

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Buah kakao merupakan salah satu komoditas pertanian yang sangat vital bagi perekonomian petani dan industri perkebunan [1]. Kualitas buah kakao memiliki dampak langsung terhadap nilai jual, hasil panen, dan stabilitas rantai pasok industri cokelat. Namun, seiring dengan kondisi cuaca dan faktor lingkungan sekitar, perkebunan kakao seringkali menghadapi ancaman hama dan penyakit yang perlu mendapat perhatian serius. Penyakit ini secara visual seringkali ditandai dengan adanya bercak kehitaman, perubahan warna dan tekstur kulit buah, hingga pembusukan akibat penyakit seperti busuk buah (*Black Pod Rot*) maupun serangan hama [2].

Berdasarkan hasil observasi di Desa Pasir Utama, Kecamatan Rambah Hilir, Kabupaten Rokan Hulu, masih terdapat masyarakat yang membudidayakan tanaman kakao pada lahan di sekitar rumah. Meskipun dilakukan dalam skala rumah tangga dan bukan perkebunan besar, tanaman kakao masih dibudidayakan oleh masyarakat. Namun, dalam proses budidayanya, tanaman kakao sering mengalami serangan hama dan penyakit, seperti busuk buah, Penggerek Buah Kakao yang dapat menurunkan kualitas hasil panen.

Secara konvensional, penentuan kesehatan buah dan identifikasi tingkat keparahan penyakit dilakukan melalui pengamatan visual manual oleh petani atau penyuluh pertanian di kebun. Meskipun dapat mendiagnosis secara langsung, metode ini memiliki kelemahan dalam hal efisiensi dan subjektivitas. Pengamatan

manual membutuhkan ketelitian, dan waktu yang cukup lama dalam proses identifikasi penyakit. Selain itu, minimnya pengetahuan petani dalam mengidentifikasi jenis penyakit seringkali berujung pada penanganan yang tidak tepat. Oleh karena itu, dibutuhkan sebuah pendekatan alternatif yang lebih cepat dan otomatis melalui teknologi visi komputer (*Computer Vision*).

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*), khususnya *Computer Vision* dengan metode *Deep Learning*, telah terbukti efektif menawarkan solusi inovatif dalam bidang pertanian [3]. Keberhasilan teknologi ini salah satunya dibuktikan oleh penelitian Hariyono dkk. (2023) yang menerapkan arsitektur *Convolutional Neural Network* (CNN) untuk mendeteksi penyakit tanaman melalui spektrum warna citra digital daun padi. Penelitian tersebut membuktikan bahwa pemanfaatan CNN mampu mengklasifikasikan jenis penyakit tanaman dengan mengekstraksi fitur morfologi secara otomatis, di mana model modifikasi (*custom*) yang dikembangkan berhasil mencapai tingkat akurasi sebesar 90%. Hal ini menegaskan keandalan algoritma *Deep Learning* sebagai sistem peringatan dini pada tanaman [4].

Penerapan visi komputer yang lebih spesifik pada deteksi penyakit komoditas kakao juga telah dilakukan oleh Ramadhan dan Udjulawa (2026) menggunakan model YOLOv11m. Model deteksi berbasis teknologi *Deep Learning* tersebut dikembangkan untuk mengenali penyakit seperti *Black Pod Rot*, *Frosty Pod Rot*, penelitian tersebut mencatat bahwa model masih memiliki keterbatasan dalam menangani objek yang bervariasi ukurannya. Model seringkali

mengalami kesalahan prediksi dan penurunan nilai *recall* secara signifikan pada citra buah atau bercak penyakit yang berukuran kecil [5].

Berdasarkan tantangan tersebut, pendekatan alternatif yang lebih tangguh dalam lokalisasi fitur visual rumit sangat diperlukan. Sebagai solusi untuk mengatasi kelemahan pada deteksi objek berukuran kecil dan kompleks, metode *Faster R-CNN* (*Region-based Convolutional Neural Network*) dapat menjadi pilihan yang tepat. Berbeda dengan pendekatan YOLO yang memprioritaskan kecepatan (*one-stage detector*), *Faster R-CNN* yang menggunakan pendekatan *two-stage detector* dikenal lebih unggul dalam tingkat ketelitian (*precision*) lokalisasi *bounding box* [6].

Penerapan metode dan logika tersebut menjadi sangat relevan untuk diimplementasikan pada kondisi perkebunan kakao di lapangan. Kondisi nyata di kebun memiliki tantangan visual yang khas, seperti pencahayaan matahari yang berubah-ubah, bayangan kanopi, serta varietas yang berbeda. Penggunaan *Faster R-CNN* yang tangguh mengekstraksi fitur kompleks diharapkan mampu mengatasi tantangan tersebut dan memiliki tingkat adaptabilitas yang tinggi terhadap kondisi nyata.

Berdasarkan rumusan permasalahan dan tinjauan literatur tersebut, penelitian ini bertujuan untuk mengimplementasikan metode *Faster R-CNN* dalam menganalisis indikator penyakit visual secara otomatis pada buah kakao, sistem diharapkan mampu mendeteksi sekaligus mengklasifikasikan tingkat keparahan penyakit buah kakao dengan sangat presisi. Penelitian ini diharapkan dapat memberikan kontribusi nyata berupa sistem pemantauan kesehatan tanaman yang

efisien guna menjaga keberlanjutan hasil panen pertanian kakao. Berdasarkan uraian di atas, penulis tertarik untuk melakukan penelitian dengan judul: “Sistem Deteksi Objek Penyakit Pada Buah Kakao Menggunakan Metode Faster R-CNN”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah pada penelitian ini adalah Bagaimana merancang, membangun, dan mengimplementasikan metode *Faster R-CNN* untuk mendeteksi objek penyakit pada citra buah kakao

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang, membangun dan mengimplementasikan metode *Faster R-CNN* untuk mendeteksi objek penyakit pada citra buah kakao

1.4 Batasan Masalah

Agar penelitian ini lebih terarah, fokus, dan tidak menyimpang dari tujuan utama yang telah ditetapkan, maka diberikan beberapa batasan masalah sebagai berikut:

1. Algoritma *Deep Learning* yang digunakan untuk deteksi objek penyakit secara eksklusif hanya menggunakan arsitektur *Faster R-CNN* (*Region-based Convolutional Neural Network*).
2. Parameter penyakit yang diteliti difokuskan secara spesifik pada indikator visual makroskopis di permukaan kulit buah kakao (seperti area bercak pembusukan akibat penyakit busuk buah, serta lubang atau kerusakan fisik akibat hama penggerek), dan tidak mencakup

pengujian kualitas biji kakao maupun uji laboratorium patologi tanaman secara mikroskopis.

3. Pengumpulan data citra yang digunakan untuk melatih (*training*) dan menguji (*testing*) model dibatasi pada citra buah kakao yang diperoleh dari *Kaggle Cacao Diseases*, dengan alokasi *dataset* sebanyak 4390 gambar.
4. Keseluruhan proses pemodelan, mulai dari prapemrosesan data (*preprocessing*), pelatihan model (*training*), hingga pengujian (*testing*) dibangun menggunakan bahasa pemrograman *Python*.
5. Perangkat lunak hasil penelitian akan dilengkapi dengan antarmuka pengguna (*User Interface*) yang dikembangkan menggunakan *framework* berbasis *Python* yaitu *Streamlit*, guna memudahkan proses unggah citra dan visualisasi hasil deteksi bagi pengguna akhir.
6. Deteksi dan klasifikasi kondisi buah kakao dalam penelitian ini dibatasi hanya pada 3 (tiga) kelas (*class*) utama sesuai dengan *dataset* yang digunakan, yaitu: Sehat (*Healthy*), Busuk Buah (*Black Pod Rot*), dan Penggerek Buah (*Pod Borer*).

1.5 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

A. Bagi Peneliti:

1. Dapat mengimplementasikan pemahaman teoritis mengenai teknologi kecerdasan buatan, khususnya *Computer Vision* dengan

metode *Faster R-CNN*, ke dalam penyelesaian masalah di sektor pertanian dan perkebunan pada studi kasus nyata.

2. Meningkatkan keterampilan teknis dalam pengembangan sistem deteksi objek berbasis *Deep Learning*, mulai dari proses pengolahan *dataset* citra buah kakao hingga tahap implementasi program antarmuka menggunakan bahasa pemrograman *Python*.

B. Bagi Objek Penelitian:

1. Memberikan alat bantu pendeteksi kesehatan buah kakao yang lebih efisien, cepat, dan objektif sebagai alternatif dari pengamatan manual yang memakan waktu dan bergantung pada pengalaman subjektif pengamat.
2. Membantu petani kakao atau penyuluh pertanian dalam mengidentifikasi secara spesifik jenis penyakit dan hama (*Black Pod Rot* dan *Pod Borer*) secara dini, guna mendukung ketepatan tindakan penanganan sehingga dapat meminimalisir risiko kerugian akibat gagal panen.

1.6 Sistematika Penulisan

Sistematika penulisan dari proposal ini terdiri dari enam bagian utama, sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi pengantar umum penelitian yang mencakup latar belakang mengenai pentingnya komoditas kakao dan ancaman penyakitnya, rumusan masalah terkait penerapan teknologi *Computer Vision*, tujuan dan manfaat penelitian, batasan masalah untuk menjaga ruang lingkup deteksi visual penyakit pada buah kakao, serta sistematika penulisan laporan.

BAB 2 LANDASAN TEORI

Bab ini menguraikan teori-teori fundamental yang menjadi landasan ilmiah penelitian. Pembahasan mencakup konsep dasar *Deep Learning*, algoritma *Convolutional Neural Network* (CNN), penjelasan mendalam mengenai arsitektur *Faster R-CNN*, serta tinjauan pustaka mengenai indikator visual penyakit dan hama pada buah kakao (seperti *Black Pod Rot* dan *Pod Borer*).

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah kerja sistematis dalam melaksanakan penelitian. Cakupannya meliputi kerangka kerja penelitian, metode pengumpulan *dataset* citra buah kakao, teknik pelabelan (*labeling*) data ke dalam 3 kelas utama, spesifikasi alat dan bahan (perangkat keras dan perangkat lunak), serta prosedur pelatihan (*training*) model *Faster R-CNN*.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berfokus pada tahap analisis kebutuhan sistem dan perancangan aplikasi. Dimulai dengan analisis kebutuhan fungsional dan non-fungsional, perancangan arsitektur sistem, pemodelan proses bisnis

menggunakan *use case diagram*, hingga perancangan antarmuka pengguna (*User Interface*) berbasis *Python*.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini memaparkan hasil dari rancangan yang telah dibuat. Bagian ini menunjukkan proses penerjemahan desain ke dalam kode program *Python*, penerapan model *Faster R-CNN* pada *dataset* citra buah kakao, serta pengujian sistem yang mencakup uji akurasi evaluasi model (*Precision*, *Recall*, *mAP*) dan pengujian fungsionalitas antarmuka menggunakan metode *Black-Box Testing*.

BAB 6 PENUTUP

Bab ini merupakan bagian penutup yang berisi kesimpulan akhir dari seluruh rangkaian penelitian guna menjawab rumusan masalah. Selain itu, bab ini memberikan saran untuk pengembangan sistem di masa depan, seperti penggunaan arsitektur model yang lebih mutakhir atau perluasan variasi *dataset* citra penyakit tanaman kakao lainnya.

BAB 2

LANDASAN TEORI

2.1 Sistem

Istilah sistem secara etimologis berakar dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*), yang merujuk pada suatu kesatuan terpadu dari berbagai elemen atau komponen yang saling terhubung guna mempermudah aliran informasi, materi, atau energi dalam mencapai tujuan tertentu. Secara konseptual, sistem menggambarkan sekumpulan entitas yang saling berinteraksi, di mana keterkaitan antar elemen tersebut sering kali dapat direpresentasikan melalui pemodelan matematika [7].

Sebagai sebuah entitas, sistem dipandang sebagai himpunan komponen fungsional yang saling berhubungan secara terorganisasi melalui metode atau proses tertentu. Keberadaan sistem menjamin integrasi berbagai fungsi agar dapat bergerak menuju sasaran yang seragam, yang secara teknis terdiri dari prosedur-prosedur khusus dalam pemrosesan data [8].

Secara operasional, sistem diartikan sebagai kumpulan variabel atau unsur yang memiliki ketergantungan satu sama lain dan bekerja sama untuk mengolah masukan (*input*) hingga menghasilkan keluaran (*output*) yang diinginkan. Dalam ranah sistem informasi, struktur ini dirancang untuk menganalisis data secara sistematis. Tujuannya adalah memfasilitasi pengumpulan serta pemahaman informasi yang relevan agar proses pengambilan keputusan menjadi lebih efisien, akurat, dan berkualitas [9].

2.2 Deteksi

Deteksi secara umum dalam konteks pengolahan citra digital adalah proses penemuan, identifikasi, atau pengenalan keberadaan suatu entitas, fitur, atau anomali tertentu di dalam sebuah data visual [10]. Proses deteksi merupakan tahapan awal yang sangat krusial sebelum sistem dapat melakukan analisis lebih lanjut, seperti pengenalan pola atau klasifikasi. Tujuan utama dari deteksi adalah memisahkan antara objek yang menjadi *Region of Interest* (RoI) dengan latar belakang (*background*) dari suatu citra [11].

Dalam praktiknya, proses deteksi mengandalkan ekstraksi fitur visual dasar seperti perubahan intensitas warna, tekstur, tepi (*edge*), dan bentuk. Sistem komputasi akan memindai keseluruhan citra untuk mencari area yang memiliki karakteristik yang sesuai dengan kriteria yang telah ditentukan [11].

2.3 Deteksi Objek

Deteksi objek (*Object Detection*) adalah salah satu cabang utama dalam teknologi *Computer Vision* yang berfokus pada penemuan instansiasi objek semantik dari kelas tertentu di dalam citra digital atau video [11]. Berbeda dengan klasifikasi citra (*image classification*) yang hanya memprediksi kelas dominan dari satu gambar secara keseluruhan, deteksi objek melakukan dua tugas komputasi secara bersamaan: pelokalan (*localization*) dan klasifikasi (*classification*) [12].

Pelokalan bertujuan untuk menemukan posisi pasti dari objek di dalam gambar, yang direpresentasikan dalam bentuk kotak pembatas (*bounding box*) berserta koordinat spasialnya (X, Y, lebar, dan tinggi). Sementara itu, tahap

klasifikasi bertugas untuk memberikan label identitas pada setiap kotak pembatas yang telah ditemukan beserta tingkat kepercayaan (*confidence score*) dari prediksi tersebut [13].

2.4 Tinjauan Umum Buah Kakao dan Penyakitnya

Kakao (*Theobroma cacao* Linnaeus .) merupakan salah satu komoditas perkebunan yang berperan dalam industri pangan global. Pertumbuhan dan perkembangan buah kakao sangat dipengaruhi oleh faktor lingkungan, perawatan, serta tingkat kerentanannya terhadap organisme pengganggu tanaman. Serangan penyakit dan hama pada buah kakao merupakan faktor utama yang dapat menurunkan kualitas biji dan kuantitas hasil panen secara drastis [14].

2.4.1 Buah Kakao Sehat

Buah kakao sehat ditandai dengan perkembangan fisik yang normal tanpa adanya indikasi infeksi patogen jamur maupun bekas serangan hama. Secara visual, kulit buah kakao sehat memiliki warna yang cerah dan merata, yang sangat bergantung pada varietas dan tingkat kematangannya. Umumnya, buah berstatus muda akan berwarna hijau atau merah keunguan, dan akan berubah menjadi kuning atau oranye secara merata saat mencapai fase masak (matang fisiologis). Permukaan kulit buah tampak bersih, padat, mengkilap, dan alur pada kulit buah terlihat tegas tanpa adanya bercak nekrotik (jaringan mati), luka lecet, atau lubang gerakan. Kondisi fisik luar yang bersih ini mengindikasikan bahwa perkembangan biji di dalamnya berlangsung secara optimal [15].



Gambar 2. 1 Buah Kakao Sehat

Untuk menjaga agar buah kakao dapat tumbuh dan berkembang secara sehat tanpa gangguan hama maupun penyakit, diperlukan tindakan pemeliharaan teknis yang rutin dan berkesinambungan. Berikut adalah tiga langkah utama dalam merawat tanaman kakao untuk memastikan pembentukan buah yang sehat secara optimal [16]:

1. Pemangkasan Teratur

Pemangkasan dilakukan dengan membuang cabang tanaman yang tidak produktif atau tidak dikehendaki agar tajuk pohon tidak terlalu tinggi dan rimbun. Pemangkasan, yang direkomendasikan pada awal atau akhir musim hujan, berfungsi untuk memperbaiki sirkulasi udara dan memastikan sinar matahari dapat menembus sela-sela daun. Pengendalian iklim mikro ini mencegah area kebun menjadi terlalu lembap, sehingga dapat menekan perkembangan patogen jamur penyebab busuk buah serta mengurangi habitat yang menguntungkan bagi perkembangbiakan hama Penggerek Buah Kakao (PBK).

2. Sanitasi Kebun

Sanitasi merupakan kegiatan membersihkan area pertanaman secara berkala guna menekan sumber infeksi (inokulum) patogen dan

memutus siklus hidup hama. Tindakan ini mencakup pembersihan gulma, pengelolaan serasah, serta yang paling utama adalah memetik dan membuang buah atau memangkas cabang yang telah terinfeksi penyakit. Menjauhkan atau menimbun buah kakao yang busuk maupun kulit buah yang berlubang karena gerakan hama sangat krusial agar larva tidak melekat atau menular ke buah yang masih sehat di sekitarnya

3. Pemupukan Berimbang

Pemberian nutrisi yang cukup memengaruhi daya tahan pohon kakao dan mencegah malnutrisi yang bisa menyebabkan bentuk buah kerdil atau rentan penyakit. Pemupukan umumnya dilakukan pada awal musim hujan untuk memaksimalkan penyerapan hara. Selain pupuk anorganik untuk menunjang hara makro (N, P, K), pemberian bahan organik sangat disarankan guna memperbaiki retensi hara (KTK tanah) dan menjaga kelembapan ketersediaan air. Tanaman dengan asupan hara yang seimbang dapat menghasilkan kulit buah yang padat, keras, dan bersih.

2.4.2 Penyakit Busuk Buah

Penyakit Busuk Buah (*Black Pod Rot*) merupakan salah satu penyakit paling merusak pada tanaman kakao yang umumnya disebabkan oleh infeksi patogen *Phytophthora palmivora*. Gejala awal dari penyakit ini ditandai dengan munculnya bercak-bercak kecil berwarna cokelat pada permukaan kulit buah. Dalam kondisi kelembapan tinggi, bercak tersebut akan membesar dengan sangat

cepat dan menyebar ke seluruh bagian permukaan, mengubah warna kulit buah menjadi cokelat tua hingga hitam pekat. Pada stadium infeksi yang lebih lanjut, buah akan membusuk secara menyeluruh. Kerusakan visual pada kulit ini berbanding lurus dengan kerusakan di bagian dalam, di mana biji kakao ikut membusuk dan tidak dapat dipanen [17].



Gambar 2. 2 Penyakit Busuk Buah Pada Kakao

Penyakit Busuk Buah umumnya disebabkan oleh infeksi patogen *Phytophthora palmivora*. Cara penanggulangan dan pengendalian yang dapat dilakukan antara lain [18]:

1. Melakukan pemetikan dan pembuangan buah yang sudah terinfeksi busuk buah agar spora jamur tidak menyebar ke buah yang sehat.
2. Mengatur kanopi pohon untuk mengurangi kelembapan kebun dan meningkatkan sirkulasi udara serta intensitas cahaya matahari yang masuk.
3. Menggunakan fungisida nabati atau kimia (berbahan aktif tembaga seperti *Copper hidroksida*) yang disemprotkan pada buah secara berkala, terutama saat musim hujan tinggi.

4. Mengaplikasikan agens hayati seperti jamur *Trichoderma sp.* pada tanah di sekitar area perakaran untuk menekan pertumbuhan patogen.

2.4.3 Hama Penggerek Buah Kakao

Penggerek Buah Kakao (PBK), yang disebabkan oleh serangan serangga *Conopomorpha cramerella*, merupakan salah satu ancaman hama terbesar pada komoditas kakao. Indikator visual serangan hama ini memiliki karakteristik yang berbeda dengan penyakit busuk buah. Gejala utama yang tampak pada kulit buah adalah terjadinya "masak awal" atau masak palsu, di mana kulit buah menguning secara tidak merata atau tampak belang-belang (campuran warna hijau, coklat, dan jingga) sebelum waktunya. Selain perubahan warna yang tidak normal, indikator visual lainnya adalah terdapatnya bekas lubang gerakan berukuran sangat kecil yang menjadi saluran keluar larva. Di sekitar lubang gerakan tersebut, seringkali ditemukan bekas kotoran hama (*frass*) atau cairan eksudat yang mengering dan berwarna kehitaman. Akibat dari serangan ini, biji kakao di dalam buah akan saling melekat kuat dan ukurannya menjadi kerdil [19].



Gambar 2. 3 Hama Penggerek Buah Pada Buah Kakao

Hama Penggerek Buah Kakao disebabkan oleh serangan serangga *Conopomorpha cramerella*. Penanganan dan pencegahan hama ini dapat diatasi melalui teknik *PsPSP* (Panen sering, Pemangkasan, Sanitasi, dan Pemupukan), yaitu [20]:

1. Memanen buah yang sudah masak setiap 1-2 minggu untuk memutus siklus hidup hama PBK sebelum larva keluar dari buah.
2. Membungkus buah kakao menggunakan plastik bening (yang ujungnya dibiarkan terbuka untuk sirkulasi) saat buah masih berukuran kecil (panjang 8-10 cm) untuk mencegah ngengat betina bertelur di permukaan buah.
3. Mengumpulkan kulit buah hasil panen dan membenamkannya ke dalam tanah atau memasukkannya ke dalam lubang rorak agar larva yang tersisa mati.
4. Melakukan penyemprotan insektisida sistemik secara selektif dan terjadwal pada buah-buahan muda jika tingkat serangan sudah sangat parah.

2.5 *Deep Learning*

Deep Learning atau pembelajaran mendalam merupakan salah satu cabang dari *Machine Learning* yang terinspirasi dari struktur dan fungsi jaringan saraf tiruan matematis (*Artificial Neural Networks*). Istilah "mendalam" (*deep*) merujuk pada jumlah lapisan (*layers*) tersembunyi yang digunakan dalam arsitektur jaringannya. Berbeda dengan jaringan saraf tiruan tradisional yang mungkin

hanya memiliki dua hingga tiga lapisan, model *Deep Learning* dapat memiliki puluhan hingga ratusan lapisan untuk memproses data [21].

Keunggulan utama dari *Deep Learning* dibandingkan dengan metode *Machine Learning* konvensional terletak pada kemampuannya dalam melakukan *feature learning* atau ekstraksi fitur secara otomatis. Pada metode konvensional, penentuan fitur (*hand-crafted features*) untuk membedakan suatu objek harus didefinisikan secara manual oleh seorang ahli (misalnya, menentukan nilai batas warna atau tekstur). Namun, *Deep Learning* mampu mempelajari representasi fitur secara hierarkis langsung dari sekumpulan data mentah. Lapisan awal pada jaringan biasanya mempelajari fitur tingkat rendah seperti garis dan tepi, sedangkan lapisan yang lebih dalam akan merangkai fitur-fitur tersebut untuk mengenali pola yang lebih kompleks [22].

Dalam domain *Computer Vision*, salah satu arsitektur *Deep Learning* yang paling revolusioner dan sering digunakan adalah *Convolutional Neural Network* (CNN). Kemampuan komputasional dari CNN dan varian tingkat lanjutnya, telah menjadikan *Deep Learning* sebagai fondasi utama dalam memecahkan masalah deteksi objek visual yang rumit [23]. Penggunaan *framework* bahasa pemrograman Python, dibantu dengan pustaka komputasi tingkat tinggi, memungkinkan pengembangan model *Deep Learning* yang efisien.

2.6 *Computer Vision*

Computer Vision merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang fokus pada pengembangan teknik agar komputer mampu memahami, menafsirkan, dan mengekstraksi informasi berharga dari data

visual, baik berupa gambar diam maupun video. Teknologi ini berupaya menduplikasi kemampuan penglihatan manusia melalui perangkat digital dan algoritma pemrosesan data [24]. Dalam perkembangannya, *Computer Vision* berkaitan erat dengan berbagai disiplin ilmu lain seperti robotika, otomasi industri, pengolahan sinyal, hingga penggunaan jaringan saraf tiruan untuk tugas-tugas kompleks seperti pengenalan wajah dan ekspresi [25].

Dalam penelitian ini, implementasi *Computer Vision* difokuskan pada dua fungsi utama:

1. Deteksi Objek: Mengidentifikasi keberadaan indikator pencemar dalam gambar dan menandai lokasinya menggunakan kotak pembatas (*bounding box*) [26].
2. Klasifikasi Objek: Menentukan kategori dari objek yang ditemukan untuk kemudian diklasifikasikan ke dalam tingkat tertentu [26].

Meskipun teknologi ini menawarkan efisiensi tinggi, penerapannya pada lingkungan perkebunan secara langsung memiliki tantangan teknis tersendiri. Kondisi nyata di lapangan cenderung memiliki variasi pencahayaan yang tidak menentu, latar belakang yang kompleks, serta varietas tanaman yang beragam, sehingga memerlukan model yang mampu beradaptasi dengan variasi data tersebut agar tetap akurat dalam mengenali ciri-ciri penyakit.

2.7 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan salah satu metode *deep learning* yang dikembangkan berdasarkan arsitektur jaringan saraf tiruan dan terbukti sangat efektif dalam mengklasifikasikan objek dengan tingkat akurasi

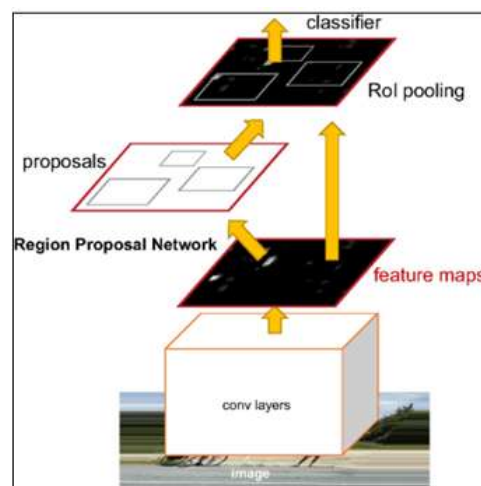
yang tinggi [27]. Dalam arsitektur deteksi objek seperti *Faster R-CNN*, CNN berperan sebagai *backbone* atau tulang belakang yang berfungsi untuk mengekstraksi fitur dari gambar masukan. Proses ini mengubah data piksel mentah menjadi sebuah *feature map* (peta fitur) agar karakteristik objek di dalam gambar lebih mudah dikenali oleh sistem computer [28].

Secara teknis, CNN bekerja dengan menerapkan filter pada gambar untuk menangkap pola visual seperti tepi, warna, hingga bentuk spesifik dari objek. *Feature map* yang dihasilkan kemudian akan diproses lebih lanjut untuk menentukan lokasi kotak pembatas (*bounding box*) dan melakukan klasifikasi terhadap objek tersebut. Beberapa poin utama mengenai peran CNN dalam penelitian ini adalah [29]:

1. Ekstraksi Fitur Otomatis: Berbeda dengan metode konvensional, CNN mampu mempelajari fitur penting dari dataset secara mandiri tanpa perlu ekstraksi manual.
2. Arsitektur Backbone: Terdapat berbagai pilihan arsitektur CNN yang dapat diintegrasikan, seperti VGG16, ZFNet, atau Inception v2, yang masing-masing memiliki karakteristik berbeda dalam hal kedalaman proses konvolusi dan beban komputasi.
3. Efisiensi Pengenalan: Penggunaan CNN memungkinkan sistem untuk mendeteksi objek dengan ukuran yang berbeda-beda namun dalam skala yang tetap serupa.

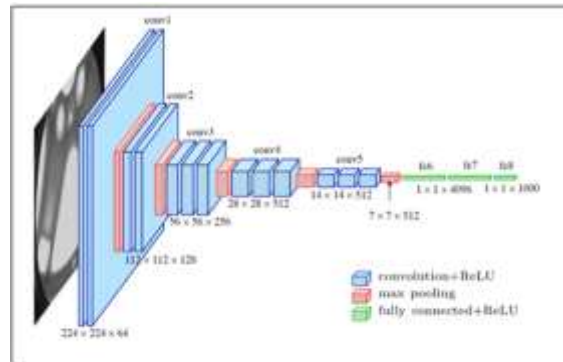
2.8 Metode *Faster R-CNN*

Faster R-CNN merupakan algoritma *deep learning* yang dirancang untuk melakukan deteksi serta klasifikasi objek secara cepat dan akurat [30]. Metode ini merupakan pengembangan dari algoritma pendahulunya, yaitu *Fast R-CNN*, dengan tetap menggunakan *Convolutional Neural Network* (CNN) sebagai tulang punggung (*backbone*) sistem. Dalam penelitian ini, *Faster R-CNN* diimplementasikan untuk mengidentifikasi keberadaan serta memetakan lokasi indikator penyakit pada buah kakao.



Gambar 2. 4 Arsitektur Metode Faster RCNN

Arsitektur *Faster R-CNN* secara umum terdiri dari tiga komponen utama, yaitu jaringan *backbone*, *neck*, dan *head*. Komponen *backbone* (seperti VGG16 atau Inception v2) bertugas mengekstraksi fitur dari gambar masukan untuk menghasilkan *feature map* [31]. Inovasi utama dalam metode ini adalah penggunaan Region Proposal Network (RPN) sebagai pengganti metode *Selective Search* yang dinilai lambat dan memperberat proses komputasi. RPN bekerja dengan memproses *feature map* untuk menentukan *Region of Interest* (RoI) atau area yang diduga kuat terdeteksi sebagai penyakit pada kakao[32].



Gambar 2. 5 Arsitektur VGG 16

2.9 Metrik Evaluasi Kinerja Model

Untuk mengukur seberapa baik model Faster R-CNN dalam mendeteksi dan mengklasifikasikan indikator penyakit pada buah kakao, digunakan beberapa metrik evaluasi standar yang didasarkan pada perhitungan *Confusion Matrix*.

1. Matriks Konfusi (*Confusion Matrix*)

Confusion Matrix adalah tabel yang merepresentasikan kinerja model dengan membandingkan nilai prediksi dengan nilai aktual (*ground truth*). Terdapat empat komponen dasar dalam matriks ini [33]:

- True Positive (TP)*: Objek dikenali dengan benar oleh model dan lokasinya sesuai dengan *ground truth* (biasanya berdasarkan nilai *Intersection over Union* / IoU di atas ambang batas tertentu).
- False Positive (FP)*: Model mendeteksi objek yang sebenarnya tidak ada, atau deteksi meleset dari *ground truth*.
- False Negative (FN)*: Objek sebenarnya ada (*ground truth*), namun model gagal mendeteksinya.

- d. *True Negative (TN)*: Model tidak mendeteksi objek pada area yang memang tidak ada objek (jarang digunakan dalam metrik *object detection*).

2. *Recall*

Recall mengukur proporsi objek positif aktual yang berhasil dideteksi dengan benar oleh model. Nilai *recall* yang tinggi menunjukkan bahwa model jarang melewatkan objek yang seharusnya dideteksi [34].

$$Recall = \frac{TP}{TP + FN}$$

3. *F1-Score*

F1-Score adalah rata-rata harmonik dari *Precision* dan *Recall*. Metrik ini sangat berguna untuk memberikan keseimbangan evaluasi jika terdapat ketidakseimbangan jumlah data antar kelas (*class imbalance*) [35].

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4. *mean Average Precision (mAP)*

mean Average Precision (mAP) adalah metrik evaluasi utama dan paling komprehensif untuk *object detection*. mAP dihitung dengan mencari luasan area di bawah kurva *Precision-Recall* (AUC-PR) untuk setiap kelas, kemudian menghitung rata-rata dari semua kelas tersebut. Semakin tinggi nilai mAP, semakin baik kemampuan model dalam melokalisasi dan mengenali objek secara akurat [36].

2.10 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan bahasa pemodelan visual standar yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak, khususnya dalam pengembangan berbasis objek [37]. UML menyediakan beragam simbol grafis yang memudahkan para pengembang dalam memvisualisasikan struktur serta perilaku sistem sebelum proses implementasi dilakukan. Dengan adanya pemodelan ini, komunikasi antara tim pengembang, analis, dan pemangku kepentingan menjadi lebih jelas karena semua pihak dapat memahami rancangan sistem melalui representasi visual yang konsisten dan sistematis [37].

Secara umum, UML terbagi menjadi dua jenis diagram utama. Pertama, diagram struktur yang menunjukkan bagian statis dari sistem, seperti *Use Case Diagram* (menjelaskan hubungan antara aktor dan sistem), *Class Diagram* (menampilkan relasi antar kelas), dan diagram komponen lainnya. Kedua, diagram perilaku, yang menggambarkan alur dan dinamika sistem, seperti *Activity Diagram* (alur proses), *Sequence Diagram* (urutan interaksi objek), serta *State Diagram* [38].

2.8.1 *Flowchart*

Dalam perancangan dan pengembangan sebuah sistem perangkat lunak, dokumentasi visual mengenai alur kerja dan logika program merupakan hal yang esensial. *Flowchart* (diagram alir) adalah representasi grafis yang menggunakan simbol-simbol standar untuk menggambarkan urutan proses, algoritma, atau instruksi kerja dari suatu sistem atau program komputer [39]. Penggunaan

flowchart bertujuan untuk menyederhanakan logika komputasi yang kompleks menjadi bentuk visual yang mudah dipahami oleh pengembang sistem maupun pemangku kepentingan (*stakeholders*) lainnya [40].

Sebuah *flowchart* disusun dari serangkaian simbol dua dimensi yang memiliki makna spesifik, yang dihubungkan dengan garis berarah (*flowlines*) untuk menunjukkan arah eksekusi aliran data atau kontrol. Beberapa simbol standar yang diakui secara internasional (*ANSI/ISO*) meliputi simbol elips untuk menandai awal (*terminator*) dan akhir program, jajaran genjang untuk proses masukan (*input*) atau keluaran (*output*) data, persegi panjang untuk mempresentasikan proses pengolahan data atau komputasi, dan belah ketupat untuk titik pengambilan keputusan (*decision*) berdasarkan kondisi tertentu [41].

2.8.2 Use Case Diagram

Use Case Diagram merupakan salah satu komponen dari *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan hubungan antara aktor dengan sistem berdasarkan fungsionalitas yang disediakan [42]. Diagram ini menekankan pada apa saja layanan atau fungsi yang harus dimiliki sistem tanpa menjelaskan bagaimana layanan tersebut dijalankan secara teknis.

Unsur utama dalam *Use Case Diagram* meliputi aktor, yaitu entitas yang berinteraksi dengan sistem, baik manusia, perangkat, maupun sistem lain; use case, yaitu representasi layanan atau fungsi yang diberikan sistem kepada aktor; serta hubungan atau *relationship* yang menghubungkan aktor dengan use case, yang dapat berupa *association*, *include*, maupun *extend* [43].

2.8.3 Sequence Diagram

Sequence Diagram merupakan jenis diagram pada *Unified Modeling Language (UML)* yang berfungsi untuk menggambarkan urutan interaksi antar objek dalam suatu sistem berdasarkan aliran waktu. Diagram ini menampilkan proses pertukaran pesan atau informasi dari awal hingga akhir, sehingga hubungan komunikasi antar objek dapat terlihat jelas [44].

Objek dalam *Sequence Diagram* digambarkan dengan lifeline (garis vertikal) yang menunjukkan keberadaannya selama proses berlangsung, sedangkan panah horizontal mewakili pesan yang dikirimkan. Alur waktu berjalan dari atas ke bawah. Diagram ini membantu menganalisis langkah-langkah proses secara kronologis, menemukan potensi hambatan, dan mempermudah pengembangan sistem [45].

2.11 Bahasa Pemrograman

Bahasa pemrograman merupakan seperangkat aturan sintaksis dan semantik yang berfungsi untuk memberikan instruksi kepada komputer dalam menjalankan suatu tugas [46]. Melalui bahasa ini, seorang pengembang perangkat lunak dapat menulis kode program yang selanjutnya akan diterjemahkan oleh *compiler* atau *interpreter* menjadi bahasa mesin sehingga dapat diproses oleh komputer. Bahasa pemrograman digunakan dalam pengembangan berbagai jenis perangkat lunak, mulai dari aplikasi, situs web, hingga sistem operasi.

Secara umum, bahasa pemrograman dibedakan menjadi bahasa tingkat rendah (*low-level language*) yang memiliki kedekatan dengan bahasa mesin, dan bahasa tingkat tinggi (*high-level language*) yang lebih mudah dipahami oleh manusia. Beberapa contoh bahasa pemrograman tingkat tinggi antara lain Python,

Java, C++, dan JavaScript. Pemilihan bahasa pemrograman biasanya mempertimbangkan tujuan pengembangan, kemudahan implementasi, serta performa yang dihasilkan.

2.12 *Python*

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido Van Rossum dan pertama kali dirilis pada tahun 1991. Saat ini, Python telah bertransformasi menjadi salah satu bahasa pemrograman yang sangat populer dan paling banyak digunakan di seluruh dunia. Keunggulan utama dari Python terletak pada sintaksnya yang sederhana, intuitif, serta sifatnya yang sangat serbaguna. Berkat dukungan ekosistem pustaka (*library*) yang masif, Python menjadi standar industri dan akademis untuk pengembangan berbagai aplikasi komputasi tingkat lanjut, khususnya di bidang *machine learning* dan *deep learning* [46].

2.13 *Streamlit*

Streamlit adalah pustaka (*library*) bersifat *open-source* berbasis bahasa pemrograman Python yang dirancang khusus untuk membangun aplikasi web kustom untuk kebutuhan *data science* dan *machine learning* secara cepat. Dalam penelitian ini, *Streamlit* digunakan sebagai kerangka kerja utama untuk membangun antarmuka pengguna (*User Interface*) yang memungkinkan warga atau peneliti di Desa Rantau Kasai berinteraksi dengan model kecerdasan buatan [47].

Beberapa keunggulan *Streamlit* yang mendukung penelitian ini meliputi [47]:

1. Pengembangan Cepat: Memungkinkan transformasi skrip Python menjadi aplikasi web interaktif dalam waktu singkat tanpa memerlukan keahlian mendalam di bidang *front-end* (seperti HTML, CSS, atau JavaScript).
2. Integrasi Model: Sangat mudah diintegrasikan dengan model *Faster R-CNN* yang dilatih menggunakan pustaka *deep learning* lainnya.
3. Visualisasi Interaktif: Mendukung tampilan hasil deteksi objek secara langsung di *browser*, sehingga memudahkan proses klasifikasi secara visual.

2.14 *OpenCV*

Open Source Computer Vision Library (OpenCV) merupakan pustaka perangkat lunak yang berfokus pada pengolahan citra digital dan visi komputer secara *real-time*. Pustaka ini bersifat serbaguna dan mendukung berbagai bahasa pemrograman, khususnya Python, untuk menjalankan fungsi-fungsi manipulasi gambar yang kompleks [48].

OpenCV berperan pada tahap prapemrosesan citra (*image pre-processing*), dengan fungsi-fungsi utama sebagai berikut:

1. Membaca Citra: Mengonversi file gambar yang diunggah pengguna menjadi *array* numerik agar dapat diproses oleh algoritma.
2. Penyesuaian Ukuran (*Resizing*): Mengubah dimensi citra menjadi standar tertentu, seperti ukuran 640x640 piksel, untuk mencapai performa pelatihan model yang optimal dan akurat.

3. Konversi Warna: Melakukan transformasi ruang warna (seperti dari BGR ke RGB) guna memastikan data visual yang diinputkan sesuai dengan format fitur yang dipelajari oleh model *Faster R-CNN*.
4. Anotasi Visual: Digunakan untuk menggambar kotak pembatas (*bounding box*) dan label teks pada hasil akhir deteksi sebelum ditampilkan di antarmuka web.

2.15 *Visual Studio Code (VSC)*

Visual Studio Code (VSC) adalah editor kode sumber yang dikembangkan oleh Microsoft dan dapat digunakan secara gratis. Editor ini mendukung berbagai bahasa pemrograman, memiliki fitur seperti *syntax highlighting*, *debugging*, integrasi *Git*, dan ekstensi yang dapat memperluas fungsinya. VSC dikenal ringan, cepat, dan dapat digunakan di berbagai sistem operasi seperti Windows, macOS, dan Linux [49].

2.16 *Web*

Web merupakan sekumpulan halaman yang saling terhubung dan dapat diakses melalui jaringan internet dengan menggunakan aplikasi peramban (*web browser*). Setiap halaman pada web biasanya dibuat dengan bahasa markup seperti HTML, dipadukan dengan CSS untuk mengatur tampilan, serta JavaScript untuk menambahkan interaktivitas [50]. Konten yang disajikan bisa berupa teks, gambar, video, audio, maupun elemen multimedia lainnya. Web sendiri dapat diakses secara global oleh pengguna di berbagai belahan dunia selama terhubung dengan jaringan internet [50].

2.17 *Blackbox Testing*

Blackbox Testing merupakan teknik pengujian perangkat lunak yang menilai fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) tanpa memperhatikan bagaimana proses internal atau kode program dijalankan. Dalam metode ini, penguji hanya berperan sebagai pengguna yang mencoba memastikan setiap fitur berjalan sesuai spesifikasi yang telah dibuat [51]. Proses ini membantu mengidentifikasi apakah hasil yang diberikan sistem sudah benar dan sesuai dengan harapan, tanpa perlu memahami struktur internal atau logika pemrogramannya.

Pengujian jenis ini umumnya diterapkan pada tahap akhir pengembangan untuk mengevaluasi sistem dari sudut pandang pengguna akhir. Keunggulan *Blackbox Testing* adalah penguji tidak memerlukan keterampilan teknis mendalam dalam pemrograman, sehingga fokus pengujian dapat diarahkan pada kelengkapan fitur, kemudahan penggunaan, serta kesesuaian fungsi dengan kebutuhan pengguna. Dengan pendekatan ini, kualitas perangkat lunak dapat dinilai secara objektif berdasarkan performa yang terlihat [52].

2.18 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Nama	Tahun	Judul	Hasil
1.	Rizky Ridho Ramadhan dan Daniel Udjulawa	2026	Deteksi dan Klasifikasi penyakit pada buah kakao menggunakan yolov11	Penelitian ini berhasil menunjukkan bahwa penerapan model YOLOv11m untuk mendeteksi dan mengklasifikasikan penyakit pada buah kakao menghasilkan peningkatan signifikan setelah diterapkan teknik augmentasi data,

				khususnya augmentasi Roboflow dan input-space oversampling. Model yang dilatih dengan teknik ini menunjukkan peningkatan besar pada metrik Precision, Recall, dan mAP, terutama untuk kelas minoritas. Namun, penurunan recall pada data uji dan kesalahan klasifikasi pada buah berukuran ke cil menunjukkan bahwa model masih memiliki keterbatasan dalam menangani ketidakseimbangan kelas pada variasi ukuran objek.
2.	Teguh Raharjo, Herman Purwoko Putro dan Herva Emilda Sari	2026	Early Detection of Tomato Plant Diseases Using the Real-Time Detection Transformer (RT-DETR) Model	Model RT-DETR menunjukkan kinerja yang sangat baik dalam mendeteksi penyakit pada tanaman tomat secara dini. Berdasarkan hasil pengujian, RT-DETR yang dilatih selama 60 epoch mampu menghasilkan akurasi sebesar 96,1%, yang mencerminkan tingkat ketepatan yang tinggi dalam mengidentifikasi area daun yang terinfeksi penyakit. Hasil ini menunjukkan efisiensi pelatihan model, mengingat pada penelitian sebelumnya dibutuhkan lebih dari 100 epoch untuk mencapai tingkat akurasi di atas 95%. Meskipun menunjukkan performa yang menjanjikan, hasil penelitian ini masih terbatas pada dataset

				berukuran 1.000 citra dan belum diuji pada kondisi lapangan secara langsung. Oleh karena itu, implementasi RT-DETR dalam sistem pertanian presisi memerlukan pengujian lanjutan pada skala yang lebih besar dan kondisi lingkungan nyata.
3.	Suastika Yulia Riska dan Achmad Noercholis	2024	Performance Comparison Of Faster R-Convolutional Neural Network (Cnn) And Efficientnet For Train Detection Under Diverse Lighting And Image Quality Conditions	Model EfficientNet dan Faster R-CNN memiliki kelebihan dan kekurangan masing-masing dalam berbagai aspek. Dalam hal efisiensi, EfficientNet lebih unggul dengan waktu training yang lebih cepat (5 jam 37 menit) dan ukuran model yang lebih kecil (321.2 MB). Sehingga, menjadikan pilihan yang tepat untuk perangkat dengan keterbatasan sumber daya. Namun, Faster R-CNN unggul dalam hal kinerja deteksi, dengan nilai Average Precision (AP) dan Average Recall (AR) yang lebih tinggi di berbagai kondisi pencahayaan dan gangguan noise. Model ini menjadi pilihan yang lebih baik untuk aplikasi yang memerlukan akurasi tinggi dan ketahanan terhadap variasi lingkungan.
4.	Gracia Yoel Christiawan, Roy Andani Putra, Azis Sulaiman, Evy Poerbaningtyas, dan Syntia	2023	Penerapan Metode Convolutional Neural Network (CNN) Dalam Mengklasifikasikan Penyakit Daun Tanaman Padi	Algoritma CNN efektif untuk pemrosesan gambar. Jumlah epoch berpengaruh pada hasil training, namun tidak signifikan. Model hasil training dapat digunakan pada sistem

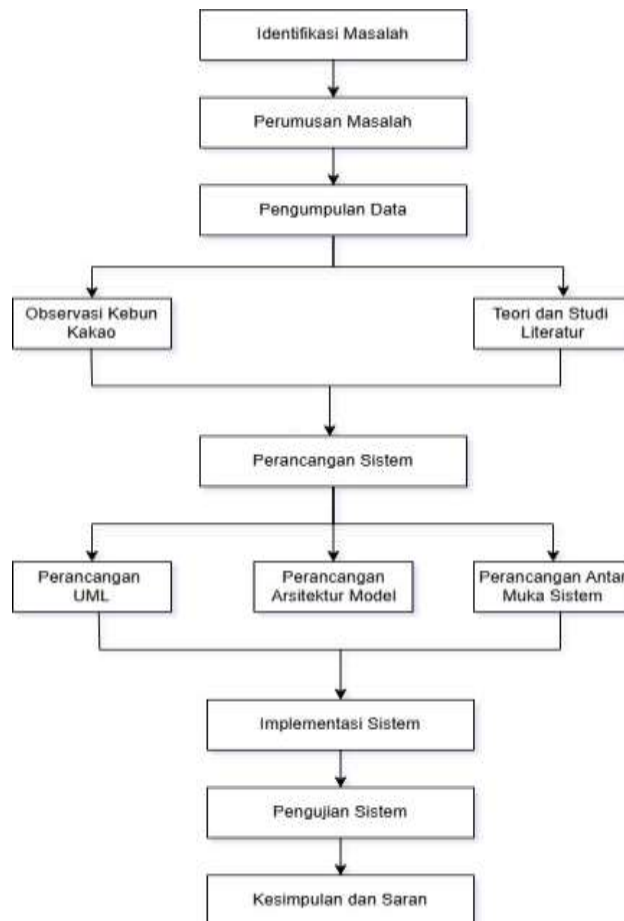
	Widyayuningtias Putri Listio			website untuk menampilkan prediksi dari model CNN yang telah dilatih. Model CNN dilatih dengan 75 epoch, batch 32, dan data augmentasi dengan pengurangan brightness 25% untuk mencapai akurasi tinggi dan berfungsi dengan baik. Saran untuk penelitian selanjutnya ialah mencoba jumlah dataset yang lebih besar. Penulis terbatas pada 500 dataset per penyakit karena keterbatasan device yang dimiliki. Memperbanyak jumlah penyakit yang dideteksi melalui dataset tanpa batasan 3 penyakit. Mengembangkan fitur sistem website untuk menunjukkan letak BPP(Badan Penyuluhan Pertanian) dan info terbaru.
5.	Rito Cipta Sigitta Hariyono, Khoerunisa, Nurul Mega Saraswati, Rizky Noor Prasetyono, dan M. Zidan Alfariki	2023	Deteksi Penyakit Bercak Coklat, Coklat Sempit Dan Hawar Melalui Spektrum Warna Citra Digital Daun Padi Menggunakan Metode Convolutional Neural Network	Hasil penelitian ini dapat disimpulkan bahwa, Tingkat akurasi yang didapatkan dari penerapan metode Convolutional Neural Network (CNN) untuk mendeteksi penyakit pada tanaman padi berdasarkan morfologi citra bercak daun padi menghasilkan akurasi yang berbeda pada masing-masing arsitektur, berdasarkan 2 jenis arsitektur yang digunakan dalam penelitian ini yaitu arsitektur Letnet-5 menghasilkan tingkat akurasi terbaik sebesar

				85% pada epoch ke 40 dan pada arsitektur Custom menghasilkan tingkat akurasi terbaik yaitu 90% pada epoch ke 29.
--	--	--	--	--

BAB 3

METODOLOGI PENELITIAN

Metodologi penelitian merupakan rangkaian langkah, prosedur, dan teknik sistematis yang disusun sebagai panduan bagi peneliti dalam mengumpulkan, mengolah, serta menganalisis data guna menjawab rumusan masalah yang telah ditetapkan. Adapun tahapan-tahapan pelaksanaan penelitian secara menyeluruh dapat dilihat pada Gambar 3.1 berikut:



Gambar 3. 1 Tahapan Metodologi Penelitian

3.1 Identifikasi Masalah

Identifikasi masalah merupakan tahap awal dalam penelitian yang bertujuan untuk menemukan, memahami, dan merumuskan permasalahan utama yang akan dikaji. Langkah ini dilakukan dengan mengamati fenomena di lapangan, menganalisis kondisi produksi kakao saat ini, serta membandingkannya dengan kondisi ideal yang diharapkan bagi produktivitas perkebunan.

Pada penelitian ini, identifikasi masalah difokuskan pada ancaman hama dan penyakit yang menyebabkan penurunan kualitas serta kuantitas hasil panen kakao secara signifikan. Masalah utama yang diidentifikasi adalah keberadaan penyakit busuk buah (*Black Pod Rot*) dan hama penggerek buah kakao (*Pod Borer*) yang seringkali sulit dibedakan oleh petani dengan pengetahuan terbatas. Kendala lainnya adalah metode identifikasi konvensional yang mengandalkan tenaga ahli seringkali memakan waktu lama dan tidak efisien untuk area perkebunan yang luas. Kesenjangan (*gap*) antara kebutuhan deteksi dini dan keterbatasan pengetahuan petani inilah yang memunculkan gagasan untuk membangun sistem deteksi otomatis berbasis *Computer Vision* sebagai solusi alternatif yang lebih akurat dan efisien.

3.2 Perumusan Masalah

Tahap perumusan masalah merupakan langkah metodologis lanjutan untuk membatasi dan menjabarkan ruang lingkup penelitian ke dalam bentuk operasional yang spesifik. Berdasarkan identifikasi masalah pada kesehatan buah kakao, tahap ini difokuskan pada perancangan skenario teknis, mulai dari penentuan arsitektur *Faster R-CNN* yang optimal, penentuan parameter *dataset*

visual penyakit, hingga perancangan skenario pengujian akurasi. Tujuannya adalah memastikan model mampu mengklasifikasikan kondisi buah ke dalam kategori yang tepat (Sehat, Busuk Buah, atau Penggerek Buah) berdasarkan fitur morfologi yang terdeteksi secara otomatis.

3.3 Pengumpulan Data

Pengumpulan data merupakan tahapan krusial dalam penelitian berbasis *Computer Vision*, karena kualitas dan kuantitas data akan sangat menentukan tingkat akurasi model *Deep Learning* yang dibangun. Dalam penelitian ini, proses pengumpulan data dibagi menjadi beberapa tahap:

A. Pengambilan Citra Primer

Dataset citra dikumpulkan dari hasil pemotretan langsung di area perkebunan maupun penggunaan *dataset* yang mewakili kondisi riil lapangan. Pengambilan citra dilakukan dengan berbagai variasi pencahayaan dan sudut pandang (*angle*) agar model memiliki kemampuan generalisasi yang baik terhadap objek buah kakao.

B. Teori dan Studi Literatur

Pengumpulan data sekunder dilakukan dengan menghimpun teori-teori dari buku, *e-book*, dan dokumentasi resmi. Fokus kajian literatur meliputi karakteristik biologis penyakit kakao, teori dasar pengolahan citra digital, serta algoritma *Deep Learning* khususnya arsitektur *Faster R-CNN*.

C. Jurnal Terdahulu

Langkah ini dilakukan dengan mengkaji berbagai artikel ilmiah atau jurnal penelitian terdahulu yang berfokus pada deteksi penyakit tanaman dan penerapan *Computer Vision* di sektor pertanian. Pengkajian ini bertujuan untuk menemukan celah penelitian (*research gap*) dan membandingkan performa model sebelumnya sebagai landasan akademis.

3.4 Perancangan Sistem

Tahap perancangan sistem memodelkan interaksi antara perangkat lunak dan algoritma sebelum memasuki tahap pengkodean (*coding*). Perancangan ini dibagi menjadi tiga aspek spesifik:

A. Perancangan UML

Perancangan ini berfokus pada pemodelan alur kerja antarmuka pengguna. Pemodelan dilakukan menggunakan *Unified Modeling Language* (UML), yang mencakup *Use Case Diagram* untuk mendefinisikan peran pengguna (petani/peneliti), serta *Activity Diagram* untuk memetakan alur kerja sistem mulai dari proses unggah foto buah kakao hingga sistem menampilkan hasil klasifikasi penyakit.

B. Perancangan Arsitektur Model

Perancangan ini berfokus pada struktur algoritma kecerdasan buatan. Prosesnya meliputi penentuan ukuran *input* citra agar komputasi lebih efisien, pemilihan arsitektur *backbone* untuk ekstraksi fitur (seperti tekstur busuk buah atau lubang hama), hingga penyesuaian *hyperparameter* pada *Region Proposal Network* (RPN) agar model

Faster R-CNN optimal dalam mendeteksi area infeksi yang sangat kecil pada kulit buah kakao.

C. Perancangan Antarmuka Sistem

Bagian ini berfokus pada perancangan tata letak (*layout*) sistem antarmuka bagi pengguna akhir. Antarmuka web dirancang agar interaktif, di mana pengguna dapat dengan mudah melihat hasil keluaran sistem berupa *bounding box* deteksi lokasi penyakit beserta label persentase kepercayaannya

3.5 Implementasi Sistem

Tahap implementasi adalah proses menerjemahkan rancangan konseptual ke dalam kode program yang dapat dieksekusi. Keseluruhan sistem dibangun menggunakan bahasa pemrograman *Python*. Tahap prapemrosesan citra memanfaatkan pustaka *OpenCV*, sementara arsitektur *Faster R-CNN* dibangun menggunakan kerangka kerja *Deep Learning TensorFlow*.

3.6 Pengujian Sistem

Sistem yang telah diimplementasikan diuji secara menyeluruh melalui dua pendekatan utama:

1. Pengujian Evaluasi Model: Bertujuan mengukur akurasi analitis model *Faster R-CNN* menggunakan *Confusion Matrix* untuk menghitung metrik performa seperti *Precision*, *Recall*, dan *Mean Average Precision* (mAP).
2. Pengujian Fungsionalitas (*Blackbox Testing*): Bertujuan menguji antarmuka *Streamlit* dari perspektif pengguna akhir, memastikan

fungsi unggah foto dan visualisasi *bounding box* hasil deteksi berjalan dengan tepat tanpa kesalahan teknis.

3.7 Kesimpulan dan Saran

Tahap terakhir adalah penarikan kesimpulan berdasarkan hasil pengujian untuk menjawab rumusan masalah yang ditetapkan pada BAB 1. Selain itu, dirumuskan saran konstruktif terkait pengembangan model di masa depan, seperti penambahan variasi *dataset* penyakit tanaman kakao lainnya atau optimasi arsitektur model agar lebih ringan saat dijalankan pada perangkat *mobile*.