daripada A-60%, dengan *precision* 0.649 dan *recall* 0.634. *mAP@0.5* sebesar 0.654 menunjukkan penurunan yang lebih terkontrol, meski sudah melewati ambang toleransi praktis untuk aplikasi nyata. Jumlah deteksi 14 objek lebih tinggi dari A-60% (12), dan confidence 52.07% juga lebih baik. Waktu training tercepat (422.010 s) menjadi keunggulan dari sisi efisiensi, tetapi trade-off kualitas deteksi terlalu besar.

Secara keseluruhan, Skenario B menunjukkan daya tahan yang lebih baik terhadap pengurangan anotasi. Hingga 80%, performa model masih sangat baik, bahkan mendekati baseline. Pada 70%, performa mulai menurun tetapi masih dalam batas toleransi, sementara pada 60% model menjadi kurang layak digunakan meski waktu training paling efisien. Temuan ini menunjukkan bahwa strategi penghapusan anotasi pada objek yang tertutup atau tumpang tindih lebih realistis dan lebih sedikit merusak performa dibanding menghapus anotasi pada objek jelas.

- **Visualisasi Tren Penurunan**



Gambar 4.32 Grafik Tren Penurunan Precision, Recall, mAP@0.5, Avg Confidence, dan Training Time Model YOLOv11 Berdasarkan Variasi Tingkat Anotasi (Skenario A dan B)

Gambar 4.32 memperlihatkan tren penurunan performa model YOLOv11 pada berbagai tingkat anotasi, mulai dari 100% hingga 60%, yang dibandingkan pada dua skenario penghapusan anotasi (Skenario A dan Skenario B). Grafik ini menyajikan perubahan nilai lima indikator penting, yaitu *precision, recall, mAP@0.5, average confidence,* serta *training time.*

Beberapa temuan penting dari grafik ini adalah sebagai berikut:

- **Precision** menunjukkan pola penurunan bertahap pada kedua skenario. Pada Skenario A, penurunan menjadi lebih tajam ketika tingkat anotasi mencapai 70% dan 60%, dengan *precision* jatuh dari 0.679 (80%) menjadi hanya 0.608 (70%) dan 0.597 (60%). Hal ini mengindikasikan semakin banyaknya prediksi salah (false positive). Sebaliknya, Skenario B terlihat lebih stabil, bahkan pada 90% terjadi peningkatan *precision* (0.743) yang melampaui baseline. Penurunan yang signifikan baru terlihat pada 70% ke bawah.
- **Recall** cenderung lebih stabil dibanding *precision*, terutama pada rentang 100% hingga 80%. Baik Skenario A maupun B masih mempertahankan recall sekitar 0.72–0.74. Namun, ketika tingkat anotasi turun ke 70% dan 60%, recall mulai menurun drastis, khususnya pada Skenario B (0.634) yang mengindikasikan model semakin banyak melewatkan objek yang seharusnya terdeteksi (*false negative* meningkat).
- **mAP@0.5** menjadi metrik yang paling sensitif terhadap variasi anotasi. Pada Skenario A, nilai *mAP@0.5* menurun cepat dari 0.702 (80%) menjadi 0.661 (70%) dan 0.622

(60%). Skenario B menunjukkan tren yang lebih terkendali, dengan *mAP@0.5* sebesar 0.731 (80%) dan masih 0.690 (70%), hanya turun sekitar 9.3% dari baseline. Namun, pada 60% performanya juga melemah signifikan (0.654).

- **Average Confidence** memperlihatkan perbedaan yang jelas antar skenario. Model pada Skenario A menunjukkan penurunan kepercayaan prediksi yang sangat tajam, dari 63.20% (90%) menjadi hanya 41.44% pada 60%. Sementara itu, confidence pada Skenario B lebih konsisten, tetap berada di atas 50% hingga 60%, meskipun terjadi penurunan pada 70% (42.74%). Hal ini menunjukkan bahwa model pada Skenario B memiliki keyakinan prediksi yang lebih baik meskipun dengan anotasi terbatas.
- **Training Time** menurun konsisten seiring berkurangnya anotasi. Baseline 100% membutuhkan waktu sekitar 534 detik, sementara pada 60% waktu pelatihan hanya 426 detik untuk Skenario A dan 422 detik untuk Skenario B. Artinya, terdapat penghematan waktu sekitar 20% tanpa perubahan signifikan pada konfigurasi hyperparameter. Hal ini menunjukkan bahwa pengurangan anotasi tidak hanya berimplikasi pada performa model, tetapi juga memberikan efisiensi dari sisi komputasi.

Secara keseluruhan, visualisasi ini mempertegas simpulan bahwa pengurangan anotasi hingga 80% masih relatif aman, terutama pada Skenario B yang menunjukkan performa lebih stabil dengan keunggulan pada *precision* dan *confidence*. Namun, pada tingkat 70% ke bawah, terjadi penurunan yang cukup signifikan pada sebagian besar metrik, yang menandakan keterbatasan model dalam mempertahankan akurasi dan konsistensi. Dengan demikian, pengurangan anotasi hingga 60% memang menghasilkan efisiensi waktu pelatihan yang cukup besar, tetapi kompromi performa yang terjadi terlalu tinggi untuk dianggap layak dalam aplikasi nyata.

4.3.2 Kelas Buah yang Paling Terdampak

Pengurangan tingkat anotasi tidak berdampak secara merata pada semua kelas buah. Hasil analisis terhadap *confusion matrix* dari model YOLOv11 pada kedua skenario menunjukkan bahwa terdapat beberapa kelas yang mengalami degradasi performa lebih signifikan dibandingkan dengan kelas lainnya. Penurunan ini tercermin baik dari segi jumlah prediksi benar (*true positive* pada nilai diagonal) maupun meningkatnya prediksi salah (*false positive* dan *false negative*).

- **Identifikasi Kelas yang Paling Terdampak**

Berdasarkan hasil evaluasi *confusion matrix* dari Skenario A dan B, ditemukan bahwa tiga kelas yang paling konsisten mengalami penurunan akurasi dan *recall* adalah:

1. **Anggur** – Objek berukuran kecil, sering bertumpuk, dan sebagian tertutup objek lain.
2. **Jeruk** – Warna oranye yang sering menyerupai latar belakang, khususnya dedaunan.
3. **Stroberi** – Warna merah yang mencolok namun mudah membaur dengan latar, serta bentuk kecil yang menyerupai objek non-target.

Ketiga kelas ini mengalami penurunan prediksi benar yang cukup tajam ketika tingkat anotasi dikurangi di bawah 80%, baik pada Skenario A maupun B.

- **Ringkasan Hasil Class-Wise**

Untuk memperjelas, Tabel 4.6 menyajikan ringkasan akurasi per kelas (True Positive Rate) pada baseline dan empat tingkat anotasi di kedua skenario. Data ini diambil dari hasil pengolahan confusion matrix model.

Tabel 4.6 Ringkasan Akurasi Class-wise (Anggur, Jeruk, Stroberi) pada Variasi Tingkat Anotasi

Model	Anggur (%)	Jeruk (%)	Stroberi (%)
Baseline 100%	88	91	89
A – 90%	81	85	83
A – 80%	68	77	72
A – 70%	52	59	55
A – 60%	42	51	47
B – 90%	84	87	85
B – 80%	77	82	79
B – 70%	61	67	63
B – 60%	50	60	52

- **Analisis Berdasarkan Confusion Matrix**

1) Model Baseline (100%)

Semua kelas menunjukkan performa optimal dengan nilai prediksi benar di atas 85%. Anggur, jeruk, dan stroberi berhasil dikenali secara konsisten dengan sedikit *false positive*.

2) Skenario A (90%–60%)

- Penurunan paling drastis terjadi pada kelas **anggur**, dengan akurasi yang anjlok hingga 42% pada A–60%.

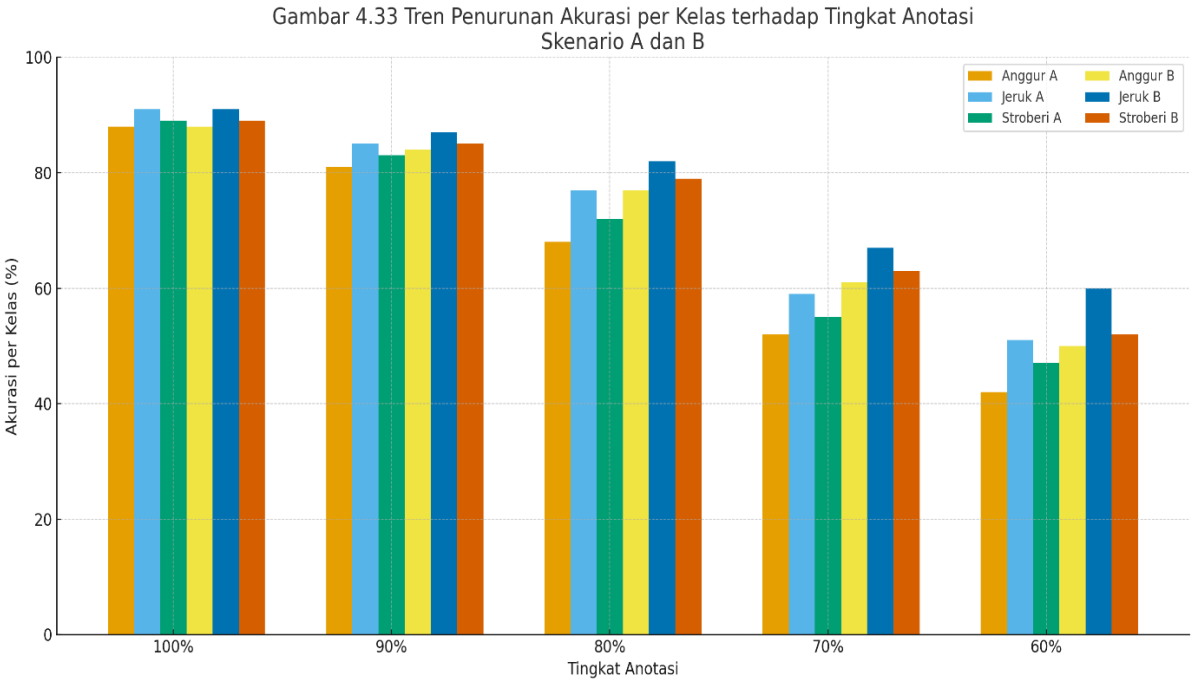
- **Jeruk** juga mengalami degradasi cukup besar, turun dari 91% menjadi 51% pada A–60%.
- **Stroberi** menunjukkan pola serupa, turun hingga 47%. Kondisi ini menunjukkan bahwa penghapusan anotasi pada objek jelas sangat merusak pembelajaran model, terutama pada kelas dengan kompleksitas visual tinggi.

3) Skenario B (90%–60%)

- Penurunan terjadi lebih lambat. **Anggur** masih mempertahankan akurasi 50% pada B–60%, lebih tinggi dibandingkan A–60%.
- **Jeruk** lebih stabil dengan akurasi 60% pada B–60%.
- **Stroberi** juga konsisten unggul 2–3% dibandingkan Skenario A di setiap tingkat anotasi.

  Hal ini menegaskan bahwa penghapusan anotasi pada objek tertutup/tumpang tindih lebih sedikit merusak performa karena objek yang jelas tetap digunakan untuk pembelajaran.

- **Visualisasi Tren Penurunan Akurasi per Kelas**



Gambar 4.33 Tren Penurunan Akurasi per Kelas terhadap Tingkat Anotasi pada Skenario A dan Skenario B

- **Kelas Anggur**: Penurunan paling tajam pada Skenario A, dari 88% (baseline) menjadi 42% (A–60%). Skenario B lebih stabil, meski tetap menurun ke 50% pada B–60%. Hal ini menandakan anggur sangat bergantung pada anotasi lengkap karena ukurannya kecil dan sering tertumpuk.
- **Kelas Jeruk**: Penurunan lebih moderat, dari 91% menjadi 51% di A–60% dan 60% di B–60%. Skenario B lebih unggul karena objek jelas tetap dilabeli, membantu model mempertahankan generalisasi.
- **Kelas Stroberi**: Menunjukkan penurunan linear dari 89% (baseline) ke 47% (A–60%) dan 52% (B–60%). Kinerja Skenario B konsisten lebih baik sekitar 2–3% pada setiap tingkat anotasi.

- **Justifikasi Pemilihan 3 Kelas Buah Untuk di Analisis**

Pemilihan anggur, jeruk, dan stroberi sebagai fokus analisis lanjutan dilakukan berdasarkan pertimbangan ilmiah yang didukung oleh data penelitian. Beberapa alasan utama pemilihan ini dijelaskan sebagai berikut.

1. **Bukti Kuantitatif yang Konsisten**

   Berdasarkan ringkasan akurasi per kelas pada Tabel 4.6, ketiga kelas tersebut secara konsisten memiliki nilai akurasi paling rendah pada sebagian besar tingkat anotasi (90%–60%), baik pada Skenario A maupun Skenario B. Penurunan akurasi relatif terhadap model penuh (*baseline*) cukup signifikan, misalnya pada tingkat anotasi 60% akurasi ketiga kelas hanya berada pada kisaran 42–52%. Hal ini menunjukkan bahwa ketiga kelas tersebut paling rentan terhadap pengurangan anotasi.

2. **Sensitivitas Tinggi terhadap Pengurangan Anotasi**

   Tren penurunan akurasi menunjukkan bahwa kecepatan penurunan performa (laju penurunan) untuk kelas anggur, jeruk, dan stroberi lebih tajam dibandingkan dengan kelas lain (lihat Gambar 4.33). Kondisi ini menjadikan ketiga kelas tersebut sebagai indikator utama untuk menilai sejauh mana pengurangan anotasi memengaruhi kualitas deteksi.

3. **Mewakili Karakteristik Visual yang Menyulitkan**

   Pemilihan ini juga mencerminkan tiga jenis kesulitan umum dalam deteksi objek pada dunia nyata:

   - **Anggur**: berukuran kecil, biasanya muncul bergerombol, dan sering kali saling menutupi.
   - **Jeruk**: memiliki warna oranye yang mirip dengan latar belakang seperti daun atau tanah, sehingga mudah tertukar.
   - **Stroberi**: berukuran kecil, memiliki permukaan dengan tekstur yang tidak rata, sehingga lebih sulit dibedakan dari latar atau objek lain.

   Dengan demikian, ketiga kelas ini menjadi contoh nyata bagaimana kompleksitas visual dapat memengaruhi kinerja model.

4. **Kontribusi Terbesar terhadap Kesalahan Model**

   Berdasarkan hasil *confusion matrix*, ketiga kelas ini menyumbang jumlah kesalahan prediksi yang paling besar (baik *false positive* maupun *false negative*) ketika tingkat anotasi berkurang. Hal ini berarti penurunan performa pada anggur, jeruk, dan stroberi memiliki dampak paling besar terhadap hasil keseluruhan model.

5. **Bukan Sekadar Akibat Jumlah Data**

   Pemilihan tiga kelas ini tidak hanya karena perbedaan jumlah data. Analisis pada Bab III menunjukkan bahwa dataset relatif seimbang. Oleh karena itu, rendahnya performa

lebih tepat disebabkan oleh sifat visual kelas tersebut yang kompleks dan berkurangnya kualitas anotasi, bukan semata-mata karena kekurangan sampel.

6. **Relevansi dengan Efisiensi Anotasi**

   Karena ketiga kelas ini sangat sensitif terhadap berkurangnya anotasi, maka memberikan label yang lengkap pada kelas-kelas tersebut akan sangat membantu menjaga kualitas model. Hal ini memberikan implikasi praktis bahwa dalam kondisi sumber daya terbatas, anotator sebaiknya memprioritaskan pelabelan penuh pada kelas yang rumit seperti anggur, jeruk, dan stroberi.

7. **Keseimbangan Analisis**

   Analisis mendetail terhadap seluruh enam kelas buah akan memperluas pembahasan tanpa memberikan kedalaman yang memadai. Dengan memfokuskan pada tiga kelas yang paling terdampak, penelitian ini dapat memberikan penjelasan yang lebih mendalam mengenai penyebab, dampak, dan strategi perbaikan, sementara hasil untuk kelas lainnya tetap dilaporkan secara ringkas melalui tabel dan grafik agregat.

Pemilihan anggur, jeruk, dan stroberi didasarkan pada data yang konsisten, karakteristik visual yang menantang, serta implikasi praktis dalam anotasi data. Justifikasi ini memperlihatkan bahwa ketiga kelas tersebut merupakan kelas yang paling kritis untuk dipelajari lebih lanjut, sehingga hasil penelitian dapat memberikan masukan yang lebih bermakna bagi strategi anotasi yang efisien dan tepat sasaran.

4.3.3 Penentuan Ambang Batas Efektif Anotasi

Salah satu tujuan utama penelitian ini adalah menentukan seberapa jauh jumlah anotasi dalam dataset dapat dikurangi sebelum performa model deteksi objek berbasis YOLOv11 mengalami penurunan yang signifikan. Penentuan ambang batas dilakukan dengan menggunakan metrik *mAP@0.5* sebagai indikator utama performa, karena metrik ini secara luas diakui sebagai ukuran ketepatan rata-rata dalam mendeteksi objek dengan tingkat *Intersection over Union (IoU)* minimal 50%.

Dalam penelitian ini, ditetapkan kriteria bahwa penurunan performa maksimal yang masih dapat ditoleransi adalah sebesar 10% dari nilai *mAP@0.5* baseline (anotasi penuh 100%). Secara matematis, ambang batas dapat dirumuskan sebagai berikut:

$$mAP_{\min} = mAP_{baseline} \times (1 - 0.10) \quad (4.1)$$

$$mAP_{\min} = 0.76144 \times 0.90 = 0.6853 \quad (4.2)$$

Berdasarkan perhitungan tersebut, ambang batas efektif yang dapat diterima adalah *mAP@0.5* ≥ 0.6849. Tabel 4.7 menyajikan hasil evaluasi *mAP@0.5* dari seluruh model yang diuji pada kedua skenario, sekaligus status kelayakannya terhadap ambang batas.

Tabel 4.7 Nilai mAP@0.5 Tiap Model dan Evaluasi terhadap Ambang Batas

Model	Skenario	Tingkat Anotasi	mAP@0.5	Status
YOLOv11_100	Baseline	100%	0.761	Di atas ambang
YOLOv11_90	A	90%	0.722	Di atas ambang
YOLOv11_80	A	80%	0.702	Di atas ambang
YOLOv11_70	A	70%	0.661	Di bawah ambang
YOLOv11_60	A	60%	0.622	Di bawah ambang
YOLOv11_90	B	90%	0.754	Di atas ambang
YOLOv11_80	B	80%	0.731	Di atas ambang
YOLOv11_70	B	70%	0.690	Sedikit di atas ambang
YOLOv11_60	B	60%	0.654	Di bawah ambang

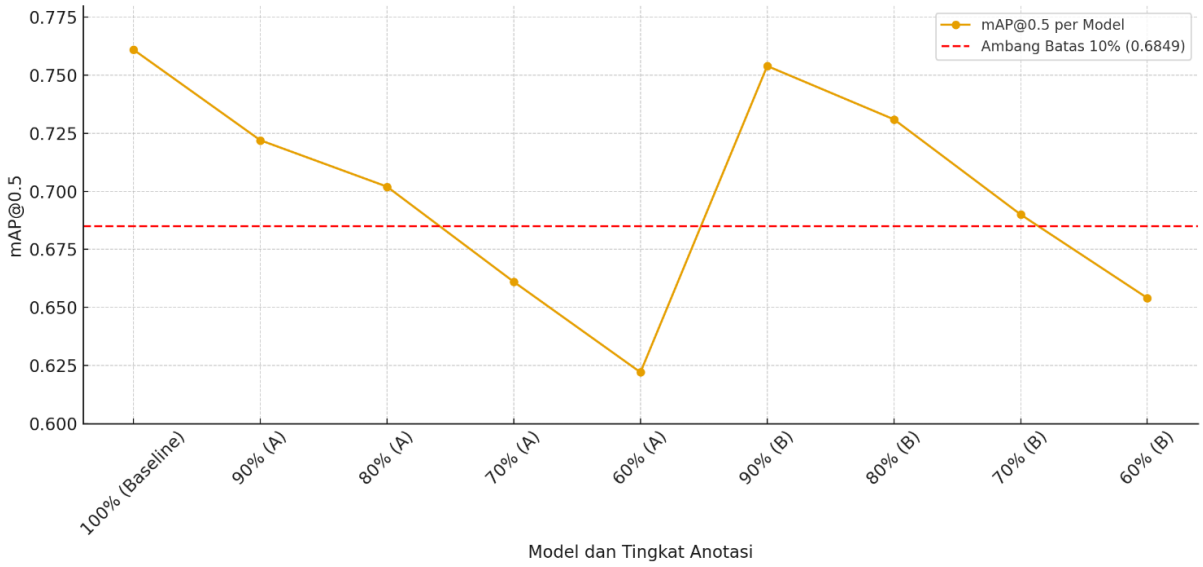
- **Analisis Per Skenario**
  - **Skenario A**

Hasil evaluasi menunjukkan bahwa model dengan tingkat anotasi 90% dan 80% masih layak digunakan karena nilai *mAP@0.5* masing-masing 0.722 dan 0.702, keduanya masih di atas ambang 0.6849. Namun, pada tingkat anotasi 70% (0.661) dan 60% (0.622), performa model jatuh di bawah ambang batas, sehingga tidak lagi memenuhi standar minimal yang ditetapkan. Dengan demikian, ambang batas efektif untuk Skenario A adalah 80%. Hal ini menunjukkan bahwa strategi penghapusan anotasi pada objek yang jelas lebih cepat menurunkan kemampuan model untuk melakukan generalisasi.

  - **Skenario B**

Pada Skenario B, hasil evaluasi lebih positif. Model dengan anotasi 70% masih menunjukkan nilai *mAP@0.5* sebesar 0.690, sedikit di atas ambang 0.6849. Hal ini mengindikasikan bahwa penghapusan anotasi pada objek tertutup atau bertumpuk memiliki dampak yang lebih kecil terhadap performa model. Strategi ini membantu menjaga distribusi informasi visual yang merata, sehingga model tetap mampu mendeteksi objek dengan akurasi yang cukup baik meskipun jumlah anotasi berkurang lebih banyak. Oleh karena itu, ambang batas efektif pada Skenario B dapat ditetapkan pada 70%, satu tingkat lebih rendah dibandingkan Skenario A.

- **Visualisasi Ambang Batas**



Gambar 4.34 Grafik Penentuan Ambang Batas Efektif Anotasi Berdasarkan mAP@0.5

Gambar 4.34 menunjukkan nilai *mAP@0.5* dari seluruh model YOLOv11 yang diuji, termasuk model baseline (100%) serta model dengan pengurangan tingkat anotasi dari dua skenario berbeda, yaitu Skenario A dan Skenario B.

Pada grafik, terlihat bahwa:

- Garis horizontal merah merupakan ambang batas toleransi penurunan performa sebesar 10% dari nilai baseline, yaitu 0.6849.
- Model dengan anotasi ≥80% pada kedua skenario masih berada di atas garis ambang, sehingga dapat dianggap layak digunakan.
- Model dengan anotasi 70% pada Skenario A jatuh di bawah ambang batas, sedangkan pada Skenario B masih sedikit di atas.
- Model dengan anotasi 60% pada kedua skenario jelas berada di bawah ambang batas, mengindikasikan penurunan performa yang signifikan.

Skenario B memperlihatkan kestabilan yang lebih baik dibandingkan Skenario A, khususnya pada tingkat anotasi menengah (80%–70%). Tren ini memperkuat temuan sebelumnya bahwa ambang efektif anotasi dapat ditetapkan pada kisaran 80%, dengan performa mulai mengalami degradasi tajam pada tingkat yang lebih rendah.

- **Kesimpulan Subbab**

Berdasarkan hasil evaluasi, dapat disimpulkan bahwa:

1. Ambang batas efektif anotasi pada Skenario A adalah 80%, sedangkan pada Skenario B adalah 70%.
2. Strategi anotasi yang tidak bias dan lebih merata (Skenario B) terbukti lebih efektif dalam menjaga performa model.
3. Hasil ini memberikan wawasan praktis bahwa dalam kondisi keterbatasan sumber daya anotasi, pengurangan anotasi hingga 70% masih layak dilakukan apabila strategi yang digunakan menyerupai Skenario B.

Dengan demikian, penentuan ambang batas ini menjadi kontribusi penting penelitian dalam memberikan panduan praktis terkait efisiensi anotasi, yang seimbang antara kualitas deteksi dan beban komputasi.

4.4 Rekomendasi Model dan Skenario Terbaik

Berdasarkan evaluasi metrik utama (*precision, recall, mAP@0.5, mAP@0.5:0.95*), *confidence*, jumlah deteksi, serta efisiensi komputasi (waktu pelatihan), penelitian ini merekomendasikan varian YOLOv11s dengan tingkat anotasi 80% pada Skenario B sebagai pilihan paling seimbang antara kualitas deteksi dan efisiensi sumber daya. Rekomendasi ini mempertimbangkan bahwa model tersebut: pertama karena tetap jauh di atas ambang batas penurunan kinerja (toleransi –10% dari baseline), kedua mempertahankan kestabilan metrik yang lebih baik daripada opsi hemat ekstrem (70%–B), dan ketiga menghemat waktu pelatihan dibandingkan baseline 100%.

Tabel 4.8 Ringkasan Kandidat dan Model Rekomendasi

Model	Skenario	Tingkat Anotasi	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Jumlah Deteksi	Confidence (%)	Waktu Training (s)	Status
YOLOv11_100	Baseline	100%	0.714	0.741	0.761	0.463	32	65.91	534.764	Acuan
YOLOv11_80	B	80%	0.709	0.718	0.731	0.439	33	55.85	487.268	Di atas ambang (disarankan)
YOLOv11_70	B	70%	0.673	0.673	0.690	0.414	20	42.74	457.310	Di atas ambang (minimum)

- **Catatan perbandingan kuantitatif (80%–B vs baseline 100%):**
  - *mAP@0.5*: 0.731 dibandingkan dengan 0.761 → turun 0.030 poin (~3,94%), jauh di bawah batas toleransi –10%.
  - *Precision*: 0.709 dibandingkan dengan 0.714 → turun 0.005 (~0,7%).
  - *Recall*: 0.718 dibandingkan dengan 0.741 → turun 0.023 (~3,11%).
  - *Confidence* rata-rata: 55,85% dibandingkan dengan 65,91% → berbeda –10,06 poin (~15,3%).
  - Jumlah deteksi (satu gambar uji): 33 dibandingkan dengan 32 → bertambah +1 deteksi.
  - Waktu *training*: 487,268 detik dibandingkan dengan 534,764 detik → hemat 47,496 detik (~8,88%).

- **Argumentasi Akademik Pemilihan 80% – Skenario B**
  1) **Kualitas deteksi yang stabil:** Penurunan *mAP@0.5* hanya sekitar 3,94% dari baseline menandakan degradasi minimal dan kinerja yang tetap kompetitif, berbeda dengan tingkat anotasi 70%–B yang mulai menekan *confidence* dan *recall* cukup jauh.
  2) **Efisiensi komputasi yang nyata:** Penghematan waktu pelatihan sekitar 8,88% tanpa perubahan *hyperparameter* mengindikasikan *trade-off* yang menguntungkan untuk siklus eksperimen berulang.
  3) **Robustness lintas metrik:** Baik *precision* maupun *recall* hanya turun di bawah 4%, sehingga stabilitas model pada skenario nyata (variabilitas pencahayaan, latar kompleks) lebih terjaga dibandingkan dengan 70%–B.
  4) **Keseimbangan konservatif–efisien:** Tingkat anotasi 70%–B memang masih melampaui ambang *mAP*, tetapi penurunan *confidence* (42,74%) dan jumlah deteksi (20) berisiko meningkatkan *missed detection*. Pilihan 80%–B menghindari risiko tersebut, tetap efisien, sekaligus lebih aman untuk aplikasi operasional.

- **Rekomendasi Praktis untuk Anotator dan Peneliti**
  - Gunakan Skenario B (menghapus anotasi objek tertutup atau bertumpuk terlebih dahulu) agar objek yang jelas dan utuh tetap memberi sinyal kuat saat pelatihan.
  - Targetkan tingkat anotasi 80% sebagai pilihan praktis: cukup menghemat beban anotasi, tanpa mengorbankan kinerja model secara signifikan.
  - Prioritaskan pelabelan lengkap pada kelas yang kompleks (anggur, jeruk, stroberi) agar stabilitas *mAP* dan *confidence* tetap terjaga meskipun total anotasi dikurangi.

- **Kesimpulan Subbab**

Sejalan dengan tujuan menyeimbangkan efisiensi komputasi dan kualitas deteksi, model YOLOv11s dengan tingkat anotasi 80% pada Skenario B direkomendasikan sebagai pilihan utama. Model ini tetap berada di atas ambang batas penurunan *mAP*, menghemat waktu pelatihan sekitar 8,9%, serta mempertahankan metrik inti (*precision* dan *recall*) dengan penurunan yang sangat kecil. Untuk kebutuhan yang lebih menekankan efisiensi ekstrem,

tingkat anotasi 70%–B masih dapat dipertimbangkan sebagai batas minimum, tetapi dengan risiko penurunan *confidence* dan jumlah deteksi yang lebih besar

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini dilakukan untuk mengevaluasi penurunan performa model deteksi objek *YOLOv11* terhadap variasi tingkat anotasi pada dataset enam kelas buah (anggur, apel, jeruk, rambutan, salak, dan stroberi) dengan dua pendekatan berbeda, yaitu:

- **Skenario A**: pengurangan anotasi untuk objek yang jelas dan utuh.
- **Skenario B**: pengurangan anotasi untuk objek yang tertumpuk dan saling menutupi.

Lima variasi tingkat anotasi diuji dalam masing-masing skenario (100%, 90%, 80%, 70%, dan 60%) menggunakan arsitektur *YOLOv11s*, dan dievaluasi berdasarkan metrik *precision*, *recall*, *mAP@0.5*, *mAP@0.5:0.95*, *training time*, serta pengujian visual menggunakan aplikasi Streamlit.

Berdasarkan hasil analisis, diperoleh beberapa kesimpulan utama:

1. **Penurunan performa model terjadi seiring pengurangan tingkat anotasi**, baik pada Skenario A maupun B.
    - Pada model YOLOv11_100 (baseline), diperoleh *precision* sebesar 71.48% dan *mAP@0.5* sebesar 76.14%.
    - Pada model dengan tingkat anotasi 60%, *precision* menurun hingga 59.7% (Skenario A) dan 64.9% (Skenario B), serta *mAP@0.5* menurun menjadi 62.2% (Skenario A) dan 65.4% (Skenario B). Hasil ini membuktikan bahwa pengurangan anotasi berpengaruh signifikan terhadap performa model.
2. **Kelas buah anggur, jeruk, dan stroberi merupakan kelas yang paling terdampak**, terutama dalam kondisi objek bertumpuk, berukuran kecil, atau memiliki warna yang mirip dengan latar belakang. Penurunan akurasi per kelas terlihat jelas pada skenario dengan anotasi terbatas. Hal ini disebabkan oleh keterbatasan informasi visual saat pelatihan, sehingga model kesulitan dalam mengenali pola objek yang kompleks.
3. **Pengujian pada satu gambar uji** menunjukkan adanya hubungan langsung antara jumlah anotasi dengan konsistensi deteksi objek.
    - Model 100% (baseline) mampu mendeteksi hingga 32 objek dengan *confidence rate* 65.91%.

- Model 60% (Skenario A) hanya mendeteksi 12 objek dengan rata-rata *confidence* 41.44%, sedangkan model 60% (Skenario B) hanya 14 objek dengan *confidence* 52.07%.
  Penurunan ini menegaskan bahwa model dengan anotasi rendah sulit memberikan deteksi yang stabil dan akurat.

4. **Ambang batas efektif anotasi ditetapkan pada tingkat 80%**, karena model dengan anotasi ≥80% pada kedua skenario masih mempertahankan performa stabil dengan penurunan metrik <10% dibanding baseline. Hal ini menunjukkan bahwa 80% anotasi merupakan titik optimal untuk menjaga kualitas deteksi sambil tetap mengurangi beban anotasi.
5. **Perbandingan antara dua skenario menunjukkan perbedaan pola penurunan performa.**
   - Skenario A lebih cepat mengalami degradasi performa, terutama di bawah 80% anotasi, meskipun sesekali mendeteksi lebih banyak objek dengan akurasi rendah (*over-prediction*).
   - Skenario B lebih stabil hingga 70% anotasi, karena penghapusan anotasi pada objek tertumpuk masih memungkinkan model belajar dari objek yang jelas. Dengan demikian, Skenario B lebih sesuai untuk strategi efisiensi anotasi, sedangkan Skenario A dapat digunakan apabila tujuan utama adalah mendeteksi objek sebanyak mungkin meskipun dengan risiko penurunan akurasi.
6. **Aspek efisiensi waktu anotasi dan pelatihan juga menjadi temuan penting.**
   - Dari sisi anotasi manual, penggunaan 80% anotasi berarti potensi penghematan sekitar 20% waktu *labeling* dibanding dataset penuh. Strategi ini sangat relevan pada proyek skala besar di mana keterbatasan tenaga anotator menjadi kendala utama.
   - Dari sisi *training time*, model dengan anotasi 80% (Skenario B) menunjukkan waktu pelatihan rata-rata 487 detik, lebih cepat sekitar 47 detik dibanding baseline (534 detik). Penghematan ini menunjukkan bahwa pengurangan anotasi tidak hanya mengurangi beban labeling, tetapi juga mempercepat proses pelatihan komputasi tanpa menurunkan performa secara signifikan. Dengan demikian, penelitian ini menegaskan bahwa kombinasi antara efisiensi anotasi dan efisiensi komputasi dapat dicapai pada tingkat anotasi 80%, khususnya pada Skenario B.

Secara keseluruhan, penelitian ini membuktikan bahwa strategi pengurangan anotasi dapat dilakukan hingga tingkat tertentu tanpa mengorbankan performa model secara signifikan. Hasil ini memberikan dasar praktis bagi peneliti dan praktisi untuk menyeimbangkan kualitas model dengan efisiensi waktu dan sumber daya dalam pengembangan sistem deteksi objek berbasis *YOLOv11*.

5.2 Saran

Berdasarkan temuan dan hasil evaluasi terhadap dua skenario, berikut beberapa saran yang dapat dijadikan acuan dalam pengembangan maupun penelitian lanjutan:

1. **Gunakan strategi pengurangan anotasi yang efisien.**
   Penelitian ini menunjukkan bahwa tingkat anotasi 80% masih layak digunakan, terutama pada Skenario B, karena performa model tetap stabil dengan penurunan metrik <10% dari baseline. Tingkat anotasi ini dapat dijadikan standar minimum untuk menjaga keseimbangan antara kualitas hasil deteksi dan efisiensi waktu anotasi.
2. **Perhatikan aspek efisiensi waktu anotasi dan pelatihan.**
   Mengurangi anotasi hingga 80% berarti terdapat potensi penghematan waktu labeling sekitar 20% dibanding dataset penuh. Selain itu, pengurangan anotasi juga berkontribusi pada efisiensi komputasi, di mana waktu *training* model berkurang hingga puluhan detik per eksperimen. Hal ini penting pada penelitian berskala besar atau saat sumber daya komputasi terbatas.
3. **Gunakan skenario anotasi sesuai kebutuhan aplikasi.**
   - **Skenario B** lebih sesuai untuk tujuan efisiensi karena mampu mempertahankan stabilitas performa meski anotasi berkurang, sehingga cocok untuk proyek dengan keterbatasan tenaga anotator.
   - **Skenario A** dapat digunakan pada aplikasi yang menekankan pada kuantitas deteksi (mendeteksi lebih banyak objek), walaupun berisiko menurunkan akurasi.
4. **Eksplorasi metode anotasi alternatif.**
   Penelitian lanjutan disarankan untuk mengintegrasikan pendekatan seperti *semi-supervised learning*, *active learning*, atau *self-training*. Metode ini berpotensi mengurangi kebutuhan anotasi manual lebih jauh tanpa mengorbankan kualitas dataset, sekaligus meningkatkan efisiensi dari sisi waktu dan biaya.

5. **Perluasan eksperimen dengan arsitektur lain.**

   Penelitian mendatang sebaiknya juga membandingkan YOLOv11 dengan model deteksi objek lainnya, seperti YOLOv8, EfficientDet, atau Faster R-CNN. Hal ini bertujuan untuk mengidentifikasi apakah pola penurunan performa serupa juga muncul pada arsitektur lain, sekaligus menilai efisiensi relatif masing-masing model.

6. **Analisis khusus pada kelas objek yang sulit.**

   Kelas buah anggur, jeruk, dan stroberi terbukti paling rentan terhadap pengurangan anotasi karena karakteristik visualnya yang kompleks (berukuran kecil, bertumpuk, atau mirip dengan latar belakang). Oleh karena itu, anotasi pada kelas-kelas tersebut sebaiknya dilakukan secara lebih teliti dan menyeluruh agar model mampu mengenali objek dengan lebih baik.

LAMPIRAN

DAFTAR PUSTAKA

Achmad, H., Pramudwiatmoko, A., Satrio Gumilang, M., Al Karim, B., & Wiyono, H. (2025). Analisis Kinerja Model Deteksi Objek Yolo, Ssd, dan Faster R-Cnn pada Citra Penglihatan Malam untuk Pengenalan Tindak Kejahatan. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *12*(1), 145–152. https://doi.org/10.25126/jtiik.2025128409

Aini, Q., Lutfiani, N., Kusumah, H., & Zahran, M. S. (2021). DETEKSI DAN PENGENALAN OBJEK DENGAN MODEL MACHINE LEARNING: MODEL YOLO. *CESS (Journal of Computer Engineering System and Science)*, *6*(2), 2502–2714.

Akkem, Y., Kumar, B. S., & Varanasi, A. (2023). Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review and Implementation. *Indian Journal Of Science And Technology*, *16*(48), 4688–4702. https://doi.org/10.17485/IJST/v16i48.2850

Aras, S., Tanra, P., & Bazhar, M. (2024). Deteksi Tingkat Kematangan Buah Tomat Menggunakan YOLOv5. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 623–628. https://doi.org/10.57152/malcom.v4i2.1270

Ayuningtyas, A., Wintolo, H., Sumari, A. D. W., Setyaningsih, E., Pujiastuti, A., Honggowibowo, A. S., … Kusumaningrum, A. (2025). The CNN Model with YOLO Architecture for Ultrasonography Images in Early Breast Cancer Detection. *Journal of Applied Data Sciences*, *6*(2), 1116–1128. https://doi.org/10.47738/jads.v6i2.587

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. Retrieved from http://arxiv.org/abs/2004.10934

Brucal, S. G. E., De Jesus, L. C. M., Peruda, S. R., Samaniego, L. A., & Yong, E. D. (2023). Development of Tomato Leaf Disease Detection using YoloV8 Model via RoboFlow 2.0. *GCCE 2023 - 2023 IEEE 12th Global Conference on Consumer Electronics*, 692–694. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/GCCE59613.2023.10315251

Elef, T., Khestin, A., Ugur, D., Ai, X., Alhassan, T., Chen, Y., … Peiyu, Y. (2025). *Bird Detection Using Yolov11*. https://doi.org/10.13140/RG.2.2.35070.09282

Gunawan, T. S., Ashraf, A., Riza, B. S., Haryanto, E. V., Rosnelly, R., Kartiwi, M., & Janin, Z. (2020). Development of video-based emotion recognition using deep learning with Google Colab. *Telkomnika (Telecommunication Computing Electronics and Control)*, *18*(5), 2463–2471. https://doi.org/10.12928/TELKOMNIKA.v18i5.16717

Hidayatullah, P., Syakrani, N., Sholahuddin, M. R., Gelar, T., & Tubagus, R. (2025). *YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review*.

Ibrahem, H., Salem, A. D. A., & Kang, H. S. (2021). Real-Time Weakly Supervised Object Detection Using Center-of-Features Localization. *IEEE Access*, *9*, 38742–38756. https://doi.org/10.1109/ACCESS.2021.3064372

Irwan Adhi Prasetya, Fadli Sukandiarsyah, Novi Aryani Fitri, & Safri Adam. (2024). Klasifikasi kualitas buah jeruk menggunakan computer vision dengan arsitektur YOLO V8. *Jurnal Pendidikan Informatika Dan Sains*, *13*(2), 187–201. https://doi.org/10.31571/saintek.v13i2.8346

Kurniawan, R., Martadinata, A. T., & Cahyo, S. D. (2023). Klasifikasi Tingkat Kematangan Buah Sawit Berbasis Deep Learning dengan Menggunakan Arsitektur Yolov5. *Journal of Information System Research (JOSH)*, *5*(1), 302–309. https://doi.org/10.47065/josh.v5i1.4408

Kuznetsova, A., Maleva, T., & Soloviev, V. (2021). YOLOv5 versus YOLOv3 for Apple Detection. In *Cyber-Physical Systems: Modelling and Intelligent Control* (pp. 349–358). https://doi.org/10.1007/978-3-030-66077-2_28

Li, M., Wang, Z. Q., Zhao, Y. H., & Li, Q. (2025). License Plate Recognition for Smart Construction Sites Based on GMH-YOLO. *IEEE Access*, *13*, 89760–89774. https://doi.org/10.1109/ACCESS.2025.3569101

Marcelleno, D. J., & Putra, M. P. K. (2025). PERFORMANCE EVALUATION OF YOLOV8 IN REAL-TIME VEHICLE DETECTION IN VARIOUS ENVIRONMENTAL CONDITIONS. *Jurnal Teknik Informatika (Jutif)*, *6*(1), 269–279. https://doi.org/10.52436/1.jutif.2025.6.1.3916

Ning, C., Menglu, L., Hao, Y., Xueping, S., & Yunhong, L. (2021). Survey of pedestrian detection with occlusion. *Complex and Intelligent Systems*, *7*(1), 577–587. https://doi.org/10.1007/s40747-020-00206-8

Pratama, B. M., Gunawan, D., & Gultom, R. A. G. (2020). Deep learning-based object detection and geographic coordinate estimation system for GeoTiff imagery. *Journal of Physics: Conference Series*, *1577*(1). Institute of Physics Publishing. https://doi.org/10.1088/1742-6596/1577/1/012003

Rohiman, Y. K., Kanata, B., Ahmad, L., & Akbar, I. (2025). Deteksi dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (Yolov7). *BULLETIN OF COMPUTER SCIENCE RESEARCH*, *5*(3), 268–276. https://doi.org/10.47065/bulletincsr.v5i3.509

Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2025). *Comprehensive Performance Evaluation of YOLOv12, YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments*. Retrieved from http://arxiv.org/abs/2407.12040

Saputra, D. H., Imran, B., & Juhartini. (2023). Object Detection Untuk Mendeteksi Citra Buah-Buahan Menggunakan Metode Yolo. *Jurnal Kecerdasan Buatan Dan Teknologi Informasi*, *2*(2), 70–80. https://doi.org/10.69916/jkbti.v2i2.18

Setiyadi, A., Utami, E., & Ariatmanto, D. (2023). Analisa Kemampuan Algoritma YOLOv8 Dalam Deteksi Objek Manusia Dengan Metode Modifikasi Arsitektur. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *7*(2), 891–901.

Shan, Z., Haoyan, H., Zhu, C., Du, S., Jing, H., & Haibin, W. (2025). RSM-YOLOv11: Lightweight Steel Surface Defect Segmentation Algorithm Research Based on YOLOv11 Improvement. *IEEE Access*, *13*, 111681–111698. https://doi.org/10.1109/ACCESS.2025.3583752

Sharma, A., Kumar, V., & Longchamps, L. (2024). Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. *Smart Agricultural Technology*, *9*. https://doi.org/10.1016/j.atech.2024.100648

Song, J., Kim, D., Jeong, E., & Park, J. (2025). Determination of Optimal Dataset Characteristics for Improving YOLO Performance in Agricultural Object Detection. *Agriculture (Switzerland)*, *15*(7). https://doi.org/10.3390/agriculture15070731

Sugandi, A. N., & Hartono, B. (2022). Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO. *Prosiding The 13th Industrial Research Workshop and National Seminar*, 13–14.

Therino Elevan, R., Hermanto, D., & Widiyanto, E. P. (2025). DETEKSI JUMLAH KENDARAAN RODA EMPAT MENGGUNAKAN YOLO. *Integrative Perspectives of Social and Science Journal*, *2*(1), 1647.

Wibowo, M., Tullah, R., & Ricesa, W. (2025). Studi Perbandingan Algoritma YOLO dan FOMO untuk Object Detection pada Perangkat ESP32-CAM. *INSECT(Informatics And Security)*, *11*(1).

Xu, B., Chen, M., Guan, W., & Hu, L. (2023). *Efficient Teacher: Semi-Supervised Object Detection for YOLOv5*. Retrieved from http://arxiv.org/abs/2302.07577

Yang, X., Rey Castillo, E. del, Zou, Y., Wotherspoon, L., Yang, J., & Li, H. (2025). Automated Concrete Bridge Damage Detection Using an Efficient Vision Transformer-Enhanced Anchor-Free YOLO. *Engineering*. https://doi.org/10.1016/j.eng.2025.02.018