

BAB II

DASAR TEORI

2.1 Tinjauan Pustaka

Dalam pengembangan sistem deteksi objek *real time* berbasis *deep learning*, khususnya pada bidang pertanian dan lingkungan, tuning *hyperparameter* telah terbukti menjadi salah satu faktor kunci dalam meningkatkan performa model YOLOv8. Berbagai studi terkini menyoroti bagaimana pemilihan dan penyetelan parameter seperti *learning rate*, *batch size*, *weight decay*, *momentum*, dan *optimizer* dapat memengaruhi secara langsung akurasi dan efisiensi model deteksi. Ramos et al[4], melakukan eksplorasi mendalam terhadap hal ini dengan mengusulkan kombinasi metode *One-Factor-At-a-Time (OFAT)* dan *Random Search* untuk mengoptimalkan YOLOv8l dalam mendeteksi asap dan kebakaran liar. Melalui proses tuning yang sistematis, model berhasil mencatat peningkatan signifikan pada seluruh metrik evaluasi, termasuk peningkatan $mAP@0.50:0.95$ sebesar 5.09%, menjadikannya lebih andal untuk penerapan dalam konteks pertanian presisi dan mitigasi bencana[4].

Studi serupa juga dilakukan oleh Lee et al[5] yang memfokuskan pada deteksi penyakit daun apel menggunakan YOLOv8s. Mereka menerapkan pendekatan tuning yang sama, yaitu OFAT diikuti dengan eksplorasi 50 konfigurasi menggunakan *Random Search*. Kombinasi parameter terbaik (C34) yang terdiri dari learning rate 0.0137, batch size 48, momentum 0.9433, dan weight decay 0.0009, mampu menghasilkan model dengan precision 0.97797, recall 0.97295, serta $mAP@0.5$ sebesar 0.98257. Hasil ini tidak hanya menegaskan efektivitas pendekatan tuning tersebut, tetapi juga memperlihatkan bagaimana konfigurasi yang tepat mampu meningkatkan *generalization* model dalam mendeteksi berbagai jenis penyakit daun secara cepat dan akurat di lingkungan nyata[5].

Di sisi lain, pendekatan YOLOv8 juga telah diimplementasikan untuk klasifikasi tingkat kematangan buah, seperti pada penelitian Rahman et al.[6] terhadap buah manggis, dan Genoveva & Syah[7] pada Tandan Buah Segar (TBS) kelapa sawit. Meskipun kedua penelitian ini belum melakukan tuning *hyperparameter* secara eksplisit atau sistematis, pemilihan varian ringan seperti YOLOv8s serta penerapan metodologi CRISP-DM menunjukkan performa yang menjanjikan. Model YOLOv8s yang digunakan dalam deteksi kematangan buah manggis misalnya, mampu mencatatkan $mAP@0.50-0.95$ sebesar 0.972 dengan recall sempurna (1.0), membuktikan bahwa meskipun tuning belum dilakukan, pemilihan struktur model dan preprocessing data berperan penting dalam performa akhir[6]. Sementara itu, pada kasus kelapa sawit, model mencatatkan mAP sebesar 0.98, dengan precision 0.945 dan recall 0.947, menjadikan YOLOv8 sebagai solusi yang efektif untuk otomasi seleksi buah matang di industri Perkebunan[7].

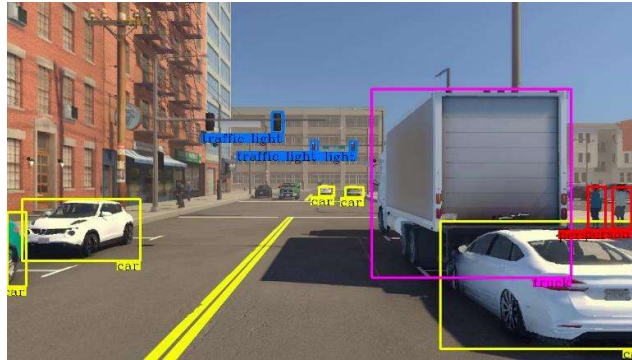
Sebagai pembandingan, studi yang lebih awal seperti oleh Yuliany et al.[8] menggunakan arsitektur *Convolutional Neural Network (CNN)* untuk klasifikasi hama tanaman padi, tetapi mengalami tantangan *overfitting*. Hal ini diakibatkan oleh pemilihan parameter model yang tidak optimal serta pembagian data yang tidak cukup robust, sehingga akurasi validasi jauh lebih rendah dibandingkan akurasi pelatihan. Hal ini menjadi bukti bahwa tanpa tuning *hyperparameter* yang terarah, model *deep learning* cenderung gagal untuk melakukan generalisasi dengan baik pada data nyata. Oleh karena itu, studi-studi yang lebih mutakhir menggarisbawahi pentingnya pendekatan tuning sistematis sebagai bagian integral dalam pengembangan sistem cerdas berbasis YOLO untuk aplikasi agrikultur[8].

Tabel 2. 1 Tabel Tinjauan Pustaka

Judul	Metode	Fokus Penelitian	Hasil
<i>Hyperparameter optimization of YOLOv8 for smoke and wildfire detection</i> (Ramos et al., 2024)	OFAT & Random Search	Deteksi asap dan kebakaran untuk keselamatan pertanian	mAP@0.5:0.95 meningkat 5.09%; peningkatan F1, recall, precision
<i>Hyperparameter optimization of apple leaf dataset using YOLOv8</i> (Lee et al., 2025)	OFAT & Random Search	Deteksi penyakit daun apel	mAP@0.5 tertinggi 0.98257; akurasi tinggi dan inferensi cepat
<i>Deteksi tingkat kematangan buah manggis menggunakan YOLOv8</i> (Rahman et al., 2024)	YOLOv8s + CRISP-DM	Klasifikasi tingkat kematangan manggis	mAP50-95 = 0.972; recall 1; precision 0.997
<i>Deteksi kematangan kelapa sawit dengan YOLOv8</i> (Genoveva & Syah, 2024)	YOLOv8 + CRISP-DM	Deteksi kematangan TBS kelapa sawit	mAP = 0.98; precision 0.945; recall 0.947

2.2 Object Detection

Object detection adalah salah satu bidang utama dalam *computer vision* yang bertujuan untuk mengidentifikasi serta menentukan lokasi objek dalam suatu gambar atau video. Proses ini mencakup dua tugas utama, yaitu *object localization* yang menentukan posisi objek dan *object classification* yang mengidentifikasi kategori dari objek tersebut. Teknologi ini memiliki berbagai aplikasi, termasuk pengenalan wajah, analisis perilaku manusia, kendaraan otonom, dan sistem keamanan. Dalam otomasi pergudangan, deteksi objek dapat digunakan untuk mengidentifikasi berbagai elemen seperti rak, kargo, kardus, pekerja, dan benda asing. Hal ini membantu sistem pendeteksian robot dalam mendeteksi hambatan di jalur pemindahan barang atau rak, sehingga meningkatkan efektivitas dan efisiensi kerja robot [9] [10].



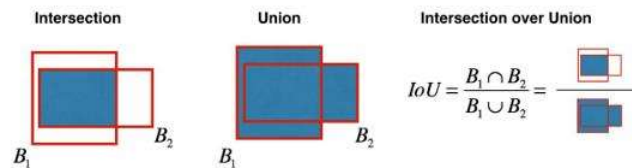
Gambar 2. 1 Object Detection [11]

Algoritma *object detection* membangun model berdasarkan contoh pelatihan untuk mengenali jenis objek tertentu. Pada gambar 2.1 merupakan hasil implementasi algoritma deteksi objek berbasis *deep learning* untuk pengamatan kondisi lalu lintas. Jika objeknya tetap dan memiliki bentuk kaku, satu contoh mungkin cukup. Namun, dalam banyak kasus, dibutuhkan ratusan hingga ribuan contoh agar model dapat memahami variasi dalam suatu kelas objek. Semakin banyak informasi tentang perbedaan dalam suatu kelas yang dimasukkan langsung ke dalam model, semakin sedikit data pelatihan yang diperlukan [12]. Secara umum, terdapat dua pendekatan utama dalam *object detection*, yaitu metode tradisional dan metode berbasis *deep learning*. Metode tradisional seperti *Histogram of Oriented Gradients* (HOG) dan *Support Vector Machines* (SVM) masih digunakan hingga kini karena kesederhanaannya, meskipun memiliki keterbatasan dalam menangani variasi objek dan kondisi pencahayaan. Selain itu, ada juga metode seperti *Deformable Part-based Models* (DPM) dan *AdaBoost*, yang menggabungkan berbagai fitur untuk meningkatkan akurasi deteksi [13]. Namun, metode-metode ini cenderung memiliki performa yang lebih rendah dibandingkan pendekatan berbasis *deep learning*, terutama dalam lingkungan yang kompleks dan dinamis.

2.2.1 Intersection Over Union (IoU)

Intersection Over Union (IoU) adalah metrik evaluasi yang digunakan untuk mengukur akurasi prediksi bounding box dalam

deteksi objek dengan membandingkan area tumpang tindih antara bounding box hasil prediksi model dan bounding box ground truth. Secara matematis, IoU dihitung dengan membagi luas area perpotongan (intersection) dari kedua bounding box dengan luas area gabungan (union) dari keduanya



Gambar 2. 2 *IoU Box*[14]

Pada Gambar 2.2 area biru pada bagian kiri menunjukkan intersection, yaitu wilayah di mana kedua bounding box saling menutupi. Sementara pada bagian tengah, area biru menggambarkan union, yaitu total luas dari kedua bounding box tanpa menghitung bagian yang tumpang tindih. Nilai IoU berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan tumpang tindih sempurna antara prediksi dan ground truth, sedangkan nilai 0 berarti tidak ada tumpang tindih sama sekali[15]. Nilai ambang batas IoU sering digunakan untuk menentukan apakah sebuah prediksi dianggap sebagai deteksi yang benar (true positive) atau tidak. Misalnya, jika IoU antara prediksi dan ground truth lebih besar dari 0,5 atau 0,7, prediksi tersebut dianggap berhasil mendeteksi objek yang benar.

2.3 *Deep Learning*

Deep Learning telah menjadi pendekatan utama dalam pengembangan sistem deteksi objek modern karena kemampuannya menangani data visual yang kompleks dan menghasilkan akurasi tinggi. Metode ini menggunakan arsitektur Convolutional Neural Networks (CNN), yang mampu mengekstrak fitur hierarkis dari gambar melalui lapisan konvolusi bertingkat. CNN memungkinkan deteksi pola lokal seperti tepi atau tekstur pada lapisan awal,

pengenalan bentuk kompleks pada lapisan lebih dalam, serta otomatisasi ekstraksi fitur tanpa memerlukan intervensi manual. Beberapa algoritma unggulan dalam deteksi objek berbasis *Deep Learning* mencakup YOLO, SSD, dan Faster R-CNN. YOLO, khususnya versi terbaru seperti YOLOv8, mampu mencapai mAP 76-81% dalam deteksi manusia dengan kecepatan hingga 45 fps pada GPU high-end, serta peningkatan akurasi hingga 5% berkat modifikasi arsitektur *head* [16].

Kelebihan utama *Deep Learning* adalah kemampuannya untuk belajar dari data besar tanpa pemrograman eksplisit, menjadikannya solusi unggul dalam berbagai aplikasi, mulai dari pengenalan wajah hingga analisis gambar medis. Namun, tantangan seperti kebutuhan dataset berkualitas tinggi dan sumber daya komputasi yang besar tetap menjadi perhatian utama. Di sisi lain, metode *deep learning* telah mengalami perkembangan pesat dan semakin mendominasi dalam deteksi objek berkat akurasi tinggi serta peningkatan kecepatan pemrosesan. Beberapa teknik yang sering digunakan dalam kategori ini meliputi R-CNN, SSD, Faster R-CNN, serta YOLO. Salah satu keunggulan utama dari pendekatan *deep learning* adalah kemampuannya dalam mengurangi kebutuhan jumlah data pelatihan dengan mengintegrasikan fitur secara eksplisit ke dalam model[12] . Namun, meskipun kemajuan ini menjanjikan, masih terdapat tantangan yang harus diatasi. Beberapa tantangan utama dalam *object detection* meliputi ketergantungan pada kualitas data pelatihan, perubahan pencahayaan dan perspektif yang dapat mempengaruhi akurasi, serta kebutuhan daya komputasi yang tinggi untuk menjalankan model *deep learning* dalam aplikasi *real time*.

2.3.1 Hyperparameter

Hyperparameter merupakan parameter kunci yang mendefinisikan arsitektur dan proses pelatihan model *deep learning*. Optimalisasi *hyperparameter* menjadi elemen kunci untuk mencapai performa model yang maksimal. Setiap dataset memiliki

karakteristik tertentu, sehingga diperlukan penyesuaian *hyperparameter* yang tepat agar model mampu mengenali pola secara efektif dan efisien. Optimalisasi ini tidak hanya memengaruhi akurasi, tetapi juga berperan dalam meningkatkan stabilitas pelatihan dan kemampuan generalisasi model terhadap data baru. Dengan mengatur *hyperparameter* secara sistematis dapat menghindari permasalahan seperti underfitting dan overfitting.

- a) Lr (Learning Rate) : mengatur seberapa besar pembaruan bobot model dilakukan selama pelatihan. Nilai yang terlalu besar dapat membuat model gagal konvergen, sementara nilai terlalu kecil memperlambat pelatihan dan rentan terjebak di minimum lokal. Nilai lr_0 yang umum digunakan berada di kisaran 0.001 hingga 0.01[17]
- b) Momentum : Momentum membantu mempercepat pelatihan dengan mempertahankan arah gradien dari iterasi sebelumnya. Ini mengurangi osilasi dan mempercepat pencapaian minimum global.[18]
- c) Batch Size : Batch size memengaruhi stabilitas pelatihan dan efisiensi komputasi. Batch kecil memberikan gradien yang lebih bervariasi (baik untuk generalisasi), sedangkan batch besar membuat pelatihan lebih stabil tetapi berisiko overfitting[19]
- d) Weight Decay : salah satu cara untuk mencegah model belajar terlalu berlebihan dari data pelatihan (overfitting). Caranya adalah dengan memberi penalti jika bobot (nilai dalam jaringan) menjadi terlalu besar. Dengan begitu, model dipaksa untuk belajar secara lebih sederhana dan tidak terlalu mengikuti detail kecil yang mungkin tidak penting [20]

2.3.2 Optimizer

Optimizer dalam proses pelatihan model *deep learning* untuk mengatur dan memperbarui bobot (weight) dari jaringan saraf

berdasarkan nilai error (loss). ujian utama optimizer adalah meminimalkan loss function, sehingga model dapat belajar dan membuat prediksi yang semakin akurat. Algoritma optimasi berikut dibandingkan dalam studi ini meliputi:

- a) SGD (Stochastic Gradient Descent): Dikenal karena stabilitas dan efektivitasnya dalam tugas visi komputer, SGD memperbarui bobot berdasarkan sebagian data, membantu menghindari minimum lokal.
- b) Adam (Adaptive Moment Estimation): Adam secara dinamis menyesuaikan kecepatan pembelajaran untuk setiap parameter, meningkatkan kecepatan konvergensi pada kumpulan data yang kompleks.
- c) AdamW (Adam with Weight Decay): Varian Adam yang mencakup peluruhan bobot, membantu mengurangi overfitting.

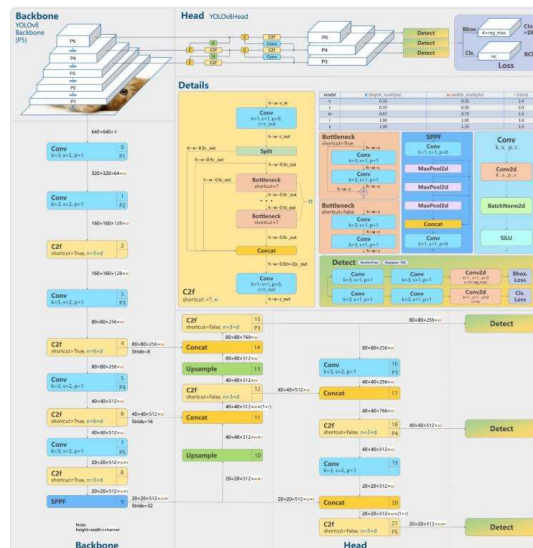
2.3.3 Preconditioning

Preconditioning memainkan peran penting dalam *object detection* ini, *Preconditioning* mencakup serangkaian teknik komprehensif yang diterapkan sebelum dan selama pelatihan model untuk meningkatkan kinerja dan akurasi. Hal ini dilengkapi dengan strategi augmentasi data yang termasuk rotasi, membalik, penskalaan, jitter warna, penyesuaian kecerahan, dan pemangkasan acak-semuanya secara artifisial memperluas kumpulan data pelatihan dan membantu model YOLOv8 menjadi tangguh terhadap variasi tampilan, orientasi, kondisi pencahayaan, dan latar belakang. Image quality enhancement techniques semakin memperkuat prapengolahan dengan menerapkan filter pengurangan noise, peningkatan kontras, dan penyeimbangan warna untuk meningkatkan ekstraksi fitur dari gambar apel. Penanganan ketidakseimbangan kelas melalui pengambilan sampel berbobot atau fungsi kehilangan khusus memastikan semua varietas apel dipelajari

dengan sama baiknya, terlepas dari representasi mereka dalam kumpulan data. *Preconditioning* arsitektur YOLOv8 sendiri melibatkan inisialisasi transfer learning, di mana bobot yang telah dilatih sebelumnya pada dataset besar memberikan titik awal yang lebih unggul dibandingkan dengan inisialisasi acak.

2.3.4 YOLOv8

Alih-alih membingkai masalah deteksi objek sebagai tantangan klasifikasi, Redmon memperkenalkan algoritma YOLO (*You Only Look Once*), yang terkenal karena kecepatannya. YOLO mampu mencapai kecepatan hingga 45 frame per detik (fps) dalam versi standar dan hingga 155 fps dalam versi cepat, dengan efisiensi tinggi dibandingkan metode deteksi objek lainnya[21]. Salah satu versi terbaru dari algoritma ini adalah YOLOv8, yang dikembangkan oleh Ultralytics. YOLOv8 dikenal karena kecepatan dan akurasi yang baik, memungkinkan penerapan yang lebih luas dalam berbagai aplikasi.

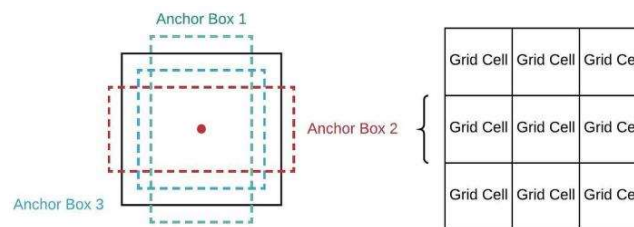


Gambar 2. 3 Arsitektur YOLO[22]

YOLOv8 merupakan arsitektur deteksi objek mutakhir yang dirancang untuk menggabungkan efisiensi komputasi dan akurasi

tinggi melalui struktur tiga komponen utamanya yaitu Backbone, Neck, dan Head yang dapat dilihat pada **Gambar 2.3**. Backbone YOLOv8 menggunakan CSPDarknet dengan modul C2f yang mengintegrasikan koneksi cross-stage partial untuk mengurangi redundansi komputasi, dilengkapi lapisan SPPF (Spatial Pyramid Pooling Fast) untuk menangani variasi skala objek, serta fungsi aktivasi yang meningkatkan kemampuan ekstraksi fitur hierarkis [23][24].

Proses ini menghasilkan lima level fitur (P1 hingga P5) yang menangkap informasi mulai dari tekstur dasar hingga bentuk objek kompleks. Bagian Neck mengombinasikan teknik PANet dan BiFPN untuk agregasi fitur multi-skala, memadukan informasi dari lapisan resolusi tinggi (detail lokal) dan rendah (konteks global) melalui upsampling dinamis dan concatenation adaptif, sehingga memungkinkan deteksi objek berukuran kecil, sedang, dan besar secara simultan [25]. YOLOv8 juga mendukung berbagai mode operasional, seperti pelatihan pada dataset khusus, validasi model, prediksi gambar atau video baru, ekspor model ke format lain, pelacakan objek secara *real time*, serta evaluasi kinerja melalui mode *benchmark*. Kombinasi fitur-fitur ini menjadikan YOLOv8 sebagai solusi yang efisien untuk deteksi objek di berbagai kondisi lingkungan [26]

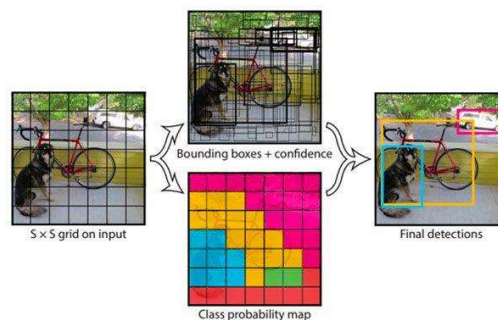


Gambar 2. 4 Anchor Box untuk masing masing *grid cell*[27]

Yolo menggunakan model terintegrasi untuk mendeteksi objek dengan membuat jaringan saraf tunggal yang memprediksi beberapa

bounding boxes dan probabilitas kelas di dalamnya secara bersamaan. Pada **Gambar 2.4** sistem Yolo bekerja dengan cara membagi gambar input ke dalam kisi-kisi $S \times S$, di mana salah satu sel *grid* bertugas mendeteksi objek jika pusatnya terletak di dalam sel *grid* tersebut[28]. Agar lebih fleksibel dalam mendeteksi berbagai bentuk dan ukuran objek, setiap grid cell dilengkapi dengan beberapa anchor box, yaitu kotak acuan dengan ukuran dan rasio aspek yang berbeda-beda. Anchor box ini bertindak sebagai titik awal prediksi, di mana model belajar menghitung offset dari anchor box agar sesuai dengan bounding box nyata (ground truth).

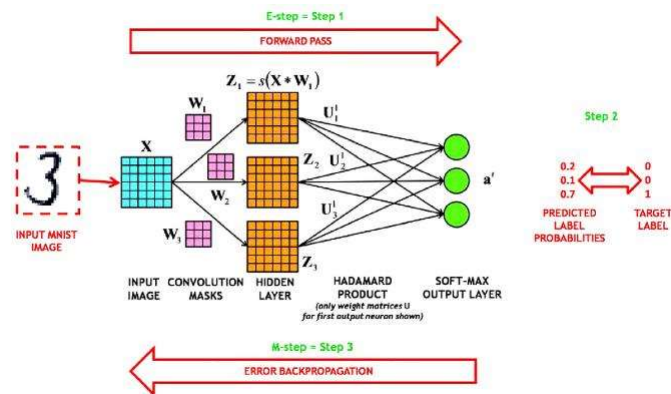
Setiap sel *grid* memprediksi *bounding box*, *class probabilities* dan *confidence score*, di mana *confidence score* menunjukkan tingkat kepercayaan dan akurasi model dalam mengidentifikasi keberadaan objek di dalam kotak[28]. Seluruh informasi ini digabungkan untuk menghasilkan bounding box akhir yang paling akurat. Model kemudian melakukan penyempurnaan menggunakan teknik *non-maximum suppression* (NMS) untuk menghilangkan prediksi yang tidak perlu dan mempertahankan hanya bounding box terbaik. Hasil akhirnya adalah output berupa gambar dengan kotak deteksi (*bounding boxes*) yang menandai lokasi objek, lengkap dengan label kelas. Proses deteksi dari model Yolo dapat dilihat pada Gambar 2.5 berikut ini.



Gambar 2. 5 Output YOLO[29]

2.3.5 Proses Training YOLOv8

Proses training YOLOv8 dilakukan melalui alur operasi algoritma (AOA) pembelajaran berbasis Convolutional Neural Network (CNN) yang bertujuan untuk mengoptimasi bobot model agar mampu melakukan deteksi dan klasifikasi objek secara akurat. YOLOv8 memanfaatkan CNN sebagai backbone untuk mengekstraksi fitur visual dari citra masukan, sehingga seluruh proses training mengikuti prinsip dasar pembelajaran CNN yang terdiri dari tahap *forward pass*, perhitungan loss, *backpropagation*, dan optimasi bobot.



Gambar 2. 6 Algoritma CNN

Pada tahap awal, citra hasil preprocessing dimasukkan ke dalam model YOLOv8 dan diproses melalui *forward pass*. Pada tahap ini, citra melewati backbone YOLOv8 yang terdiri dari lapisan konvolusi, fungsi aktivasi non-linear, serta mekanisme pooling untuk mengekstraksi fitur visual tingkat rendah hingga tingkat tinggi. Fitur yang dihasilkan kemudian diteruskan ke bagian *neck* untuk dilakukan penggabungan fitur multi-skala, sebelum akhirnya diproses oleh *head* YOLOv8 yang menghasilkan prediksi berupa bounding box, kelas objek, dan nilai confidence. Tahapan ini berfungsi untuk membentuk representasi fitur dan menghasilkan keluaran prediksi, namun belum melakukan pembaruan bobot model.

Hasil prediksi YOLOv8 selanjutnya dibandingkan dengan label ground truth menggunakan fungsi loss khusus YOLO yang mencakup kesalahan lokasi bounding box, kesalahan klasifikasi, dan kesalahan confidence. Nilai loss ini digunakan sebagai indikator seberapa besar perbedaan antara prediksi model dan kondisi sebenarnya. Setelah loss dihitung, proses dilanjutkan dengan *backpropagation*, yaitu mekanisme penyebaran gradien kesalahan dari bagian head kembali ke seluruh lapisan jaringan, termasuk backbone dan neck, untuk menghitung arah perbaikan bobot model.

Tahap pembaruan bobot dilakukan oleh algoritma optimasi (*optimizer*), seperti Adam, yang memanfaatkan nilai gradien hasil *backpropagation*. Pada tahap inilah hyperparameter berperan secara langsung dalam proses training. Hyperparameter seperti learning rate mengatur besar langkah pembaruan bobot, batch size menentukan jumlah data yang diproses dalam satu iterasi, epoch menentukan jumlah pengulangan proses training, serta weight decay berfungsi sebagai regularisasi untuk mencegah terjadinya overfitting. Hyperparameter ini tidak ditempatkan pada lapisan tertentu seperti pooling atau fungsi aktivasi, melainkan bekerja secara global pada tahap optimasi bobot di seluruh jaringan YOLOv8.

Rangkaian proses *forward pass*, perhitungan loss, *backpropagation*, dan optimasi bobot ini diulang secara terus-menerus selama sejumlah epoch hingga model mencapai kondisi konvergen atau menunjukkan performa optimal pada data validasi. Setelah proses training selesai, bobot dan parameter jaringan yang telah teroptimasi disimpan sebagai model YOLOv8 terlatih. Model ini kemudian siap digunakan untuk proses inferensi secara real-time pada sistem penyortiran buah apel, di mana hasil deteksi dan

klasifikasi digunakan sebagai dasar pengambilan keputusan mekanisme penyortiran otomatis.

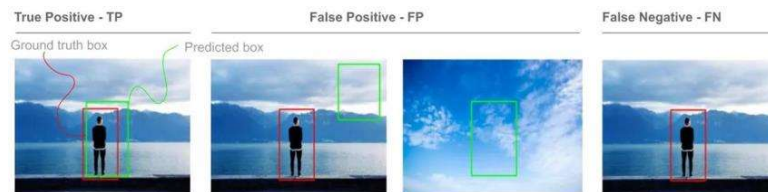
Selama proses training berlangsung, performa model dipantau secara berkala menggunakan data validasi yang tidak digunakan dalam proses pembelajaran. Metrik evaluasi yang digunakan mencakup mean Average Precision (mAP), precision, recall, dan F1-score untuk mengukur akurasi deteksi objek dan klasifikasi kategori buah apel. Data validasi berfungsi sebagai indikator objektif untuk menilai kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Monitoring dilakukan pada setiap epoch atau interval tertentu dengan mencatat nilai loss pada data training dan data validasi. Jika nilai loss pada data training terus menurun namun loss pada data validasi mulai meningkat, hal ini mengindikasikan terjadinya overfitting, yaitu kondisi di mana model terlalu menyesuaikan diri dengan pola spesifik data training sehingga kehilangan kemampuan generalisasi. Untuk mencegah overfitting, teknik regularisasi seperti weight decay, dropout, dan data augmentation diterapkan selama proses training. Selain itu, mekanisme early stopping juga dapat diimplementasikan untuk menghentikan proses training secara otomatis ketika performa pada data validasi tidak lagi menunjukkan peningkatan signifikan setelah sejumlah epoch tertentu.

2.3.6 Confusion Matrix

Matriks konfusi merupakan instrumen evaluasi yang digunakan untuk mengukur efektivitas model klasifikasi dengan cara membandingkan hasil prediksi dengan nilai sebenarnya dari data uji. Matrix ini menunjukkan perbandingan antara klasifikasi sebenarnya dengan klasifikasi yang diprediksi oleh model. Pada deteksi objek, confusion matrix membantu mengevaluasi efektivitas model dalam

mendeteksi dan mengklasifikasikan objek dengan benar. Elemen utama dalam confusion matrix untuk object detection meliputi:

- a) True Positive (TP) : Deteksi yang benar dari bounding box ground-truth. Ini berarti model berhasil mengidentifikasi dan mengklasifikasikan objek yang benar-benar ada dalam gambar dengan tepat.
- b) False Positive (FP) : Deteksi yang salah dari objek yang tidak ada atau deteksi yang salah tempat dari objek yang sebenarnya ada. Ini menunjukkan kasus di mana model menghasilkan deteksi palsu atau keliru menempatkan bounding box pada objek yang ada.
- c) False Negative (FN) : Bounding box ground-truth yang tidak terdeteksi oleh model. Ini menunjukkan objek yang ada dalam gambar tetapi tidak berhasil dikenali atau dideteksi oleh model.



Gambar 2. 7 Ilustrasi True Positive (TP), False Positive (FP), dan False Negative (FN) pada deteksi objek[30]

Pada **Gambar 2.7** tidak ada elemen True Negative (TN), karena hasil TN tidak berlaku karena terdapat jumlah bounding box yang tidak terdeteksi dalam gambar yang tidak terbatas. Dengan kata lain, tidak mungkin untuk mengidentifikasi semua area dalam gambar yang tidak mengandung objek sebagai True Negative karena jumlah potensi area yang harus dipertimbangkan adalah tak terhingga[31]. Dengan menganalisis TP, FP, dan FN, kita dapat mengukur seberapa baik model mendeteksi objek yang ada, mengidentifikasi kesalahan dalam deteksi, dan meningkatkan performa model untuk mengurangi deteksi yang salah serta meningkatkan kemampuan deteksi objek yang sebenarnya ada.

a) Presisi

Presisi adalah metrik yang mengukur kemampuan model untuk hanya mengidentifikasi objek yang relevan. Dalam deteksi objek, presisi menunjukkan persentase prediksi positif yang benar dari total prediksi positif yang diklaim oleh model [32]. Secara matematis, presisi dihitung dengan Persamaan:

$$Presisi = \frac{TP}{TP + FP} \quad (2.1)$$

Presisi merepresentasikan tingkat ketepatan model dalam melakukan identifikasi, dengan berfokus pada kemampuannya menghindari deteksi yang tidak seharusnya (false positive). Semakin tinggi nilai presisi, semakin sedikit jumlah deteksi palsu yang dihasilkan model, menunjukkan bahwa model lebih efektif dalam meminimalkan kesalahan deteksi. Namun, untuk mendapatkan gambaran menyeluruh tentang kinerja model, presisi sering dikombinasikan dengan metrik lain seperti recall dan F1 Score.

b) Recall

Recall adalah metrik evaluasi yang mengukur kemampuan model untuk mendeteksi semua objek yang relevan dalam sebuah gambar[33]. Pada deteksi objek, recall menggambarkan seberapa baik model dalam menemukan dan mengidentifikasi semua instance dari objek yang sebenarnya ada, tanpa melewatkan satu pun. Secara matematis, recall dihitung dengan Persamaan 2.2:

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

Recall memberikan gambaran tentang kemampuan model dalam mengurangi kesalahan deteksi dengan memastikan bahwa semua objek yang relevan berhasil ditemukan. Metrik ini

mengukur kemampuan sistem untuk meminimalisir kegagalan deteksi dengan berhasil menemukan proporsi maksimal dari objek yang relevan. Tingginya nilai recall mencerminkan kapasitas model untuk mengenali mayoritas objek yang ada, sehingga meminimalkan insiden *false negative* situasi dimana objek yang seharusnya terdeteksi justru terlewatkan.

c) F1-Score

F1-Score adalah metrik evaluasi dalam machine learning yang menggabungkan dua metrik penting, yaitu presisi (precision) dan recall, ke dalam satu nilai yang merepresentasikan keseimbangan antara keduanya. F1-Score dihitung sebagai rata-rata harmonis dari presisi dan recall, dengan **Persamaan 2.3**:

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \quad (2.3)$$

Nilai F1-Score berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan performa sempurna (presisi dan recall sama-sama tinggi). Di sisi berlawanan, nilai F1-Score yang mendekati atau mencapai 0 mengindikasikan kinerja yang sangat lemah, mencerminkan ketidakmampuan model dalam mencapai keseimbangan efektif antara presisi dan recall. Skor rendah ini menandakan bahwa model memiliki defisiensi signifikan baik dalam hal banyaknya false positive, false negative, atau kombinasi keduanya.

d) *Average Precision* (AP)

Average Precision (AP) adalah metrik yang digunakan untuk mengevaluasi kinerja model deteksi objek dengan mengukur seberapa baik model dalam mengidentifikasi dan mengklasifikasikan objek pada berbagai ambang batas IoU. AP dihitung sebagai area di bawah kurva Precision Recall (PR)[34]. Kurva ini dihasilkan dengan memvariasikan nilai ambang batas

IoU yang digunakan untuk menentukan apakah suatu prediksi dianggap sebagai True Positive atau tidak. Semakin besar area di bawah kurva ini, semakin baik performa model dalam mendeteksi objek. Secara matematis, AP dapat dihitung menggunakan persamaan berikut:

$$AP = \int_0^1 P(r)dr \quad (2.4)$$

Dimana $P(r)$ adalah presisi sebagai fungsi dari recall r . Dalam praktiknya, AP dihitung dengan merata-ratakan presisi pada beberapa titik recall diskrit yang dihasilkan saat ambang batas bervariasi. Nilai AP untuk setiap kelas objek dalam dataset kemudian dirata-ratakan untuk menghasilkan Mean Average Precision (mAP) yang memberikan gambaran umum tentang kinerja model di seluruh kelas yang ada. Nilai AP berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan kinerja model yang lebih baik.

e) *Mean Average Precision (mAP)*

Mean Average Precision (mAP) adalah metrik yang digunakan untuk mengevaluasi performa keseluruhan sistem deteksi objek di seluruh kelas dalam dataset. mAP dihitung dengan merata-ratakan nilai Average Precision (AP) untuk setiap kelas yang mencerminkan efektivitas model dalam mendeteksi berbagai jenis objek[35]

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.5)$$

dimana N adalah jumlah kelas dalam dataset dan AP_i adalah Average Precision untuk kelas ke- i . mAP memberikan gambaran umum tentang kinerja model dalam mendeteksi berbagai jenis objek. Nilai mAP yang tinggi menunjukkan bahwa model memiliki performa deteksi yang baik dengan tingkat kesalahan yang rendah, sehingga metrik ini sangat berguna untuk membandingkan dan meningkatkan model deteksi objek.

f) Loss

Loss merupakan metrik fundamental dalam sistem deteksi objek yang mengkuantifikasi disparitas antara prediksi yang dihasilkan model dengan ground truth atau nilai aktual. Fungsi loss berperan sebagai kompas yang mengarahkan proses pembelajaran model, menyediakan sinyal numerik tentang seberapa akurat prediksi saat ini dan bagaimana model harus menyesuaikan parameter parameternya. Berikut beberapa jenis loss yang digunakan:

- Box Loss

Box loss adalah jenis bounding box regression loss yang berfungsi untuk membantu model memahami posisi dan ukuran yang tepat dari bounding box yang mengelilingi objek yang terdeteksi. Loss ini mengukur kesalahan dalam prediksi koordinat dan dimensi bounding box dibandingkan dengan ground truth dengan tujuan mengurangi perbedaan antara kotak yang diprediksi dan yang sebenarnya.

- Class Loss

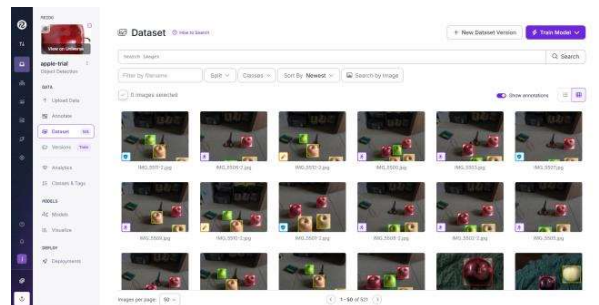
Class loss berkaitan dengan akurasi klasifikasi objek yang terdeteksi. Tujuan utama class loss adalah memastikan bahwa setiap objek yang terdeteksi diklasifikasikan dengan benar ke dalam salah satu kategori yang telah ditentukan. Loss ini diukur berdasarkan perbedaan antara probabilitas kelas yang diprediksi dan label kelas ground truth.

- Distribution Focal Loss (DFL)

Distribution focal loss berfungsi untuk meningkatkan akurasi model, terutama dalam skenario deteksi objek yang kompleks dengan menekankan pada objek yang sulit dideteksi. Loss ini menyesuaikan bobot kesalahan secara berbeda berdasarkan tingkat kesulitan, sehingga model dapat lebih baik dalam mengestimasi kategori objek. Loss ini juga mengatasi masalah ketidakseimbangan kelas dengan memberikan fokus lebih pada contoh-contoh yang sulit diklasifikasikan.

2.4 Roboflow

Roboflow adalah platform end-to-end yang dirancang khusus untuk memudahkan pengembangan model computer vision, dengan fokus pada pengelolaan dataset, preprocessing gambar, dan deployment model. Platform ini menyediakan serangkaian alat yang komprehensif untuk mengotomatisasi dan mengoptimalkan alur kerja dalam proyek computer vision, termasuk sistem deteksi objek seperti YOLOv8 yang Anda implementasikan untuk deteksi dan klasifikasi apel.



Gambar 2. 8 Roboflow

Seperti pada **Gambar 2.8** Fitur utama Roboflow mencakup manajemen dataset yang intuitif, memungkinkan pengguna untuk mengimpor, mengorganisir, dan memberi anotasi pada gambar dengan efisien. Platform ini menawarkan kemampuan anotasi yang kuat untuk berbagai tugas computer vision seperti deteksi objek, segmentasi, dan klasifikasi. Roboflow juga menyediakan berbagai teknik augmentasi data yang dapat dikonfigurasi

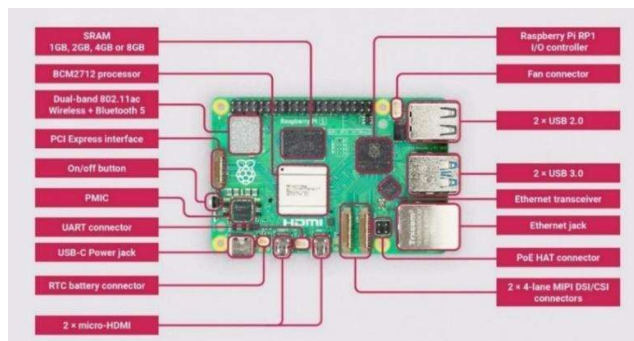
untuk meningkatkan keragaman dataset, termasuk rotasi, penskalaan, pemotongan, dan penyesuaian kecerahan yang sangat berguna untuk melatih model deteksi apel yang robust terhadap variasi kondisi pengambilan gambar.

2.5 Ultralytics

Ultralytics adalah perusahaan yang dikenal karena pengembangan model deteksi objek berbasis algoritma YOLO (You Only Look Once). Didirikan oleh Glenn Jocher, Ultralytics telah merilis beberapa versi dari YOLO, termasuk YOLOv11. Model-model ini dirancang untuk memberikan kinerja tinggi dalam deteksi objek secara *real time*, dengan fokus pada efisiensi komputasi dan akurasi yang lebih baik dibandingkan versi sebelumnya[36]. YOLOv8, yang dirilis pada Januari 2023, membawa inovasi tambahan dengan menghilangkan penggunaan anchor boxes dalam prediksi, sehingga mengurangi beban komputasi. Model ini juga menggunakan modul C2f untuk meningkatkan akurasi melalui pemrosesan fitur kontekstual.

2.6 Raspberry Pi 5

Raspberry Pi 5 merupakan single board computer (SBC) generasi terbaru yang dirancang khusus untuk komputasi berperforma tinggi dengan konsumsi daya rendah. Raspberry Pi 5 memiliki spesifikasi yang lebih tinggi dengan CPU/VideoCore VII GPU yang telah ditingkatkan, yang membuatnya ideal untuk aplikasi AI di edge computing. Pada **Gambar 2.9** memperlihatkan pemetaan visual dari seluruh komponen utama pada papan sirkuit Raspberry Pi 5.



Gambar 2. 9 Board Diagram Raspberry Pi 5[37]

Perangkat ini memiliki ukuran yang kompak setara kartu kredit namun mampu menjalankan sistem operasi lengkap seperti Linux Ubuntu atau Raspberry Pi OS. Raspberry Pi 5 memungkinkan implementasi computer vision *real time* seperti object detection langsung pada perangkat, mengeliminasi kebutuhan akan sumber daya cloud. Raspberry Pi 5 memiliki konsumsi daya rendah dengan performa tinggi, membuatnya sempurna untuk deployment AI di edge computing dan Raspberry Pi 5 dapat menggunakan berbagai sistem operasi dan menyediakan GPIO pins untuk integrasi mudah dengan sensor, aktuator, dan komponen hardware lainnya.

Tabel 2. 2 Spesifikasi Raspberry PI 5

Parameter	Keterangan
GPU	VideoCore VII GPU OpenGL ES 3.1, Vulkan 1.2 4Kp60 HEVC decode H.264 decode dan encode hingga 1080p30
CPU	Broadcom BCM2712, Quad-core Cortex-A76 (ARM v8) 64-bit SoC @ 2.4GHz
Memory	4GB dan 8GB LPDDR4X-4267 SDRAM

2.7 Hailo-8 (13 TOPS)



Gambar 2. 10 Hailo 8

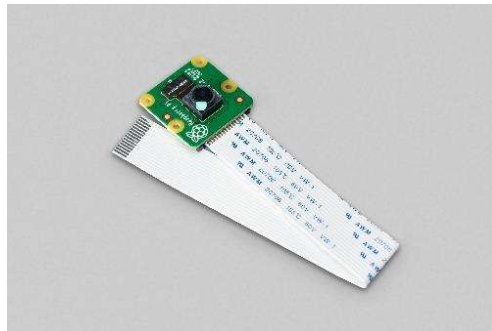
Hailo-8 adalah modul akselerator kecerdasan buatan (*AI accelerator*) dari Hailo AI yang dirancang khusus untuk aplikasi edge computing.

Perangkat ini berfungsi mempercepat proses inferensi *deep learning* pada papan pengembangan seperti Raspberry Pi 5, sehingga memungkinkan pemrosesan model visi komputer berperforma tinggi dengan konsumsi daya rendah. Hailo-8 memiliki kemampuan komputasi hingga 13 TOPS (Tera Operations Per Second) dengan efisiensi energi yang sangat baik, sehingga dapat menjalankan model neural network kompleks secara *real time*.

Tabel 2. 3 Spesifikasi Hailo-8

Parameter	Deskripsi
Prosesor AI	Hailo-8 Neural Network Processor
Performa	13 TOPS
Antarmuka	M.2 Key-M (PCIe Gen3 x2)
Konsumsi Daya	1,5 – 3 W
Dukungan Software	HailoRT, Hailo TAPPAS, integrasi TensorFlow / PyTorch / ONNX

2.8 Camera Pi V3



Gambar 2. 11 Camera Pi V3

Camera Pi V3 Wide adalah modul kamera resmi generasi ketiga dari Raspberry Pi Foundation yang menggunakan sensor Sony IMX708 dengan sudut pandang lebih lebar. Varian ini dirancang untuk aplikasi yang membutuhkan cakupan area yang luas, seperti computer vision, robotika, pemantauan, dan sistem otomatisasi. Dibandingkan dengan varian standar, V3 Wide mampu menangkap area lebih besar tanpa perlu menambah jarak

kamera, sehingga sangat cocok digunakan pada conveyor atau sistem sortir buah yang membutuhkan deteksi objek dalam area lebar. Selain itu, Camera Pi V3 Wide tetap membawa semua keunggulan generasi ketiga, yaitu autofocus, dukungan HDR, serta kinerja yang lebih baik pada kondisi cahaya rendah.

Tabel 2. 4 Spesifikasi Camera

Parameter	Deskripsi
Resolusi Gambar	4608 × 2592 piksel (12 Megapiksel)
Sensor	Sony IMX708
Interface	CSI-2 (Camera Serial Interface) melalui flat ribbon (22-pin pada Raspberry Pi 5, adaptor tersedia untuk 15-pin lama)
Field of View	102° diagonal (wide-angle)
Fitur	Autofocus, HDR, performa cahaya rendah lebih baik

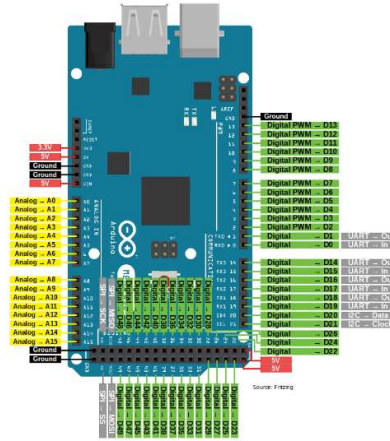
Pada penggunaannya, Raspberry Pi Camera v3 Wide dipasang tegak lurus di atas lintasan. Ketinggian dipilih agar lebar bidang pandang menutup seluruh lebar conveyor dengan sedikit margin kiri–kanan sekaligus menjaga ketajaman citra. Untuk menghitung ketinggian kamera kita bisa menggunakan persamaan berikut

$$H = \frac{W}{2 \cdot \tan\left(\frac{HFOV}{2}\right)} \quad (2.6)$$

Pada **Persamaan 2.6**, Untuk Raspberry Pi Camera V3 yang memiliki HFOV (Horizontal Field of View) sebesar 66° sesuai datasheet, tinggi kamera dari conveyor dapat dihitung menggunakan prinsip trigonometri. W merupakan lebar conveyor yang dalam kasus ini adalah 40 cm, sedangkan HFOV adalah sudut pandang kamera pada arah mendatar. Dimana H adalah tinggi kamera dari permukaan conveyor. Persamaan ini memberikan titik

awal untuk mengetahui tinggi kamera agar dapat menangkap semua apel sebagai objek saat berada di conveyor.

2.9 Arduino Mega 2560



Gambar 2. 12 Arduino [38]

Arduino Mega 2560 adalah sebuah board mikrokontroler yang berbasis ATmega2560. Pada **Gambar 2.12** Arduino Mega memiliki 54 pin input/output digital yang mana 15 pin dapat digunakan sebagai output PWM, 16 input analog, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino Mega memiliki kelebihan tersendiri dibanding board mikrokontroler yang lain. Selain bersifat open source, Arduino Mega juga mempunyai bahasa pemrogramannya sendiri yang berupa bahasa C++. Dalam board Arduino Mega sudah terdapat bootloader yang terintegrasi melalui port USB sehingga memudahkan kita ketika memprogram mikrokontroler.

Tabel 2. 5 Spesifikasi Arduino

Parameter	Keterangan
Tegangan Input	7-12V
Pin I/O Digital	54 (15 di antaranya mendukung PWM)
Pin Input Analog	16
Pin Input Digital	54
Arus DC per Pin I/O	20 mA
Dimensi	101.5 mm × 53.4 mm

Pada penelitian ini, Arduino Mega 2560 digunakan untuk mengendalikan servo motor. Servo motor, yang digunakan untuk kontrol sudut presisi (0° – 180°), bekerja dengan sinyal PWM (frekuensi 50 Hz, periode 20 ms) yang dihasilkan oleh pin PWM Arduino, memungkinkan pengaturan posisi akurat melalui library Servo.h.

2.10 Motor Servo MG996R



Gambar 2. 13 Servo MG996R[39]

Servo MG996R adalah servo motor berbasis gear logam yang dirancang untuk aplikasi yang membutuhkan torsi tinggi dan durabilitas, sering digunakan dalam robotika, model RC, dan proyek otomatisasi. Motor ini juga dilengkapi dengan sistem umpan balik (feedback), di mana posisi aktual poros motor akan dikembalikan sebagai sinyal ke rangkaian kendali. Hal ini memungkinkan Motor Servo untuk mengatur dan mempertahankan posisi sudut secara presisi sesuai dengan sinyal kendali yang diterima. Struktur utama dari Motor Servo terdiri atas motor DC, rangkaian gear, potensiometer sebagai sensor posisi, dan rangkaian kontrol internal. Pengaturan sudut dilakukan dengan mengatur lebar pulsa sinyal (PWM) yang diberikan pada pin kendalinya. Selain itu, servo dipilih karena memiliki torsi yang memadai untuk mengubah arah apel ke wadah. Adapun spesifikasi servo yang digunakan tertera pada **Tabel 2.6**

Tabel 2. 6 Spesifikasi Servo

Parameter	Deskripsi
Tegangan Operasi	4.8V – 7.2V DC
Kecepatan	0.19 detik/60° jika 4.8V 0.14 detik/60° jika 6.0V
Torsi Maksimal	9.4 kg·cm/0.92 Nm jika 4.8V 11 kg·cm/1.08 Nm jika 6.0V

2.11 Motor Power Window

**Gambar 2. 14** Motor Power Window[40]

Motor power windows yang didesain khusus untuk aplikasi conveyor memiliki karakteristik dan spesifikasi yang berbeda dari motor yang digunakan dalam kendaraan. Motor ini dirancang untuk menggerakkan sistem pengangkut barang atau material secara kontinyu dan efisien. Motor ini pada dasarnya adalah motor DC yang dilengkapi dengan gearbox terintegrasi. Gearbox inilah yang menjadi kunci mengapa motor power window cocok untuk aplikasi penggerak seperti conveyor. Dengan karakteristik ini, motor power window mampu memberikan daya dorong yang cukup untuk menggerakkan sabuk conveyor dan beban yang relatif ringan di atasnya, meskipun desain awalnya adalah untuk mengangkat dan menurunkan kaca jendela mobil secara intermiten. Pemilihan motor ini didasarkan karena memiliki torsi yang cukup tinggi sehingga memungkinkan pergerakan yang stabil dalam menggerakkan conveyor *belt*. Selain itu, motor ini mudah

dikendalikan dengan Dimmer yang dapat dikontrol manual menggunakan potensio. Berikut merupakan spesifikasi dari Motor DC *Power Window* yang digunakan terdapat pada **Tabel 2.7**.

Tabel 2. 7 Spesifikasi Motor Power Window	
Parameter	Deskripsi
Tegangan Masukan	12V DC
<i>Weight</i>	290 gram
<i>Load Current</i>	4.5 A
Kecepatan Putaran Motor	100 RPM
<i>Peak Current</i>	10A
<i>Rated Torque</i>	3 Nm

Berdasarkan data yang diperoleh dari **Tabel 2.7**, dapat dilakukan analisis menggunakan model matematika untuk menentukan kebutuhan daya motor dalam menggerakkan sistem konveyor. Perhitungan ini tidak hanya bertujuan untuk menentukan daya motor yang diperlukan, tetapi juga berfungsi sebagai dasar untuk menghitung torsi motor dan efisiensi sistem transmisi dengan perhitungan daya listrik sebagai berikut.

$$P = I.V \quad (2.7)$$

Persamaan 2.7 menjelaskan mengenai daya (P) yang digunakan motor dengan dinyatakan adalah hasil dari arus mengalir (I) dikalikan terhadap tegangan (V). Persamaan ini menunjukkan bahwa daya (P) merupakan hasil perkalian antara tegangan (V) yang diberikan dan arus listrik (I) yang mengalir. Semakin besar nilai arus atau tegangan yang disuplai ke motor, semakin besar pula daya masukan listriknya. Selanjutnya dengan dihasilkan nilai daya pada motor, maka dapat dicari torsi (τ) dengan pendekatan persamaan sebagai berikut.

$$\tau = \frac{P}{2\pi N} \quad (2.8)$$

Untuk menghitung nilai torsi (τ) pada motor yang digunakan untuk menentukan beban maksimum yang dapat diangkat, digunakan Persamaan dengan membagi daya (P) terhadap hasil perkalian kecepatan rotasi (N). Kecepatan rotasi awalnya dalam satuan putaran per detik (rps) dikonversi terlebih dahulu ke putaran per menit (rpm) dengan mengalikannya dengan 60. Setelah dikonversi, nilai tersebut kemudian dikalikan dengan 2π , di mana π diambil sebesar 3,14. Jika sudah mencari nilai torsi (τ) maka dapat digunakan untuk mencari nilai torsi lainnya untuk motor sebagai penggerak conveyor sebagai berikut.

$$\tau = r \times F \quad (2.9)$$

Dibutuhkan nilai r , yaitu jarak jari-jari vektor antara pusat putaran motor (tempat torsi diukur) dan titik di mana gaya bekerja (titik beban), yang dalam hal ini menggunakan sistem transmisi berupa pulley dan belt. Besarnya gaya (F) yang dihasilkan akan menciptakan gaya linear pada permukaan pulley. Dengan mengetahui nilai jari-jari dan torsi (τ), maka selanjutnya dapat dihitung total beban dalam satuan kilogram (kg) yang sanggup diangkat oleh motor, dengan memperhitungkan pengaruh percepatan gravitasi (g) sebesar $9,81 \text{ m/s}^2$. Beban maksimum yang dapat diangkat oleh motor sebagai berikut:

$$m = \frac{F}{g} \quad (2.10)$$

Nilai gaya (F) dalam persamaan 2.9 diperoleh berdasarkan hasil dari persamaan 2.8 yang digunakan untuk menghitung gaya angkat. Gaya ini kemudian dimanfaatkan untuk menentukan beban maksimum yang mampu diangkat oleh motor. Persamaan (2.9) digunakan untuk menghitung massa (m), di mana gaya (F) yang dihasilkan berbanding lurus terhadap massa dan dibagi dengan percepatan gravitasi (g) sebesar $9,81 \text{ m/s}^2$.

$$P_{out} = \tau \times \omega \quad (2.11)$$

Persamaan 2.11 menjelaskan mengenai daya keluar (Pout) dari motor yang dihasilkan berkaitan dengan torsi (τ) dan kecepatan sudut (ω) pada saat motor bekerja. Semakin tinggi nilai torsi (τ) dan kecepatan sudut utama (ω) maka semakin besar daya keluar dihasilkan). Untuk mencari nilai kecepatan sudut utama (ω) dengan melihat **Persamaan 2.12** dan **Persamaan 2.13** berikut ini.

$$\omega_1 = \frac{2\pi N}{60} \quad (2.12)$$

Dalam Persamaan ini, kecepatan sudut utama (ω_1) diperoleh dari hasil perkalian 2π (dengan π bernilai 3,14) dengan kecepatan putaran motor (N) dalam satuan putaran per detik, kemudian dibagi dengan 60 untuk mengkonversi ke dalam putaran per menit. Persamaan (2.11) untuk mencari kecepatan sudut juga dapat digunakan untuk mencari persamaan transmisi rotasi dengan persamaan sebagai berikut.

$$\omega_2 = \frac{\omega_1 R_1}{R_2} \quad (2.13)$$

Persamaan ini digunakan untuk mencari kecepatan sudut output (ω_2) yang dihasilkan dari transmisi. Nilainya diperoleh dengan mengalikan kecepatan sudut utama (ω_1) dengan jari-jari roda penggerak (R_1), lalu membaginya dengan jari-jari roda yang digerakkan (R_2). Ini mencerminkan hubungan terbalik antara kecepatan sudut dan ukuran jari-jari pulley pada sistem.

$$Pin = V \times I \quad (2.14)$$

Daya masuk (Pin) diperoleh dari hasil perkalian antara tegangan terukur (V) dan arus terukur (I). Semakin tinggi nilai tegangan dan arus, maka daya masuk (Pin) akan semakin besar. Untuk menghitung efisiensi (η), bagi nilai daya keluar (Pout) dengan daya masuk (Pin), kemudian kalikan hasilnya dengan 100%, sesuai dengan **Persamaan 2.15**.

$$\eta = \frac{P_{out}}{P_{in}} \times 100\% \quad (2.15)$$

Nilai perhitungan motor juga akan terpengaruh oleh sistem transmisi yang beroperasi. Dalam penelitian ini, digunakan transmisi berbasis pulley dan belt, dengan mempertimbangkan rasio pulley terhadap kecepatan poros utama dan poros yang digerakkan, serta panjang belt yang diperlukan.

2.12 Pulley and Belt

Pada sistem konveyor, terdapat mekanisme penggerak yang terdiri dari pulley dan V-belt, yang berfungsi untuk mentransmisikan gerakan rotasi dari motor. Kecepatan putaran dapat diatur, baik dipercepat maupun diperlambat, yang pada akhirnya juga mempengaruhi torsi yang dihasilkan. Torsi ini bergantung pada rasio diameter antara pulley penggerak dan pulley yang digerakkan[41].



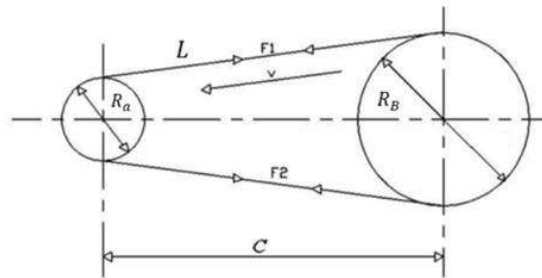
Gambar 2. 15 Pulley dan Belt[42]

Bahan dasar V-belt biasanya terbuat dari karet, yang memberikan fleksibilitas tinggi dan memiliki desain berbentuk trapesium pada penampangnya. Pulley sendiri berfungsi sebagai media untuk mentransmisikan daya dari sumber penggerak menuju komponen yang akan digerakkan. Sementara itu, V-belt bertugas meneruskan rotasi dari pulley motor ke pulley pada poros penerima. Untuk menentukan diameter pulley pada poros yang akan digerakkan, diperlukan persamaan perhitungan yang mempertimbangkan perbandingan kecepatan dan torsi guna memastikan efisiensi dalam sistem transmisi daya. Diperlukan persamaan untuk menghitung hubungan antara diameter pulley dan kecepatan motor, yang

berfungsi untuk mempercepat atau memperlambat transmisi putaran. Hubungan ini dijelaskan melalui persamaan berikut:

$$\frac{R_a}{R_b} = \frac{V_b}{V_a} \quad (2.16)$$

Prinsip kerja pulley didasarkan pada sistem transmisi daya. Ketika pulley penggerak dan pulley yang digerakkan terhubung melalui sabuk, perubahan ukuran diameter masing-masing akan memengaruhi kecepatan putaran. Nilai kecepatan pulley yang digerakkan V_b diperoleh dari hasil pembagian antara jari-jari pulley penggerak R_a dengan jari-jari pulley yang digerakkan R_b , kemudian dikalikan dengan kecepatan pulley penggerak V_a .



Gambar 2. 16 Scematik Pulley and Belt

Selain itu, panjang sabuk (v-belt) yang diperlukan juga bergantung pada rasio pulley dan kecepatan motor, serta dipengaruhi oleh parameter seperti daya (HP) dan kecepatan putaran sistem. Langkah selanjutnya adalah menghitung panjang sabuk yang dibutuhkan dalam sistem transmisi pulley pada conveyor, dengan menggunakan Persamaan berikut:

$$L = 2 \cdot c + \left[R_a + R_b \cdot \frac{\pi}{2} \right] + \left[\frac{(R_a + R_b)^2}{4 \cdot c} \right] \quad (2.17)$$

Dengan pernyataan panjang *belt* (L), jarak sumbu (c) antara kedua *pulley*, jari-jari *pulley* penggerak (R_a), jari-jari *pulley* yang digerakkan (R_b), dan nilai konstanta π bernilai 3,14. **Persamaan 2.17** mencakup perhitungan panjang bagian lurus (*Straight Length*), panjang bagian melengkung (*Arc Length*), dan

koreksi panjang (*Correction Length*). Jika jarak sumbu c bertambah maka nilai L juga akan meningkat.

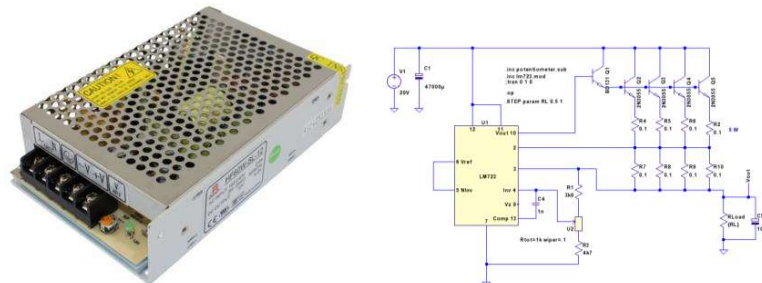
Setelah menentukan nilai kecepatan putar pulley yang digerakkan (Vb) serta kebutuhan panjang belt, aspek penting lainnya dalam memastikan sistem transmisi bekerja secara efisien pada proses sortir apel adalah perhitungan kecepatan linear (V). Sistem transmisi bergerak secara melingkar, di mana permukaan luar pulley bersinggungan langsung dengan belt. Oleh sebab itu, perhitungan kecepatan linear diperlukan untuk mengetahui laju perpindahan objek sepanjang lintasan lurus conveyor.

$$V = \pi . D . n \quad (2.18)$$

Dengan mengetahui nilai putaran pulley berdasarkan spesifikasi motor (n) dan diameter pulley yang digerakkan (D), kecepatan linear dapat diperoleh dengan mengalikan nilai π sebesar 3,14, diameter pulley, serta kecepatan putaran (n) yang telah dikonversi terlebih dahulu ke satuan RPS. Melalui proses tersebut, diperoleh nilai kecepatan linear (V) dalam satuan m/s.

2.13 Power Supply

Power Supply atau catu daya merupakan komponen elektronik yang menyuplai energi listrik ke perangkat yang terhubung. Secara fundamental, catu daya berfungsi sebagai penyearah dan filter karena mengonversi arus bolak-balik (AC) menjadi arus searah (DC). Dalam penelitian ini digunakan switching power supply 12 V 20 A, yang bekerja melalui empat tahap utama penyearahan, penyaringan, regulasi, dan proteksi internal.



Gambar 2. 17 *Power Supply*[43]

PSU ini dipilih karena mampu menyediakan arus besar secara kontinu, serta memiliki stabilitas tegangan yang baik untuk supply mesin sortir apel. Tegangan keluaran 12 V dari PSU selanjutnya didistribusikan jalur daya terpisah agar setiap kelompok aktuator memperoleh suplai yang sesuai dengan spesifikasi komponennya. Dengan desain seperti ini, sistem catu daya mampu memberikan suplai yang stabil meskipun seluruh aktuator bekerja secara bersamaan. Spesifikasi Power Supply disajikan pada **Tabel 2.8**.

Tabel 2. 8 Spesifikasi Power Supply

Parameter	Deskripsi
Output Voltage	12 Volt DC
Max Output Current	20 Ampere
Daya Maksimum (Power Output)	240 Watt
Jenis Output	DC Regulated Output

2.14 Stepdown DC-DC 300W 20A

**Gambar 2. 18** Stepdown DC-DC 20A

Buck converter adalah rangkaian DC-DC step-down yang menurunkan tegangan DC lebih tinggi menjadi lebih rendah secara efisien umumnya >85–90% sambil menjaga kestabilan tegangan melalui mekanisme pensaklaran cepat, induktor, dan kapasitor yang diatur oleh umpan balik. Dibanding penurun tegangan linear, buck menghasilkan panas lebih kecil pada arus

besar, sehingga cocok untuk beban dinamis. buck converter berfungsi menurunkan keluaran PSU 12 V 20 A menjadi rel 6,0 V DC teratur untuk memasok enam servo MG996R. Pendekatan ini memberikan tegangan kerja yang sesuai spesifikasi servo sekaligus cadangan arus (headroom) saat beberapa servo bergerak bersamaan—arus kerja per servo dapat mencapai ~0,5–1 A dan sesaat melonjak mendekati ~2 A.

Tabel 2. 9 Spesifikasi Buck Converter 20A

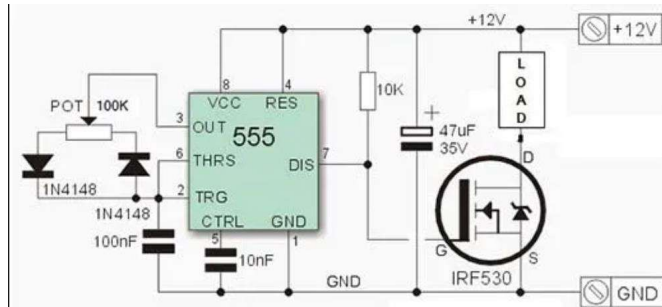
Parameter	Deskripsi
Tegangan Masukan	8–36 V DC
Tegangan Keluaran	4.8–8.0 V <i>adjustable</i>
Arus Maksimal	20 A
Suhu Operasi	80–85 °C

2.15 Dimmer DC 12V-40V

Dimmer DC adalah suatu rangkaian elektronika daya yang berfungsi untuk mengatur atau mengontrol tegangan DC (Direct Current) keluaran yang diberikan kepada beban, sehingga dapat mengendalikan daya yang diterima oleh beban tersebut. Dimmer DC bekerja dengan prinsip modulasi lebar pulsa (Pulse Width Modulation) untuk mengubah nilai rata-rata tegangan yang diberikan kepada beban tanpa mengubah sumber tegangan input.



Gambar 2. 19 Dimmer DC 12V-40V1



Gambar 2. 20 Schematik Dimmer DC 12V-40V[44]

Dimmer DC menggunakan komponen semikonduktor daya seperti MOSFET atau transistor daya sebagai saklar elektronik. Komponen ini akan di-switching dengan frekuensi tinggi untuk menghasilkan tegangan keluaran yang bervariasi. Dengan mengatur duty cycle dari sinyal PWM, maka nilai rata-rata tegangan keluaran dapat dikontrol dari 0% hingga 100% dari tegangan input. Untuk Spesifikasi dapat dilihat pada **Tabel 2.10**.

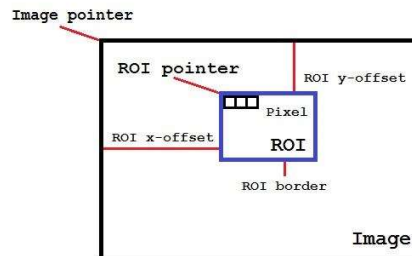
Tabel 2. 10 Spesifikasi Dimmer

Parameter	Deskripsi
Operating Voltage	DC 12V-40V
PWM Duty Cycle	0–100%
Control Input	Potensiometer
Maximum Current	10A
Rated current	8A

2.16 ROI (Region of Interest)

Region of Interest (ROI) merupakan suatu area tertentu pada citra digital yang dipilih untuk diproses lebih lanjut karena dianggap mengandung informasi penting yang relevan dengan tujuan analisis. Dalam sistem visi komputer, ROI digunakan untuk membatasi area pemrosesan hanya pada bagian citra yang menjadi fokus pengamatan, sementara area lain yang tidak relevan diabaikan. Pendekatan ini bertujuan untuk meningkatkan efisiensi

komputasi, mengurangi gangguan, serta meningkatkan akurasi hasil pemrosesan citra.



Gambar 2. 21 Simulasi Cara Kerja ROI

Pada sistem penyortiran buah apel berbasis computer vision, ROI digunakan untuk menentukan area lintasan apel pada conveyor yang menjadi fokus pengambilan citra dan analisis. Area ini dipilih sedemikian rupa agar apel berada pada posisi yang optimal untuk dideteksi dan diklasifikasikan berdasarkan tingkat kematangannya.

Secara sederhana, Region of Interest (ROI) umumnya berbentuk persegi panjang (*rectangular ROI*) pada citra digital. ROI ini dapat didefinisikan menggunakan empat parameter koordinat sebagai berikut:

$$ROI = (x1, y1, x2, y2) \quad (2.19)$$

Dimana, $x1, y1$ merupakan koordinat titik kiri atas dari ROI, dan $x2, y2$ merupakan koordinat titik kanan bawah dari ROI. Koordinat tersebut dinyatakan dalam sistem koordinat citra digital, di mana titik $(0,0)$ berada pada sudut kiri atas citra, sumbu x bertambah ke arah kanan, dan sumbu y bertambah ke arah bawah. Area citra yang berada di dalam batas koordinat tersebut selanjutnya diproses sebagai ROI, sedangkan area di luar ROI diabaikan. Pendefinisian ROI menggunakan pendekatan ini memungkinkan sistem untuk memfokuskan proses deteksi dan klasifikasi objek hanya pada area yang relevan, sehingga dapat meningkatkan efisiensi komputasi serta mengurangi gangguan dari latar belakang yang tidak berkaitan dengan objek utama.