

BAB II

METODE PENELITIAN

2.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan sejak disetujuinya judul penelitian pada bulan Januari hingga Oktober 2025. Lokasi pengambilan data dan penelitian dilakukan di Laboratorium *Artificial Intelligence*, Kampus Teknik, Universitas Hasanuddin. Lokasi penelitian ditunjukkan pada Gambar 10,



Gambar 10. Lokasi penelitian

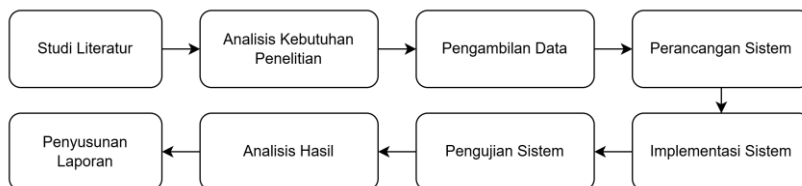
2.2 Instrumen Penelitian

Berikut instrumen atau alat yang digunakan dalam penelitian ini:

1. Perangkat lunak:
 - a. Logitech Capture
 - b. Roboflow
 - c. Colab
 - d. Kaggle
 - e. Google Spreadsheet
 - f. Microsoft Word
 - g. Microsoft PowerPoint
 - h. Chrome
 - i. Sistem Operasi Windows 11 64-bit
 - j. Mendeley Reference Manager
2. Perangkat keras:
 - a. Logitech Brio UltraHD
 - b. Lenovo Ideapad Gaming 3 15ARH7, *Processor* AMD Ryzen 5 7535HS with Radeon Graphic, *Graphic Card* NVIDIA® GeForce RTX™ 2050Ti, RAM 16GB.
 - c. Konveyor
 - d. *Belt conveyor*
 - e. *DC motor*
 - f. *Power supply*
 - g. *Tachometer*

- h. *LED Strip*
- i. *Box* kamera
- 3. Bahasa pemrograman
 - a. Python 3.11.3
- 4. *Library*:
 - a. Ultralytics v8.3.203 dikembangkan oleh Ultralytics LLC, menyediakan implementasi algoritma YOLO (You Only Look Once) untuk tugas seperti deteksi objek, segmentasi, dan klasifikasi citra.
 - b. Roboflow v1.2.9 dikembangkan oleh Roboflow, Inc., digunakan sebagai platform untuk manajemen, anotasi, dan persiapan dataset visi komputer.
 - c. Matplotlib v3.7.2 oleh John D. Hunter & Michael Droettboom (dan komunitas Matplotlib), digunakan untuk visualisasi data yang komprehensif, digunakan untuk membuat plot dan grafik statis.
 - d. Skimage (scikit-image) v0,25.0 dikembangkan oleh komunitas scikit-image. Berisi kumpulan algoritma untuk pemrosesan gambar. Modul *restoration* (bagian dari v0,25.0) secara khusus menyediakan fungsi untuk restorasi (pemulihan) gambar, seperti *denoising* (menghilangkan *noise*) dan *deconvolution*.
 - e. Pandas v.2.2.3 oleh tim The Pandas Development, digunakan untuk manipulasi dan analisis data, khususnya menyediakan struktur data berperforma tinggi dan mudah digunakan seperti *DataFrame*.
 - f. Numpy v1.18.5 oleh Travis E. Oliphant (dan komunitas Numpy) digunakan untuk komputasi numerik di Python, menyediakan dukungan fundamental untuk *array* dan matriks multidimensi besar serta fungsi matematis untuk menanganinya.
 - g. Opencv-python v4.10,0 dikembangkan oleh komunitas OpenCV dengan fungsi untuk memproses gambar dan video.
 - h. Albumentations v2.0,8 dikembangkan tim Albumentations, digunakan untuk augmentasi data gambar yang cepat dan fleksibel, sering diintegrasikan dalam *pipeline training* model *deep learning* untuk meningkatkan ketahanan model.

2.3 Tahapan Penelitian



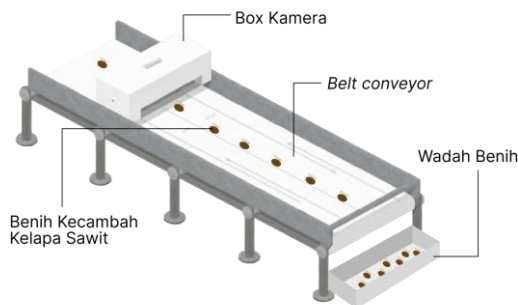
Gambar 11. Alur penelitian

Proses penelitian ini terdiri atas beberapa tahapan sebagaimana ditunjukkan pada Gambar 11. Tahap awal dimulai dengan studi literatur untuk menelaah berbagai referensi yang relevan terkait topik penelitian, yaitu metode klasifikasi benih dan perbaikan citra dengan mencari literatur dari jurnal internasional, artikel, dan laporan penelitian sebagai dasar penguatan teori. Selanjutnya dilakukan analisis kebutuhan guna merumuskan permasalahan, menetapkan tujuan penelitian, serta menentukan data dan metode yang akan digunakan.

Setelah itu dilakukan pengambilan data sebagai bahan utama yang bersumber dari Pusat Penelitian Kelapa Sawit yang kemudian diambil menggunakan konveyor dan akan digunakan dalam pengembangan sistem. Data yang diperoleh kemudian dimanfaatkan pada tahap perancangan sistem, yang mencakup penyusunan alur kerja serta identifikasi komponen yang diperlukan. Rancangan sistem tersebut selanjutnya diimplementasikan dan diuji untuk menilai kinerja serta tingkat akurasi. Hasil pengujian dianalisis lebih lanjut untuk mengevaluasi efektivitas sistem, dan keseluruhan proses penelitian disusun secara terstruktur dalam bentuk laporan akhir.

2.4 Teknik Pengambilan Data

Keseluruhan data yang digunakan dalam penelitian ini merupakan data primer yang diperoleh peneliti secara langsung dari lokasi pengambilan data. Dengan jumlah benih kecambah kelapa sawit sebanyak 250 butir, yang telah diseleksi langsung oleh Pusat Penelitian Kelapa Sawit untuk memisahkan benih kecambah normal dan abnormal.



(a)



(b)

Gambar 12. (a) Proses pengambilan data; (b) Contoh dataset

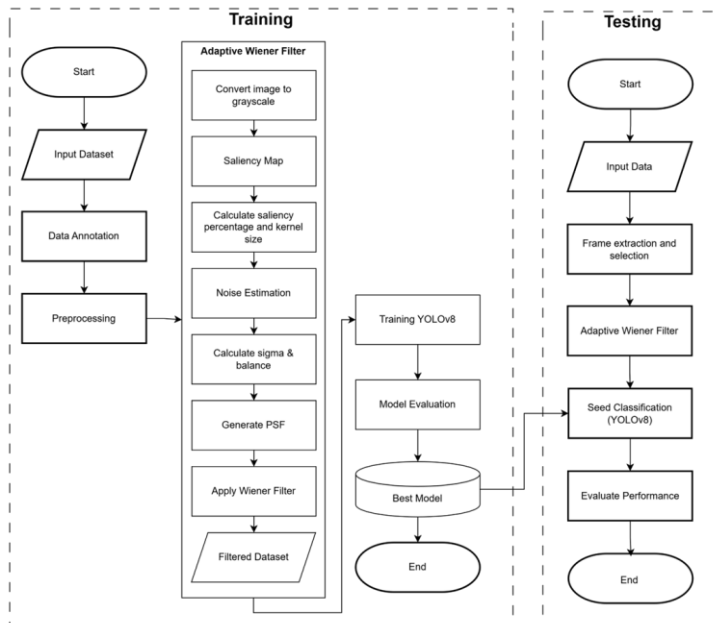
Pengambilan data dilakukan dengan menggunakan konveyor yang dilengkapi kamera Logitech Brio Ultra HD. Video direkam dengan posisi kamera tegak lurus terhadap objek yang diletakkan di atas *belt conveyor*. Jarak kamera dengan objek diatur pada 5cm dan 10cm dengan tambahan lampu LED *strip* sebagai

pencahayaan. *Belt* yang digunakan berwarna putih sebagai latar belakang objek. Benih kecambah kelapa sawit kemudian diletakkan dengan jarak tertentu satu sama lain di atas *belt conveyor*. Konveyor akan bergerak dengan kecepatan 60, 70, 80, 90, 100 *revolutions per minute* (RPM) menuju posisi kamera yang terpasang pada kotak. Hasil rekaman disimpan dalam format (.MP4) dengan 60 *frame per second* (FPS) dengan resolusi 1920×1080 piksel. Terdapat beberapa skenario pengambilan data yang diatur berdasarkan kombinasi jarak tinggi kamera terhadap objek dan kecepatan konveyor, ditunjukkan pada tabel 2. Data video yang telah direkam kemudian diekstraksi untuk diolah lebih lanjut, contoh dataset terdapat pada gambar 12.

Tabel 2. Skenario pengambilan data.

Skenario ke-	Kecepatan Konveyor (RPM)	Jarak kamera (cm)
1	60	5
2	60	10
3	70	5
4	70	10
5	80	5
6	80	10
7	90	5
8	90	10
9	100	5
10	100	10

2.5 Perancangan dan Implementasi Sistem



Gambar 13. Perancangan alur sistem

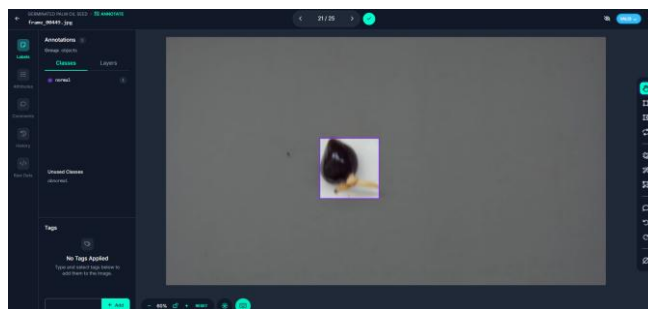
2.5.1 Tahap *training*

1. Input *Dataset*

Seluruh video yang telah diakuisisi kemudian diekstraksi untuk mengubah video menjadi *frame* citra. Keluaran yang dihasilkan berupa *frame* citra berformat RGB. *Frame* yang telah diekstrak selanjutnya akan diseleksi dan hanya menyimpan *frame* yang menampilkan seluruh bagian objek kecambah kelapa sawit secara utuh. Hasil seleksi *frame* per skenario akan disimpan ke dalam folder berdasarkan skenario kecepatan konveyor dan jarak kamera yang digunakan. Total citra yang diperoleh berjumlah 300 citra per skenario yang berisi objek benih kecambah kelapa sawit.

2. Anotasi Data

Setiap gambar kemudian dianotasi secara manual menggunakan platform Roboflow untuk menandai lokasi dan kelas objek benih. Setelah pelabelan, dataset kemudian dibagi menjadi data latih dan data validasi dengan rasio (80:20), yang menghasilkan 240 citra untuk data latih dan 60 citra untuk data validasi per skenario. Untuk memastikan model dilatih dengan berbagai skenario kecepatan konveyor dan jarak kamera, maka 240 citra per skenario digabung hingga menghasilkan 2400 citra untuk data latih. Data latih akan menjadi bahan model untuk mengenali dan mempelajari pola dari kedua kelas benih kecambah kelapa sawit, sementara data validasi akan digunakan untuk mengukur performa model terhadap data yang tidak dikenali dengan berbagai kombinasi skenario. Contoh pelabelan data menggunakan Roboflow ditunjukkan pada Gambar 14.



Gambar 14. Proses pelabelan data pada platform Roboflow

3. *Preprocessing*

Sebelum memasuki tahap *training* model dataset akan melalui tahapan *preprocessing* terlebih dahulu, dimulai dari augmentasi data, *filtering* menggunakan *Adaptive Wiener Filter*, dan *resizing*. Augmentasi data hanya dilakukan untuk data latih. Pada penelitian ini ada 6 jenis augmentasi data, yaitu *horizontal flip*, *vertical flip*, rotasi, penskalaan, translasi, *shear*. Setelah proses augmentasi, data yang sebelumnya berjumlah 2400 citra menjadi 3000 citra.

Horizontal flip. *Horizontal flip* mencerminkan gambar dari kiri ke kanan. Tujuan utamanya adalah melatih model agar invariabel terhadap arah objek. Dengan demikian, model diharapkan mampu mendeteksi benih secara konsisten, baik ketika benih menghadap ke kiri maupun ke kanan saat bergerak di atas *belt conveyor*.



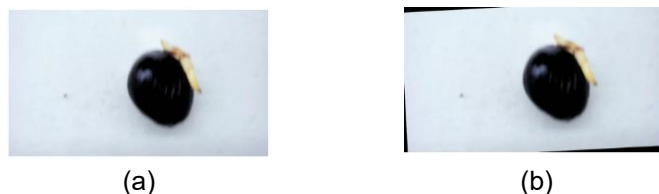
Gambar 15. (a) Original image; (b) Hasil *horizontal flip*

Vertical flip. *Vertical flip* mencerminkan gambar dari atas ke bawah. Augmentasi ini melatih model agar model tidak sensitif terhadap perubahan orientasi vertikal benih. Dengan begitu, model akan tetap dapat mengenali benih dengan baik, baik yang posisinya normal (misal, pucuk di atas) maupun yang terbalik di atas *belt conveyor*.



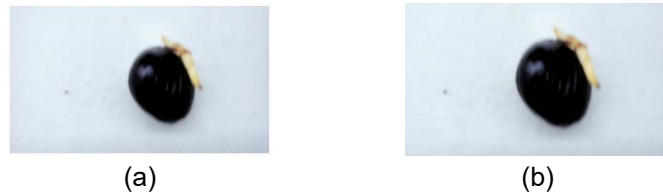
Gambar 16. (a) Original image; (b) Hasil *vertical flip*.

Rotasi. Rotasi memutar gambar sedikit pada porosnya (antara -3° dan $+3^\circ$). Ini bertujuan membuat model menjadi tangguh (*robust*) terhadap kemiringan kecil. Dalam praktiknya, ini mensimulasikan kondisi nyata di mana benih mungkin tidak berada dalam posisi lurus sempurna saat bergerak di atas *belt conveyor*.



Gambar 17. (a) Original image; (b) Hasil rotasi.

Penskalaan. Penskalaan memperbesar atau memperkecil gambar sedikit (antara 97% hingga 103%). Ini membuat model menjadi toleran terhadap variasi ukuran. Teknik ini melatih model untuk tetap mengenali benih, baik yang secara alami sedikit lebih besar/kecil maupun yang jaraknya sedikit berbeda dari kamera.



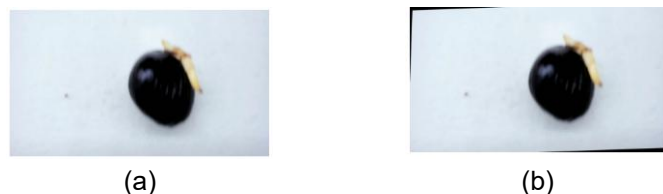
Gambar 18. (a) *Original image*; (b) Hasil penskalaan.

Translasi. Translasi menggeser posisi gambar hingga sekitar 3% dari posisi awal. Teknik ini melatih model untuk mendeteksi objek di berbagai lokasi dalam *frame*, tidak hanya di bagian tengah. Ini juga sangat membantu model agar dapat mengenali benih yang berada di pinggir *belt conveyor* atau yang terlihat terpotong sebagian.



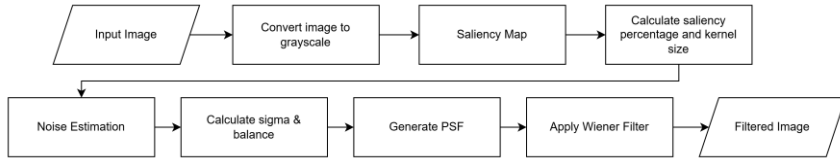
Gambar 19. (a) *Original image*; (b) Hasil translasi.

Shear. *Shear* memiringkan gambar seolah-olah didorong dari satu sisi, mengubah bentuknya. Augmentasi ini membuat model toleran terhadap distorsi perspektif. Ini berguna untuk mensimulasikan kondisi jika sudut pandang kamera tidak diposisikan tegak lurus sempurna di atas *belt conveyor*.



Gambar 20. (a) *Original image*; (b) Hasil *shear*.

Setelah proses augmentasi, selanjutnya dilakukan perbaikan citra *blur* dengan menggunakan *Adaptive Wiener Filter*. Proses ini dilakukan karena *blur* pada objek akan mempengaruhi proses klasifikasi kualitas benih kecambah kelapa sawit. Berikut langkah-langkah yang terlibat dalam proses *Adaptive Wiener Filter*:

Gambar 21. Alur *Adaptive Wiener Filter*

Estimasi *Adaptive Parameter*. Tahap krusial pertama adalah estimasi parameter adaptif yang terdiri dari tiga komponen utama: ukuran *kernel*, *sigma*, dan *balance*. Proses ini dimulai dengan mengonversi gambar menjadi *grayscale* lalu pembuatan *saliency map* menggunakan deteksi tepi *Canny* yang mengidentifikasi area-area penting dalam citra. Persentase saliency dihitung dari rasio piksel tepi terhadap total piksel, yang kemudian digunakan untuk menentukan ukuran *kernel* secara berbanding terbalik; semakin tinggi saliency (banyak detail), semakin kecil *kernel* yang dibutuhkan, dengan rentang antara 3 hingga 51 piksel. Estimasi *noise* dilakukan dengan menghitung standar deviasi dari histogram intensitas citra, yang kemudian digunakan untuk menentukan nilai *sigma* dan *balance* parameter. Estimasi *noise* dihitung menggunakan simpangan baku histogram intensitas karena simpangan baku merupakan ukuran penyebaran yang praktis untuk merepresentasikan level derau global pada citra, sekaligus ringan dan efisien dihitung tanpa proses iteratif. Pendekatan statistik berbasis varians/simpangan baku ini umum dipakai dalam pemodelan *noise* (misalnya *noise Gaussian*) dan sebagai parameter pada restorasi citra seperti *Wiener filtering*, sebagaimana dibahas dalam buku *Digital Image Processing* (Gonzalez, 2018).

Pembuatan PSF (*Point Spread Function*). Setelah menentukan parameter, PSF berbentuk *gaussian* akan dibuat berdasarkan ukuran *kernel* dan *sigma* yang telah dihitung. PSF ini merepresentasikan fungsi degradasi yang diasumsikan telah terjadi pada citra. Fungsi *Gaussian* dibuat menggunakan *meshgrid* untuk menghasilkan *kernel* 2D yang simetris, kemudian dinormalisasi agar jumlah seluruh elemennya sama dengan satu, sehingga total area di bawah kurva bernilai 1.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (7)$$

Di mana:

$G(x, y)$: nilai intensitas *kernel* pada titik (x, y)

σ : standar deviasi

e : bilangan Euler (~ 2.71828)

$\frac{1}{2\pi\sigma^2}$: normalisasi matematis agar total area di bawah kurva 1

Penerapan Wiener Filter. Citra masukan dinormalisasi ke rentang 0 - 1 untuk memastikan stabilitas numerik dan konsistensi dalam perhitungan matematis pada domain frekuensi. Karena citra memiliki tiga *channel* warna (RGB), maka proses *Wiener Filter* dilakukan pada setiap *channel* warna secara terpisah. Untuk setiap *channel*, fungsi *Wiener* diaplikasikan dengan parameter citra, PSF, dan balance (K). Secara matematis *Wiener Filter* di domain frekuensi dirumuskan sebagai berikut:

$$\hat{F}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + K} G(u, v) \quad (8)$$

Di mana:

$\hat{F}(u, v)$: spektrum (FFT) dari citra hasil restorasi (estimasi citra tajam) pada frekuensi (u, v) .

(u, v) : indeks frekuensi (koordinat di domain frekuensi hasil FFT)

$G(u, v)$: FFT dari citra masukan pada frekuensi (u, v)

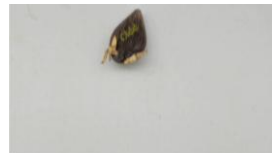
$H(u, v)$: FFT dari PSF pada frekuensi (u, v) .

K: : parameter keseimbangan (balance parameter) yang mengontrol tingkat regularisasi pada proses pemulihan citra.

Berikut merupakan gambar sebelum dan sesudah melalui proses *Adaptive Wiener Filtering*:



(a)



(b)

Gambar 22. (a) *Original image*; (b) *Filtered image*.

4. Training YOLOv8

YOLOv8 dirancang untuk memberikan kinerja mutakhir (*cutting-edge performance*) dalam hal akurasi dan kecepatan. Struktur arsitektur YOLOv8 secara umum dibagi menjadi tiga bagian utama: *Backbone*, *Neck*, dan *Head*:

Backbone. *Backbone* bertindak sebagai ekstraktor fitur utama dalam arsitektur pembelajaran mendalam, dengan tugas mengekstraksi fitur berbeda pada berbagai tingkat resolusi. Arsitektur *Backbone* tersusun dari lapisan konvolusi awal (seringkali dengan stride 2 untuk mengurangi resolusi spasial, misalnya dari 640x640 menjadi 320x320), dan serangkaian blok C2f. Blok C2f sendiri terdiri dari blok konvolusi (yang menggabungkan lapisan konvolusi 2D, normalisasi *batch* 2D, dan fungsi aktivasi SiLU) serta blok *bottleneck*.

Neck. *Neck* berfungsi untuk menggabungkan fitur-fitur yang diperoleh dari berbagai lapisan *Backbone*. Fitur gabungan ini menjadi lebih kaya dan fleksibel untuk mendukung prediksi yang akurat. Komponen penting di sini adalah blok *Spatial Pyramid Pooling Fast* (SPPF), yang ditempatkan setelah lapisan konvolusi terakhir pada *Backbone*. Fungsi utama SPPF, yang merupakan modifikasi dari SPP dengan kecepatan yang lebih tinggi, adalah menghasilkan representasi spasial objek dengan ukuran bervariasi tanpa perlu mengubah ukuran citra. Proses penggabungan fitur melibatkan penggunaan lapisan *Upsample* untuk meningkatkan resolusi peta fitur, diikuti oleh operasi *Concatenation* (*concat*) yang menjumlahkan jumlah channel sambil mempertahankan resolusi spasial. Blok C2f juga dimanfaatkan dalam struktur *Neck* ini.

Head. *Head* bertanggung jawab untuk menghasilkan prediksi akhir, yang mencakup kelas objek dan wilayah *bounding box*. YOLOv8 mengadopsi fitur utama berupa *Anchor-free Split Ultralytics Head*, yang berarti model ini tidak lagi mengandalkan kotak jangkar (*anchor boxes*) yang telah ditentukan sebelumnya, sebuah perubahan yang diklaim berkontribusi pada akurasi yang lebih baik dan proses deteksi yang lebih efisien. Deteksi sesungguhnya terjadi di dalam *Detect Block*, yang terbagi menjadi dua jalur prediksi yang berbeda: satu jalur untuk prediksi *bounding box* dan jalur lainnya untuk prediksi kelas. *Detect Block* ini juga dispesialisasikan untuk mendeteksi objek dari berbagai ukuran, termasuk objek kecil, sedang, dan besar.

Training dilakukan melalui platform Kaggle yang menawarkan akses ke sumber daya komputasi berbasis GPU guna mempercepat proses

pembelajaran *deep learning*. *Training* dilakukan dengan 3 skenario kombinasi *hyperparameter*, sebagai berikut:

Tabel 3. Skenario *hyperparameter training* model YOLOv8n.

Skenario ke-	Image Size	Agnostic NMS	IoU	Close_mosaic
1	640	False	0,7	10
2	1024	True	0,3	10
3	1024	True	0,3	0

Dalam penelitian ini, model YOLOv8n dilatih menggunakan tiga skenario *hyperparameter* sebagaimana ditunjukkan pada Tabel 3. Tujuan pengaturan ini adalah untuk menilai pengaruh kombinasi parameter terhadap stabilitas serta konsistensi deteksi benih kelapa sawit pada citra konveyor, baik pada dataset tanpa AWF maupun dengan AWF.

Skenario pertama menggunakan konfigurasi *default* YOLOv8n, yaitu ukuran citra 640 piksel, *agnostic_nms=False*, ambang *Intersection over Union* (IoU) sebesar 0,7, dan *Close_mosaic* bernilai 10, Skenario ini dijadikan *baseline* guna memperoleh performa awal model tanpa modifikasi parameter. Namun, perbedaan performa model dengan dataset AWF belum terlihat, serta hasil *training* dan inferensi memperlihatkan adanya fenomena *double detection*, yakni satu objek terdeteksi lebih dari sekali. Hal ini muncul baik dalam bentuk beberapa *bounding box* dengan kelas yang sama maupun berbeda, yang menandai objek identik. Kondisi tersebut menunjukkan bahwa diperlukan proses *Non-Maximum Suppression (NMS)* sehingga model mampu secara efektif menyaring prediksi yang saling tumpang tindih.

Skenario kedua dilakukan dengan meningkatkan ukuran citra menjadi 1024 piksel untuk mempertegas detail fitur yang dipelajari model sekaligus menonjolkan efek peningkatan kualitas citra hasil AWF, mengaktifkan *Agnostic_NMS* dan menurunkan ambang IoU menjadi 0,3. Pengaturan *Agnostic_NMS = True* membuat NMS bekerja lintas kelas, sehingga *bounding box* yang bertumpang tindih tetap dieliminasi meskipun berasal dari kelas berbeda. Sementara itu, penurunan nilai IoU memperketat kriteria tumpang tindih, sehingga membantu mengurangi deteksi ganda pada area objek yang sama. Peningkatan resolusi ini bertujuan agar model dapat menangkap detail fitur visual benih dengan lebih baik, seperti tekstur permukaan, bentuk kecambah, dan kontras antar bagian.

Terakhir, skenario ketiga dilakukan dengan menonaktifkan *Close_mosaic* (nilai 0), sementara *Agnostic_NMS* dan IoU tetap mengikuti skenario kedua. Penonaktifan *mosaic augmentation* dimaksudkan agar objek benih tidak muncul dalam kondisi terpotong atau tergabung dengan bagian objek lain akibat proses *mosaic*. Hal ini penting agar model mempelajari bentuk kecambah secara utuh, sehingga mampu membedakan karakteristik morfologi antar kelas secara konsisten. Dengan demikian,

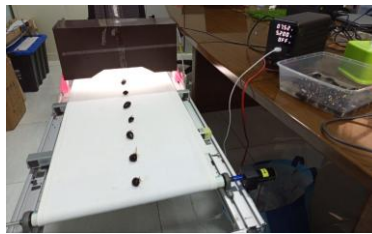
skenario ini memastikan model memperoleh representasi objek yang lebih realistis sesuai kondisi aktual di konveyor.

Training untuk setiap skenario dijalankan selama 500 *epochs* dengan *batch size* sebesar 16, *Stochastic Gradient Descent* (SGD) sebagai *optimizer*, dan *learning rate* awal sebesar 0,01 untuk mengoptimalkan bobot model secara iteratif.

2.5.2 Tahap *testing*

1. Input dataset uji

Proses pengujian diawali dengan memasukkan input berupa rekaman video yang diambil dengan merekam 50 benih kecambah kelapa sawit yang terdiri atas 25 benih normal dan 25 benih abnormal di atas *belt conveyor* yang bergerak dengan kombinasi 5 variasi kecepatan konveyor dan 2 jenis jarak kamera. Gambar proses pengambilan data uji ditunjukkan pada Gambar 23.



Gambar 23. Proses pengambilan data uji.

Video yang telah diakuisisi kemudian diekstraksi menjadi *frame-frame* citra dan diseleksi untuk memilih satu *frame* yang dapat menjadi representasi satu benih. Berikut skenario pengambilan data uji:

Tabel 4. Skenario data uji

Skenario ke-	Kecepatan Konveyor (RPM)	Jarak kamera (cm)
1	60	5
2	60	10
3	70	5
4	70	10
5	80	5
6	80	10
7	90	5
8	90	10
9	100	5
10	100	10

2. *Preprocessing*

Proses *preprocessing* pada tahap pengujian bertujuan untuk meningkatkan kualitas citra sehingga diperoleh hasil yang optimal. Tahapan *preprocessing* yang digunakan selama pengujian serupa dengan tahap yang diterapkan pada proses *training*, dimulai dari ekstraksi video menjadi beberapa *frame* citra hingga perbaikan citra yang mengalami *blur* menggunakan metode AWF.

3. Klasifikasi benih kecambah kelapa sawit.

Setelah citra digital diproses melalui tahap *preprocessing*, langkah selanjutnya adalah klasifikasi kualitas benih kecambah kelapa sawit. Pada tahap ini, model yang telah dilatih sebelumnya digunakan untuk mendeteksi dan mengklasifikasi setiap kecambah menjadi dua kelas: normal dan abnormal.

4. *Output*

Hasil dari proses klasifikasi ditampilkan sebagai video yang telah terklasifikasi dengan label kelasnya. Setiap benih kecambah kelapa sawit dalam video akan ditandai dengan label yang menunjukkan apakah benih kecambah tersebut termasuk dalam kategori normal atau abnormal. Video *output* ini memberikan hasil klasifikasi, di mana setiap kecambah yang direkam akan ditampilkan bersama dengan label prediksi kualitasnya.

5. Evaluasi sistem.

Evaluasi sistem dilakukan untuk menguji performa sistem dalam mengklasifikasikan benih kecambah kelapa sawit berdasarkan kualitasnya jika dibandingkan dengan kualitas aktual. Metrik evaluasi yang digunakan adalah *confusion matrix*, dengan metrik turunannya yaitu *accuracy*, *precision*, *recall*, dan *F1 score*. Metrik ini akan memberikan informasi tentang seberapa baik model dalam mengklasifikasikan data dari video

6. Pengujian Ablasi

Hasil pengujian ablasi (*ablation testing*) merupakan metodologi eksperimental yang secara sistematis menghilangkan atau memodifikasi komponen spesifik dari model deep learning untuk mengukur kontribusi individual setiap komponen terhadap performa sistem. Pada penelitian ini, pengujian ablasi dilakukan dengan cara memperketat nilai *confidence score* (ambang kepercayaan) pada saat inferensi untuk mengamati pengaruhnya terhadap kinerja klasifikasi.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kelapa sawit merupakan salah satu komoditas perkebunan strategis yang memberikan kontribusi signifikan terhadap pertumbuhan ekonomi Indonesia. Berdasarkan data Gabungan Pengusaha Kelapa Sawit Indonesia (GAPKI), industri kelapa sawit berhasil menyumbang devisa sebesar Rp600 triliun sepanjang tahun 2023, yang menjadi pencapaian tertinggi dalam sejarah (GAPKI, 2024). Pencapaian tersebut menunjukkan bahwa sektor kelapa sawit memiliki peran penting dalam menopang perekonomian nasional dan masih memiliki potensi besar untuk terus dikembangkan. Seiring meningkatnya permintaan pasar domestik maupun global terhadap produk turunan kelapa sawit, pemerintah berupaya meningkatkan kapasitas produksi melalui penambahan luas lahan dan peningkatan produktivitas. Presiden Republik Indonesia, Prabowo Subianto, dalam Musrenbangnas di Bappenas (Nurcahyo, 2024), menyampaikan bahwa Indonesia perlu menambah luasan perkebunan kelapa sawit untuk menjaga ketersediaan pasokan dan daya saing di pasar dunia. Upaya tersebut perlu didukung dengan penerapan teknologi pertanian modern agar proses produksi lebih efisien dan berkelanjutan.

Kementerian Pertanian turut mendorong penerapan konsep *smart farming* sebagai solusi peningkatan efisiensi dan produktivitas sektor perkebunan. Konsep ini melibatkan pemanfaatan teknologi seperti *Internet of Things* (IoT), *artificial intelligence* (AI), robotika, dan otomatisasi mekanisasi pertanian (Waranggani, 2023). Penerapan smart farming pada sektor kelapa sawit diharapkan dapat mengoptimalkan proses budidaya, mulai dari tahap pembibitan hingga panen, melalui pengumpulan data dan analisis yang akurat berbasis sistem digital. Oleh karena itu, diperlukan metode klasifikasi benih yang mampu menilai kualitas kecambah secara akurat dan konsisten.

Saat ini, proses klasifikasi benih kecambah kelapa sawit masih banyak dilakukan secara manual. Metode manual membutuhkan waktu yang lama, bersifat subjektif, dan rentan terhadap kesalahan manusia. Berdasarkan Standar Operasional Prosedur (SOP), proses seleksi terhadap 12.000 kecambah dapat memakan waktu hingga 8-10 hari kerja dengan tingkat akurasi yang rendah akibat bias penglihatan petani (Serikat Petani Kelapa Sawit, 2016). Kondisi ini menunjukkan perlunya sistem klasifikasi otomatis yang lebih efisien dan objektif.

Pemanfaatan teknologi visi komputer menjadi salah satu solusi potensial untuk meningkatkan efisiensi dalam proses klasifikasi benih kecambah kelapa sawit. Teknologi ini memungkinkan sistem mengenali dan mengklasifikasi kecambah berdasarkan ciri visual secara cepat dan konsisten. Standar Nasional Indonesia (SNI) 8211:2015 yang diterbitkan oleh Badan Standardisasi Nasional (BSN) menetapkan kriteria mutu fisik kecambah kelapa sawit meliputi bobot biji, panjang radikula dan plumula, warna, arah tumbuh, serta kenampakan yang jelas (Badan Standardisasi Nasional, 2015). Dengan acuan standar tersebut, sistem berbasis visi

komputer dapat dirancang untuk melakukan penilaian mutu secara otomatis dan terukur.

Dalam penerapannya, sistem klasifikasi otomatis sering memanfaatkan konveyor sebagai media pengangkut kecambah agar proses pengambilan citra dapat dilakukan secara berkelanjutan. Konveyor merupakan sistem transportasi material yang bergerak secara kontinu dan teratur, terdiri atas motor penggerak, sabuk atau rantai sebagai media pengangkut, serta roller atau pulley sebagai penopang gerak (Diva et al., 2024). Kecepatan pergerakan konveyor menjadi faktor penting karena memengaruhi kualitas citra yang diambil oleh kamera. Jika kecepatan terlalu tinggi, citra yang dihasilkan dapat menjadi kabur.

Citra kabur (*blur*) terjadi ketika gambar yang diambil tidak fokus atau saat objek mengalami pergerakan selama proses pengambilan gambar (Bazame et al., 2021). Salah satu jenisnya, yaitu *motion blur*, disebabkan oleh pergerakan objek yang terekam di antara *frame*, sehingga objek tampak memanjang mengikuti arah gerakan (Li et al., 2022). Dalam konteks klasifikasi benih kecambah kelapa sawit pada konveyor berkecepatan tinggi, efek *blur* menjadi tantangan serius karena dapat mengaburkan detail visual penting yang membedakan kecambah normal dan abnormal (Furqan, 2024).

Berdasarkan uraian di atas, permasalahan utama yang dihadapi adalah penurunan kualitas citra akibat *motion blur* pada konveyor berkecepatan tinggi yang berdampak pada akurasi klasifikasi benih kecambah kelapa sawit. Penelitian sebelumnya umumnya berfokus pada peningkatan kinerja klasifikasi benih tanpa secara khusus mengatasi degradasi citra akibat *blur* pada sistem berbasis konveyor. Oleh karena itu, penelitian ini mengusulkan penerapan *Adaptive Wiener Filter* sebagai metode *deblurring* yang dikombinasikan dengan model deteksi objek YOLOv8 untuk meningkatkan akurasi klasifikasi benih kecambah kelapa sawit pada berbagai kecepatan konveyor.

1.2 Landasan Teori

1.2.1 Kelapa sawit

Kelapa sawit (*Elaeis guineensis* Jacq.) merupakan spesies tanaman dari famili *Arecaceae* (palem-paleman) yang berasal dari kawasan Afrika Barat dan kini telah dibudidayakan secara luas di berbagai wilayah tropis, termasuk Indonesia (Situmorang, 2024). Secara morfologis, tanaman kelapa sawit memiliki habitus pohon dengan batang tunggal yang dapat tumbuh hingga mencapai ketinggian lebih dari 20 meter, serta daun majemuk menyirip dengan panjang sekitar 3-5 meter. Tanaman ini tumbuh optimal pada dataran rendah tropis dengan curah hujan tahunan berkisar antara 1.500-4.000 mm, suhu udara stabil di kisaran 24-28°C, kelembapan 80-90%, dan kondisi tanah yang gembur dengan drainase baik (PT. Binasawit Makmur, 2024).



Gambar 1. Kelapa Sawit
Sumber: (Nurhadi, 2024)

Kelapa sawit memiliki peranan yang sangat penting dalam sektor pertanian dan perekonomian nasional, terutama sebagai sumber utama minyak sawit (*crude palm oil*) dan minyak inti sawit (*palm kernel oil*). Indonesia merupakan produsen kelapa sawit terbesar di dunia dengan luas areal perkebunan mencapai sekitar 16,8 juta hektare pada tahun 2023 (Situmorang, 2024). Kontribusi ekonominya sangat signifikan, tercermin dari perolehan devisa ekspor sebesar USD 25,61 miliar pada tahun yang sama, serta kemampuannya dalam menyerap tenaga kerja dan mendukung kesejahteraan jutaan petani di sektor perkebunan (Sehusman, 2024).

1.2.2 Kualitas Benih Kecambah Kelapa Sawit

Benih kelapa sawit dijelaskan dalam Keputusan Menteri Pertanian Republik Indonesia Nomor 321/Kpts/KB.020/10/2015 sebagai tanaman atau bagian tanaman kelapa sawit yang digunakan untuk memperbanyak dan/atau mengembangbiakkan tanaman kelapa sawit. Dalam praktik budidaya, benih bermutu umumnya diperoleh dari hasil persilangan antara tetua Dura (D), yang memiliki cangkang buah tebal, dengan tetua Pisifera (P), yang tidak memiliki cangkang dan mempunyai tandan betina steril. Hasil persilangan tersebut menghasilkan tipe Tenera, yaitu varietas unggul yang diharapkan karena memiliki produktivitas dan rendemen minyak yang lebih tinggi (Ayuningtyas et al., 2022).

Untuk menjamin kualitas benih yang beredar, Badan Standardisasi Nasional (BSN) telah menetapkan persyaratan mutu kecambah kelapa sawit yang tercantum dalam SNI 8211:2015. Standar ini membagi mutu kecambah ke dalam tiga tolok ukur utama, yaitu mutu fisik, fisiologis, dan genetik, yang masing-masing memiliki kriteria penilaian tersendiri (Badan Standardisasi Nasional, 2015). Tolok ukur dan persyaratan mutu kecambah kelapa sawit tersebut dapat dilihat pada Tabel 1.

Tabel 1. Persyaratan mutu kecambah

No	Tolok Ukur	Persyaratan
1	Mutu genetik	
	a. Asal bahan tanaman	Kebun benih yang telah ditetapkan oleh Direktur Jendral Perkebunan
	b. Varietas	Benih bina

No	Tolok Ukur	Persyaratan
	c. Kemurnian	Tidak kurang dari 98% kecambah menghasilkan kelapa sawit tipe Tenera (cangkang tipis)
2	Mutu Fisiologis	
	a. Kesehatan	Bebas OPT
3	Mutu fisik	
	a. Bobot biji	Minimal 0,8g
	b. Radikula dan plumula	
	- Panjang	Minimal 0,5 cm
	- Warna	Putih kekuningan
	- Arah tumbuh	Berlawanan arah
	- Kenampakan	Dapat dibedakan dengan jelas

Benih kelapa sawit memiliki peranan penting sejak fase awal pertumbuhan tanaman. Kualitas benih menentukan vigor, keseragaman pertumbuhan, ketahanan terhadap cekaman lingkungan dan serangan organisme pengganggu tanaman, serta efisiensi penyerapan unsur hara (Ayuningtyas et al., 2022). Dampaknya berlanjut hingga fase produksi, karena tanaman yang berasal dari benih unggul umumnya memiliki produktivitas tandan buah segar yang lebih tinggi serta rendemen minyak yang lebih baik. Dengan demikian, pemilihan dan penanganan benih yang tepat menjadi fondasi utama dalam upaya menghasilkan tanaman kelapa sawit yang sehat, produktif, dan berkelanjutan.

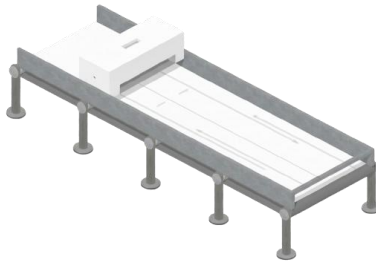


Gambar 2. Benih Kecambah Kelapa Sawit
Sumber: (BKS-PPS, 2020)

1.2.3 Konveyor

Konveyor merupakan alat mekanis yang berfungsi untuk memindahkan barang atau material dari satu titik ke titik lainnya secara teratur dan berkesinambungan (Sindak Hutauruk et al., 2021). Prinsip kerja konveyor serupa dengan jalur bergerak yang membawa objek di sepanjang lintasan yang telah ditentukan. Secara umum, sistem konveyor terdiri atas beberapa komponen utama, yaitu rangka atau struktur penopang sebagai kerangka utama, motor penggerak yang menyediakan tenaga

untuk menggerakkan sistem, serta sabuk atau rantai yang bergerak di atas rol atau katrol sebagai media pemindahan material (Sindak Hutauruk et al., 2021).



Gambar 3. Konveyor

Dalam penerapannya, konveyor digunakan secara luas di berbagai sektor industri, seperti manufaktur, logistik, pertambangan, dan pengolahan makanan. Pada industri manufaktur, konveyor berperan penting dalam proses perakitan dan pengemasan produk. Di bidang logistik dan pergudangan, sistem ini membantu mempercepat proses distribusi barang dari satu area ke area lainnya. Sementara itu, pada sektor pertambangan, konveyor dimanfaatkan untuk mengangkut material berat seperti batu bara, pasir, maupun bijih mineral (Mulyana et al., 2024).

Keunggulan utama penggunaan konveyor terletak pada kemampuannya dalam meningkatkan efisiensi proses pemindahan material, menjaga keteraturan aliran produksi, serta memungkinkan pengaturan kecepatan sesuai kebutuhan operasional (Khatib Sulaiman et al., 2025). Selain itu, sistem konveyor juga dapat mengurangi ketergantungan terhadap tenaga kerja manusia, menekan biaya operasional, serta meningkatkan keselamatan kerja dengan meminimalkan risiko cedera akibat pemindahan manual (Siahaan et al., 2024). Dengan demikian, konveyor menjadi salah satu komponen penting dalam mendukung otomatisasi dan kelancaran proses industri modern.

1.2.4 Visi komputer

Visi komputer merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan mesin untuk menginterpretasikan dan memahami informasi visual dari gambar maupun video digital. Sistem ini dirancang agar mampu mengenali pola, mengekstraksi fitur, serta melakukan analisis terhadap data visual dengan tujuan meniru kemampuan penglihatan manusia (Lee & Ahn, 2023). Melalui teknologi ini, komputer dapat melakukan berbagai fungsi seperti deteksi objek, klasifikasi citra, pengenalan wajah, dan pelacakan gerak. Aplikasi visi komputer kini telah meluas ke berbagai bidang, seperti kendaraan otonom, kesehatan, pertanian, sistem pengawasan, hingga industri hiburan, yang menunjukkan besarnya pengaruh teknologi ini dalam kehidupan modern (Gendy & Patel, 2024).

Perkembangan pesat dalam bidang pembelajaran mesin, khususnya *deep neural network* (DNN), telah memberikan peningkatan signifikan terhadap

kemampuan sistem visi komputer. Liu et al. (2025) menyatakan bahwa kemajuan berkelanjutan dalam model pembelajaran mendalam mampu meningkatkan akurasi pengenalan gambar dan memperluas kemampuan sistem dalam menyelesaikan berbagai tugas visual secara kompleks. Metode modern berbasis *convolutional neural networks* (CNN) memungkinkan sistem mencapai performa tinggi bahkan secara *real-time*, berkat kombinasi algoritma canggih dan kemajuan perangkat keras komputasi yang efisien (Liu et al., 2025). Dengan kemampuan tersebut, visi komputer kini menjadi teknologi fundamental yang mendorong inovasi di berbagai sektor industri dan menjadi solusi terhadap berbagai permasalahan yang melibatkan analisis visual di dunia nyata (Gendy & Patel, 2024). Dalam konteks penelitian ini, visi komputer dimanfaatkan untuk mengotomatisasi proses klasifikasi benih kecambah kelapa sawit berdasarkan ciri visual pada acitra yang diambil dari sistem konveyor.

1.2.5 Pengolahan citra

Pengolahan citra merupakan bidang ilmu yang berfokus pada analisis, manipulasi, serta interpretasi data visual melalui penerapan berbagai teknik komputasi canggih. Disiplin ini terus mengalami perkembangan pesat seiring dengan kemajuan teknologi informasi dan kecerdasan buatan. Menurut Khairunnisa et al. (2025), perkembangan terkini dalam pengolahan citra ditandai oleh kemajuan signifikan dalam pengembangan algoritma untuk peningkatan kualitas citra, segmentasi objek, serta pengenalan pola. Sebagian besar inovasi tersebut didukung oleh integrasi metode pembelajaran mesin (*machine learning*) dan pembelajaran mendalam (*deep learning*), yang secara substansial meningkatkan akurasi dan efisiensi pada berbagai aplikasi kritis seperti pencitraan medis, sistem otonom, serta penginderaan jauh.

Tahap prapemrosesan citra merupakan komponen penting dalam keseluruhan proses pengolahan, karena berfungsi untuk meminimalkan *noise* dan meningkatkan kualitas data sehingga fitur citra dapat diekstraksi secara optimal. Optimalisasi pada tahap ini menjadi semakin penting seiring meningkatnya kebutuhan akan kemampuan pemrosesan citra secara *real-time*. Selain itu, penelitian mutakhir juga menunjukkan adanya integrasi pengolahan citra dengan teknik lain, seperti pemodelan tiga dimensi dan analisis multispektral, yang memperluas cakupan dan dampak penerapannya dalam berbagai bidang ilmiah maupun industri (Khairunnisa et al., 2025).

1.2.6 Adaptive Wiener Filter

Adaptive Wiener Filter (AWF) merupakan pengembangan dari *Wiener Filter* yang dirancang untuk meningkatkan kualitas citra, khususnya dalam proses penghilangan *noise* (*denoising*) dan *blur* (*deblurring*) secara adaptif. Sifat adaptif pada filter ini memungkinkan sistem menyesuaikan estimasi variansi *noise* dan *blur* secara lokal pada setiap area citra. Dengan demikian, filter dapat merespons perbedaan tingkat gangguan pada berbagai bagian citra sehingga menghasilkan proses restorasi yang

lebih optimal tanpa menghilangkan detail penting (Salehi & Vahidi, 2021). Berbeda dengan *Wiener Filter* konvensional yang mengasumsikan kondisi noise dan *blur* seragam di seluruh citra, AWF mampu memberikan hasil yang lebih presisi dalam kondisi citra yang kompleks dan bervariasi.

Seiring perkembangan teknologi pengolahan citra, berbagai penelitian telah mengusulkan modifikasi terhadap *Wiener Filter* untuk meningkatkan kinerjanya. Beberapa pendekatan modern mengintegrasikan metode *Discrete Wavelet Transform* (DWT) guna memperkirakan tingkat noise secara lebih akurat, sementara pendekatan lain memanfaatkan teknik kolaboratif yang mengandalkan kemiripan patch dalam citra untuk proses denoising dan *deblurring*. Selain itu, beberapa studi juga menggabungkan *Wiener Filter* dengan jaringan saraf tiruan (*neural network*) atau *autoencoder*, sehingga proses restorasi tidak hanya efisien dan dapat dijelaskan secara matematis, tetapi juga mampu memanfaatkan kekuatan pembelajaran mesin untuk menghasilkan kualitas citra yang lebih baik. Dalam penelitian ini, *Adaptive Wiener Filter* digunakan untuk mengurangi efek *motion blur* pada citra kecambah kelapa sawit yang diambil dari konveyor, sehingga kualitas citra yang masuk ke model deteksi objek menjadi lebih baik.

1.2.7 Image Quality Assessment (IQA)

Image Quality Assessment (IQA) adalah langkah vital untuk mengkuantifikasi perubahan kualitas (baik degradasi maupun perbaikan) pada sebuah citra. Terdapat dua metodologi utama: subjektif, yang berdasarkan persepsi manusia, dan objektif, yang menggunakan perhitungan matematis (Gao et al., 2025). Meskipun metode subjektif dianggap acuan utama (standar emas) karena paling akurat merefleksikan penglihatan manusia, ia memiliki kelemahan signifikan dalam hal biaya, waktu, dan reliabilitas (sulit direplikasi). Sebagai solusinya, metode objektif menawarkan evaluasi otomatis. Salah satu metrik objektif paling fundamental dan dominan adalah *Peak Signal-to-Noise Ratio* (PSNR). PSNR merupakan metrik referensi penuh (*full-reference*), yang berarti evaluasinya bergantung pada ketersediaan citra asli (ground truth) untuk dibandingkan dengan citra hasil olahan.

Mean Squared Error (MSE). *Mean Squared Error* (MSE) didefinisikan sebagai rata-rata kuadrat selisih antara nilai piksel pada citra asli dengan nilai piksel pada citra aproksimasinya di koordinat yang bersesuaian (Gao et al., 2025). Untuk citra monokrom I berukuran $m \times n$ dan citra aproksimasinya K , MSE diformulasikan sebagai berikut:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (1)$$

Semakin kecil nilai MSE, maka semakin kecil pula galat kumulatif antara kedua citra, yang mengidentifikasi tingkat kesamaan atau fiselitas yang lebih tinggi. Nilai MSE sama dengan nol menandakan bahwa kedua citra tersebut identik secara piksel. Meskipun sederhana, MSE memiliki

keterbatasan terkait ketergantungan pada skala, hal ini membuat MSE sulit diinterpretasikan secara universal.

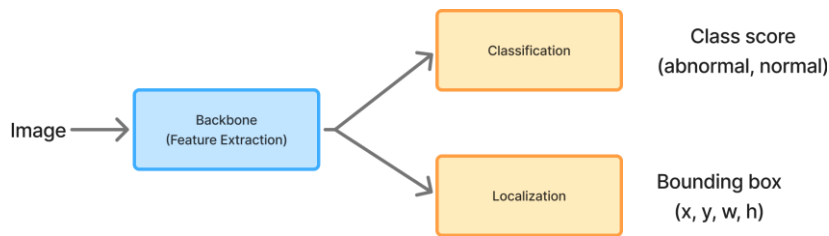
Peak Signal-to-Noise Ratio (PSNR). *Peak Signal-to-Noise Ratio (PSNR)* mengatasi keterbatasan MSE dengan menormalisasi galat terhadap daya maksimum sinyal yang mungkin. PSNR adalah rasio antara daya maksimum yang mungkin dari sebuah sinyal dengan daya derau (*noise*) yang merusak fidelitas representasinya (Gao et al., 2025). Dalam konteks IQA, sinyal merujuk pada citra asli, sedangkan “derau” adalah galat yang diperkenankan selama proses pengolahan, seperti kompresi atau transmisi. Kualitas citra dinyatakan baik jika nilai PSNR-nya berada pada rentang 30-50dB. PSNR diekspresikan dalam skala logaritmik desibel (dB) dan dihitung dari MSE dengan formula berikut (untuk *image* RGB):

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right) \quad (2)$$

1.2.8 Deteksi objek

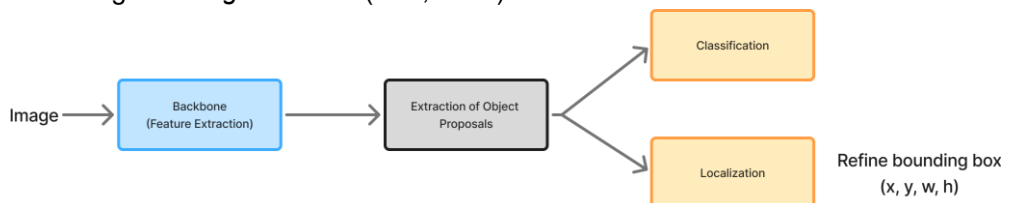
Deteksi objek merupakan salah satu bidang fundamental dalam visi komputer yang bertujuan untuk mengidentifikasi jenis objek yang terdapat pada suatu citra sekaligus menentukan posisi keberadaannya misalnya melalui penandaan kotak pembatas di sekitar objek (Carranza-García et al., 2021). Dengan demikian, deteksi objek tidak hanya berfokus pada klasifikasi kategori, tetapi juga menyertakan informasi spasial mengenai lokasi objek, berbeda dengan klasifikasi citra konvensional yang hanya menghasilkan label kelas tanpa informasi posisi. Perkembangan metode berbasis deep learning telah menghasilkan peningkatan akurasi yang signifikan, sehingga teknik ini kini banyak dimanfaatkan dalam berbagai aplikasi, seperti kendaraan otonom, sistem pengawasan, hingga analisis citra medis untuk mendeteksi dan melacak objek secara otomatis dalam waktu nyata. Secara umum, pendekatan deteksi objek modern dapat dikategorikan ke dalam dua kelompok utama, yaitu *one-stage detectors* dan *two-stage detectors*, yang masing-masing memiliki keunggulan serta keterbatasan (Carranza-García et al., 2021).

One-stage detector melakukan proses deteksi secara langsung dengan memprediksi kelas objek dan kotak pembatas dalam satu langkah tanpa melalui tahap penentuan lokasi. Dengan kata lain, hanya diperlukan satu jaringan syaraf untuk melaksanakan tugas klasifikasi sekaligus lokalisasi (Carranza-García et al., 2021). Namun, kecepatan tersebut sering kali dicapai dengan mengorbankan sebagian akurasi, khususnya dalam mendeteksi objek berukuran kecil atau objek yang tertutup sebagian (Han, 2025). Beberapa contoh model yang termasuk dalam kategori ini adalah SSD (*Single Shot MultiBox Detector*), YOLO (*You Only Look Once*), dan *RetinaNet* (Carranza-García et al., 2021).



Gambar 4. Proses *one-stage detector* (Thaku, 2023)

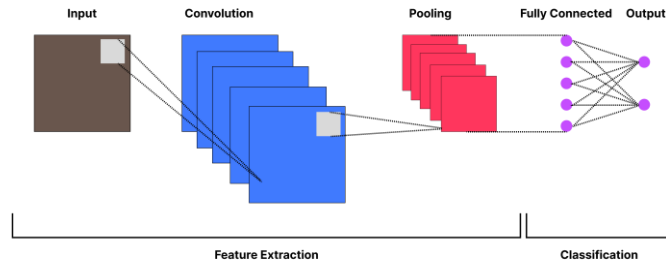
Berbeda dengan pendekatan *one-stage detector*, *two-stage detector* membagi proses deteksi menjadi dua fase. Tahap pertama menghasilkan kandidat lokasi yang berpotensi mengandung objek, kemudian tahap kedua melakukan klasifikasi serta penyempurnaan kotak pembatas. Pendekatan bertahap ini, sebagaimana diimplementasikan pada algoritma R-CNN, umumnya menghasilkan akurasi yang lebih tinggi karena model dapat melakukan analisis lebih mendalam pada sejumlah kandidat lokasi yang terbatas (Carranza-García et al., 2021). Akan tetapi, peningkatan akurasi tersebut disertai dengan konsekuensi berupa kebutuhan komputasi yang lebih besar dan kecepatan inferensi yang relatif lebih rendah dibanding *one-stage detector* (Han, 2025).



Gambar 5. Proses *two-stage detector* (Thaku, 2023)

1.2.9 Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) merupakan sistem kecerdasan buatan berbasis jaringan saraf tiruan yang berlapis banyak dan mampu melakukan identifikasi, pengenalan, serta klasifikasi objek, sekaligus mendeteksi dan melakukan segmentasi objek pada citra (Lanjewar & Gurav, 2022). CNN atau ConvNet, merupakan salah satu arsitektur *deep learning* diskriminatif yang populer karena dapat dipelajari langsung dari data masukan tanpa memerlukan proses ekstraksi fitur secara manual oleh manusia (Alzubaidi et al., 2021).



Gambar 6. Arsitektur CNN
(Gurucharan, 2025)

Jaringan ini banyak digunakan dalam bidang seperti pengenalan visual, analisis citra medis, segmentasi gambar, pemrosesan bahasa alami (NLP), dan beragam aplikasi lainnya, karena dirancang khusus untuk menangani data berbentuk dua dimensi. Dibandingkan dengan jaringan saraf konvensional, CNN lebih efektif karena mampu secara otomatis mengenali elemen-elemen penting dari data masukan tanpa keterlibatan manusia dalam proses penentuan fitur (Krichen, 2023). Terdapat 4 tipe lapisan utama dalam CNN, yaitu:

Convolutional layer. *Convolutional layer* merupakan inti CNN. Filter (*kernel*) digunakan untuk mengekstraksi fitur dari citra masukan. Proses konvolusi menghasilkan *feature map* yang merepresentasikan pola tertentu. Dengan *training*, bobot *kernel* disesuaikan sehingga mampu mengenali fitur yang relevan (Krichen, 2023).

Pooling layer. *Pooling layer* berfungsi mengurangi dimensi spasial *feature map* dengan tetap mempertahankan informasi penting. Teknik yang umum digunakan adalah *max pooling* dan *average pooling*. *Pooling* membantu mempercepat komputasi, mengurangi jumlah parameter, dan membuat model lebih tahan terhadap pergeseran kecil pada citra (Savyakhosla, 2025).

Activation function (Non-linearity layer). *Activation function (Non-linearity layer)* memperkenalkan non-linearitas ke dalam jaringan, memungkinkan CNN mempelajari hubungan kompleks. Fungsi yang sering digunakan antara lain Sigmoid, Tanh, dan ReLU. ReLU menjadi pilihan utama karena sederhana, efisien, dan mengurangi masalah *vanishing gradient* (Sanjaya, 2023).

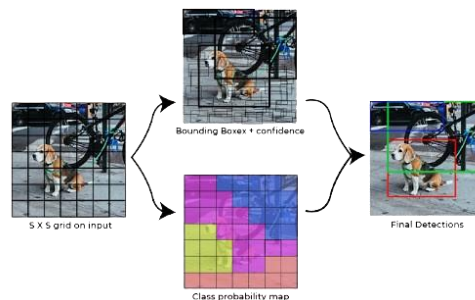
Fully connected layer. *Fully connected layer* biasanya berada di bagian akhir CNN. Semua neuron pada lapisan ini terhubung dengan neuron pada lapisan sebelumnya. Fungsinya adalah menggabungkan fitur yang telah diekstraksi untuk menghasilkan keputusan klasifikasi akhir (Vakalopoulou et al., 2023).

1.2.10 You Only Look Once (YOLO)

Algoritma *You Only Look Once* (YOLO) merupakan salah satu metode deteksi objek berbasis deep learning yang dirancang untuk melakukan identifikasi objek secara cepat dan efisien (Murat & Kiran, 2025). YOLO mengubah permasalahan deteksi objek menjadi regresi tunggal, di mana citra masukan dibagi ke dalam grid, dan setiap sel grid bertanggung jawab memprediksi *bounding box* serta probabilitas kelas objek. Dengan pendekatan ini, proses deteksi dapat dilakukan hanya dalam satu kali inferensi, sehingga YOLO mampu bekerja secara *real-time* bahkan pada data video (Murat & Kiran, 2025).

Selain itu, penggunaan informasi global dari citra membuat YOLO memiliki kemampuan generalisasi yang baik untuk berbagai jenis objek (Ali & Zhang, 2024). Namun demikian, YOLO juga memiliki keterbatasan, misalnya kesulitan dalam mendeteksi objek berukuran kecil atau objek yang saling berdekatan (Fauzan Arif et al., 2023).

Secara keseluruhan, YOLO menjadi salah satu algoritma deteksi objek yang paling banyak digunakan dalam berbagai bidang, mulai dari pengawasan lalu lintas, pertanian presisi, hingga aplikasi industri. Sifatnya yang ringan, efisien, dan mudah diimplementasikan menjadikan YOLO pilihan utama untuk sistem yang membutuhkan deteksi objek secara cepat dan akurat. Dalam penelitian ini, YOLO dimanfaatkan sebagai dasar arsitektur deteksi objek untuk mengidentifikasi kecambah kelapa sawit pada citra konveyor.



Gambar 7. Gambaran umum proses deteksi YOLO
Sumber: (M, 2025)

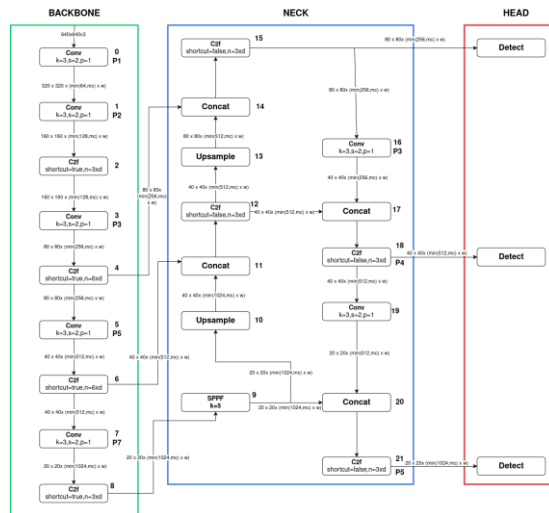
1.2.11 YOLOv8

Algoritma YOLOv8 merupakan generasi ke-8 dari keluarga *You Only Look Once* (YOLO) yang dikembangkan oleh Ultralytics. Model ini dirancang untuk meningkatkan kinerja deteksi objek melalui kombinasi arsitektur yang lebih efisien dan strategi *training* yang adaptif. YOLOv8 mengadopsi pendekatan *anchor-free detection head*, yang berbeda dari versi sebelumnya yang berbasis *anchor box*. Pendekatan ini menyederhanakan proses prediksi dengan langsung memperkirakan

koordinat *bounding box*, sehingga mengurangi kompleksitas komputasi sekaligus meningkatkan akurasi pada objek dengan variasi ukuran dan bentuk (Diko Aprilyanto et al., 2025).

Selain fungsi utama dalam deteksi objek, YOLOv8 juga mendukung berbagai tugas visi komputer lain, seperti segmentasi *instance*, klasifikasi citra, serta deteksi pose (pose estimation) (Diko Aprilyanto et al., 2025). Fleksibilitas ini diperoleh melalui desain arsitektur modular yang memungkinkan adaptasi terhadap berbagai jenis data dan kebutuhan aplikasi. Lebih jauh, YOLOv8 dilengkapi dengan model pralatih (*pre-trained models*) yang telah dioptimalkan pada dataset berskala besar, sehingga dapat digunakan secara langsung (*out-of-the-box*) maupun disesuaikan lebih lanjut melalui proses *fine-tuning* (Diko Aprilyanto et al., 2025).

Secara keseluruhan, YOLOv8 menekankan keseimbangan antara kecepatan inferensi dan akurasi deteksi, menjadikannya relevan untuk implementasi pada sistem *real-time* maupun perangkat dengan keterbatasan sumber daya. Dengan dukungan kerangka kerja PyTorch dan dokumentasi yang komprehensif, YOLOv8 tidak hanya berfungsi sebagai model deteksi mutakhir, tetapi juga sebagai fondasi penelitian dan pengembangan lebih lanjut dalam bidang visi komputer. YOLOv8 digunakan sebagai model utama deteksi objek dalam penelitian ini karena menawarkan kombinasi kecepatan dan akurasi yang sesuai untuk aplikasi berbasis konveyor.



Gambar 8. Arsitektur YOLOv8
Sumber: (Timilsina, 2024)

1.2.12 Confusion matrix

Confusion matrix merupakan salah satu metrik evaluasi yang sering digunakan untuk menilai kinerja model klasifikasi. Matriks ini berbentuk tabel yang membandingkan hasil prediksi model dengan label sebenarnya (*ground truth*), sehingga memberikan

gambaran yang lebih rinci mengenai performa model. Pada kasus klasifikasi biner, *confusion matrix* terdiri atas empat komponen utama, yaitu *True Positive* (TP), prediksi benar terhadap kelas positif; *True Negative* (TN), prediksi benar terhadap kelas negatif; *False positive* (FP), kesalahan ketika model memprediksi positif padahal sebenarnya negatif; dan *False negative* (FN), yaitu kesalahan ketika model gagal mengenali kelas positif yang sebenarnya. Jadi, *confusion matrix* tidak hanya menunjukkan tingkat akurasi, namun juga mengungkapkan jenis kesalahan dalam proses klasifikasi.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 9. *Confusion matrix*
(Nugroho, 2019)

Dari keempat komponen *confusion matrix*, terdapat empat metrik tunggal yang bisa dihitung, seperti *accuracy*, *precision*, *recall*, dan *F1 score*.

Accuracy. Akurasi menunjukkan seberapa banyak prediksi model yang benar dibandingkan dengan seluruh prediksi yang dilakukan. Metrik ini memberikan gambaran umum tentang kinerja model, tetapi dapat menyesatkan apabila terdapat ketidakseimbangan kelas.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Precision. Presisi berfokus pada kualitas prediksi positif yang dihasilkan model. Metrik ini mengukur seberapa banyak prediksi positif yang benar-benar sesuai dengan kelas positif. Presisi penting ketika kesalahan *false positive* perlu diminimalkan.

$$precision = \frac{TP}{TP + FP} \quad (4)$$

Recall. Sensitivitas mengukur sejauh mana model mampu mengenali seluruh kasus positif yang ada. Metrik ini menunjukkan proporsi *true positive* yang berhasil dideteksi dari semua data positif sebenarnya.

$$recall = \frac{TP}{TP + FN} \quad (5)$$

F1 score. *F1 score* menggabungkan presisi dan *recall* ke dalam satu metrik untuk menyeimbangkan *trade-off* di antara keduanya. Metrik ini sangat berguna pada dataset yang tidak seimbang, di mana baik *false positive* maupun *false negative* sama-sama penting.

$$F1\ Score = \frac{2 \times precision \times recall}{precision + recall} \quad (6)$$

1.3 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah:

1. Bagaimana implementasi *Adaptive Wiener Filter* dalam sistem klasifikasi benih kecambah kelapa sawit yang mengalami efek *blur* ketika bergerak cepat melalui konveyor?
2. Bagaimana kinerja sistem dalam mengklasifikasi kualitas benih kecambah kelapa sawit pada konveyor?

1.4 Tujuan dan Manfaat Penelitian

1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah:

1. Mengimplementasikan *Adaptive Wiener Filter* dalam sistem klasifikasi benih kecambah kelapa sawit yang terpengaruh efek *blur* akibat pergerakan cepat di atas konveyor.
2. Mengevaluasi kinerja sistem dalam mengklasifikasi kualitas benih kecambah kelapa sawit pada konveyor.

1.4.2 Manfaat

Adapun manfaat dari penelitian ini adalah:

1. Penelitian ini diharapkan dapat menjadi referensi yang berguna bagi akademisi dan peneliti yang tertarik untuk mengkaji topik serupa.
2. Penelitian ini juga diharapkan dapat menjadi inovasi yang berkontribusi pada peningkatan kualitas dan konsistensi klasifikasi benih kecambah kelapa sawit, sekaligus mendorong penerapan teknologi visi komputer di sektor pertanian.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini adalah:

1. Mengklasifikasikan kecambah kelapa sawit berdasarkan bentuk dan ukuran dari radikula dan plumula sesuai persyaratan mutu fisik yang distandardisasi secara nasional.
2. Kecambah diklasifikasikan dalam dua kelas yaitu normal dan abnormal.
3. Kamera yang digunakan adalah kamera Logitech Brio Ultra HD, statis, dan tidak menggunakan *auto focus*.
4. Data yang digunakan yaitu data citra berupa gambar dan video kecambah kelapa sawit yang diambil menggunakan *belt konveyor*.