DANI SASMOKO, S.T., M. ENG.

# COMPUTER VISION MODERN

## MODEL, ARSITEKTUR, DAN APLIKASI

YAYASAN PRIMA AGUS TEKNIK

DANI SASMOKO, S.T., M. ENG.

# COMPUTER VISION MODERN

## MODEL, ARSITEKTUR, DAN APLIKASI

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-51-5 (PDF)
9 786347 227515

**COMPUTER VISION MODERN**
**Model, Arsitektur dan Aplikasi**

**Penulis :**
Dani Sasmoko, S.T., M.Eng

**ISBN : 978-634-7227-51-5**

**Editor :**
Reni Veliyanti, S.Kom, M.Kom

**Penyunting :**
Irdha Yunianto, S.Ds., M.Kom

**Desain Sampul dan Tata Letak :**
Dani Sasmoko, S.T., M.Eng

**Penebit** :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

**Anggota IKAPI No:** 279 / ALB / JTE / 2023

**Redaksi :**
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

**Distributor Tunggal :**
**Universitas STEKOM**
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : info@stekom.ac.id

## KATA PENGANTAR / UCAPAN TERIMAKASIH

### Kata Pengantar

Puji syukur kita panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya buku yang berjudul "COMPUTER VISION MODERN" dapat diselesaikan dengan baik. Buku ini hadir sebagai upaya untuk memberikan pemahaman mendalam mengenai perkembangan, konsep, serta penerapan computer vision di era digital yang semakin maju.

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence/AI) dan pembelajaran mesin (Machine Learning) telah membawa dampak signifikan pada berbagai bidang kehidupan. Salah satu cabang yang berkembang pesat adalah computer vision, yaitu kemampuan komputer untuk "melihat", mengenali, serta menafsirkan informasi visual dari gambar maupun video. Kehadiran teknologi ini telah memberikan kontribusi besar dalam berbagai sektor, mulai dari kesehatan, transportasi, keamanan, industri manufaktur, hingga hiburan.

Buku ini disusun untuk memberikan gambaran yang komprehensif mengenai dasar-dasar computer vision, perkembangan metode klasik hingga pendekatan modern berbasis deep learning, serta implementasinya dalam berbagai aplikasi nyata. Dengan penyajian yang sistematis, penulis berharap pembaca dapat memahami evolusi teknologi ini sekaligus menguasai konsep-konsep penting yang menjadi fondasi penerapannya.

Ucapan terima kasih penulis sampaikan kepada semua pihak yang telah memberikan dukungan, baik berupa saran, masukan, maupun motivasi selama proses penulisan buku ini. Semoga buku ini dapat bermanfaat bagi mahasiswa, peneliti, praktisi, maupun masyarakat umum yang tertarik mendalami bidang computer vision.

Akhir kata, penulis menyadari bahwa buku ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan pada edisi berikutnya.

Semarang, 25 Juli 2025

Dani Sasmoko

## Daftar Isi

# BAB I
# PERKEMBANGAN ALGORITMA VISION

## 1.1 Definisi Computer Vision

Dalam dunia yang semakin terdigitalisasi, peran data visual menjadi sangat penting. Foto, video, dan berbagai bentuk citra digital kini hadir di hampir setiap aspek kehidupan manusia. Mulai dari kamera pengawas di ruang publik, sistem navigasi kendaraan otonom, hingga pemeriksaan medis berbasis citra digital—semuanya melibatkan proses interpretasi terhadap informasi visual. Di sinilah peran *Computer Vision* menjadi sangat vital. Computer Vision atau visi komputer merupakan salah satu cabang dari kecerdasan buatan (AI) yang fokus pada bagaimana komputer bisa memperoleh, memproses, dan memahami informasi dari gambar atau video—layaknya cara kerja sistem visual manusia.

### 1.1.1 Pengertian Dasar

Secara ringkas, Computer Vision adalah disiplin ilmu yang berusaha mengembangkan system yang dapat mengamati dan memahami dunia visual seperti yang dilakukan oleh manusia. Namun, berbeda dengan mata manusia yang secara biologis terhubung ke otak melalui system saraf, sistem penglihatan komputer bergantung pada perangkat keras (seperti kamera dan sensor) serta perangkat lunak (seperti algoritma dan model kecerdasan buatan) untuk menginterpretasikan citra digital.

Dalam definisi yang lebih formal, visi komputer dapat dipahami sebagai proses otomatisasi dan reproduksi system persepsi visual manusia melalui pemrograman komputer. Tujuannya adalah untuk memungkinkan mesin mengenali pola, mengklasifikasikan objek, mendeteksi gerakan, memahami konteks spasial, dan pada tahap lebih lanjut mengambil keputusan atau bertindak berdasarkan informasi visual tersebut.

Selain itu Computer Vision (CV) adalah cabang ilmu komputer yang berfokus pada bagaimana komputer dapat meniru, memahami, dan bahkan melampaui kemampuan manusia dalam melihat. “Melihat” di sini tidak sekadar berarti menangkap gambar melalui kamera, melainkan mencakup proses lebih kompleks: mulai dari menginterpretasi, mengklasifikasi, hingga mengambil keputusan berbasis informasi visual.

Dalam kehidupan sehari-hari, manusia mengandalkan penglihatan untuk 70–80% aktivitas kognitifnya. Kemampuan mengenali wajah teman, membaca rambu lalu lintas, hingga memahami ekspresi lawan bicara semuanya adalah hasil dari sistem visual yang luar biasa kompleks. Computer Vision mencoba mereplikasi hal ini dengan menggunakan algoritma, model matematis, dan kecerdasan buatan.

Namun, ada perbedaan fundamental antara cara manusia dan mesin “melihat”. Manusia menggunakan pengalaman, konteks, dan intuisi; mesin menggunakan data, aturan, serta pola numerik. Computer Vision tidak “melihat” dengan mata, melainkan “menghitung” citra menjadi angka dan matriks, lalu menemukan pola yang bermakna.

### 1.1.2 Posisi Computer Vision dalam Spektrum Ilmu

Computer Vision tidak bisa dilepaskan dari disiplin ilmu lain. Dari sisi teknologi, ia berdiri di antara pengolahan citra digital, kecerdasan buatan, dan pembelajaran mesin. Dari sisi sains dasar, ia bertumpu pada matematika, fisika cahaya, serta statistika. Dari sisi biologi dan psikologi, computer vision berhutang inspirasi pada studi tentang bagaimana manusia dan hewan memproses informasi visual. Bahkan, dari sisi ilmu sosial, computer vision bersinggungan dengan hukum (misalnya regulasi privasi), etika, hingga filsafat teknologi.

Keterhubungan multidisipliner ini menunjukkan bahwa belajar computer vision tidak hanya mengasah kemampuan teknis, tetapi juga menuntut pemahaman lintas bidang. Seorang insinyur computer vision yang baik bukan hanya piawai mengutak-atik kode, melainkan juga mampu melihat dampak sosial dan etis dari teknologi yang ia kembangkan.

Computer vision adalah bidang multidisiplin yang bersinggungan langsung dengan berbagai disiplin ilmu lainnya, antara lain:

- **Pengolahan Citra Digital**
  Di sinilah asal teknis computer vision. Tugas pengolahan citra adalah memodifikasi gambar agar lebih mudah dianalisis—misalnya dengan meningkatkan kontras, menghilangkan noise, atau mengekstraksi fitur tertentu.
- **Kecerdasan buatan**
  Dalam konteks AI, computer vision adalah salah satu bentuk penerapan kecerdasan. Sistem AI dipakai untuk mengerti isi gambar atau video dan membuat keputusan berdasar pemahaman itu.
- **Pembelajaran Mesin**
  Modern computer vision ebanyakan berhubungan dengan machine learning algorithms, terutama deep learning. Model ini dilatih dengan dataset besar untuk mempelajari representasi visual dan membuat prediksi yang tepat.
- **Statistik dan Matematika Terapan**
  Banyak teknik dalam computer vision yang memanfaatkan konsep statistik, seperti estimasi probabilitas, pengenalan pola, dan transformasi matematis Fourier transform atau wavelet analysis.

### 1.1.3 Tujuan dan Fungsi Utama

Tujuan utama pemahaman visual komputer adalah meniru dan melampaui kemampuan persepsi visual manusia untuk memahami dunia nyata. Pengenalan Objek: Mendeteksi dan mengklasifikasikan objek dalam video atau foto. Mengidentifikasi orang, mobil, atau barang di rak toko adalah contohnya.

- **Deteksi dan Pelacakan Objek (Object Detection and Tracking)** termasuk mengidentifikasi objek, menemukan lokasinya, dan melacak pergerakannya sepanjang waktu.
- **Segmentation Gambar**: Membagi beberapa area dalam gambar dengan garis atau fitur tertentu, seperti membedakan background dari subjek utama.
- **Analisis Gerakan,** juga dikenal sebagai "analisis gerakan", adalah proses melacak perubahan tempat objek dalam urutan gambar atau video untuk mengidentifikasi arah, kelajuan, atau jenis aktivitas.

- **Rekonstruksi 3D** menghasilkan model tiga dimensi dari data gambar dua dimensi. Ini berguna untuk aplikasi robotik, AR, dan pemetaan.

Secara umum, ada tiga tujuan besar computer vision: deteksi, pemahaman, dan pengambilan keputusan.

1. Deteksi → sistem harus mampu menemukan objek atau fitur visual tertentu, misalnya mendeteksi wajah di tengah kerumunan.
2. Pemahaman → sistem harus memahami konteks, misalnya membedakan antara " seseorang sedang berjalan di zebra cross" dengan "seseorang berdiri di trotoar".
3. Pengambilan keputusan → hasil dari proses vision digunakan untuk tindakan lanjut, misalnya mobil otonom yang mengerem saat mendeteksi pejalan kaki.

Di luar itu, computer vision juga berfungsi sebagai jembatan antara manusia dan mesin. Contoh sederhana adalah teknologi augmented reality (AR) yang memungkinkan kita berinteraksi dengan objek digital seolah-olah ada di dunia nyata.

### .1.1.4 Analog Visualisasi Manusia vs Mesin

Seringkali mahasiswa baru bingung membedakan image processing dengan computer vision. Image processing berfokus pada transformasi citra agar kualitasnya lebih baik atau informasi tertentu bisa diambil (misalnya memperbaiki kontras, mengurangi noise). Sebaliknya, computer vision berfokus pada pemahaman citra.

Analogi sederhana: image processing ibarat "membersihkan kacamata agar gambar lebih jelas", sedangkan computer vision adalah "menggunakan mata untuk mengerti isi gambar itu".

Meskipun memiliki tujuan yang sama, mekanisme sistem visual manusia dan mesin sangat berbeda. Mata manusia mengambil cahaya melalui lensa dan otak menerjemahkannya menjadi arti. Sebaliknya, gambar ditangkap oleh kamera dalam bentuk piksel digital yang perlu dianalisis secara numerik. Untuk meniru makna yang secara alami dimengerti oleh manusia, seperti membedakan antara gambar wajah asli dan gambar wajah, atau mengenali objek yang tampak sebagian tertutup, sistem visi komputer harus bekerja ekstra keras.

Fakta bahwa informasi visual bersifat ambigu, kompleks, dan sangat kontekstual merupakan masalah utama dalam penglihatan komputer. Objektif yang sama mungkin tampak berbeda tergantung pada pencahayaan, sudut pandang, rotasi, dan kondisi lingkungan. Oleh karena itu, sistem visi komputer membutuhkan algoritma pembelajaran yang dapat digeneralisasi dengan baik dan representasi data yang sangat kaya.

### 1.1.5 Implementasi dalam Kehidupan Nyata

Banyak aplikasi modern bergantung pada visi komputer. Contoh nyata dari aplikasinya adalah sebagai berikut:

- **Kendaraan otonom:** Mobil tanpa pengemudi yang menggunakan kamera dan sistem visi untuk membaca marka jalan, mengenali pejalan kaki, dan menghindari tabrakan.
- **Kesehatan Digital:** Algoritma penglihatan digunakan dalam analisis radiologi, seperti deteksi tumor pada MRI atau CT-scan.
- **Pertanian Presisi:** Petani kontemporer memantau kesehatan tanaman secara real-time melalui penggunaan drone dan sistem penglihatan digital.

- **Sistem Keamanan:** AI dapat mendeteksi perilaku mencurigakan dan mengirimkan peringatan otomatis.
- **Industri Manufaktur:** Proses inspeksi kualitas dilakukan oleh kamera pintar yang secara otomatis menemukan kesalahan pada produk.

Implementasi computer vision sudah sangat dekat dengan kehidupan kita. Hampir setiap mahasiswa mungkin sudah menggunakannya tanpa sadar:

- Face unlock di smartphone → contoh nyata sistem pengenalan wajah.
- Filter Instagram/TikTok → mendeteksi titik-titik wajah untuk menambahkan efek.
- Google Lens → mengenali teks, objek, hingga menerjemahkan bahasa.
- Aplikasi kesehatan → mendiagnosis penyakit kulit hanya dari foto.

Di Indonesia, computer vision banyak dimanfaatkan untuk pertanian cerdas, seperti deteksi penyakit pada daun padi atau sawi, serta untuk transportasi cerdas, seperti kamera lalu lintas yang otomatis merekam pelanggaran.

### 1.1.6 Perbedaan dengan Image Processing

Seberapa pun canggihnya, computer vision menghadapi keterbatasan. Misalnya, sistem sering gagal jika kondisi pencahayaan buruk atau objek tertutup sebagian. Tantangan lain adalah kebutuhan data yang sangat besar; model deep learning modern memerlukan jutaan gambar berlabel, sesuatu yang sulit dipenuhi jika konteksnya spesifik seperti penyakit tanaman tropis.

Selain itu, ada isu bias algoritmik. Jika model dilatih dengan data yang tidak beragam, hasilnya bisa diskriminatif. Misalnya, sistem pengenalan wajah yang lebih akurat untuk kulit terang dibanding kulit gelap. Hal ini menimbulkan pertanyaan serius tentang keadilan dalam teknologi. Selain itu terdapat juga kebingungan antara computer vision dan image processing sering terjadi. Tujuan akhir mereka membedakan mereka satu sama lain. Computer vision berusaha untuk memahami gambar dan membuatnya lebih jelas, sedangkan pengolahan citra berkonsentrasi pada mengubah gambar agar lebih informatif atau mudah dilihat. Misalnya, memperjelas gambar buram termasuk dalam domain pengolahan gambar, sementara mengenali wajah dalam gambar termasuk dalam domain penglihatan komputer.

### 1.1.7 Tantangan dan Isu Kontemporer

Masa depan computer vision diprediksi akan mengarah ke integrasi dengan AI multimodal, yang menggabungkan penglihatan, bahasa, dan suara. Model seperti CLIP (OpenAI) menunjukkan bahwa sistem bisa memahami hubungan antara gambar dan teks. Foundation models seperti SAM (Segment Anything Model) bahkan mampu melakukan segmentasi objek baru tanpa pelatihan khusus.

Dalam konteks ini, computer vision tidak lagi berdiri sendiri, melainkan menjadi bagian dari ekosistem AI yang lebih luas.

Computer vision masih menghadapi beberapa masalah meskipun teknologi telah berkembang pesat. Beberapa di antaranya adalah sebagai berikut:

- **Generalisasi Model:** Banyak sistem bekerja dengan baik pada data pelatihan, tetapi tidak dapat mengidentifikasi variasi dunia nyata.
- **Bias Data**: Model dapat membuat prediksi yang tidak akurat jika dataset pelatihan tidak representatif.
- **Kebutuhan Komputasi Tinggi:** Model deep learning seperti CNN atau Transformer membutuhkan GPU dan sumber daya besar.
- **Privasi dan Etika:** Penggunaan kamera visi di tempat publik menimbulkan masalah hukum dan etis.

### 1.1.8 Masa Depan Computer Vision

Inspirasi computer vision banyak berasal dari penelitian neurosains. Penemuan tentang sel reseptor di korteks visual mamalia, misalnya, mengilhami desain lapisan convolution pada CNN. Dari sisi psikologi kognitif, studi tentang persepsi wajah membantu pengembangan algoritma pengenalan wajah. Dari sisi linguistik, computer vision berperan penting dalam sistem penerjemahan teks berbasis kamera. Dengan demikian, computer vision adalah bidang yang kaya kolaborasi lintas disiplin, bukan domain tunggal.

Fokus penelitian saat ini mulai bergerak dari sekadar "penglihatan" ke pemahaman penuh tentang konteks visual secara semantik dan logis, dan tren masa depan menunjukkan bahwa kecerdasan buatan akan semakin berintegrasi dengan teknologi lain seperti edge computing, sensor Internet of Things, dan multimodal AI. Perkembangan model dasar seperti CLIP (OpenAI) atau DINOv2 (Meta) menandai era baru di mana sistem visi dapat berinteraksi dengan bahasa alami dan logika dunia nyata dengan lebih cerdas.

### 1.1.9 Dimensi Interdisipliner Computer Vision

Computer Vision bukanlah bidang yang berdiri sendiri, melainkan tumbuh dari hasil dialog dengan berbagai disiplin ilmu. Dari sisi neurosains, misalnya, penelitian mengenai cara kerja mata dan otak manusia dalam mengenali objek menjadi inspirasi bagi arsitektur jaringan saraf tiruan. Mekanisme sel reseptor di retina, jalur sinyal menuju korteks visual, hingga bagaimana otak menyusun persepsi spasial, banyak dijadikan analogi dalam merancang lapisan-lapisan convolutional neural network.

Dari sisi psikologi kognitif, computer vision berhubungan erat dengan teori persepsi manusia. Eksperimen tentang bagaimana manusia mengenali wajah, pola, atau huruf memberikan wawasan tentang “fitur” visual apa yang sebenarnya penting. Demikian pula, linguistik ikut berperan ketika computer vision dikaitkan dengan pemahaman bahasa alami, terutama dalam sistem multimodal yang harus menghubungkan teks dengan citra.

Melalui keterkaitan ini, jelas bahwa computer vision bukan hanya domain teknik komputasi, melainkan hasil kerja bersama antara ilmu komputer, biologi, psikologi, matematika, hingga ilmu sosial. Hal ini membuat computer vision bersifat multidimensi: ia meniru manusia, melampaui batas biologis, dan sekaligus menghadirkan persoalan filosofis baru tentang batasan kecerdasan buatan.

Pertanyaan ini tidak bisa diabaikan. Mesin memang mampu mengenali objek dengan akurasi tinggi, tetapi apakah itu berarti ia benar-benar “melihat”? Atau hanya menghitung pola matematis?

Filsuf teknologi membedakan antara persepsi (kemampuan sensoris) dan pemahaman (kemampuan konseptual). Mesin mungkin memiliki persepsi, tetapi apakah ia punya pemahaman? Diskusi ini penting karena akan memengaruhi bagaimana kita memperlakukan teknologi: apakah hanya alat, atau entitas yang memiliki “cara pandang” tersendiri.

## 1.2 Sejarah Singkat Perkembangan Algoritma Vision

Evolusi teknologi komputer tidak mempengaruhi kemajuan visi komputer. Bidang ini telah mengalami transformasi luar biasa sejak upaya awal untuk mengenali pola sederhana dalam gambar hingga munculnya sistem berbasis pembelajaran mendalam yang mampu mengalahkan manusia dalam beberapa tugas pengenalan visual. Sangat penting bagi mahasiswa atau peneliti untuk memahami sejarah perkembangan algoritma visi agar mereka dapat memahami "cara kerja" sistem saat ini dan memahami mengapa pendekatan tertentu muncul, berkembang, dan kadang-kadang tergantikan.

### 1.2.1 Periode Awal: Pionir (1960–1980-an)

Peneliti di MIT, Stanford, dan universitas lain di Amerika Serikat mulai menyelidiki kemampuan komputer untuk mengenali pola visual pada dekade 60-an, yang memulai perkembangan bidang ilmiah yang dikenal sebagai visi komputer. Algoritma visi saat ini masih eksperimental dan sangat terbatas.

Sejarah computer vision tidak bisa dilepaskan dari lahirnya disiplin kecerdasan buatan (AI) di pertengahan abad ke-20. Pada tahun 1966, di MIT, seorang profesor bernama Marvin Minsky memberikan proyek musim panas kepada mahasiswanya dengan judul Summer Vision Project. Tujuannya sederhana: membuat komputer yang mampu mengenali objek dalam ruangan menggunakan kamera. Proyek ini diperkirakan hanya butuh waktu tiga bulan.

Namun, kenyataannya jauh lebih rumit. Mahasiswa-mahasiswa tersebut segera menyadari bahwa mengajarkan komputer untuk mengenali objek bukanlah perkara sederhana. Objek bisa tampak berbeda tergantung pencahayaan, sudut pandang, atau kondisi lingkungan. Dari kegagalan inilah muncul kesadaran bahwa “penglihatan” adalah salah satu tantangan terbesar dalam AI.

"Proyek Visi Musim Panas" MIT adalah proyek yang sangat terkenal yang bertujuan untuk membuat sistem yang dapat mengenali objek sehari-hari dari gambar 2D. Proyek ini menargetkan pencapaian besar ini dalam waktu tiga bulan, yang menunjukkan betapa rumitnya visi komputer pada saat itu. Meskipun proyek ini tidak mencapai tujuannya, ia berfungsi sebagai momen penting untuk memberi tahu komunitas ilmiah bahwa penglihatan komputer adalah masalah yang jauh lebih sulit daripada yang dipikirkan.

Pada dekade 1970-an, penelitian berfokus pada representasi sederhana seperti tepi (edge) dan bentuk geometris. Metode seperti edge detection (Canny, Sobel) mulai populer. Konsep segmentasi citra juga berkembang, meskipun masih terbatas pada citra sederhana.

Pelajaran dari era pionir adalah: visi komputer bukan sekadar masalah teknis, melainkan masalah kompleks yang membutuhkan pemahaman mendalam tentang persepsi dan kognisi.

Algoritma awal seperti deteksi tepi membantu komputer menemukan batas antar objek. Algoritma Canny Edge Detector (1986) adalah salah satu penemuan awal yang signifikan. Masih digunakan hingga hari ini karena kemampuan untuk mendeteksi tepi yang halus dan kurang suara.

### 1.2.2 Era Metode Berbasis Fitur (1980–2000)

Dekade 1980–2000 ditandai oleh lahirnya algoritma berbasis fitur. Para peneliti menyadari bahwa komputer tidak harus memahami seluruh citra sekaligus, melainkan cukup mengenali fitur-fitur penting yang stabil terhadap rotasi, skala, dan cahaya.

Metode berbasis fitur juga dikenal sebagai metode berbasis fitur beralih menjadi metode utama dalam penglihatan komputer pada tahun 1980-an dan awal 2000-an. Untuk mengekstraksi fitur visual yang dianggap penting dari sebuah gambar, seperti *Corner* (sudut) dan *Edges* (tepi), *Texstur* (pola permukaan), warna histogram, dan deskripsi bentuk.

Algoritma mulai dikembangkan. Pada saat ini, para peneliti fokus pada desain fitur-fitur tersebut secara manual. Metode transformasi fitur skala-invariant SIFT, yang diusulkan oleh David Lowe pada tahun 1999, merupakan penemuan besar. SIFT mendeteksi fitur lokal yang stabil meskipun skala, rotasi, atau pencahayaan berubah. Algoritma serupa seperti SURF dan ORB kemudian mengikuti SIFT.

Algoritma seperti SIFT (Scale-Invariant Feature Transform) yang diperkenalkan David Lowe (1999), SURF (Speeded-Up Robust Features), dan ORB (Oriented FAST and Rotated BRIEF) menjadi tonggak penting. Dengan algoritma ini, komputer mampu mencocokkan dua gambar berbeda (misalnya logo yang dipotret dari sudut berbeda) dengan akurasi tinggi.

Di sisi lain, metode HOG (Histogram of Oriented Gradients) banyak dipakai untuk deteksi pejalan kaki dalam bidang keamanan dan otomotif. Pendekatan ini dianggap revolusioner karena mampu bekerja dalam kondisi dunia nyata yang kompleks.

Kelebihan era ini adalah efisiensi. Algoritma bisa berjalan di komputer dengan sumber daya terbatas. Namun kelemahannya, semua fitur harus dirancang manual oleh manusia. Hal ini membatasi fleksibilitas sistem, terutama untuk masalah yang lebih kompleks

Ciri utama era ini adalah proses klasifikasi dan ekstraksi fitur dilakukan secara terpisah. Untuk melakukan pengenalan objek setelah fitur diekstrak, algoritma pembelajaran mesin seperti SVM (Support Vector Machine) atau k-Nearest Neighbor digunakan. Setiap tahap prosedur (deteksi fitur, reduksi dimensi, dan klasifikasi) dilakukan secara modular dalam pendekatan yang disebut pendekatan pipeline. Metode ini sangat bergantung pada keahlian manusia dalam merancang fitur yang relevan (fitur yang dibuat tangan), sehingga kurang fleksibel terhadap variasi objek atau latar belakang, meskipun berhasil menangani banyak aplikasi sederhana.

### 1.2.3 Munculnya Deep Learning (2012–Sekarang)

Pada tahun 2012, tim yang dipimpin oleh Geoffrey Hinton dari University of Toronto memperkenalkan model pembelajaran mendalam bernama AlexNet dalam kompetisi ImageNet Large Scale Visual Recognition Challenge (ILSVRC), yang mengubah paradigma. Dengan margin akurasi yang sangat besar, AlexNet dapat mengalahkan pendekatan konvensional. Ini menandai permulaan era baru dalam computer vision berbasis deep convolutional neural networks (CNN).

Titik balik terbesar terjadi pada tahun 2012 dalam kompetisi ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Tim yang dipimpin oleh Geoffrey Hinton, dengan arsitektur yang dikenal sebagai AlexNet, berhasil menurunkan error rate pengenalan objek secara drastis dibanding metode konvensional.

AlexNet menggunakan Convolutional Neural Network (CNN) yang sebenarnya sudah diperkenalkan Yann LeCun pada 1990-an (LeNet), tetapi saat itu gagal berkembang karena keterbatasan data dan komputasi. Keberhasilan AlexNet dimungkinkan karena tiga faktor:

1. Dataset besar (ImageNet, jutaan gambar).
2. GPU komputasi (NVIDIA CUDA) yang mempercepat training.
3. Arsitektur deep learning yang lebih dalam dan kompleks.

Setelah itu, arsitektur CNN berkembang pesat: VGGNet, ResNet, Inception, EfficientNet. Kemampuan deteksi objek juga meningkat lewat R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO, RetinaNet, dan seterusnya.

Era ini sering disebut sebagai “revolusi deep learning” karena hampir semua kompetisi pengenalan citra dimenangkan oleh jaringan saraf dalam.

**Mengapa Deep Learning Mengubah Segalanya?**

Sistem dapat menggunakan model deep learning, terutama CNN, untuk belajar langsung dari data mentah tanpa perlu mendesain fitur secara manual. Pada CNN, lapisan konvolusi secara otomatis mengekstraksi pola visual dari tingkat rendah (seperti garis) ke tingkat tinggi (seperti bentuk kompleks atau wajah manusia). Selain itu, kemajuan dalam GPU dan cloud computing mendukung kemampuan untuk melakukan pelatihan pada data dalam jumlah besar.

Berbagai arsitektur CNN telah dikembangkan sejak AlexNet. Beberapa di antaranya adalah VGGNet (2014), yang menggunakan arsitektur yang lebih dalam dengan filter 3x3. GoogLeNet/Inception (2014–2016): menerapkan konsep pemrosesan skala multi dalam satu jaringan. ResNet (2015): menawarkan koneksi residual untuk membuat pelatihan jaringan dalam (hingga 152 lapisan) lebih stabil. DenseNet, EfficientNet, dan lainnya.

Berbagai tugas vision kontemporer, seperti deteksi objek (YOLO, SSD, Faster R-CNN), segmentasi semantik (U-Net, Mask R-CNN), pengenalan wajah (FaceNet, ArcFace), dan rekonstruksi 3D dan estimasi pose, didasarkan pada CNN.

### 1.2.4 Transformasi Menuju Vision Transformer (2020–Kini)

Terlepas dari dominasi CNN selama hampir sepuluh tahun, metode baru mulai muncul yang mengubah arsitektur transformasional bidang pemrosesan bahasa alami. Untuk menggantikan konvolusi dengan mekanisme self-attention, tim penelitian Google memperkenalkan Vision Transformer (ViT) pada tahun 2020. Transformers memungkinkan model untuk menangkap hubungan global antar bagian gambar dengan efisiensi yang lebih tinggi dalam skala besar. ViT dan turunannya, seperti Swin Transformer dan DeiT, sangat disukai untuk tugas-tugas visi komputer yang kompleks dan multimodal karena keunggulannya.

Seperti model CLIP OpenAI atau Flamingo DeepMind, menggabungkan visi dan bahasa memungkinkan pengembangan sistem yang tidak hanya "melihat" tetapi juga "memahami" dan "menjelaskan" apa yang dilihat.

Meskipun CNN mendominasi hampir satu dekade, tahun 2020 menjadi titik awal lahirnya paradigma baru: Vision Transformer (ViT). Peneliti Google memperkenalkan ide untuk menggunakan transformer architecture yang sebelumnya sukses besar di pemrosesan bahasa alami ke dalam domain visual. ViT memandang gambar sebagai kumpulan "patch" kecil, mirip cara transformer membaca urutan kata. Dengan mekanisme self-attention, ViT mampu memahami hubungan antarbagian citra secara lebih global. Hasilnya, ViT bahkan mampu mengalahkan CNN dalam banyak benchmark, terutama dengan data skala besar. Selain ViT, lahir pula varian seperti Swin Transformer dan DeiT (Data-Efficient Image Transformer) yang lebih efisien. Era ini menandai pergeseran fokus riset dari sekadar pengenalan objek menuju pemahaman visual yang lebih kaya dan multimodal.

### 1.2.5 Aplikasi Lintas Bidang dan Evolusi Algoritma Khusus

Perkembangan terkini adalah lahirnya foundation models, yaitu model berskala sangat besar yang dilatih dengan data multimodal. Contohnya:

- CLIP (Contrastive Language-Image Pretraining) dari OpenAI, yang menghubungkan gambar dengan deskripsi teks.
- Segment Anything Model (SAM) dari Meta, yang mampu melakukan segmentasi objek baru tanpa perlu dilatih ulang.
- DINO, BEiT, Flamingo → model-model lain yang memperluas cakupan computer vision ke arah general AI.

Foundation models ini bukan hanya mampu mengenali objek, tetapi juga memahami konteks dan bahkan melakukan tugas yang sebelumnya membutuhkan pelatihan khusus.

Selain pengenalan umum, banyak bidang khusus dalam computer vision yang berkembang dengan algoritma-algoritma khas:

• **Face Recognition**: Dari metode Eigenfaces (1990-an) hingga deep metric learning (FaceNet, ArcFace). • Medical Imaging: Dengan model CNN dan transformer yang khusus dilatih untuk MRI, CT, dan X-ray.

• **Object Tracking dan Video Analytics**: Algoritma seperti SORT, Deep SORT, dan ByteTrack digunakan untuk pelacakan multi-objek real-time.

**Vision untuk Robotika**: Perkembangan visual SLAM (Simultaneous Localization and Mapping) dan visual servoing untuk robot.

### 1.2.6 Tren Masa Kini: Foundation Models dan Zero-shot Learning

Tren terbaru menunjukkan pergeseran menuju foundation models, yaitu model berskala besar yang dilatih pada berbagai tugas dan dataset untuk membangun "pemahaman umum" tentang dunia visual. Contoh yang signifikan adalah:

• CLIP (Contrastive Language-Image Pretraining): Mapping text and images into the same representational space.

• DINO, SAM (Segment Anything Model): Menunjuk pada model penglihatan yang lebih universal dan fleksibel.

Modern vision models with zero-shot and few-shot learning do not need to be fully retrained for new tasks; rather, with just minimal guidance, they can directly perform classification or segmentation.

Tabel 1.1 Perjalanan sejarah algortima vision

| Era / Tahun | Tokoh / Model | Kontribusi Utama | Kelebihan | Keterbatasan |
|---|---|---|---|---|
| 1960–1980 (Pionir) | Marvin Minsky, MIT | Summer Vision Project, edge detection awal | Membuka kesadaran kompleksitas | Gagal implementasi nyata |
| 1980–2000 (Fitur) | David Lowe, Dalal | SIFT, SURF, HOG | Efisien, hand-crafted | Kurang fleksibel |
| 2012 (Deep Learning) | Hinton, Krizhevsky | AlexNet, CNN, ResNet, YOLO | Akurasi tinggi, end-to-end | Butuh data besar & GPU |
| 2020 (Transformer) | Dosovitskiy, Google | Vision Transformer (ViT), Swin | Global attention, scalable | Training mahal |
| 2022+ (Foundation) | OpenAI, Meta | CLIP, SAM, multimodal models | Zero-shot, generalizable | Data & komputasi masif |

Tabel 1.2.6 menyajikan garis besar perkembangan algoritma computer vision dari masa pionir hingga era foundation models. Masing-masing baris mewakili periode tertentu, tokoh kunci atau model utama, kontribusi yang dibawa, serta kelebihan dan keterbatasannya.

**Era Pionir (1960–1980)**

- Tokoh/Model: Marvin Minsky dan tim MIT melalui Summer Vision Project merupakan representasi era ini. Fokusnya masih pada pendekatan sederhana seperti edge detection.
- Kontribusi: Membuka kesadaran bahwa visi komputer jauh lebih rumit daripada sekadar memproses citra. Walaupun sederhana, metode deteksi tepi meletakkan dasar penting bagi representasi visual.
- Kelebihan: Memberi pemahaman awal tentang kompleksitas masalah vision.
- Keterbatasan: Tidak mampu menyelesaikan masalah dunia nyata; sebagian besar eksperimen gagal jika diterapkan pada citra kompleks.

**Era Berbasis Fitur (1980–2000)**

- Tokoh/Model: David Lowe dengan SIFT, Dalal & Triggs dengan HOG, serta varian seperti SURF dan ORB.
- Kontribusi: Menghadirkan fitur lokal yang robust terhadap rotasi, skala, dan pencahayaan. Cocok untuk aplikasi seperti pengenalan objek dan image matching.
- Kelebihan: Efisien, bisa berjalan di komputer dengan kapasitas terbatas.

- Keterbatasan: Semua fitur harus dirancang manual (hand-crafted), sehingga kurang fleksibel untuk data kompleks.

**Era Deep Learning (2012–Sekarang)**

- Tokoh/Model: Geoffrey Hinton, Alex Krizhevsky, dan arsitektur AlexNet, diikuti ResNet, VGGNet, Inception, YOLO, dan Mask R-CNN.
- Kontribusi: Menunjukkan bahwa CNN mampu mengungguli metode tradisional secara signifikan. Era ini menandai lahirnya end-to-end learning—sistem yang bisa otomatis belajar fitur tanpa rekayasa manual.
- Kelebihan: Akurasi sangat tinggi, dapat menyelesaikan berbagai tugas vision (klasifikasi, deteksi, segmentasi).
- Keterbatasan: Membutuhkan dataset raksasa dan GPU dengan daya komputasi besar.

**Era Vision Transformer (2020–Kini)**

- Tokoh/Model: Alexey Dosovitskiy dan tim Google dengan Vision Transformer (ViT), kemudian Swin Transformer dan DeiT.
- Kontribusi: Memperkenalkan mekanisme self-attention dalam domain visual, sehingga model bisa memahami hubungan antarbagian gambar secara global.
- Kelebihan: Mampu menggeneralisasi lebih baik pada dataset besar, fleksibel, dan menjadi dasar multimodal AI.
- Keterbatasan: Training sangat mahal, butuh data dan komputasi lebih besar daripada CNN.

**Era Foundation Models (2022+)**

- Tokoh/Model: OpenAI dengan CLIP, Meta dengan SAM, serta model multimodal seperti Flamingo dan DINO.
- Kontribusi: Munculnya general-purpose vision models yang bisa melakukan berbagai tugas tanpa pelatihan khusus. CLIP, misalnya, bisa menghubungkan teks dengan citra; SAM mampu melakukan segmentasi "apapun" hanya dengan prompt sederhana.
- Kelebihan: Mampu melakukan zero-shot learning, adaptif, dan dapat dipakai lintas domain.
- Keterbatasan: Sangat bergantung pada dataset raksasa dan daya komputasi yang mahal; masih menjadi tantangan bagi peneliti di negara berkembang.

### 1.2.7 Penutup: Belajar dari Sejarah

Sejarah perkembangan algoritma vision menunjukkan pola umum:

1. Dimulai dari pendekatan manual berbasis fitur dan aturan;

2. Beralih ke metode pembelajaran mesin terstruktur;

3. Bergeser ke pendekatan pembelajaran representasi otomatis melalui deep learning;

Sekarang memasuki zaman model multimodal yang dapat "melihat dan memahami" secara kontekstual.

Siapa pun yang ingin mengembangkan atau menerapkan algoritma vision secara efektif harus memahami jalur sejarah ini. Ia tidak hanya memberikan konteks teknis, tetapi juga membantu

menghindari pengulangan kesalahan masa lalu, serta membuka sudut pandang terhadap kemungkinan inovasi di masa depan. Sejarah perkembangan computer vision sering digambarkan sebagai serangkaian lompatan teknologi. Namun, penting dipahami bahwa setiap era memiliki kekuatan dan keterbatasan. Era pionir (1960–1980) misalnya, meskipun gagal memenuhi ambisi "mesin yang bisa mengenali dunia dalam tiga bulan", justru membekali komunitas ilmiah dengan kesadaran bahwa penglihatan komputer adalah masalah yang sangat kompleks. Tanpa kegagalan ini, mungkin tidak akan lahir penelitian serius di dekade berikutnya.

Metode berbasis fitur (1980–2000) mengajarkan pentingnya abstraksi matematis. Algoritma seperti SIFT atau HOG membuktikan bahwa informasi lokal dapat menjadi kunci untuk mengenali objek, bahkan ketika skala, rotasi, atau cahaya berubah. Pendekatan ini memang manual, tetapi sangat berharga untuk aplikasi sederhana yang membutuhkan interpretabilitas.

Kemunculan deep learning pada 2012 melalui kemenangan AlexNet di kompetisi ImageNet adalah bukti bahwa data dalam jumlah besar ditambah daya komputasi GPU mampu mengubah lanskap ilmu. Kejadian ini sering disebut sebagai "ImageNet moment" dan menjadi titik balik sejarah.

Kini, era Vision Transformer (2020–sekarang) menunjukkan bahwa computer vision terus bergerak ke arah model yang lebih umum, fleksibel, dan multimodal, bukan hanya pengenalan objek tetapi juga pemahaman konteks visual secara semantik. Dari sejarah ini, mahasiswa dapat belajar bahwa ilmu berkembang bukan dengan jalan lurus, melainkan melalui kombinasi ambisi, kegagalan, dan inovasi yang saling berlapis.

### 1.3 Peran Strategis Computer Vision di Era Industri 4.0

Rapid digital technology development in the last two decades has brought mankind into a new phase of industrial revolution called Industry 4.0. Fase ini ditandai oleh integrasi antara dunia fisik dan dunia digital melalui sistem siber-fisik, Internet of Things (IoT), komputasi awan, kecerdasan buatan (AI), serta big data analytics. Dalam kerangka besar ini, computer vision muncul sebagai salah satu komponen strategis yang sangat penting dalam mendorong otomatisasi, efisiensi, dan kecerdasan sistem di berbagai sektor industri.

#### 1.3.1 Industri 4.0 dan Transformasi Digital

Konsep Industri 4.0 mengacu pada revolusi dalam dunia manufaktur dan bisnis, di mana jaringan mesin cerdas yang saling terhubung dan mampu mengambil keputusan secara otonom menggantikan proses produksi, distribusi, dan layanan yang tidak lagi bergantung pada kerja manual atau sistem konvensional. Ini dicapai dengan menggabungkan dunia fisik (sensor, aktuator, robotik) dan digital (AI, data analytics, cloud computing).

Komputerisasi persepsi, atau kemampuan sistem untuk mengenali dan memahami lingkungan sekitar melalui data sensorik—baik suara, suhu, tekanan, maupun citra visual—adalah salah satu pilar utama transformasi ini. Computer vision's role becomes very strategic here: this technology lets machines "see" and "understand" environments with high accuracy, real-time, and without human intervention.

Revolusi Industri 4.0 ditandai oleh integrasi teknologi siber-fisik, Internet of Things (IoT), big data, dan kecerdasan buatan. Computer vision berperan sebagai "indera penglihatan" bagi sistem cerdas, sehingga mesin dapat mengobservasi dunia nyata dan meresponsnya secara otomatis.

Contohnya, dalam smart factory, kamera dan algoritma vision digunakan untuk mengawasi jalur produksi. Sistem dapat langsung mendeteksi produk cacat, menghentikan mesin, atau mengirimkan peringatan. Hal ini meningkatkan efisiensi, presisi, dan kualitas tanpa perlu campur tangan manusia terus-menerus.

### 1.3.2 Computer Vision sebagai Teknologi Kunci

Peran strategis computer vision tidak hanya teknis, tetapi juga ekonomi dan sosial. Teknologi ini mendukung berbagai sektor:

- Manufaktur: kontrol kualitas otomatis, robot kolaboratif dengan sistem penglihatan.
- Kesehatan: diagnosa berbasis citra medis, seperti deteksi tumor dari MRI.
- Pertanian: drone vision untuk pemantauan lahan, deteksi penyakit daun padi atau jagung.
- Transportasi: sistem traffic monitoring, kendaraan otonom, deteksi pelanggaran lalu lintas.
- Keamanan: pengawasan publik dengan kamera cerdas.

Computer vision menjadi salah satu pilar utama otomatisasi, sejajar dengan IoT dan machine learning.

Computer vision bukan hanya mendukung tetapi juga mengakselerasi pelaksanaan Industri 4.0 dalam beragam dimensi. Beberapa peran strategis utamanya meliputi:

#### a. Otomatisasi Visual dalam Manufaktur

Computer vision digunakan dalam industri manufaktur kontemporer untuk melakukan inspeksi kualitas produk secara otomatis. Kamera resolusi tinggi yang terintegrasi dengan algoritma deteksi objek dan segmentasi dapat menemukan cacat produksi, penyimpangan warna, ukuran, atau bentuk dengan kecepatan dan konsistensi yang tidak dapat ditemukan oleh pengawasan manusia.

YOLO, singkatan dari You Only Look Once, digunakan dalam sistem produksi elektronik untuk memeriksa setiap produk yang bergerak di jalur produksi secara real-time untuk memastikan bahwa semua komponen terpasang dengan benar. Hal ini meningkatkan efisiensi dan mengurangi kesalahan dan biaya perbaikan.

#### b. Robotika Vision-Guided

Computer vision menjadi indra utama bagi robot dalam robotik industri. Sistem vision-guided robotics memungkinkan robot untuk menemukan posisi objek, menghitung kedalaman, dan merespon secara dinamis terhadap perubahan lingkungan. Robot dapat mengambil dan meletakkan benda dengan tepat, melakukan perakitan, atau bahkan mencari di ruang kerja yang kompleks berkat kombinasi vision dan AI.

**c. Pemeliharaan Prediktif dan Monitoring Visual**

Pemeliharaan prediktif memerlukan visi komputer. Sensor visual dan kamera termal mendeteksi anomali mesin seperti kebocoran, retakan, atau perubahan suhu permukaan. Analisis pola visual komponen mesin memungkinkan sistem untuk memprediksi kerusakan sebelum terjadi dan menjadwalkan perawatan secara proaktif untuk menghindari downtime.

**d. Keselamatan dan Keamanan Kerja**

Computer vision digunakan untuk mendeteksi pelanggaran prosedur keselamatan di tempat kerja yang berisiko tinggi, seperti tambang atau pabrik kimia. Misalnya, sistem dapat mengetahui apakah karyawan mengenakan helm, rompi, atau perlindungan mata dan mengeluarkan peringatan jika terjadi kelalaian. Sistem juga dapat mengetahui jika ada orang di area berbahaya dan secara otomatis menghentikan mesin untuk mencegah kecelakaan.

**e. Visual Analytics untuk Optimalisasi Proses**

Computer vision digunakan untuk mendeteksi pelanggaran prosedur keselamatan di tempat kerja yang berisiko tinggi, seperti tambang atau pabrik kimia. Misalnya, sistem dapat mengetahui apakah karyawan mengenakan helm, rompi, atau perlindungan mata dan mengeluarkan peringatan jika terjadi kelalaian. Sistem juga dapat mengetahui jika ada orang di area berbahaya dan secara otomatis menghentikan mesin untuk mencegah kecelakaan.

### 1.3.3 Penerapan Lintas Sektor dalam Ekosistem Industri 4.0

Peran strategis computer vision tidak terbatas pada produksi. Berbagai bagian dari industri 4.0 telah menggunakan teknologi ini:

**a. Sektor Pertanian (Smart Farming)**

Computer vision digunakan di bidang pertanian kontemporer untuk menemukan penyakit tanaman, menghitung populasi tanaman, dan mengukur tingkat kematangan buah. Traktor pintar atau drone memiliki kamera yang dapat memindai area pertanian secara otomatis untuk membuat peta kesehatan tanaman. Petani menggunakan teknologi ini untuk membuat keputusan tentang hal-hal seperti penanganan hama yang efektif, irigasi terarah, dan pemupukan yang tepat

Di Indonesia, computer vision semakin banyak digunakan untuk **pertanian presisi**. Misalnya, penggunaan kamera yang dipasang pada **drone** atau **ESP32-CAM** untuk mendeteksi penyakit daun secara otomatis. Sistem ini tidak hanya menghemat waktu, tetapi juga meningkatkan produktivitas petani karena penyakit bisa dideteksi sejak dini.

Studi lain menunjukkan bahwa sistem vision mampu menghitung jumlah tanaman, memantau pertumbuhan, hingga memperkirakan hasil panen. Dengan demikian, computer vision berkontribusi pada ketahanan pangan sekaligus modernisasi sektor pertanian.

.**b. Kesehatan (Smart Healthcare)**

Computer vision menjadi alat bantu diagnosis yang sangat efektif dalam sistem kesehatan berbasis kecerdasan buatan. Contohnya adalah mendeteksi kanker kulit dari gambar dermatoskopi, mengidentifikasi diabetes melalui analisis retina, dan menemukan pola pada scan CT-Scan dan MRI untuk mendeteksi penyakit otak sejak dini. Sistem pemantauan berbasis

kamera juga digunakan dalam perawatan jarak jauh untuk melacak pasien yang lebih tua dan penderita penyakit kronis.

Selain itu computer vision juga memainkan peran penting dalam **radiologi digital**. Algoritma CNN mampu mendeteksi kelainan pada X-ray, CT-Scan, atau MRI dengan akurasi mendekati bahkan melebihi dokter spesialis.

Misalnya, pada kasus kanker paru, sistem vision dapat membantu menyaring ribuan citra medis dengan cepat, sehingga dokter hanya fokus pada kasus yang paling mencurigakan. Hal ini mempercepat diagnosis dan mengurangi risiko human error.

Selain itu, aplikasi kesehatan berbasis smartphone kini juga memanfaatkan vision. Contoh: aplikasi yang bisa mengidentifikasi kondisi kulit hanya dengan memotret bagian tubuh yang bermasalah.

**c. Transportasi dan Logistik**

Computer vision digunakan dalam industri logistik untuk pengenalan plat nomor, deteksi kemacetan, dan penghitungan dan pelacakan barang secara otomatis di gudang pintar. Kamera visi digunakan sebagai "mata" mobil dalam sistem kendaraan otonom untuk mendeteksi jalur, pejalan kaki, rambu lalu lintas, dan kendaraan lain. Sistem persepsi multimodal menggunakan sensor lidar dan radar untuk memungkinkan navigasi aman dan mandiri.

Dalam transportasi modern, computer vision digunakan untuk **Intelligent Transportation Systems (ITS)**. Kamera jalan yang terhubung dengan sistem vision dapat mendeteksi kemacetan, kecelakaan, dan pelanggaran lalu lintas secara otomatis.

Pada kendaraan otonom, vision menjadi sensor utama untuk mendeteksi rambu lalu lintas, pejalan kaki, atau kendaraan lain. Kombinasi vision dengan sensor lain seperti LiDAR membuat mobil dapat mengambil keputusan dalam hitungan milidetik. Bagi smart city, sistem ini membantu menciptakan lingkungan kota yang lebih aman, teratur, dan efisien.

**d. Ritel dan E-commerce**

Penggunaan computer vision memberikan dampak luas:

1. Ekonomi: meningkatkan produktivitas dan efisiensi, tetapi juga berpotensi mengurangi lapangan kerja manual.
2. Sosial: membantu meningkatkan kualitas hidup, misalnya melalui deteksi dini penyakit. Namun, ada pula risiko privasi akibat pengawasan masif.
3. Pendidikan dan keterampilan: muncul kebutuhan tenaga kerja baru yang terampil di bidang AI, data science, dan vision engineering.

Dengan demikian, computer vision bukan hanya soal teknologi, tetapi juga soal transformasi sosial. Toko-toko tanpa kasir seperti Amazon Go menggunakan sistem computer vision untuk secara otomatis mengidentifikasi barang yang dibeli konsumen. Sebaliknya, teknologi ini juga digunakan untuk menganalisis perilaku pelanggan di toko fisik melalui kamera. Ini melacak jalur pergerakan pelanggan, waktu yang dihabiskan untuk berinteraksi dengan produk, dan ekspresi wajah pelanggan saat berbelanja. Semua ini dilakukan untuk meningkatkan pengalaman pelanggan dan strategi pemasaran.

**e. Lingkungan dan Energi**

Computer vision digunakan dalam pengawasan lingkungan untuk mendeteksi kebakaran hutan, mengawasi pencemaran air, dan menghitung jumlah hewan liar. Di bidang energi, teknologi ini digunakan untuk memeriksa jaringan listrik, turbin angin, dan panel surya dengan drone dan kamera termal yang diproses secara otomatis.

### 1.3.4 Keterkaitan dengan Teknologi Lain

Computer vision jarang berdiri sendiri dalam praktik. Ia bekerja sama dengan teknologi lain dalam ekosistem Industri 4.0. Misalnya, dengan IoT: gambar atau video yang diambil oleh kamera IoT diproses secara lokal (edge computing) atau dikirim ke cloud untuk analisis lanjutan.

Dengan machine learning: model pembelajaran mendalam (CNN, Transformer) digunakan untuk klasifikasi, deteksi, dan segmentasi visual.

Dengan AR/VR: Visi komputer membantu dalam pengenalan gerakan, pelacakan objek, dan interaksi manusia-komputer.

Dengan kerja sama ini, sistem cerdas yang tidak hanya reaktif, tetapi juga prediktif dan adaptif.

### 1.3.5 Tantangan dan Arah Masa Depan

Meskipun penglihatan komputer sangat membantu industri 4.0, masih ada beberapa masalah yang perlu ditangani:

• Kualitas Data Visual: Pencahayaan yang buruk, distorsi kamera, atau kondisi ekstrim dapat mengganggu akurasi deteksi;

• Kebutuhan Komputasi Tinggi: Model deep learning membutuhkan GPU dan sumber daya besar, terutama untuk proses real-time;

• Privasi dan Etika: Penggunaan kamera di ruang publik dan tempat kerja menimbulkan masalah privasi dan pengawasan bergerak.

Masa depan dari computer vision di dalam Industri 4.0 ini akan ditandai dengan penerapan model visi dasar, peningkatan AI tepi, dan peningkatan integrasi dengan pemrosesan dan penalaran bahasa alami. Seiring waktu, peran manusia akan beralih dari operator ke pengawas sistem cerdas yang sepenuhnya otonom.

### 1.3.6 Dampak Sosial dan Ekonomi Computer Vision

Penerapan computer vision membawa dampak luas, tidak hanya dalam dunia industri tetapi juga dalam kehidupan masyarakat. Dari sisi ekonomi, teknologi ini memungkinkan otomatisasi proses produksi dan distribusi, yang pada gilirannya meningkatkan efisiensi dan menurunkan biaya. Namun, otomatisasi juga dapat menimbulkan kekhawatiran terkait berkurangnya lapangan kerja manual. Contoh nyata adalah penggunaan kamera pintar untuk inspeksi kualitas di pabrik, yang dapat menggantikan peran puluhan pekerja manusia. Untuk menidentifikasi kualitas produk pada lini produksi sudah dikembangkan berbasis kamera apakah benda yang di produksi sudah sesuai dengan demensi dan bentuk yang di inginkan hal ini menggantikan pengawasan mata manusia yang kadang mengalami kelelahan dan error kesalahan, hal ini tentu nya akan meningkat kan kapasitas produksi pada pabrik.

Di sisi lain, computer vision juga menciptakan lapangan kerja baru dalam bidang teknologi: mulai dari pengembang algoritma, analis data visual, hingga pakar keamanan digital. Hal ini menunjukkan adanya pergeseran keterampilan yang dibutuhkan di era digital yang awal nya pekerjaan manusia sudah di gantikan dengan algortima vision tentu nya hal ini akan memberikan dampak pada masyarakat tentang penguasaan skill yang harus dimiliki.

Dari sisi sosial, computer vision berkontribusi pada peningkatan kualitas hidup, misalnya melalui sistem deteksi dini penyakit, navigasi kendaraan otonom yang lebih aman, dan sistem keamanan cerdas di ruang publik. Namun, muncul juga tantangan etika seperti privasi, bias algoritmik, dan potensi penyalahgunaan teknologi untuk pengawasan massal. Oleh karena itu, pembahasan computer vision tidak cukup hanya pada sisi teknis, tetapi juga harus mencakup dimensi sosial-ekonomi.

Selain masalah tadi tentu nya masalah database dan pemilikan data dari computer vision sangat menjadi isu yang penting sekali, misal nya penguasaan computer vision dalam pengawasan kejahatan hal ini jika di salah gunakan bisa berbahaya , selain itu misal nya masalah pengenalan wajah jika di manipulasi data yang di dapat kan bisa menjadikan fals error yang berbahasa dalam mendeteksi wajah penjahat dan teroris. Sama hal nya dengan bidang lain computer vision masih mengalami kendala margin error ini masih dalam penelitian lebih lanjut.

### 1.3.7 Peran Vision dalam Kehidupan Sehari-hari Mahasiswa

Bagi mahasiswa, computer vision bukanlah sesuatu yang jauh dan abstrak. Justru, banyak aktivitas sehari-hari yang diam-diam sudah melibatkan teknologi ini. Misalnya:

- Face unlock di smartphone adalah contoh sistem pengenalan wajah yang berbasis CNN.
- Fitur filter Instagram dan TikTok memanfaatkan face landmark detection untuk melacak posisi mata, hidung, dan bibir.
- Google Lens menggunakan object detection dan image retrieval untuk mengenali teks, barang, hingga tumbuhan hanya dengan foto.

Aplikasi e-learning memanfaatkan computer vision untuk proctoring online exam, yaitu mengawasi ujian daring dengan deteksi wajah dan gerakan.

Contoh-contoh sederhana ini penting untuk membuat mahasiswa merasa dekat dengan materi yang dipelajari. Dengan menyadari bahwa mereka sudah berinteraksi dengan computer vision hampir setiap hari, pembelajaran tidak hanya menjadi teoritis, tetapi juga praktis dan relevan dengan realitas kehidupan mereka.

### 1.3.8 Revolusi Industri 5.0: Human-Centered AI

Saat ini dunia mulai bergerak menuju Revolusi Industri 5.0, yang menekankan kolaborasi antara manusia dan mesin. Jika Industri 4.0 berfokus pada otomatisasi, maka Industri 5.0 menekankan human-centered AI.

Dalam konteks ini, computer vision tidak hanya digunakan untuk menggantikan manusia, tetapi juga untuk memberdayakan manusia. Misalnya, sistem vision yang membantu dokter menganalisis citra medis, bukan menggantikannya. Atau aplikasi vision yang membantu petani kecil memantau lahan mereka, bukan hanya perusahaan besar. Dengan pendekatan ini, computer vision dipandang bukan sebagai ancaman, melainkan mitra manusia dalam menciptakan masa depan yang lebih adil dan inklusif.

# BAB II
# Dasar Matematika dan Statistik untuk Computer Vision

## Pendahuluan

Setiap bidang ilmu memiliki fondasi yang menopang perkembangannya. Bagi computer vision, fondasi itu terletak pada matematika dan statistika. Tanpa pemahaman yang cukup mengenai aljabar linear, probabilitas, transformasi sinyal, dan optimisasi, sulit bagi kita untuk benar-benar memahami mengapa algoritma vision bekerja sebagaimana mestinya.

Mahasiswa sering kali langsung mempelajari pemrograman dengan framework seperti OpenCV, TensorFlow, atau PyTorch. Mereka dapat menjalankan kode, melatih model, bahkan menghasilkan sistem deteksi objek. Namun, banyak di antara mereka yang masih bertanya-tanya: "Mengapa filter convolution bisa menonjolkan tepi gambar?", atau "Mengapa kita butuh ratusan iterasi gradient descent hanya untuk melatih satu model?". Pertanyaan-pertanyaan semacam itu hanya bisa dijawab dengan memahami fondasi matematis.

Bab ini bertujuan untuk membangun jembatan antara teori dan praktik. Kita akan melihat bagaimana citra digital direpresentasikan sebagai matriks, bagaimana statistik membantu mengenali pola, bagaimana transformasi Fourier digunakan dalam filtering, bagaimana algoritma optimisasi bekerja saat melatih jaringan saraf, serta bagaimana tensor digunakan untuk merepresentasikan data visual di era deep learning.

## 2.1 Aljabar Linear dalam Computer Vision

### 2.1.1 Citra sebagai Matriks

Citra digital adalah representasi matematis dari dunia visual. Dalam citra grayscale, setiap piksel direpresentasikan oleh sebuah angka yang menunjukkan intensitas cahaya, biasanya berkisar antara 0 hingga 255. Angka 0 berarti hitam total, sedangkan 255 berarti putih total. Jika disusun secara keseluruhan, piksel-piksel ini membentuk matriks intensitas.

Sebagai contoh, sebuah citra berukuran 640 × 480 piksel dapat dilihat sebagai matriks dengan 480 baris dan 640 kolom. Setiap elemen matriks memuat angka intensitas. Jika citra berwarna, maka representasi yang digunakan adalah tiga matriks sekaligus: merah (R), hijau (G), dan biru (B). Kombinasi ketiganya membentuk warna yang kita lihat.

Konsep citra sebagai matriks inilah yang menjadi pintu masuk bagi aljabar linear ke dalam dunia vision. Segala operasi, mulai dari rotasi, translasi, hingga filtering, dapat dipahami sebagai operasi matriks.

### 2.1.2 Operasi Dasar Matriks

Aljabar linear menyediakan alat untuk memanipulasi citra. Operasi yang tampak abstrak dalam teori, ternyata sangat nyata dalam aplikasi vision.

1. Penjumlahan Matriks: Dua citra dengan ukuran sama dapat dijumlahkan elemen demi elemen. Misalnya, untuk membuat efek transparansi atau penggabungan dua gambar.
2. Perkalian Skalar: Mengalikan matriks citra dengan angka tertentu dapat meningkatkan atau menurunkan tingkat kecerahan.
3. Transpose: Menukar baris dan kolom dari sebuah matriks, yang dalam citra berarti memutar gambar terhadap diagonal.
4. Inverse: Dalam konteks transformasi geometris, inverse matriks memungkinkan kita mengembalikan citra yang sudah dirotasi atau digeser ke posisi semula.

Dengan cara ini, konsep dasar aljabar linear yang dipelajari di kelas matematika menjadi relevan bagi pengolahan citra.

### 2.1.3 Transformasi Linear: Rotasi, Skala, Translasi

- Transformasi geometris adalah salah satu aplikasi paling intuitif dari aljabar linear.
- Rotasi: Rotasi citra dapat dilakukan dengan mengalikan koordinat piksel dengan matriks rotasi. Misalnya, rotasi 90° ke kanan.
- Skala: Perbesaran atau pengecilan citra dilakukan dengan matriks skala.

Translasi: Pergeseran posisi citra juga dapat dipahami sebagai operasi penambahan vektor pada koordinat.

Sebagai contoh, jika kita ingin memutar titik $(x, y)$ sebesar sudut $\theta$, maka koordinat baru $(x', y')$ diperoleh dari perkalian dengan matriks rotasi:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Inilah alasan mengapa transformasi visual dapat direpresentasikan secara kompak menggunakan aljabar linear.

### 2.1.4 Eigenvalue dan Eigenvector: PCA

Konsep eigenvalue dan eigenvector sering dianggap abstrak oleh mahasiswa. Namun, dalam computer vision, konsep ini sangat aplikatif. Salah satu penggunaannya adalah Principal Component Analysis (PCA).

PCA digunakan untuk mereduksi dimensi data visual. Misalnya, sebuah citra wajah berukuran 100 × 100 piksel memiliki 10.000 dimensi. Dengan PCA, kita dapat merepresentasikan wajah

tersebut dalam ruang berdimensi lebih rendah, misalnya 50 dimensi, tanpa kehilangan ciri penting.

Metode ini melahirkan konsep Eigenfaces: representasi wajah dalam bentuk kombinasi eigenvector. Secara praktis, sistem dapat mengenali wajah bukan dengan melihat piksel per piksel, tetapi dengan memproyeksikan wajah ke ruang eigen yang lebih ringkas.

Bagi mahasiswa, ini menunjukkan bahwa teori eigenvalue yang dipelajari di kelas bukan sekadar rumus, melainkan fondasi dari teknologi pengenalan wajah yang mereka gunakan sehari-hari.

### 2.1.5 Citra sebagai Ruang Vektor

Selain dipandang sebagai matriks, citra juga bisa dilihat sebagai vektor berdimensi tinggi. Misalnya, citra berukuran 28 × 28 piksel (seperti pada dataset MNIST) dapat diproyeksikan ke dalam vektor 784 dimensi. Setiap piksel adalah satu elemen vektor. Pendekatan ini membuka pintu bagi penerapan teknik aljabar linear lain, seperti PCA, SVD (Singular Value Decomposition), atau metode embedding. Dengan pandangan ini, computer vision tidak sekadar bekerja dengan gambar, tetapi dengan ruang vektor yang sangat besar. Inilah yang menjelaskan mengapa reduksi dimensi penting: tidak mungkin manusia atau komputer memproses data berdimensi puluhan ribu tanpa kompresi atau representasi baru.

### 2.1.6 Operasi Matriks dalam Filtering

Salah satu aplikasi paling sederhana dari operasi matriks adalah filtering. Filter citra dapat direpresentasikan sebagai matriks kernel kecil yang dikalikan dengan bagian tertentu dari citra. Contohnya, filter 3 × 3 untuk deteksi tepi Sobel.

Secara matematis, proses ini adalah bentuk khusus dari perkalian matriks. Operasi convolution yang populer dalam CNN pun pada dasarnya adalah perkalian matriks berulang. Dengan penjelasan ini, mahasiswa dapat memahami bahwa filtering hanyalah variasi dari operasi matriks dasar.

2.1.Tabel Perbandingan Operasi Matriks

| Operasi Matriks | Makna Matematis | Aplikasi Vision |
|---|---|---|
| Penjumlahan | Menjumlahkan dua matriks elemen demi elemen | Penggabungan dua gambar (blending) |
| Perkalian Skalar | Mengalikan setiap elemen dengan konstanta | Mengubah tingkat kecerahan gambar |
| Transpose | Menukar baris dan kolom | Rotasi sederhana / transformasi koordinat |
| Perkalian Matriks | Transformasi linear | Rotasi, translasi, skala citra |
| Determinan | Luas/volume dalam ruang vektor | Transformasi geometris, kompresi |
| Eigenvalue/Vektor | Arah dominan dari transformasi | PCA, Eigenfaces, kompresi citra |

Pada table 2.1 terlihat Operasi dasar matriks dalam aljabar linear memiliki keterkaitan langsung dengan berbagai aplikasi computer vision. Misalnya, penjumlahan matriks yang secara matematis berarti menjumlahkan elemen-elemen pada posisi yang sama, dalam konteks citra dapat digunakan untuk menggabungkan dua gambar. Teknik ini lazim dipakai dalam image blending, ketika dua citra dikombinasikan untuk menghasilkan efek transparansi atau superimposisi. Demikian pula, perkalian skalar yang berarti mengalikan setiap elemen matriks dengan bilangan konstan, dalam citra berfungsi untuk menyesuaikan tingkat kecerahan. Jika konstanta lebih besar dari satu, seluruh gambar tampak lebih terang, sedangkan jika antara nol hingga satu, gambar menjadi lebih gelap.

Operasi transpose, yang awalnya hanya dimaknai sebagai pertukaran baris dan kolom, juga memiliki makna visual. Transpose dapat dianggap sebagai rotasi terhadap diagonal utama, sehingga bermanfaat dalam transformasi koordinat dan manipulasi citra. Sementara itu, perkalian matriks memegang peranan yang lebih luas, karena memungkinkan terjadinya transformasi linear seperti rotasi, translasi, dan skala. Dalam pengolahan citra, perkalian matriks digunakan untuk image registration atau penyamaan dua citra yang diambil dari sudut berbeda, dan dalam animasi komputer, hampir semua transformasi objek dilakukan melalui operasi ini.

Selain itu, konsep determinan meskipun sering dianggap abstrak, memiliki interpretasi penting dalam dunia visual. Determinan dari suatu matriks transformasi menunjukkan bagaimana luas atau volume berubah setelah transformasi diterapkan. Jika determinan sama dengan satu, berarti tidak ada perubahan ukuran; jika lebih kecil dari satu, objek mengalami penyusutan; dan jika lebih besar dari satu, objek membesar. Pemahaman ini krusial dalam analisis transformasi geometris maupun kompresi citra.

Terakhir, konsep eigenvalue dan eigenvector menunjukkan arah dominan dari suatu transformasi. Dalam computer vision, keduanya menjadi dasar bagi teknik reduksi dimensi

seperti Principal Component Analysis (PCA). Aplikasi klasik dari pendekatan ini adalah metode Eigenfaces dalam pengenalan wajah. Dengan memanfaatkan eigenvector dari computer vision berupa kumpulan citra wajah, sistem dapat merepresentasikan wajah dalam bentuk ciri-ciri dominan, sehingga proses pengenalan lebih efisien dan tidak bergantung pada semua piksel. Dengan kata lain, teori aljabar linear yang tampak abstrak ternyata memiliki peran fundamental dalam menggerakkan berbagai algoritma vision modern.

### 2.1.7 Mini Studi Kasus: Eigenfaces

Metode Eigenfaces adalah contoh klasik pengenalan wajah berbasis PCA. Ide dasarnya adalah mengumpulkan banyak citra wajah, kemudian menghitung eigenvector dari kovariansi citra tersebut. Eigenvector yang dominan disebut eigenfaces, karena jika divisualisasikan akan terlihat menyerupai wajah samar-samar.

Dalam aplikasi, wajah baru diproyeksikan ke ruang eigenfaces untuk dibandingkan dengan wajah yang sudah ada. Walaupun sederhana dibanding deep learning modern, metode ini sangat penting karena memperlihatkan bagaimana aljabar linear murni dapat dipakai dalam pengenalan pola visual.

## 2.2 Probabilitas dan Statistik untuk Pengenalan Pola

### 2.2.1 Pendahuluan

Probabilitas dan statistika adalah fondasi lain yang sangat penting dalam computer vision. Hampir semua algoritma modern, baik yang berbasis machine learning maupun deep learning, berakar pada teori peluang. Hal ini karena citra digital bersifat penuh ketidakpastian: pencahayaan bisa berubah, objek bisa tertutup sebagian, atau sensor kamera bisa menghasilkan noise. Untuk menghadapi ketidakpastian inilah probabilitas digunakan.

Statistika membantu kita menganalisis pola dari data visual, sedangkan probabilitas memberi kerangka untuk membuat prediksi. Dalam pengenalan wajah, misalnya, sistem tidak pernah bisa 100% yakin bahwa gambar tertentu benar-benar milik seseorang. Yang bisa dilakukan adalah menghitung peluang bahwa citra tersebut cocok dengan data wajah yang tersimpan. Dengan demikian, hasil pengenalan selalu berupa probabilitas, bukan kepastian mutlak.

2.2.2 Konsep Dasar Probabilitas

Probabilitas dapat dipahami sebagai ukuran seberapa besar kemungkinan sebuah peristiwa terjadi. Dalam computer vision, peristiwa ini bisa berupa “piksel bernilai 255 adalah bagian dari objek putih” atau “gambar ini adalah kucing”.

Secara matematis, probabilitas didefinisikan sebagai:

$$P(A) = \frac{\text{Jumlah kejadian A}}{\text{Jumlah total kejadian}}$$

Contohnya, jika dari 1000 gambar terdapat 200 gambar kucing, maka probabilitas munculnya kucing adalah

$$P(\text{kucing}) = 200/1000 = 0.2.$$

Konsep dasar ini kemudian diperluas ke distribusi probabilitas, yang menunjukkan sebaran kemungkinan dari suatu variabel.

### 2.2.3 Distribusi Probabilitas dalam Citra

Dalam computer vision, distribusi probabilitas sering digunakan untuk memodelkan noise, intensitas piksel, atau kemunculan fitur.

Tabel 2.2 Distribusi probabilitas Noise

| Distribusi | Karakteristik | Aplikasi dalam Vision |
|---|---|---|
| Normal (Gaussian) | Simetris, berbentuk lonceng | Model noise kamera, filter Gaussian |
| Uniform | Semua nilai sama kemungkinannya | Model noise seragam, sampling piksel |
| Binomial | Dua kemungkinan (sukses/gagal) | Deteksi objek biner (ada/tidak ada) |
| Multinomial | Banyak kategori | Klasifikasi multi-kelas (misal, mengenali 10 digit MNIST) |

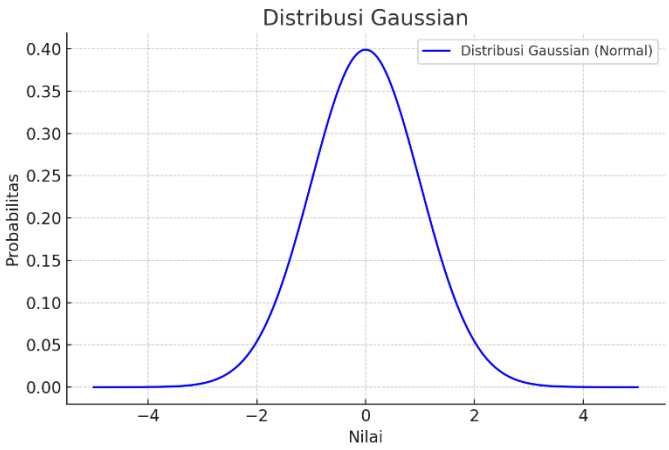
Pada table 2.2 Sebagai contoh, noise pada sensor kamera sering dimodelkan dengan distribusi normal. Hal ini menjelaskan mengapa filter Gaussian efektif untuk mengurangi noise: karena ia sesuai dengan sifat distribusi probabilitas yang dimodelkan. Dalam computer vision, berbagai jenis distribusi probabilitas digunakan untuk memodelkan karakteristik data visual. Distribusi normal atau Gaussian merupakan yang paling umum dijumpai. Distribusi ini berbentuk lonceng simetris dan banyak digunakan untuk merepresentasikan noise kamera. Misalnya, ketika sensor kamera bekerja dalam kondisi pencahayaan rendah, piksel yang terekam sering kali berfluktuasi secara acak. Fluktuasi ini biasanya mengikuti pola Gaussian,

sehingga filter Gaussian efektif digunakan untuk mereduksi noise tanpa menghilangkan detail penting.

Distribusi lain yang sederhana adalah uniform distribution, di mana setiap nilai memiliki peluang yang sama. Dalam konteks citra, distribusi ini sering digunakan untuk memodelkan noise seragam atau proses sampling piksel. Jika sebuah citra terkontaminasi noise uniform, maka intensitas piksel bisa berubah secara acak dengan kemungkinan yang sama pada seluruh rentang nilai.

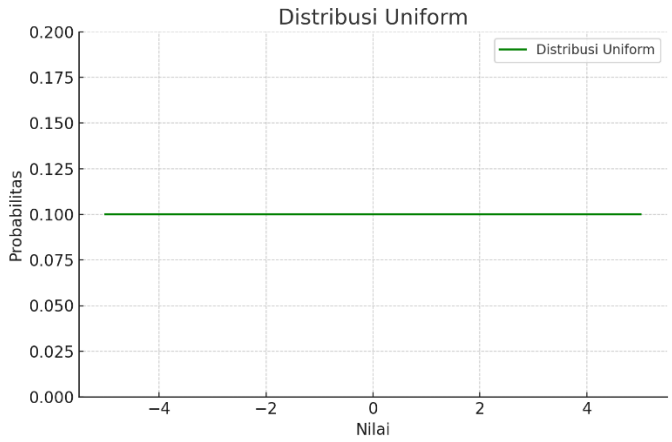
Selanjutnya, terdapat binomial distribution, yang merepresentasikan peristiwa dengan dua kemungkinan, seperti sukses dan gagal. Dalam computer vision, distribusi ini dapat digunakan pada kasus deteksi objek biner, misalnya keberadaan atau ketidakberadaan suatu objek dalam citra. Contoh sederhananya adalah klasifikasi antara "objek ada" dan "objek tidak ada" pada sistem keamanan berbasis kamera.

Untuk kasus klasifikasi yang lebih kompleks, digunakan multinomial distribution, yang memperluas konsep binomial ke banyak kategori. Distribusi ini relevan dalam klasifikasi multi-kelas, seperti pengenalan digit pada dataset MNIST yang memiliki sepuluh kelas (0–9). Setiap digit dipandang sebagai sebuah kategori, dan probabilitasnya dihitung berdasarkan seberapa sering pola tertentu muncul pada data pelatihan.
Dengan memahami berbagai distribusi ini, mahasiswa dapat melihat bahwa probabilitas bukan sekadar teori abstrak, melainkan alat praktis untuk menangani ketidakpastian pada data visual. Distribusi membantu kita merancang model yang lebih sesuai dengan sifat data, sehingga sistem vision menjadi lebih akurat dan tahan terhadap variasi kondisi nyata.



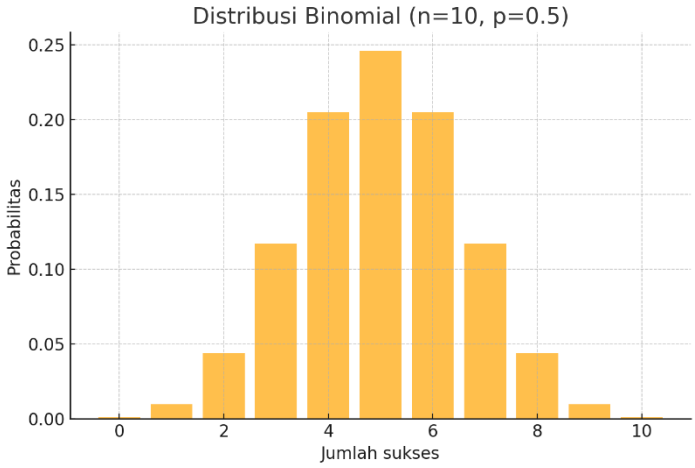

Gambar Distribusi Gaussian

Gambar di atas memperlihatkan distribusi Gaussian atau distribusi normal, yang berbentuk kurva lonceng simetris. Distribusi ini sangat penting dalam computer vision karena banyak fenomena visual dapat dimodelkan dengan pola Gaussian. Misalnya, noise kamera akibat cahaya rendah sering menyebar mengikuti distribusi normal, sehingga nilai piksel yang terekam berfluktuasi di sekitar rata-rata dengan penyebaran tertentu. Inilah alasan mengapa filter Gaussian digunakan secara luas untuk mereduksi noise pada citra: karena filter ini sesuai dengan sifat distribusi data yang dimodelkan.




Gambar Distribusi Uniform

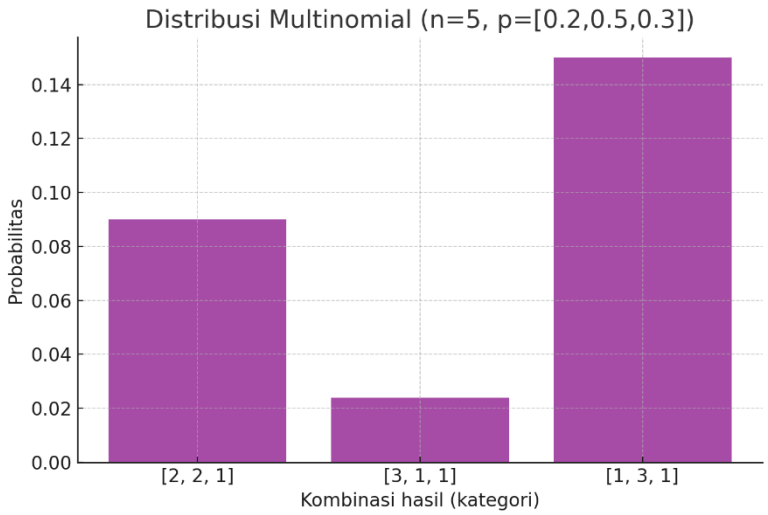
Distribusi uniform dicirikan oleh peluang yang sama pada setiap nilai dalam suatu rentang. Pada grafik terlihat bahwa semua nilai memiliki probabilitas konstan. Dalam konteks citra, distribusi ini relevan ketika piksel terkontaminasi oleh noise seragam, di mana setiap intensitas dalam rentang tertentu sama-sama mungkin muncul. Distribusi uniform juga banyak digunakan dalam proses *sampling*, misalnya saat memilih piksel secara acak untuk pelatihan model.




Gambar Binomial

Distribusi binomial menggambarkan peluang terjadinya sejumlah keberhasilan dari percobaan yang bersifat biner, yaitu hanya memiliki dua kemungkinan hasil: sukses atau gagal. Pada ilustrasi ditunjukkan distribusi binomial dengan $n = 10$ percobaan dan probabilitas sukses $p = 0.5$.

Dalam computer vision, distribusi ini dapat dipakai untuk memodelkan deteksi objek sederhana yang hanya menghasilkan dua kelas: objek ada atau objek tidak ada. Misalnya, sistem keamanan berbasis kamera yang hanya perlu mengidentifikasi apakah ada orang di ruangan atau tidak.




Gambar Multinomial

Distribusi multinomial memperluas konsep binomial ke lebih dari dua kategori. Grafik di atas menunjukkan probabilitas berbagai kombinasi hasil untuk tiga kategori dengan peluang [0.2,0.5,0.3] dalam lima percobaan. Dalam computer vision, distribusi ini sangat relevan untuk klasifikasi multi-kelas, misalnya mengenali angka dalam dataset MNIST yang memiliki sepuluh kelas (0–9). Setiap kelas (digit) dianggap sebagai kategori, dan probabilitasnya dihitung dari seberapa sering pola visual tertentu muncul dalam data latih. Dengan pendekatan ini, sistem dapat menentukan kelas mana yang paling mungkin sesuai dengan data input.

### 2.2.4 Statistik Deskriptif dalam Analisis Citra

Statistik deskriptif digunakan untuk merangkum informasi dari citra.

- Mean (rata-rata): menunjukkan tingkat kecerahan rata-rata sebuah gambar.
- Variance (ragam): menunjukkan seberapa beragam intensitas piksel dalam citra.
- Histogram: distribusi nilai intensitas, sering dipakai untuk analisis kontras.

Contoh: sebuah citra dengan mean tinggi kemungkinan terang, sedangkan citra dengan variance tinggi menunjukkan adanya banyak detail dan perbedaan kontras. Histogram intensitas bahkan bisa dipakai untuk segmentasi sederhana. Jika ada dua puncak (bimodal), kita bisa menggunakan *thresholding* untuk memisahkan objek dari latar belakang.

### 2.2.5 Teorema Bayes dan Pengenalan Pola

Teorema Bayes adalah salah satu pilar utama probabilitas yang banyak digunakan dalam pengenalan pola visual. Rumusnya adalah:

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)}$$

Di sini, $H$ adalah hipotesis (misalnya gambar adalah kucing), dan D adalah data yang diamati (piksel-piksel gambar). Teorema ini menghitung probabilitas hipotesis berdasarkan data yang masuk. Aplikasi sederhananya adalah Naive Bayes Classifier, yang sering digunakan untuk klasifikasi sederhana seperti mengenali digit angka pada dataset MNIST. Meskipun metode ini sederhana, hasilnya cukup baik dan dapat menjadi pengantar mahasiswa untuk memahami bagaimana probabilitas dipakai dalam klasifikasi citra.

### 2.2.6 Contoh Kasus: Pengenalan Digit MNIST dengan Naive Bayes

Dataset MNIST berisi 70.000 gambar digit tangan (0–9). Dengan Naive Bayes, setiap piksel dipandang sebagai fitur, lalu sistem menghitung probabilitas setiap digit muncul berdasarkan distribusi nilai piksel.

Hasilnya, meskipun hanya menggunakan probabilitas sederhana, akurasi klasifikasi bisa mencapai lebih dari 80%. Angka ini memang lebih rendah dibanding CNN modern yang bisa mencapai >99%, tetapi contoh ini sangat penting untuk menunjukkan bahwa teori probabilitas bisa langsung diaplikasikan ke dalam computer vision.

### 2.2.7 Variansi, Kovariansi, dan PCA

Statistika juga digunakan untuk memahami hubungan antar fitur visual. Variansi menunjukkan seberapa besar penyebaran data, sedangkan kovariansi menunjukkan hubungan antar variabel. Dalam PCA (Principal Component Analysis), kovariansi digunakan untuk menemukan sumbu utama penyebaran data. Dengan menghitung eigenvalue dan eigenvector dari matriks kovariansi, kita bisa menemukan dimensi baru yang lebih representatif.

Contoh aplikasinya adalah Eigenfaces dalam pengenalan wajah. Di sini, wajah manusia direpresentasikan bukan dengan ribuan piksel, tetapi dengan kombinasi beberapa komponen utama.

### 2.2.8 Estimasi Parameter

Banyak algoritma vision membutuhkan estimasi parameter distribusi. Misalnya, jika kita mengasumsikan piksel mengikuti distribusi normal, maka kita perlu menghitung mean dan variansinya. Estimasi ini bisa dilakukan dengan metode Maximum Likelihood Estimation (MLE).

Sebagai contoh, dalam segmentasi warna, distribusi RGB suatu objek bisa dimodelkan sebagai distribusi Gaussian. Dengan MLE, kita dapat menghitung mean dan varians warna objek, lalu menggunakannya untuk membedakan objek dari latar belakang.

Dengan memahami konsep-konsep probabilitas, mahasiswa tidak hanya belajar cara menghitung peluang, tetapi juga mengembangkan intuisi bahwa computer vision bekerja dalam kerangka “kemungkinan” alih-alih “kepastian mutlak”.

## 2.3 Fourier Transform, DCT, dan Wavelet dalam Computer Vision

### 2.3.1 Pendahuluan

Salah satu keunikan citra digital adalah bahwa ia dapat dipahami bukan hanya dalam domain ruang (spatial domain), tetapi juga dalam domain frekuensi (frequency domain). Domain ruang merepresentasikan citra sebagai susunan piksel yang terlihat, sedangkan domain frekuensi merepresentasikannya sebagai gabungan gelombang dengan frekuensi berbeda.

Mengubah citra ke domain frekuensi memungkinkan kita melihat pola tersembunyi yang tidak mudah dikenali secara langsung di domain ruang. Teknik inilah yang mendasari berbagai aplikasi penting, mulai dari pengurangan noise, deteksi tepi, hingga kompresi JPEG.

### 2.3.2 Fourier Transform dalam Citra

Transformasi Fourier adalah alat matematis untuk memecah sebuah sinyal atau citra menjadi komponen frekuensinya. Secara sederhana, Fourier menyatakan bahwa setiap sinyal kompleks dapat dibangun dari kombinasi gelombang sinus dan cosinus.

Dalam citra digital, frekuensi rendah menggambarkan bagian halus atau area rata (seperti latar belakang langit biru), sedangkan frekuensi tinggi menggambarkan detail tajam (seperti tepi gedung atau rambut). Dengan memahami perbedaan ini, kita dapat memanipulasi citra lebih efektif.

Sebagai contoh, jika kita menghapus komponen frekuensi tinggi, citra akan tampak lebih halus (*low-pass filtering*). Sebaliknya, jika kita hanya mempertahankan frekuensi tinggi, citra yang dihasilkan akan menonjolkan tepi (*high-pass filtering*).

### 2.3.3 Contoh Visual Domain Frekuensi

Sebuah citra sederhana, misalnya wajah manusia, jika ditransformasikan dengan Fourier, akan menghasilkan spektrum frekuensi. Bagian tengah spektrum merepresentasikan frekuensi rendah, sedangkan bagian pinggir berisi frekuensi tinggi. Filter Gaussian di domain frekuensi bekerja dengan memotong bagian tertentu dari spektrum ini, kemudian mengembalikannya ke domain ruang melalui invers Fourier transform.

### 2.3.4 Discrete Cosine Transform (DCT)

DCT adalah variasi khusus dari Fourier Transform yang menggunakan hanya fungsi cosinus. Kelebihan DCT adalah efisiensinya dalam merepresentasikan informasi citra dengan jumlah koefisien yang lebih sedikit.

Inilah yang menjadikan DCT sebagai dasar dari algoritma JPEG compression. Dalam JPEG, citra dipecah menjadi blok 8×8 piksel, lalu setiap blok ditransformasikan dengan DCT. Hasilnya berupa koefisien-koefisien yang sebagian besar bernilai mendekati nol. Koefisien yang tidak signifikan dapat diabaikan, sehingga ukuran file berkurang drastis tanpa mengorbankan kualitas visual secara besar-besaran.

### 2.3.5 Wavelet Transform

Berbeda dengan Fourier yang hanya memecah sinyal berdasarkan frekuensi global, wavelet transform memungkinkan analisis pada berbagai skala (*multi-resolution analysis*). Dengan wavelet, kita bisa melihat citra pada resolusi rendah (untuk struktur global) sekaligus pada resolusi tinggi (untuk detail lokal).

Wavelet digunakan dalam berbagai aplikasi vision, termasuk kompresi citra (format JPEG2000), denoising, dan pengenalan pola. Keunggulannya adalah kemampuannya menangkap informasi lokal yang tidak bisa dilakukan oleh Fourier murni.

### 2.3.6 Tabel Perbandingan

Tabel 2.3 Perbandingan Metode

| Metode | Kelebihan | Kekurangan | Aplikasi |
|---|---|---|---|
| Fourier Transform | Representasi frekuensi global yang akurat | Tidak menangkap informasi lokal | Filtering, analisis spektrum |
| DCT | Efisien, koefisien cepat menurun | Tidak fleksibel untuk sinyal non-stasioner | Kompresi JPEG |
| Wavelet | Multi-resolusi, menangkap detail lokal | Lebih kompleks, membutuhkan komputasi lebih | JPEG2000, denoising, analisis pola |

Tabel 2.3 ini memperlihatkan bahwa setiap metode memiliki kekuatan dan keterbatasan masing-masing, sehingga pemilihan metode bergantung pada aplikasi yang ingin dicapai.

Setiap metode transformasi sinyal dalam computer vision memiliki kelebihan dan kekurangan yang menjadikannya sesuai untuk aplikasi tertentu. **Fourier Transform** unggul dalam merepresentasikan informasi frekuensi global dari sebuah citra. Artinya, ia mampu menggambarkan dengan sangat baik seberapa banyak komponen frekuensi rendah dan tinggi yang terkandung dalam gambar. Namun, kelemahannya adalah Fourier tidak bisa memberikan informasi lokal. Misalnya, kita tahu bahwa ada frekuensi tinggi pada sebuah citra, tetapi tidak tahu di bagian mana frekuensi itu muncul. Karena itu, Fourier biasanya lebih cocok digunakan untuk filtering global atau analisis spektrum.

**Discrete Cosine Transform (DCT)** memiliki sifat khusus karena hanya menggunakan fungsi cosinus. Hal ini membuat representasi data menjadi lebih efisien: sebagian besar informasi citra dapat ditangkap oleh sejumlah kecil koefisien pertama. Inilah yang menjadikan DCT sangat efektif untuk kompresi, khususnya pada format JPEG. Kekurangannya, DCT tidak cukup fleksibel jika sinyal atau citra mengandung perubahan lokal yang signifikan, sehingga kadang menimbulkan artefak blok (blocking artifact) pada kompresi gambar. Berbeda dengan keduanya,

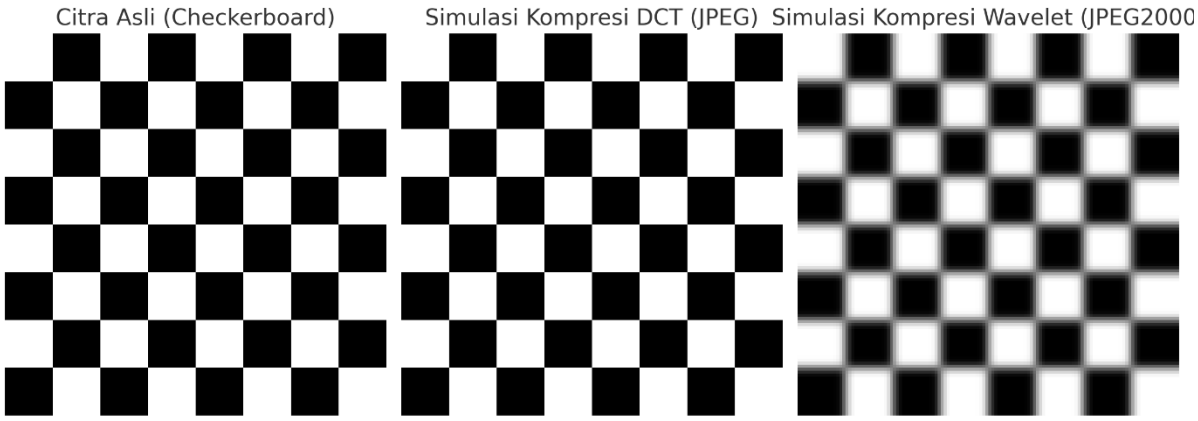
**Wavelet Transform** menawarkan pendekatan multi-resolusi. Dengan wavelet, citra dapat dianalisis baik pada tingkat resolusi global maupun detail lokal. Hal ini menjadikannya lebih unggul dibanding Fourier dalam mendeteksi perubahan lokal, misalnya tepi atau tekstur halus. Namun, kelemahannya adalah perhitungan wavelet lebih kompleks dan membutuhkan sumber daya komputasi lebih banyak. Meski begitu, wavelet sangat berguna dalam aplikasi yang membutuhkan keseimbangan antara detail lokal dan global, seperti kompresi JPEG2000, denoising citra, dan pengenalan pola visual.Secara keseluruhan, tabel ini memperlihatkan bahwa tidak ada satu metode yang sepenuhnya unggul. Fourier baik untuk analisis frekuensi global, DCT efisien untuk kompresi, sementara wavelet unggul untuk analisis multi-resolusi. Pemilihan metode harus disesuaikan dengan kebutuhan aplikasi, sehingga mahasiswa dapat memahami logika di balik penggunaan transformasi tertentu dalam computer vision.

### 2.3.7 Contoh Kasus: Kompresi JPEG

Dalam kehidupan sehari-hari, hampir semua gambar digital yang kita temui berbentuk JPEG. Proses di balik JPEG menunjukkan bagaimana matematika berperan dalam teknologi sehari-hari.

Langkah-langkahnya adalah:

- Citra dibagi ke dalam blok 8×8 piksel.
- Setiap blok ditransformasikan menggunakan DCT.
- Koefisien hasil DCT diurutkan berdasarkan pentingnya.
- Koefisien dengan kontribusi kecil dibuang.
- Data dikodekan ulang dalam bentuk bitstream.
- Hasil akhirnya adalah citra dengan ukuran file lebih kecil, namun secara visual masih dapat dikenali dengan baik.




Gambar Kompresi JPEG (blok DCT) vs JPEG2000 (wavelet)

Gambar perbandingan di atas memperlihatkan bagaimana perbedaan metode transformasi menghasilkan kualitas kompresi yang berbeda. Pada citra asli dengan pola kotak-kotak (*checkerboard*), detail garis tegas masih tampak jelas dan teratur. Ketika citra dikompresi menggunakan **Discrete Cosine Transform (DCT)** seperti pada format JPEG, pola kotak tetap terlihat, namun muncul efek samping berupa garis-garis blok yang disebut *blocking artifact*. Artefak ini muncul karena JPEG bekerja dengan membagi citra ke dalam blok-blok kecil (8×8 piksel), lalu membuang sebagian koefisien frekuensi tinggi pada tiap blok. Meskipun efisien, cara ini sering menghasilkan batas blok yang terlihat jelas terutama pada gambar dengan pola tajam.

Sebaliknya, pada hasil kompresi dengan **Wavelet Transform** yang digunakan dalam format JPEG2000, artefak blok hampir tidak terlihat. Wavelet menganalisis citra pada berbagai skala (*multi-resolution*), sehingga struktur global dan detail lokal dapat dipertahankan dengan lebih halus. Kekurangannya, beberapa detail tajam sedikit berkurang, tetapi secara keseluruhan hasil kompresi lebih natural dibandingkan JPEG. Perbandingan ini menunjukkan keunggulan wavelet dalam menjaga kualitas visual meskipun ukuran file diperkecil, sekaligus menjelaskan mengapa metode ini banyak dipakai dalam aplikasi yang menuntut kualitas tinggi, seperti citra medis atau arsip digital.

Dengan demikian, gambar ini menegaskan bahwa pemilihan transformasi tidak sekadar soal matematika, melainkan juga berdampak nyata pada kualitas visual. Fourier, DCT, dan wavelet masing-masing menawarkan pendekatan berbeda, dan keberhasilan aplikasinya sangat bergantung pada konteks kebutuhan kompresi atau analisis citra.

### 2.3.8 Contoh Kasus: Filtering Noise

Salah satu tantangan utama dalam pengolahan citra adalah keberadaan noise, yaitu gangguan acak yang mengotori informasi visual. Noise dapat muncul dari berbagai sumber, misalnya keterbatasan sensor kamera, kondisi pencahayaan yang buruk, transmisi data yang terganggu, atau bahkan kompresi digital yang berulang. Kehadiran noise membuat citra kehilangan kejernihannya, sehingga detail penting seperti tepi objek atau tekstur halus menjadi sulit dikenali.

Fourier dan wavelet menjadi alat penting dalam upaya reduksi noise karena keduanya memungkinkan analisis citra di domain frekuensi. Jika sebuah citra tercemar oleh noise dengan pola frekuensi tertentu, maka noise tersebut dapat difilter di domain frekuensi. Sebagai contoh, noise berupa bintik-bintik halus (salt and pepper noise atau Gaussian noise) biasanya muncul pada komponen frekuensi tinggi. Dengan menerapkan low-pass filtering, komponen frekuensi tinggi dapat dilemahkan sehingga noise berkurang tanpa menghilangkan struktur utama citra.

Sebaliknya, jika yang ingin dipertahankan adalah detail tajam, seperti pada citra medis atau citra satelit, penggunaan wavelet transform lebih disukai. Wavelet memiliki kemampuan multi-resolution analysis, yang memungkinkan citra dipisahkan ke dalam representasi frekuensi rendah dan tinggi di berbagai skala. Dengan cara ini, noise dapat dikurangi hanya pada level tertentu, sementara detail penting tetap dipertahankan. Misalnya, pada citra MRI otak, reduksi noise dengan wavelet dapat meningkatkan kejelasan batas antara jaringan sehat dan jaringan yang mengalami kelainan.

Selain pendekatan klasik, filtering noise juga memiliki implikasi penting dalam deep learning. Sebelum citra dilatih ke dalam model CNN, sering kali dilakukan preprocessing berupa smoothing untuk mengurangi pengaruh noise. Tanpa preprocessing ini, model bisa salah mengenali pola, karena kebisingan visual dianggap sebagai fitur yang relevan.

Untuk memperjelas perbedaan pendekatan, dapat dibuat perbandingan sederhana:

- Filtering di domain ruang (spatial domain): menggunakan filter rata-rata (mean filter) atau filter median. Cocok untuk noise sederhana, tetapi sering mengaburkan detail.
- Filtering di domain frekuensi (Fourier): menekankan manipulasi spektrum citra. Cocok untuk noise yang jelas menempati frekuensi tertentu.
- Filtering dengan wavelet: fleksibel untuk berbagai jenis noise, mampu mempertahankan detail lokal, dan lebih adaptif terhadap variasi dalam citra.

Sebagai ilustrasi nyata, bayangkan sebuah citra satelit yang merekam permukaan sawah. Karena pengaruh atmosfer, citra tersebut penuh dengan bintik halus. Dengan low-pass filtering berbasis Fourier, noise bisa dihaluskan, tetapi detail pematang sawah ikut hilang. Sementara itu, jika digunakan wavelet, detail garis pematang masih terlihat, sementara noise bintik-bintik berkurang signifikan.

Kasus lain adalah dalam fotografi digital. Kamera ponsel yang digunakan pada malam hari sering menghasilkan foto dengan noise tinggi akibat sensor kekurangan cahaya. Aplikasi kamera modern memanfaatkan prinsip filtering frekuensi untuk membersihkan foto. Sebagian bahkan mengombinasikan filtering klasik dengan model pembelajaran mesin agar hasilnya tampak lebih alami.

Dengan demikian, filtering noise tidak sekadar proses teknis, tetapi juga strategi ilmiah untuk memastikan informasi penting dalam citra tetap dapat diakses meskipun lingkungan pengambilan gambar penuh gangguan. Fourier memberi kita lensa global untuk melihat spektrum noise, sedangkan wavelet menyediakan alat yang lebih halus untuk memilih bagian mana dari citra yang harus dibersihkan.

# BAB III
# CITRA DIGIAL

## 3.1 Representasi Citra Digital

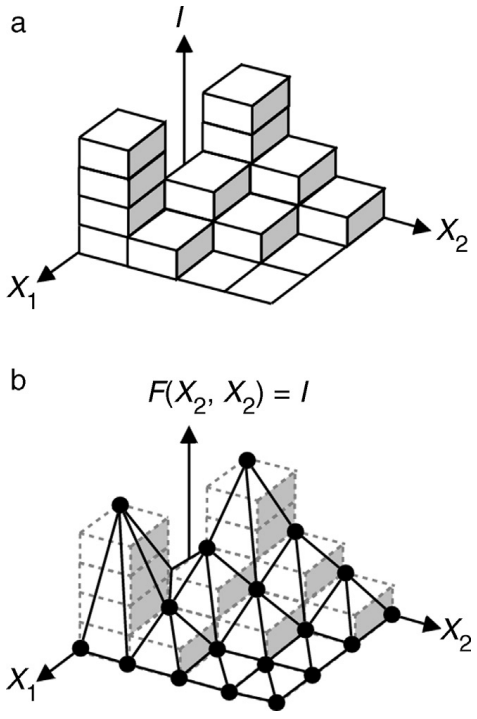
### 3.1.1 Apa Itu Citra Digital?

Citra digital adalah representasi dua dimensi dari dunia nyata atau objek visual yang telah diubah menjadi bentuk numerik sehingga komputer dapat menyimpan, mengolah, dan menganalisisnya. Setiap piksel citra
digital terdiri dari grid dua dimensi yang disebut pixel (picture element). Informasi tentang intensitas atau warna terkandung di setiap piksel.

### 3.1.2 Struktur Data pada Citra

Citra Grayscale

Setiap piksel citra grayscale pada gambar 1 memiliki nilai intensitas yang berkisar antara 0 (total hitam) dan 255 (total putih), dan citra
grayscale hanya memiliki satu kanal informasi untuk menunjukkan tingkat kecerahan (brightness) dari hitam ke putih.



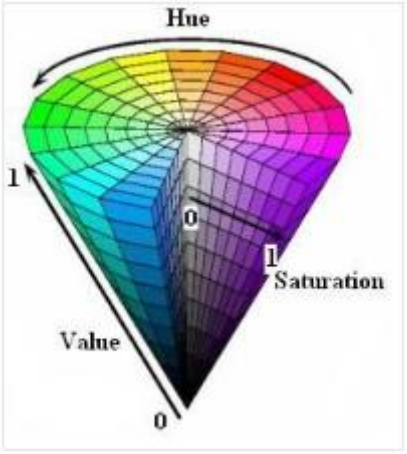

Gambar 1 menunjukkan representasi citra grayscale sebagai array piksel diskrit dan fungsi kontinu piecewise-linear.

Gambar1 menjelaskan dua cara representasi citra digital grayscale:

- Bagian (a) menunjukkan citra sebagai array piksel diskrit, di mana setiap kotak mewakili sebuah piksel dengan intensitas tertentu (ditunjukkan oleh tinggi balok). Koordinat X1X_1X1 dan X2X_2X2 menyatakan posisi piksel dalam bidang spasial, sedangkan III menunjukkan nilai intensitas.
- Bagian (b) menunjukkan citra yang sama sebagai fungsi kontinu F(X1,X2)=IF(X_1, X_2) = IF(X1,X2)=I, di mana intensitas diinterpolasi menjadi permukaan yang mulus

(piecewise-linear). Representasi ini menekankan bahwa citra digital, meskipun berupa sampel diskrit, dapat dipandang sebagai aproksimasi fungsi intensitas kontinu.

Citra Berwarna (RGB)
Setiap piksel citra RGB pada gambar 2 memiliki tiga kanal warna: merah (Red), hijau (Green), dan biru (Blue). Tiga nilai intensitas untuk
setiap piksel menunjukkan kombinasi proporsi dari ketiga kanal tersebut.




Gambar 2. Representasi Piksel Citra RGB

Gambar 2 memperlihatkan representasi piksel citra dalam ruang warna HSV (Hue, Saturation, Value) yang sering digunakan sebagai alternatif dari model RGB. Dalam model ini, warna diuraikan berdasarkan tiga komponen utama.

Hue (H) menggambarkan jenis warna itu sendiri, misalnya merah, hijau, biru, atau kuning. Hue dinyatakan dalam bentuk sudut melingkar seperti roda warna, sehingga memudahkan visualisasi perbedaan warna. Pada gambar, hue ditunjukkan sebagai lingkaran spektrum yang mengelilingi kerucut, di mana setiap posisi sudut merepresentasikan warna berbeda.

Saturation (S) menunjukkan tingkat kemurnian warna. Nilai saturasi 1 berarti warna berada pada kondisi paling jenuh dan tajam, sedangkan nilai mendekati 0 berarti warna semakin pudar menuju abu-abu. Pada ilustrasi, saturasi ditampilkan sebagai jarak dari pusat kerucut ke arah luar. Semakin jauh dari pusat, semakin kuat intensitas warnanya.

Value (V) berkaitan dengan tingkat kecerahan atau terang-gelap suatu warna. Nilai 0 berarti hitam total, sedangkan nilai 1 berarti warna berada pada tingkat kecerahan maksimal. Pada gambar, value divisualisasikan sebagai sumbu vertikal dari dasar kerucut ke bagian atas.

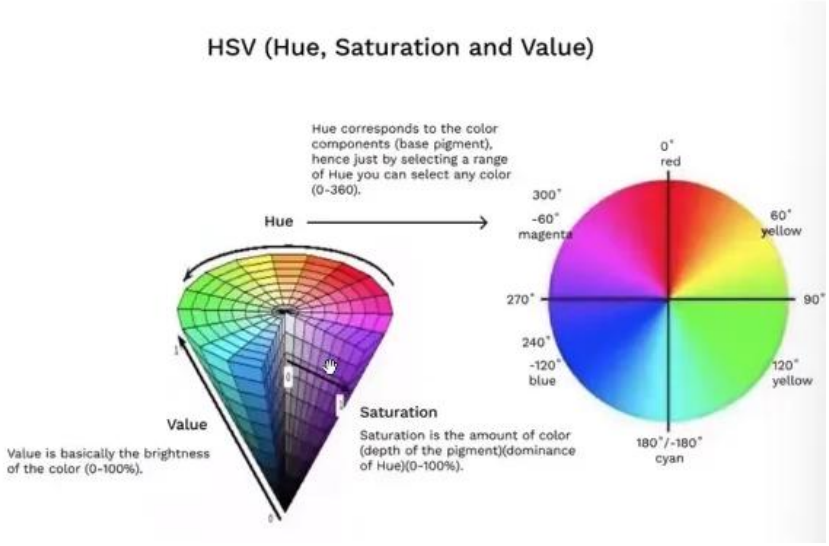
Model HSV banyak dipakai dalam pengolahan citra karena lebih sesuai dengan cara manusia memahami warna. Jika model RGB berfokus pada kombinasi tiga cahaya dasar (merah, hijau, biru), maka HSV menawarkan pendekatan yang lebih intuitif. Dengan menggunakan HSV, misalnya, kita dapat dengan mudah memisahkan objek berdasarkan warna dominan tanpa terlalu dipengaruhi oleh perubahan pencahayaan.

Dengan demikian, gambar ini tidak hanya memperlihatkan representasi visual piksel, tetapi juga menggambarkan bagaimana warna dapat dipetakan secara sistematis untuk mendukung

berbagai aplikasi pengolahan citra digital, mulai dari segmentasi objek hingga analisis pola warna.

### 3.1.3 Format Lain Representasi Warna

HSV (Hue, Saturation, Value): membedakan aspek warna (hue) dari intensitas (value) dan kejenuhan (saturation) warna. Ini digunakan sesuai dengan aplikasi.




Gambar 3. Diagram Model Warna RGB vs HSV

Gambar 3 menampilkan perbandingan antara dua representasi warna yang paling umum digunakan dalam pengolahan citra digital, yaitu RGB (Red, Green, Blue) dan HSV (Hue, Saturation, Value).

Model RGB didasarkan pada kombinasi tiga cahaya primer: merah, hijau, dan biru. Setiap warna dihasilkan dari pencampuran ketiganya dengan intensitas tertentu, sehingga ruang warna RGB berbentuk kubus tiga dimensi. Model ini sangat sesuai untuk perangkat tampilan digital seperti monitor, kamera, dan sensor karena bekerja langsung dengan sinyal cahaya. Namun, RGB tidak selalu intuitif ketika digunakan untuk analisis warna yang melibatkan persepsi manusia, misalnya dalam segmentasi objek berdasarkan warna.

Untuk mengatasi keterbatasan tersebut, dikembangkan model HSV. Dalam model ini, warna direpresentasikan dengan tiga parameter yang lebih dekat dengan persepsi visual manusia:

Hue (H) menggambarkan jenis warna dasar, direpresentasikan dalam bentuk sudut melingkar 0°–360°. Misalnya, 0° mewakili merah, 120° hijau, dan 240° biru. Pada gambar, hue divisualisasikan sebagai roda warna.

Saturation (S) menunjukkan tingkat kejenuhan warna, yaitu seberapa murni atau dominan suatu warna dibandingkan campuran abu-abu. Nilai mendekati 100% berarti warna sangat jenuh, sedangkan nilai rendah menghasilkan warna yang lebih pucat.

Value (V) merepresentasikan tingkat kecerahan (brightness) warna, dari 0% (hitam total) hingga 100% (terang maksimum).

Secara visual, ruang warna HSV ditunjukkan dalam bentuk kerucut (atau kerucut terpotong), di mana hue melingkar di sekeliling kerucut, saturasi bergerak dari pusat ke tepi, dan value ditunjukkan pada sumbu vertikal.

Perbedaan mendasar antara RGB dan HSV adalah orientasinya: RGB lebih bersifat teknis, berfokus pada bagaimana warna dibentuk dari cahaya primer, sementara HSV lebih bersifat perseptual, menekankan bagaimana manusia mengenali warna berdasarkan jenis, kejenuhan, dan tingkat kecerahan. Oleh karena itu, model HSV sering digunakan dalam aplikasi deteksi objek, pengolahan citra medis, dan computer vision, karena memungkinkan pemisahan warna yang lebih stabil terhadap perubahan intensitas cahaya.

Dengan demikian, gambar ini menjelaskan bahwa meskipun RGB merupakan dasar teknis pembentukan warna pada perangkat digital, HSV memberikan cara representasi yang lebih intuitif untuk memahami dan mengolah warna sesuai dengan cara manusia mempersepsinya.

### 3.1.4 Resolusi dan Dimensi Citra

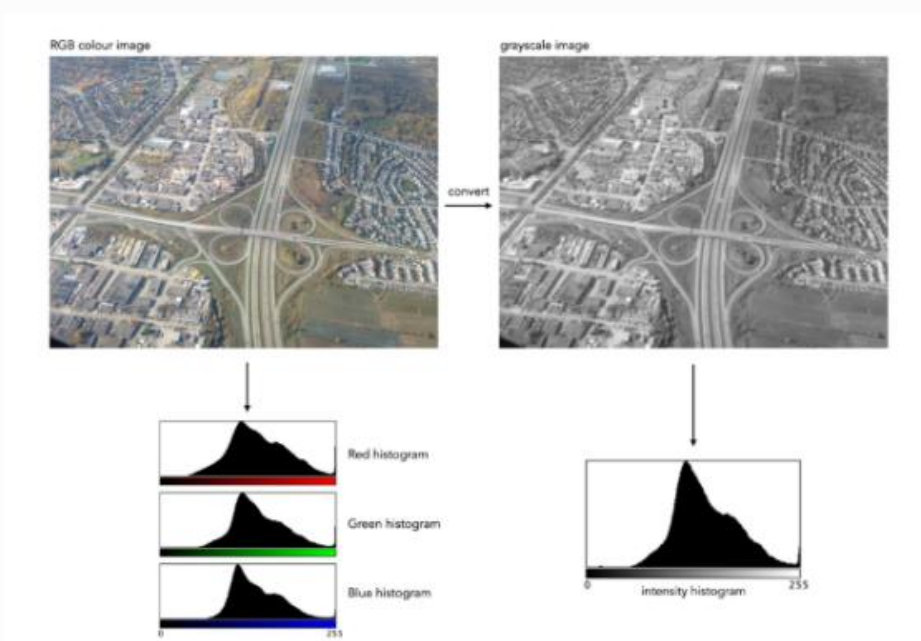
Resolusi sebuah gambar menunjukkan jumlah piksel dalam arah horizontal dan vertikal, seperti 1920 x 1080 (Full HD). Semakin tinggi resolusi sebuah gambar, semakin banyak informasi visual yang dapat ditangkap.

### 3.1.5 Representasi Numerik dalam Komputer

Ciri digital diwakili sebagai array multidimensi dalam pemrograman. Misalnya, citra grayscale adalah matriks dua dimensi yang berisi nilai intensitas, sedangkan citra RGB adalah tensor 3 dimensi yang berukuran tinggi, lebar, dan tiga sumbu, di mana sumbu ketiga berfungsi sebagai kanal warna.

### 3.1.6 Histogram Citra

Histogram menunjukkan distribusi nilai intensitas dalam gambar secara grafis. Untuk gambar grayscale, histogram menunjukkan frekuensi yang muncul untuk masing-masing level keabuan dari 0 hingga 255.Hal
ini terlihat pada gambar 4. Yang terlihat berupa Histogram Citra Grayscale




Gambar 4 menunjukkan histogram citra sebelum dan sesudah proses equalization.

Gambar 4 memperlihatkan perbedaan distribusi intensitas citra sebelum dan sesudah melalui proses histogram equalization. Pada sisi kiri ditampilkan citra berwarna dalam ruang RGB, lengkap dengan tiga histogram yang merepresentasikan distribusi nilai intensitas masing-masing kanal merah, hijau, dan biru. Setiap histogram menunjukkan sebaran piksel terhadap tingkat kecerahan, di mana konsentrasi piksel pada rentang tertentu menandakan dominasi warna atau intensitas tertentu dalam citra.

Setelah citra dikonversi menjadi grayscale, distribusi intensitas piksel digambarkan dalam bentuk satu histogram gabungan. Histogram ini memperlihatkan penyebaran tingkat keabuan mulai dari 0 (hitam pekat) hingga 255 (putih terang). Pada tahap ini, variasi intensitas lebih mudah dianalisis karena seluruh informasi warna telah disederhanakan menjadi satu dimensi kecerahan.

Proses equalization dilakukan untuk meningkatkan kualitas visual citra dengan cara meratakan distribusi intensitas. Tujuannya adalah memperluas rentang kontras sehingga detail yang sebelumnya kurang terlihat dapat muncul lebih jelas. Pada citra hasil equalization, histogram tampak lebih merata dibandingkan sebelum proses, menandakan distribusi intensitas yang lebih seimbang. Hal ini membuat perbedaan terang-gelap pada citra menjadi lebih jelas, sehingga objek-objek dalam gambar dapat dikenali dengan lebih baik.

Dengan demikian, gambar ini menegaskan peran histogram equalization sebagai salah satu teknik penting dalam pengolahan citra digital. Teknik ini tidak hanya meningkatkan kontras

secara visual, tetapi juga mempermudah tahap analisis lanjutan, seperti segmentasi, deteksi tepi, maupun klasifikasi objek.

### 3.1.7 Tantangan dalam Representasi Visual

Beberapa masalah yang sering muncul dalam representasi gambar digital adalah sebagai berikut: suara: gangguan visual berupa titik-titik acak; perubahan cahaya: intensitas cahaya yang tidak merata; shadow dan occlusion: bagian objek yang tertutup atau terlindung bayangan; dan skala dan rotasi: objek yang sama mungkin memiliki representasi piksel yang berbeda jika dilihat dari sudut atau jarak yang berbeda.

### 3.1.8 Pentingnya Pra-pemrosesan Citra

Sebelum gambar digital dianalisis oleh algoritma vision, tahap pra-pemrosesan dilakukan dengan tujuan meningkatkan kualitas visual untuk mengatasi variasi dan noise.

## 3.2 Transformasi dan Operasi Dasar pada Citra Digital

### 3.2.1 Transformasi Geometris

Proses yang mengubah posisi, ukuran, atau orientasi piksel dalam gambar disebut transformasi geometris. Proses ini sangat penting untuk berbagai tujuan, seperti koreksi distorsi, registrasi gambar, dan augmentasi data.

**a. Translasi**

Translasi adalah proses menggerakkan seluruh piksel dalam gambar dalam suatu jarak tertentu dalam arah horizontal dan/atau vertikal. Untuk menggerakkan gambar sejauh lima puluh piksel ke kanan dan tiga puluh piksel ke bawah, matriks translasi yang ditunjukkan di bawah ini digunakan:

$$T = \begin{bmatrix} 1 & 0 & 50 \\ 0 & 1 & 30 \\ 0 & 0 & 1 \end{bmatrix}$$

Matriks ini digunakan dalam transformasi homogen untuk menghitung posisi baru setiap piksel.


Gambar 5. Contoh Output dari Image Translation

Gambar 5 menampilkan contoh hasil dari proses image translation, yaitu salah satu transformasi geometrik dalam pengolahan citra digital. Pada sisi kiri ditunjukkan citra asli (original image), sedangkan pada sisi kanan diperlihatkan citra hasil translasi (image translation).

Secara konsep, image translation berarti menggeser posisi citra ke arah tertentu dalam bidang koordinat tanpa mengubah bentuk, ukuran, maupun orientasi objek di dalamnya. Pergeseran ini didefinisikan dengan menambahkan vektor translasi $(t_x, t_y)$ pada setiap piksel citra, di mana $t_x$ menunjukkan pergeseran horizontal dan $t_y$ menunjukkan pergeseran vertikal.
Pada hasil yang ditampilkan, citra asli bergeser ke arah kanan bawah sehingga sebagian area di sisi kiri dan atas menjadi kosong (biasanya diisi dengan warna latar, misalnya hitam).
Meskipun posisi citra berubah, karakteristik objek seperti warna, tekstur, dan pola tetap sama persis dengan citra aslinya.

Transformasi translasi memiliki fungsi penting dalam berbagai aplikasi pengolahan citra dan computer vision. Misalnya, pada augmentasi data untuk pelatihan model deep learning, translasi digunakan untuk menghasilkan variasi posisi objek sehingga model lebih robust terhadap pergeseran spasial. Selain itu, translasi juga sering diterapkan pada sistem pelacakan objek, registrasi citra medis, hingga koreksi pergeseran pada citra satelit atau drone.

Dengan demikian, gambar ini memberikan ilustrasi sederhana namun jelas mengenai bagaimana translasi dapat mengubah posisi citra tanpa memengaruhi sifat visual objek yang terkandung di dalamnya.

**b. Rotasi**
Rotasi adalah proses memutar gambar ke arah pusat atau sudut tertentu. Untuk sudut θ, matriks rotasi adalah:
Aplikasi seperti rotasi gambar medis atau orientasi objek dalam visi komputer menggunakan rotasi.

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$


Gambar 6. Contoh Image Rotation

Gambar 6 memperlihatkan contoh penerapan transformasi geometrik berupa rotasi citra (image rotation). Pada sisi kiri ditampilkan citra asli (original image), sementara pada sisi kanan ditunjukkan hasil citra setelah diputar (image rotation).

Secara konsep, rotasi citra dilakukan dengan memutar seluruh koordinat piksel terhadap suatu titik pusat rotasi, yang umumnya terletak di tengah citra. Perubahan ini dapat dinyatakan dengan persamaan transformasi linear menggunakan matriks rotasi:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

di mana $(x, y)$ adalah koordinat piksel sebelum rotasi, $(x', y')$ adalah koordinat setelah rotasi, dan $\theta$ merupakan sudut rotasi.

Pada hasil yang ditampilkan, citra asli diputar dengan sudut tertentu sehingga menghasilkan orientasi baru. Akibat dari proses ini, sebagian area di luar batas asli citra menjadi kosong dan biasanya diisi dengan warna latar, seperti hitam. Walaupun posisinya berubah, karakteristik objek di dalam citra tetap sama, baik dari segi warna, tekstur, maupun bentuk.

Transformasi rotasi memiliki peranan penting dalam banyak aplikasi pengolahan citra. Misalnya, pada augmentasi data, rotasi membantu meningkatkan variasi sudut pandang objek agar model pembelajaran mesin lebih tahan terhadap perubahan orientasi. Selain itu, rotasi juga banyak digunakan pada registrasi citra medis, penyelarasan citra satelit, sistem pengenalan pola, serta aplikasi grafis dan desain.

Dengan demikian, gambar ini menggambarkan bagaimana rotasi dapat mengubah orientasi citra tanpa mengubah informasi visual dasar yang terkandung di dalamnya.

**c. Skala (Scaling)**

Skala mengubah ukuran citra dengan faktor tertentu pada sumbu x dan y. Matriks skala dengan faktor sxs_xsx dan sys_ysy adalah:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Skala digunakan untuk memperbesar atau memperkecil citra sesuai kebutuhan aplikasi.

**d. Shearing**

Shearing menggeser satu sumbu citra secara proporsional terhadap sumbu lainnya menghasilkan efek miring. Matriks shearing adalah:

$$Ref_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

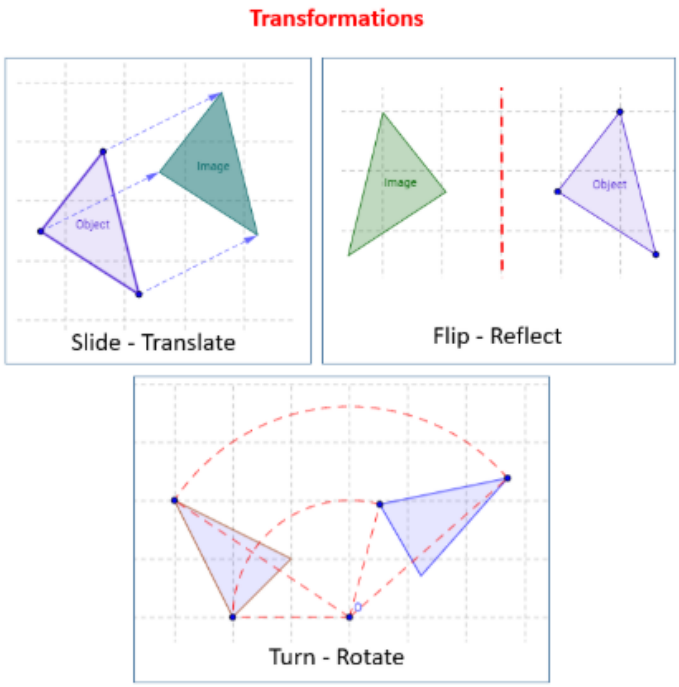
Shearing digunakan dalam efek artistik atau koreksi perspektif.

**e. Refleksi**

Refleksi mencerminkan citra terhadap sumbu tertentu. Misalnya, refleksi terhadap sumbu y menggunakan matriks:

$$Ref_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Refleksi digunakan dalam augmentasi data dan analisis simetri.




Gambar 7. Contoh Transformasi Geometris

Gambar 7 memperlihatkan tiga jenis transformasi geometris dasar dalam pengolahan citra digital, yaitu translasi (translation), refleksi (reflection), dan rotasi (rotation). Transformasi ini digunakan untuk memodifikasi posisi, orientasi, atau tampilan citra tanpa mengubah struktur internal dari objek yang direpresentasikan.

1. Slide – Translate (Translasi)
   Translasi merupakan proses menggeser citra dari satu posisi ke posisi lain dalam bidang koordinat. Pergeseran dilakukan dengan menambahkan vector $(t_x, t_y)$ ke setiap piksel citra.
   Hasilnya, objek tetap mempertahankan ukuran, bentuk, dan orientasi, hanya posisinya saja yang berubah.

2. Flip – Reflect (Refleksi)
   Refleksi adalah transformasi yang menghasilkan bayangan cermin dari suatu citra terhadap garis tertentu, misalnya sumbu vertikal atau horizontal. Proses ini sering digunakan untuk membalik orientasi citra, misalnya dari kiri ke kanan (horizontal flip) atau dari atas ke bawah (vertical flip). Refleksi banyak dimanfaatkan pada augmentasi data untuk menambah variasi arah pandang objek.

3. Turn – Rotate (Rotasi)
   Rotasi dilakukan dengan memutar citra terhadap titik pusat tertentu dengan sudut $\theta$. Transformasi ini mengubah orientasi citra, namun tidak memengaruhi bentuk maupun ukuran objek. Rotasi digunakan secara luas dalam sistem pengenalan pola dan computer vision untuk membuat model lebih robust terhadap perbedaan sudut pandang.

Transformasi geometris seperti yang ditunjukkan dalam gambar ini memiliki peranan penting baik dalam analisis citra maupun dalam augmentasi data pada machine learning. Dengan melakukan translasi, refleksi, dan rotasi, data pelatihan menjadi lebih beragam sehingga model dapat mengenali objek dengan lebih baik meskipun posisi atau orientasinya berubah.

Dengan demikian, gambar ini tidak hanya menjelaskan konsep matematis dari transformasi geometris, tetapi juga menunjukkan bagaimana prinsip tersebut diaplikasikan untuk mendukung berbagai kebutuhan dalam pengolahan citra digital.

### 3.2.2 Operasi Titik (Point Operations)

Digunakan untuk menyesuaikan kontras, kecerahan, dan thresholding, operasi titik mengubah nilai intensitas setiap piksel secara mandiri tanpa memperhitungkan piksel tetangganya.

#### a. Penyesuaian Kecerahan dan Kontras

Dengan menambahkan nilai konstan ke setiap piksel, kecerahan ditingkatkan, sedangkan dengan mengalikan setiap piksel dengan faktor tertentu, kontras ditingkatkan:

$$I_{baru}(x, y) = \alpha \cdot I(x, y) + \beta$$

Di mana α\alphaα adalah faktor kontras dan β\betaβ adalah nilai kecerahan.

**b. Negatif Citra**
Untuk mendapatkan nilai negatif gambar, setiap nilai piksel harus dikurangkan dari nilai tertinggi, misalnya 255 untuk gambar 8-bit:

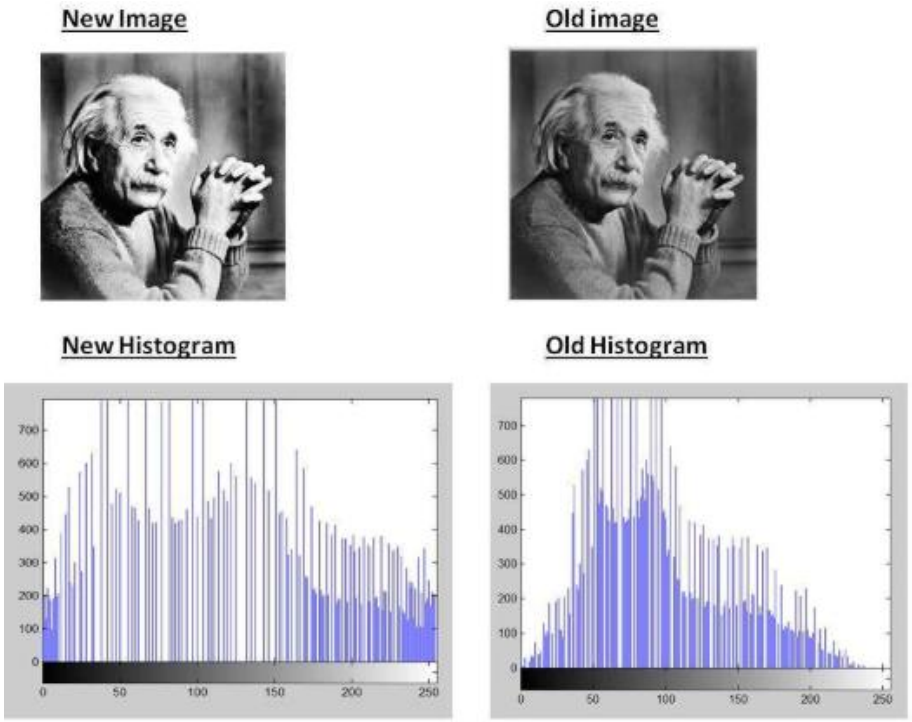
$$I_{negatif}(x, y) = 255 - I(x, y)$$

Negatif citra digunakan dalam analisis medis dan fotografi.

**c. Thresholding**
Ambang mengubah gambar grayscale menjadi gambar biner:

$$I_{biner}(x, y) = \begin{cases} 1, & \text{jika } I(x, y) \geq T \\ 0, & \text{jika } I(x, y) < T \end{cases}$$

Thresholding digunakan dalam segmentasi objek dan deteksi tepi.




Gambar 8. Contoh Operasi Titik

Gambar 8 memperlihatkan contoh penerapan operasi titik (point operation) dalam pengolahan citra digital. Operasi titik merupakan teknik dasar di mana transformasi nilai intensitas dilakukan secara langsung pada setiap piksel citra, tanpa mempertimbangkan hubungan dengan piksel tetangga. Artinya, nilai keluaran sebuah piksel hanya ditentukan oleh nilai masukannya melalui fungsi transformasi tertentu.

Pada bagian atas, diperlihatkan perbandingan antara citra lama (old image) dan citra baru (new image) setelah melalui operasi titik. Citra hasil transformasi tampak memiliki tingkat kontras yang lebih seimbang, sehingga detail wajah dan tekstur lebih jelas terlihat dibandingkan citra aslinya.

Bagian bawah menunjukkan perbandingan histogram dari kedua citra. Old histogram memperlihatkan distribusi intensitas yang tidak merata, dengan konsentrasi piksel pada rentang tertentu sehingga menyebabkan citra terlihat kurang optimal dari segi kontras. Setelah dilakukan operasi titik, new histogram menjadi lebih tersebar merata di sepanjang rentang intensitas. Hal ini menunjukkan bahwa nilai piksel telah ditransformasi sehingga mencakup rentang kecerahan yang lebih luas, yang berdampak pada peningkatan kualitas visual citra.
Beberapa jenis operasi titik yang umum digunakan antara lain:

Kontras stretching, untuk memperlebar rentang intensitas dan menonjolkan perbedaan terang-gelap.

- Thresholding, untuk memisahkan objek dari latar berdasarkan ambang batas tertentu.
- Negasi citra, untuk membalikkan nilai intensitas sehingga area terang menjadi gelap dan sebaliknya.
- Logarithmic dan power-law transformation, untuk menyesuaikan distribusi intensitas sesuai kebutuhan aplikasi tertentu.

Dengan demikian, gambar ini menegaskan bahwa operasi titik berperan penting sebagai langkah awal dalam pengolahan citra digital. Teknik ini tidak hanya meningkatkan kualitas tampilan citra secara visual, tetapi juga mempermudah proses analisis lanjutan seperti segmentasi, deteksi tepi, maupun pengenalan pola.

### 3.2.3 Operasi Spasial (Spatial Operations)

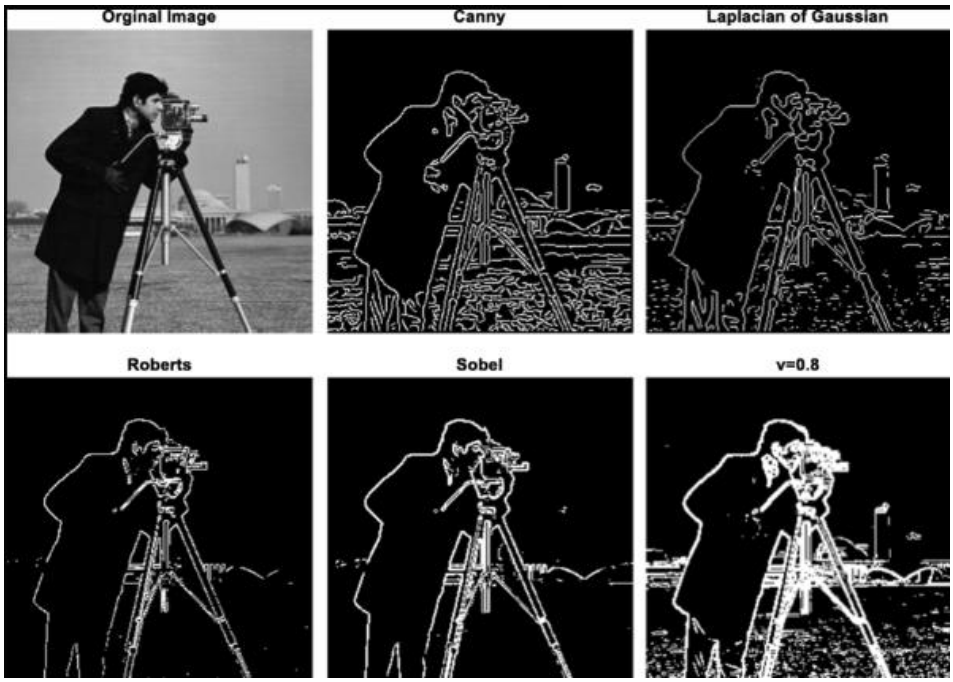
Operasi spasial mempertimbangkan nilai piksel tetangga untuk memodifikasi nilai piksel target. Operasi ini termasuk filtering dan deteksi tepi.

#### a. Filtering

Filtering menghaluskan dan menajamkan citra. Contoh filter termasuk: Filter Median: Mengganti setiap piksel dengan nilai rata-rata tetangganya; Filter Gaussian: Menghaluskan gambar dengan menggunakan distribusi Gaussian;
dan Filter Median: Mengganti setiap piksel dengan nilai median tetangganya, yang efektif untuk menghilangkan suara.

#### b. Deteksi Tepi

Deteksi tepi menunjukkan batas objek dengan mendeteksi perubahan intensitas yang tajam dalam gambar. Operator umum termasuk:
Operator Sobel: Mendeteksi tepi horizontal dan vertikal menggunakan kernel;
Operator Prewitt: Mirip dengan Operator Sobel, tetapi menggunakan kernel yang berbeda; dan
Operator Tepi Canny: Metode multi-langkah yang menghasilkan deteksi tepi yang akurat.



Gambar 9. Contoh Deteksi Tepi

Gambar 9 menunjukkan beberapa contoh hasil deteksi tepi pada citra menggunakan berbagai operator yang berbeda. Setiap bagian gambar memperlihatkan bagaimana tepian objek dalam citra ditangkap dengan karakteristik yang unik sesuai dengan metode yang digunakan.

Pada bagian "Organ Image", terlihat hasil deteksi tepi yang halus dan detail, cocok untuk aplikasi medis atau analisis citra biologis. Bagian "Galaxy" menampilkan deteksi tepi pada citra astronomi, dengan pola-pola yang menyerupai struktur galaksi, menunjukkan kemampuan deteksi tepi dalam menangkap objek-objek kompleks dan berjarak jauh.

"Laplacian of Gaussian" menghasilkan tepian yang lebih halus dan teredam, mengurangi noise sekaligus menjaga detail penting. Sementara itu, operator "Roberts" dan "Sobel" masing-masing
menampilkan deteksi tepi dengan ketebalan dan ketajaman yang berbeda, di mana Sobel cenderung lebih halus dan Roberts lebih menonjolkan tepian yang kuat.

Terakhir, bagian yang diberi label "y=0.5" mungkin menunjukkan hasil deteksi tepi dengan suatu parameter atau threshold tertentu, yang menghasilkan tampilan tepian dengan tingkat kepercayaan atau
ketebalan yang berbeda. Gambar ini secara keseluruhan memberikan pemahaman visual yang jelas tentang bagaimana berbagai metode deteksi tepi bekerja dalam kondisi dan aplikasi yang beragam

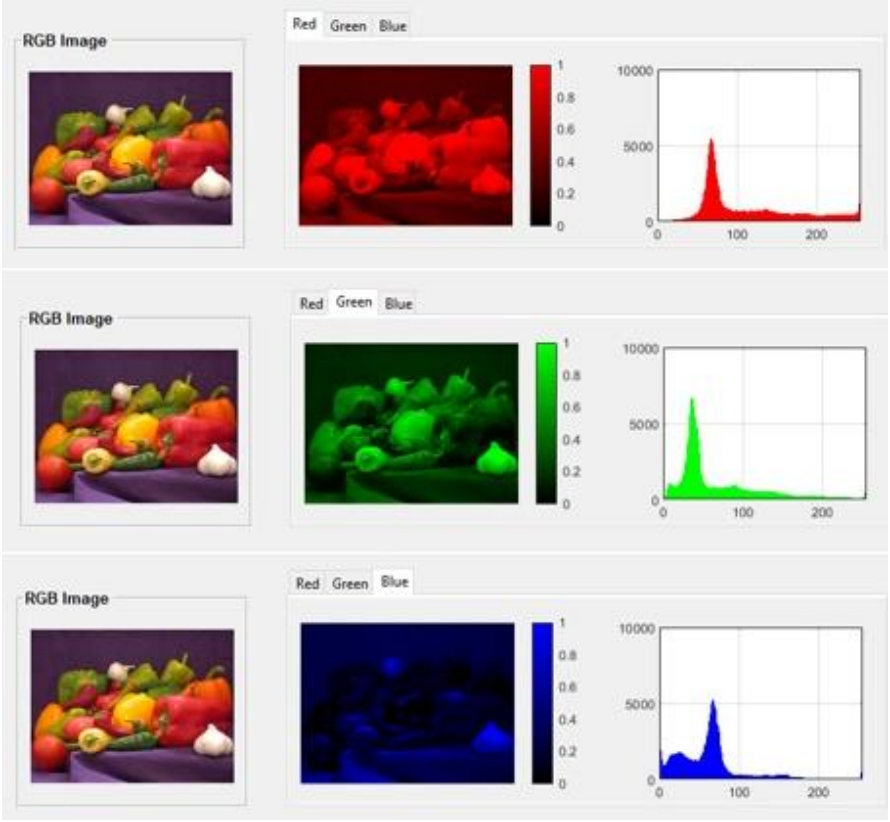
### 3.2.4 Transformasi Intensitas

Untuk meningkatkan kualitas visual atau mempersiapkan gambar untuk analisis lebih lanjut, transformasi intensitas mengubah distribusi nilai piksel dalam gambar.

**a. Histogram Equalization**

Histogram equalization meningkatkan kontras gambar karena menyebarkan distribusi intensitas secara merata.

**b. Logarithmic dan Power-Law Transformations**
Nilai intensitas di area terang atau gelap gambar dapat diperluas dengan menggunakan transformasi logaritmik dan hukum kekuatan.



Gambar 10. Contoh Transformasi Intensitas

Gambar 10 menunjukkan contoh transformasi intensitas yang diterapkan pada suatu citra. Pada gambar tersebut, terdapat tiga buah grafik yang masing-masing merepresentasikan distribusi intensitas untuk saluran warna merah, hijau, dan biru. Sumbu horizontal pada grafik menunjukkan nilai intensitas piksel
yang berkisar antara 0 hingga 200, sedangkan sumbu vertikal menunjukkan jumlah kemunculan atau frekuensi dari masing-masing nilai intensitas tersebut.

Dari ketiga grafik tersebut, dapat diamati bahwa setiap saluran warna memiliki karakteristik distribusi yang berbeda-beda. Grafik warna merah menunjukkan puncak distribusi yang lebih tinggi dibandingkan dengan kedua saluran lainnya, yang mengindikasikan bahwa warna merah mendominasi dalam citra
ini. Sementara itu, saluran hijau dan biru memiliki distribusi yang lebih rendah, dengan puncak yang tidak setinggi saluran merah.

Transformasi intensitas yang dilakukan bertujuan untuk mengubah nilai-nilai intensitas pada citra asli menjadi nilai intensitas baru sesuai dengan fungsi transformasi yang diinginkan. Proses ini dapat digunakan untuk meningkatkan
kualitas citra, seperti meningkatkan kecerahan, kontras, atau menyesuaikan keseimbangan warna. Hasil dari transformasi ini akan memengaruhi tampilan

visual citra, di mana distribusi intensitas pada masing-masing saluran warna akan berubah sesuai dengan fungsi transformasi yang diterapkan.

### 3.2.5 Transformasi Warna

Transformasi warna mengubah representasi warna gambar. Salah satu contohnya adalah perubahan dari RGB ke grayscale atau HSV.
Perubahan ini sangat penting untuk deteksi objek dan segmentasi warna.

**a. Konversi RGB ke Grayscale**

Konversi ini menggabungkan kanal merah, hijau, dan biru menjadi satu kanal intensitas:

$$I_{gray}(x, y) = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B$$

**b. Konversi RGB ke HSV**

Konversi ini memisahkan informasi warna (hue), kejenuhan (saturation), dan nilai (value), berguna dalam deteksi warna yang lebih stabil terhadap pencahayaan.


Gambar 11. Contoh Transformasi Warna

Gambar 11 memperlihatkan contoh hasil dari berbagai transformasi warna pada citra digital. Transformasi ini dilakukan untuk mengubah representasi gambar ke dalam bentuk yang lebih sesuai dengan kebutuhan analisis. Pada citra biner, gambar hanya terdiri dari dua nilai intensitas, yaitu hitam dan putih, yang biasanya dihasilkan melalui proses thresholding. Representasi ini sangat berguna untuk memisahkan objek utama dari latar belakang secara sederhana. Selanjutnya, citra asli ditampilkan dalam bentuk citra berwarna (RGB) yang merupakan tampilan umum dengan tiga kanal warna merah, hijau, dan biru. Transformasi berikutnya adalah citra grayscale (gray), di mana setiap piksel hanya memiliki satu nilai intensitas dari hitam hingga putih. Bentuk ini sering digunakan untuk mempermudah analisis dengan mengurangi kompleksitas informasi warna. Terakhir, ditunjukkan transformasi ke dalam ruang warna HSV (Hue, Saturation, Value) yang memisahkan informasi warna (hue), tingkat kejenuhan (saturation), dan kecerahan (value). Representasi HSV sangat berguna dalam

proses deteksi objek berbasis warna karena lebih stabil terhadap perubahan pencahayaan dibandingkan model RGB. Keempat contoh ini menggambarkan bagaimana sebuah citra dapat dimodifikasi ke dalam berbagai format untuk mendukung proses pemrosesan dan analisis visual pada computer vision.

# BAB IV
# Algoritma Vision Klasik vs Modern

## 4.1 Algoritma Klasik: Thresholding, Contour, dan Template Matching

Pendahuluan

Berbagai algoritma klasik telah menjadi dasar dalam analisis dan pemrosesan gambar sebelum dunia visi komputer didominasi oleh pendekatan berbasis pembelajaran mesin dan deep learning. Metode-metode ini jelas, matematis, dan tidak memerlukan data pelatihan seperti model kontemporer. Pengetahuan domain dan teknik pemrosesan gambar konvensional yang telah teruji selama bertahun-tahun menjadi dasar algoritma klasik tersebut.

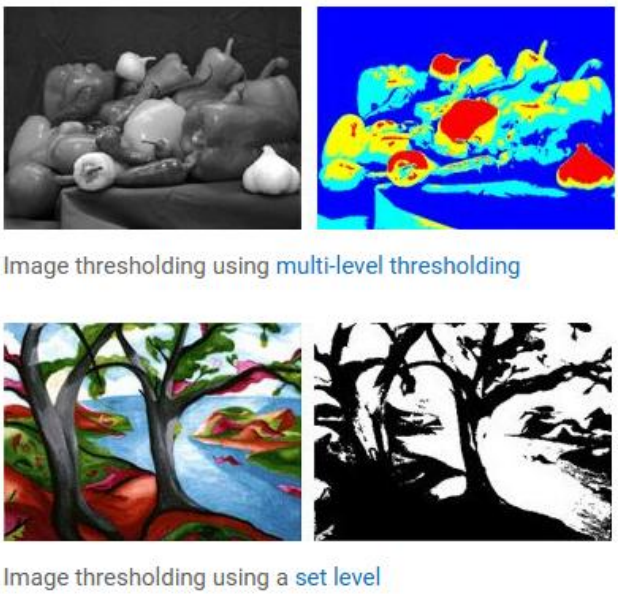
Algoritma vision klasik masih sangat penting untuk aplikasi yang membutuhkan efisiensi tinggi, interpretabilitas, dan sumber daya komputasi yang terbatas. Ini meskipun popularitasnya saat ini mulai menurun karena model pembelajaran berbasis data.

Thresholding, contour detection, dan template matching adalah tiga pilar utama algoritma visi klasik.

### 1. Thresholding

Metode segmentasi citra paling dasar, thresholding, digunakan untuk memisahkan objek dari latar belakang berdasarkan intensitas pikselnya. belakang tergantung pada apakah nilainya di atas atau di bawah ambang tersebut.

• Thresholding Global: Menggunakan satu nilai ambang tetap untuk seluruh gambar. Misalnya, metode Otsu secara otomatis menemukan ambang terbaik dengan mengurangi varians intra-kelas. • Adaptive Thresholding: Menyesuaikan ambang secara lokal berdasarkan statistik di sekitar tiap piksel, cocok untuk gambar dengan pencahayaan yang tidak merata.Prinsip utamanya adalah menentukan ambang, atau
ambang, dan mengkategorikan setiap piksel sebagai bagian dari objek atau latar




Gambar 12: Ilustrasi thresholding global dan adaptif

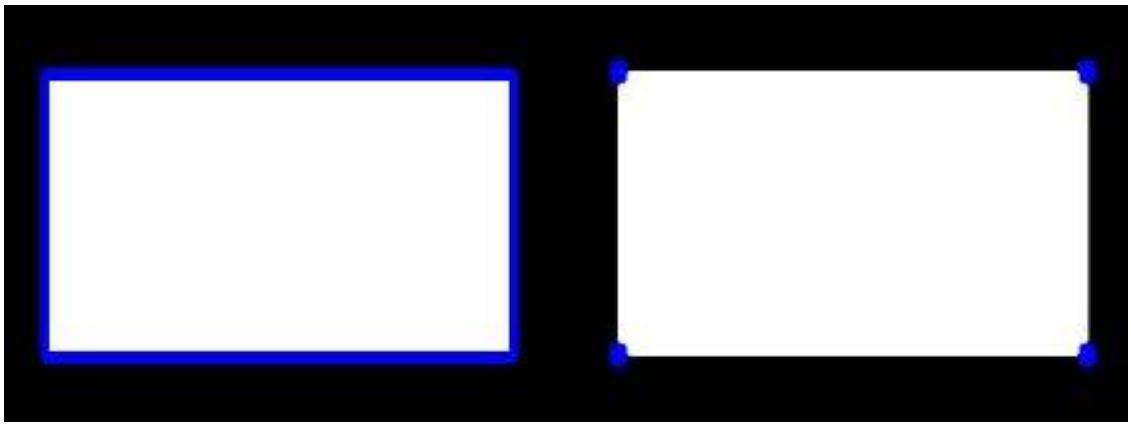
Thresholding pada gambar 12 sangat efektif untuk citra biner dan aplikasi industri seperti inspeksi kualitas produk.

**2. Deteksi Kontur (Contour Detection)**

Kurva yang menghubungkan titik-titik pada batas objek yang memiliki intensitas yang sama disebut kontur. Setelah thresholding, deteksi kontur sering digunakan untuk mengekstrak bentuk objek dan mengenalinya.

Fungsi findContours, yang dapat mengekstraksi kontur dari gambar biner, tersedia di OpenCV, salah satu pustaka komputer vision yang paling populer. Kontur dapat digunakan untuk mengukur luas, bentuk, dan orientasi objek, dan mengidentifikasi objek berdasarkan bentuk geometrisnya.

Berikut contoh ilustrasi hasil deteksi kontur:



Gambar 13. Hasil deteksi kontur pada objek berbentuk geometris

Deteksi kontur Gambar 13 sangat berguna dalam aplikasi pengenalan objek sederhana, pelacakan gerakan, serta pengukuran dimensi fisik objek di dunia nyata.

**3. Template Matching**

Pada gambar 14 terdapat Metode pencocokan gambar yang dikenal sebagai template matching digunakan untuk menentukan lokasi sebuah patch (template) dalam gambar yang lebih besar. Prinsipnya adalah menggunakan
teknik korelasi atau perbedaan piksel untuk membandingkan template dengan semua lokasi potensial dalam citra yang dimaksud.
Beberapa teknik yang umum digunakan adalah korelasi lintasan, yang mengukur kesamaan antara template dan subregion gambar. Korelasi normalisasi, yang menyesuaikan nilai korelasi dengan perbedaan intensitas.
Untuk mengidentifikasi objek dengan bentuk dan ukuran yang tetap, seperti karakter huruf, simbol, atau bagian mesin, template matching sangat berguna.

Contoh visualisasi:


Gambar14. Lokasi template yang cocok ditandai dengan kotak merah

Metode ini memiliki banyak kelebihan, seperti kesederhanaannya dan kemampuan untuk digunakan dalam sistem real-time tanpa pengalaman sebelumnya. Namun, metode ini sensitif terhadap perubahan dalam skala, rotasi, dan pencahayaan.

Kesimpulan: Algoritma klasik masih sangat penting untuk aplikasi dengan sumber daya terbatas atau prediksi yang transparan. Thresholding, contour detection, dan template matching adalah teknik dasar yang masih digunakan secara luas dalam berbagai bidang. Sebelum mempelajari pendekatan vision berbasis pembelajaran mesin yang lebih kompleks, pemahaman mendalam tentang algoritma ini juga sangat penting.

**4.2 Peralihan ke Machine Learning dalam Computer Vision**

Pendahuluan: Para peneliti dan praktisi mulai beralih dari pendekatan berbasis aturan eksplisit ke pendekatan berbasis data karena kebutuhan akan sistem penglihatan komputer yang lebih akurat, fleksibel, dan efisien meningkat. Ini adalah pergeseran dari algoritma penglihatan konvensional ke penggunaan machine learning (ML). Metode machine learning jauh lebih fleksibel dibandingkan metode konvensional karena memungkinkan sistem untuk belajar dari data dan memperbaiki performa secara bertahap.

Bagian ini akan membahas secara rinci bagaimana kemajuan teknologi mendorong penggunaan machine learning dalam sistem visi, menjelaskan perubahan paradigma dalam pendekatan pemrosesan gambar, dan membahas berbagai metode ML awal yang memainkan peran penting dalam pembentukan ekosistem visi kontemporer.

Alasan Peralihan dari Algoritma Klasik ke Algoritma Penginderaan Mesin Ada sejumlah faktor utama yang mendorong pergeseran dari
metode klasik ke pendekatan penginderaan mesin dalam kecerdasan buatan.

1. Keterbatasan Algoritma Klasik: Tidak dapat beradaptasi dengan variasi data, sulit diatur untuk lingkungan kompleks, dan sulit diintegrasikan antar domain.

2. Peningkatan Ketersediaan Data dan Daya Komputasi: Proliferasi kamera digital, Internet of Things, dan internet menghasilkan volume data citra/video yang sangat besar.
3. Peningkatan kemampuan GPU memungkinkan pelatihan model kompleks dalam waktu yang relatif singkat.
4. Persyaratan untuk Prediksi Non-Linear dan Kompleks: Tugas seperti deteksi wajah, pengenalan objek, atau klasifikasi gambar
5. memerlukan pemodelan hubungan yang kompleks yang tidak dapat dicapai oleh metode berbasis aturan eksplisit.

**Perkembangan Metode Machine Learning dalam Vision**

Computer vision telah menggunakan banyak teknik ML lama sebelum deep learning muncul. Berikut adalah beberapa metode yang paling umum:

**1. Support Vector Machines (SVM)**

SVM sering digunakan dalam klasifikasi fitur visual seperti:

• Ciri tekstur dan warna

• Deskriptor fitur seperti SIFT, HOG, atau SURF.

SVM bekerja dengan mencari hyperplane optimal yang memisahkan data ke dalam dua kelas contoh nya sepertipada gambar 15.


- Gambar 15. Sistem Pengenalan Wajah SVM

Pada gambar 15 sistem pengenalan wajah awal menggunakan HOG sebagai fitur dan SVM sebagai klasifikator.

**2. K-Nearest Neighbors (KNN)**
Algoritma non-parametrik ini menggunakan mayoritas label dari tetangga terdekat untuk mengklasifikasikan sampel. KNN masih digunakan dalam sistem data yang teratur meskipun mudah dipahami.

**3. Decision Trees dan Random Forest**
Model pohon keputusan membagi ruang fitur secara rekursif berdasarkan persyaratan tertentu. Penggabungan pohon acak meningkatkan stabilitas dan mengurangi overfitting.

**4. K-Means dan Clustering**
Algoritma clustering seperti K-Means digunakan dalam tugas segmentasi citra untuk mengelompokkan piksel berdasarkan kesamaan warna atau tekstur..

Tabel 3.1 berikut memperlihatkan perbandingan singkat algoritma ML klasik dalam vision:

| Algoritma | Tipe Masalah | Kelebihan | Kelemahan |
|---|---|---|---|
| SVM | Klasifikasi | Akurat, bekerja baik pada data kecil | Tidak efisien di dataset besar |
| KNN | Klasifikasi | Sederhana, tanpa pelatihan | Lambat saat prediksi |
| Decision Tree | Klasifikasi/Regresi | Mudah diinterpretasi | Cenderung overfit |
| K-Means | Clustering | Efisien untuk segmentasi | Butuh inisialisasi yang baik |

pada table 3.1 memaparkan perbandingan empat algoritma machine learning inti yang biasa diterapkan dalam computer vision. SVM (Support Vector Machine) unggul dalam menangani tugas klasifikasi dengan tingkat akurasi yang tinggi dan dapat bekerja secara efektif bahkan dengan volume data yang relatif kecil. Namun, kelemahan utama SVM terletak pada efisiensinya yang menurun drastis ketika dihadapkan pada dataset yang sangat masif. KNN (K-Nearest Neighbors), algoritma klasifikasi lain, terkenal karena kesederhanaan konsepnya yang hanya membandingkan kemiripan data dan tidak memerlukan fase pelatihan yang eksplisit. Akan tetapi, kemudahan ini dibayar dengan kecepatan prediksi yang lambat karena ia harus menganalisis seluruh data yang ada untuk setiap prediksi baru.

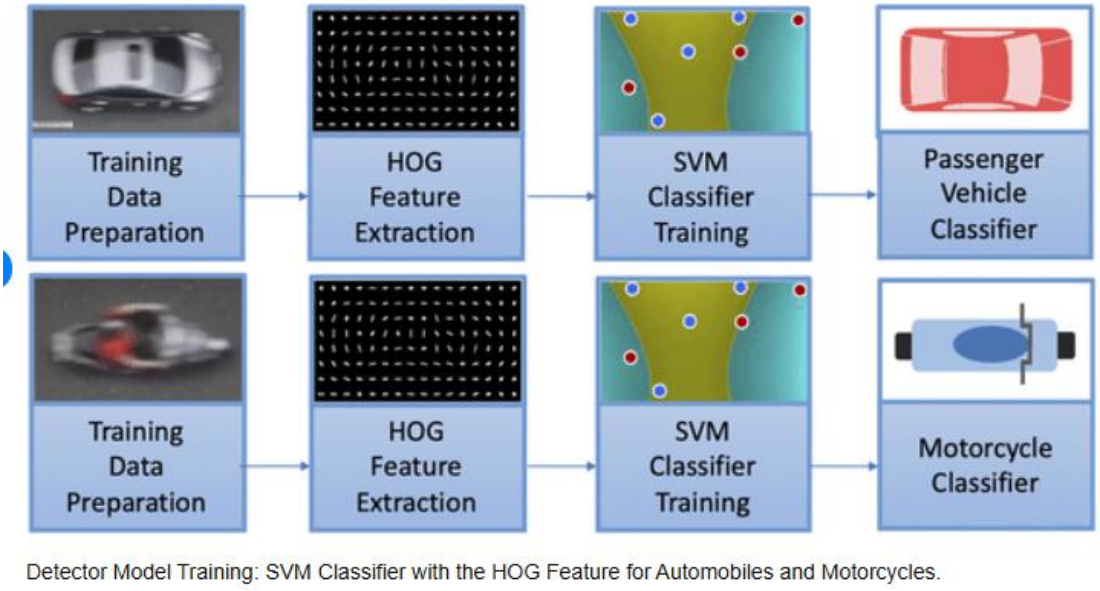
Sementara itu, Decision Tree (Pohon Keputusan) menawarkan fleksibilitas dengan mampu menangani masalah klasifikasi dan regresi. Keunggulan terbesarnya adalah kemudahan untuk diinterpretasikan oleh manusia, membuatnya transparan seperti diagram alur. Namun, model ini memiliki kecenderungan alami untuk overfitting, yaitu menjadi terlalu spesifik terhadap data latih sehingga performanya pada data baru sering kali mengecewakan. Terakhir, K-Means adalah algoritma andalan untuk tugas clustering atau pengelompokan tanpa pengawasan, seperti dalam segmentasi gambar. Algoritma ini sangat efisien dan cepat, tetapi hasil

pengelompokannya sangat bergantung pada inisialisasi atau penentuan titik awal yang tepat. Jika inisialisasinya buruk, maka kualitas cluster yang dihasilkan pun akan tidak optimal.

**Contoh Aplikasi Awal**

Beberapa aplikasi awal kecerdasan buatan dalam Computer Vision termasuk:

- OCR (Optical Character Recognition): Menggunakan SVM untuk klasifikasi hasil segmentasi karakter;
- •Face Recognition: PCA (Principal Component Analysis) digunakan untuk mengekstraksi fitur wajah, kemudian diklasifikasikan dengan SVM atau KNN; dan
- Deteksi Kendaraan: Menggunakan HOG + SVM untuk mengidentifikasi mobil dalam video lalu lintas contoh nya pada gambar 16.
-




Citra 16. Deteksi kendaraan dengan fitur Histogram of Oriented Gradients (HOG) dan SVM

Tantangan dalam Adopsi Awal

Integrasi awal pembelajaran mesin ke penglihatan memiliki beberapa kendala. Beberapa di antaranya adalah sebagai berikut:

- Ekstraksi fitur masih dilakukan secara manual: kinerja model sangat bergantung pada kemampuan engineer dalam merancang fitur.
- Sulit untuk menangani variasi dan suara yang kompleks: misalnya, pencahayaan tidak seragam, rotasi objek, atau latar belakang yang kompleks.
- Bergantung pada pra-pemrosesan yang luas: normalisasi, segmentasi awal, dan penghilangan suara sangat penting untuk performa akhir.

**Transisi Menuju Deep Learning**

Keterbatasan ini akhirnya memungkinkan penggunaan deep learning, yang menghilangkan kebutuhan untuk mengekstrak fitur secara manual. Neural network dalamnya meningkatkan akurasi dan fleksibilitas dengan belajar representasi langsung dari data mentah.

Proses ini menggabungkan prinsip ML klasik dengan teknik modern, bukan pengganti langsung. Banyak sistem terus menggunakan ML klasik sebagai dasar atau modul bantu di awal pipeline sebelum proses deep learning.

Kesimpulan: Pergeseran dari algoritma penglihatan klasik ke pendekatan pengajaran komputer merupakan tahap penting dalam evolusi penglihatan komputer. Metode seperti Decision Tree, SVM, dan KNN meningkatkan kemampuan sistem visi untuk mendeteksi pola kompleks dan menyesuaikannya dengan berbagai situasi dunia nyata. Dalam bagian berikutnya, kami akan membahas bagaimana deep learning—sebuah jenis lanjutan dari pembelajaran mesin—mengubah pengolahan visual secara signifikan dan meningkatkan cakupan aplikasi visi ke tingkat yang lebih tinggi.

### 4.3 Perbandingan Pendekatan Tradisional dan Deep Learning

Ada dua pendekatan utama untuk mengembangkan visi komputer. Yang pertama adalah pendekatan tradisional, yang bergantung pada fitur dan aturan eksplisit; yang kedua adalah pendekatan modern, yang bergantung pada pembelajaran mendalam. Pilihan metode yang tepat sangat bergantung pada kompleksitas tugas, ketersediaan data, dan kebutuhan sistem untuk beroperasi. Tujuan dari bagian ini adalah untuk membandingkan kedua metode tersebut dari berbagai sudut pandang. Ini termasuk arsitektur, kebutuhan data, fleksibilitas, performa, dan interpretabilitas.

**Karakteristik Pendekatan Tradisional**
Metode tradisional terdiri dari dua langkah utama: ekstraksi fitur secara manual dan penggunaan algoritma konvensional untuk klasifikasi. Pendekatan ini memiliki beberapa ciri khas, seperti berikut:

- Ekstraksi Fitur Tersurat: Metode seperti SIFT, HOG, dan SURF digunakan untuk mendapatkan informasi tentang bentuk, tekstur, atau warna.
- Penggunaan Algoritma Klasik: Setelah fitur diekstraksi, algoritma seperti SVM, KNN, atau Decision Tree digunakan.
- Kebutuhan Pra-pemrosesan Tinggi: Langkah-langkah pra-pemrosesan seperti segmentasi, normalisasi, dan pengurangan suara sangat penting.

**Keunggulan:** Mudah difahami, tidak membutuhkan banyak sumber daya komputasi, dan efektif untuk tugas-tugas sederhana dan dalam domain terbatas.

Keterbatasan: Sulit menangani kompleksitas visual yang tinggi; sangat bergantung pada kemampuan desain fitur; dan tidak fleksibel dengan variasi data baru

**Karakteristik Pendekatan Deep Learning**
Bidang visi telah berubah karena pendekatan deep learning yang menggunakan Convolutional Neural Networks (CNN). CNN memiliki kemampuan untuk mengekstrak fitur secara otomatis dan membuat representasi hirarkis dari data visual.

**Keunggulan:** • Pembelajaran End-to-End: Anda tidak perlu mendesain fitur secara manual. Skalabilitas tinggi menunjukkan kemampuan untuk menangani jutaan data dengan variasi kompleks. Performa tinggi menunjukkan kemampuan untuk mengalahkan metode tradisional dalam deteksi objek, segmentasi, dan klasifikasi gambar.

**Keterbatasan:** Kebutuhan Data Besar: Membutuhkan dataset berlabel dalam jumlah besar; Komputasi Intensif: Pelatihan memerlukan GPU dan waktu yang lama; dan Kurang Interpretabel: Sulit untuk menjelaskan alasan keputusan model kotak hitam.

**Perbandingan Berdasarkan Dimensi Utama**

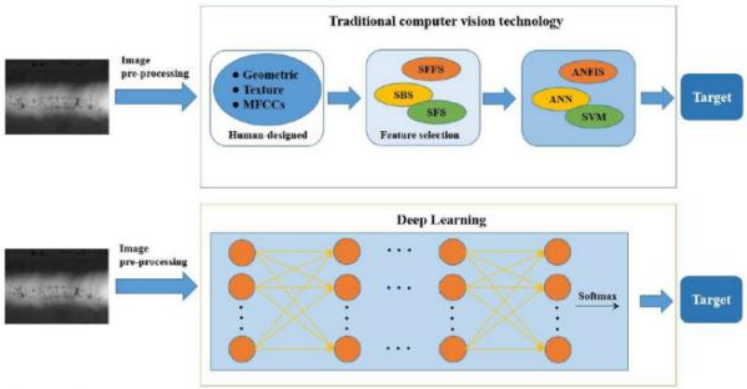
Tabel 3.2 berikut menunjukkan perbandingan menyeluruh antara pendekatan tradisional dan deep learning:

| Aspek | Pendekatan Tradisional | Deep Learning |
|---|---|---|
| Ekstraksi Fitur | Manual (HOG, SIFT, dll.) | Otomatis (melalui CNN) |
| Interpretabilitas | Tinggi (mudah ditelusuri) | Rendah (black box) |
| Kebutuhan Data | Relatif kecil | Sangat besar |
| Ketergantungan Pra-proses | Tinggi | Minimal |
| Kinerja pada tugas kompleks | Terbatas | Sangat tinggi |
| Sumber daya komputasi | Rendah | Tinggi (butuh GPU) |

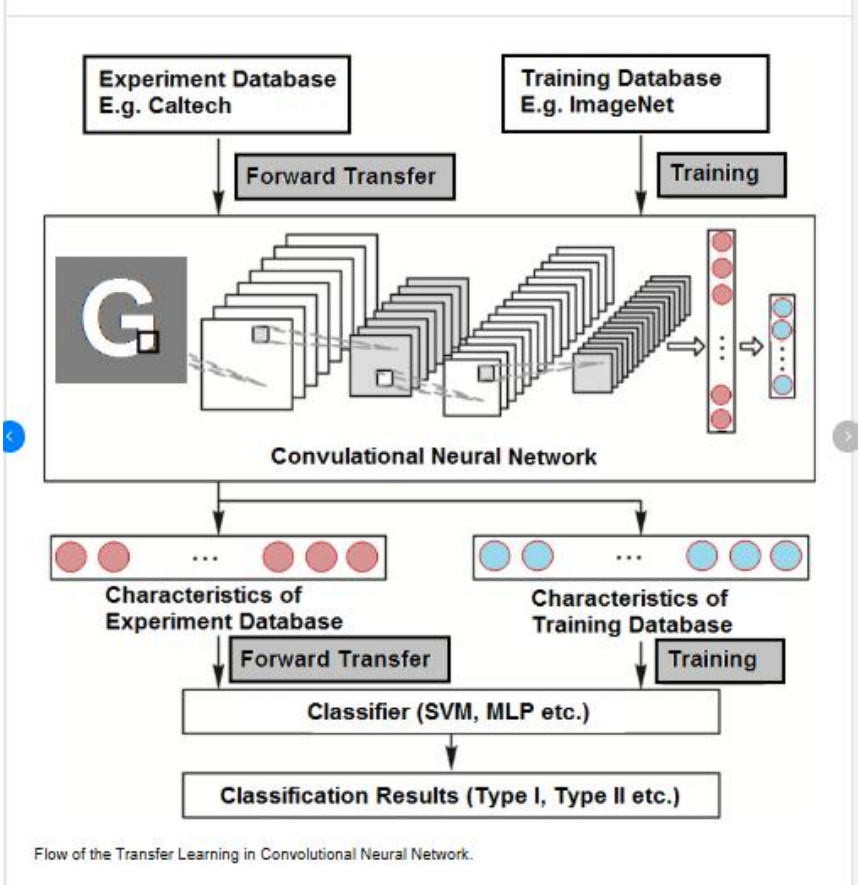
Pada table 3.2 Perbandingan antara pendekatan tradisional dan deep learning menunjukkan perbedaan yang sangat mendasar dalam filosofi dan caranya bekerja. Dalam pendekatan tradisional, para ahli harus secara manual merancang dan mengekstraksi fitur-fitur dari data menggunakan teknik tertentu, sebuah proses yang membutuhkan banyak waktu dan keahlian. Keuntungannya, langkah-langkahnya sangat transparan dan mudah untuk ditelusuri alasan di balik sebuah keputusan. Pendekatan ini juga bisa bekerja dengan baik meski data yang tersedia tidak terlalu banyak dan tidak memerlukan perangkat komputasi yang sangat canggih. Namun, kemampuannya sering kali terbatas ketika menghadapi masalah yang sangat rumit dan kompleks.

Di sisi lain, deep learning mengambil jalan yang berlawanan dengan mengotomatisasi proses yang paling rumit tersebut. Jaringan neural dalam deep learning, seperti CNN, mampu belajar dan mengekstraksi fitur-fitur penting secara langsung dari data mentah, sehingga sangat mengurangi ketergantungan pada pra-pemrosesan data yang rumit. Kehebatan utama deep learning adalah kemampuannya untuk mencapai kinerja yang sangat tinggi pada tugas-tugas yang sangat kompleks, yang sebelumnya mustahil ditangani. Akan tetapi, kehebatan ini datang dengan harga. Deep learning memerlukan jumlah data yang sangat besar untuk belajar, membutuhkan sumber daya komputasi yang masif seperti GPU, dan modelnya sering dianggap sebagai "kotak hitam" karena sangat sulit untuk memahami alasan spesifik di balik prediksi yang dihasilkannya.

**Visualisasi Perbandingan**




Gambar 17. Diagram yang mengilustrasikan alur pendekatan tradisional vs deep learning:




Gambar 18. CNN Workflow:

**Studi Kasus Perbandingan**

Sebagai ilustrasi terlihat pada gambar 17 dan gambar 18 , studi kasus pengenalan kendaraan pada citra lalu lintas:

- Metode Tradisional: HOG digunakan untuk mendeskripsikan bentuk mobil → diklasifikasi dengan SVM.
- Deep Learning: CNN dilatih end-to-end dari gambar ke label tanpa desain fitur manual.

CNN secara konsisten lebih baik daripada kombinasi HOG + SVM pada dataset besar seperti COCO atau ImageNet. Ini terutama berlaku dalam situasi kompleks seperti rotasi objek, skala bervariasi, dan latar belakang berisik.

**Kesimpulan**

Pendekatan deep learning menawarkan fleksibilitas dan akurasi, terutama dalam lingkungan berskala besar dan kompleks. Sebaliknya, pendekatan tradisional vision menawarkan efisiensi dan interpretabilitas pada tugas-tugas yang lebih sederhana dan lingkungan yang lebih terbatas. Masing-masing memiliki tempatnya sendiri dan berfungsi untuk aplikasinya. Bahkan dalam sistem hybrid, kombinasi keduanya sering digunakan. Salah satu contohnya adalah penggunaan preprocessing konvensional sebelum input

# BAB V
# Arsitektur Deep Learning dalam Computer Vision

## 5.1 Convolutional Neural Network (CNN)

Pendahuluan: Mengapa CNN?

Convolutional Neural Network (CNN) dibangun untuk meniru cara kerja sistem visual biologis, terutama cara otak manusia melakukan

pengolahan visual, yang menyusun dan mengekstraksi informasi spasial secara hierarkis. Konsep dasar CNN adalah bahwa informasi

visual memiliki struktur spasial dan hierarkis, yang merupakan dasar dari pengolahan citra berbasis deep learning.Struktur Umum CNN

CNN terdiri dari beberapa jenis lapisan utama yang membentuk arsitektur dasarnya:

**Convolutional Layer**

Ini adalah tempat fitur lokal diekstraksi. Dalam gambar, filter (atau kernel) diterapkan untuk membuat peta fitur yang mewakili pola seperti tepi, sudut, tekstur, dan objek kompleks di lapisan yang lebih dalam.

**Activation Function (ReLU)**

Agar jaringan dapat memodelkan hubungan yang kompleks, fungsi aktivasi seperti ReLU menambahkan non-linearitas.

**Pooling Layer (Subsampling)**

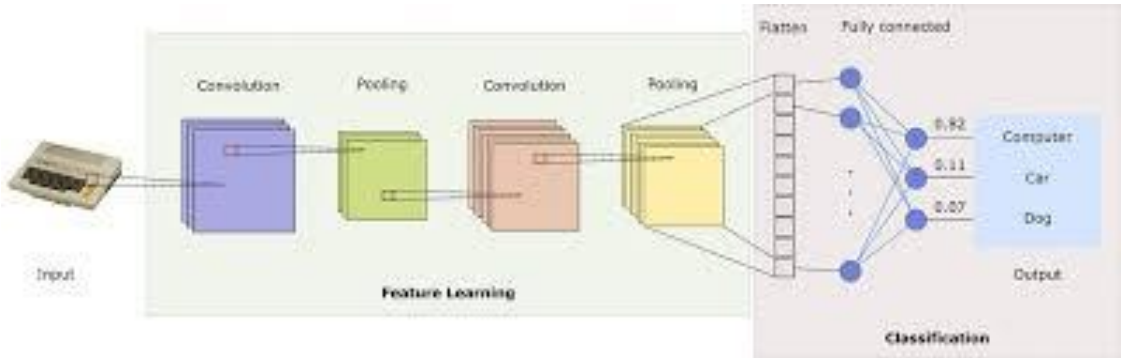
Pooling mengurangi dimensi spasial feature maps, yang membuat representasi lebih efisien dan tahan terhadap distorsi kecil. Max Pooling adalah metode yang paling umum digunakan.Fully .

**Connected Layer (FC)**

Pada gambar 20 Dengan menggunakan fungsi seperti softmax, lapisan akhir CNN diubah menjadi bentuk vektor, yang kemudian diklasifikasikan oleh lapisan penuh terhubung.

**Output Layer**

Lapisan ini menghasilkan probabilitas klasifikasi untuk setiap label target.




Gambar 20 Arsitektur CNN Umum

Gambar 20 menggambarkan arsitektur umum Convolutional Neural Network (CNN), yang merupakan salah satu pendekatan paling populer dalam bidang computer vision. Proses kerja CNN secara garis besar terdiri dari dua tahap utama, yaitu feature learning dan classification.

1) **Input**
   Tahap awal dimulai dengan pemberian citra masukan (input). Citra ini bisa berupa gambar objek nyata, misalnya foto komputer, mobil, atau anjing seperti yang ditunjukkan pada ilustrasi.
2) **Feature Learning**
   Pada tahap ini, CNN melakukan proses ekstraksi ciri (feature extraction) dari citra. Ekstraksi dilakukan melalui dua jenis lapisan utama:
   Convolution Layer: berfungsi untuk mengekstrak fitur dasar dari citra, seperti garis, tepi, atau tekstur. Proses konvolusi menggunakan filter atau kernel yang bergerak di seluruh citra untuk menghasilkan peta fitur (feature map).
   Pooling Layer: bertugas mereduksi dimensi dari peta fitur sekaligus mempertahankan informasi yang paling penting. Jenis pooling yang sering digunakan adalah max pooling, yaitu memilih nilai maksimum dari suatu area tertentu.
   Proses konvolusi dan pooling biasanya dilakukan berulang kali dalam beberapa lapisan, sehingga jaringan mampu membangun representasi fitur yang semakin kompleks, mulai dari fitur rendah (edge, garis) hingga fitur tingkat tinggi (bentuk objek tertentu).
3) **Classification**
   Setelah melewati beberapa lapisan konvolusi dan pooling, hasil akhirnya diratakan (flattening) menjadi vektor satu dimensi. Vektor ini kemudian diproses pada lapisan fully connected (dense layer) untuk melakukan klasifikasi.
   Setiap node pada lapisan ini terhubung penuh dengan node pada lapisan sebelumnya.
   Proses ini mirip dengan jaringan saraf tiruan klasik, di mana bobot dan bias digunakan untuk menentukan keluaran.

Akhirnya, lapisan output menghasilkan probabilitas untuk setiap kelas yang telah ditentukan sebelumnya. Misalnya, jika citra input adalah gambar hewan, maka jaringan akan memberikan skor probabilitas terhadap kategori "komputer", "mobil", atau "anjing". Kategori dengan nilai probabilitas tertinggi dipilih sebagai hasil prediksi akhir.

### 5.2 Proses Pelatihan CNN

Dua tahap utama dalam proses pelatihan CNN adalah forward pass dan backpropagation. Tahap forward pass menghasilkan prediksi dari data masukan (gambar) yang diproses melalui berbagai lapisan untuk
menghasilkan prediksi. Tahap backpropagation menghitung perbedaan antara prediksi dan label asli (kehilangan) dan digunakan untuk
memperbarui bobot jaringan dengan menggunakan algoritma optimisasi seperti Stochastic Gradient Descent (SGD) atau Adam.

**Aplikasi CNN dalam Computer Vision**

Table 5.1 Penggunaan CNN dalam berbagai tugas computer vision:

| Tugas | Penjelasan Singkat |
|---|---|
| Klasifikasi Gambar | Mengklasifikasikan gambar ke dalam satu kategori |
| Deteksi Objek | Menemukan dan memberi label objek dalam gambar |
| Segmentasi Citra | Membagi citra menjadi wilayah dengan makna semantik |
| Pengenalan Wajah | Mencocokkan wajah individu berdasarkan fitur deep |
| Analisis Medis | Deteksi kanker, penyakit kulit, dan lainnya dari gambar medis |

Pada tabel 5.1 adalah penjelasan tentang Jaringan saraf convolutional (CNN) yang memiliki peran sangat penting dan serbaguna dalam bidang computer vision, dengan aplikasi yang menyentuh berbagai aspek kehidupan. Kemampuan utamanya dimulai dari Klasifikasi Gambar, dimana sebuah CNN dapat melihat sebuah gambar dan secara akurat menggolongkannya ke dalam satu kategori tertentu, seperti membedakan gambar kucing dari anjing.

Lebih dari sekadar memberi label pada seluruh gambar, CNN juga mampu untuk Deteksi Objek. Di sini, tugasnya menjadi lebih detail, yaitu untuk menemukan lokasi semua objek yang menarik dalam sebuah gambar dan melingkari mereka sambil memberi label pada setiap objek yang ditemukan. Kemampuan ini ditingkatkan lagi melalui Segmentasi Citra, yang bertindak seperti memberikan "peta" yang sangat detail untuk sebuah gambar. Teknik ini membagi setiap pixel dalam gambar ke dalam wilayah atau kategori yang berbeda, misalnya memisahkan jalan, mobil, dan pejalan kaki dalam sebuah foto lalu lintas.

Aplikasi praktis yang sangat populer adalah dalam Pengenalan Wajah, dimana CNN digunakan untuk mengidentifikasi atau memverifikasi identitas seseorang dengan menganalisis fitur-fitur wajah yang unik dan kompleks. Akhirnya, di bidang yang sangat kritis yaitu Analisis Medis, CNN menunjukkan nilai yang luar biasa dengan membantu tenaga medis menganalisis gambar seperti foto rontgen atau pemindaian MRI untuk mendeteksi tanda-tanda penyakit, termasuk kanker dan berbagai
kondisi kulit, sehingga membantu dalam proses diagnosis yang lebih cepat dan akurat.

Tabel 5.2 Kelebihan dan Kekurangan CNN

| Aspek | Kelebihan | Kekurangan |
|---|---|---|
| Akurasi | Tinggi dalam klasifikasi dan deteksi | Membutuhkan data latih yang besar |
| Fleksibilitas | Dapat diadaptasi untuk berbagai tugas vision | Arsitektur dan hiperparameter sering butuh eksperimen |
| Interpretasi | Visualisasi fitur membuat interpretasi dimungkinkan | Masih kurang transparan dibanding metode statistik klasik |
| Efisiensi | Ekstraksi fitur otomatis | Komputasi tinggi, memerlukan GPU |

Pada table 5.2 Convolutional Neural Networks (CNN) telah menjadi tulang punggung dalam dunia computer vision, menawarkan sejumlah keunggulan sekaligus tantangan. Dari segi akurasi, CNN dikenal sangat handal dan unggul dalam tugas-tugas seperti mengklasifikasikan gambar atau mendeteksi objek, sering kali melampaui kemampuan manusia. Namun, untuk

mencapai tingkat ketepatan yang demikian, CNN memerlukan 'bahan bakar' yang sangat besar, yaitu volume data latih yang masif untuk dipelajari.

Fleksibilitas adalah nilai jual lainnya; arsitektur CNN dapat dimodifikasi dan disesuaikan untuk menyelesaikan beragam masalah, mulai dari yang sederhana hingga yang sangat kompleks. Akan tetapi, untuk menemukan konfigurasi yang optimal, diperlukan eksperimen yang cukup panjang dan trial-and-error dalam menentukan struktur jaringan dan pengaturan hyperparameter-nya.

Meskipun terdapat teknik visualisasi fitur yang membantu kita memahami apa yang 'dilihat' oleh model, interpretasi atau kejelasan alasan di balik suatu keputusan tetap menjadi titik lemah CNN. Model ini masih dianggap kurang transparan dan lebih sulit dipahami dibandingkan dengan metode statistik tradisional yang lebih sederhana.

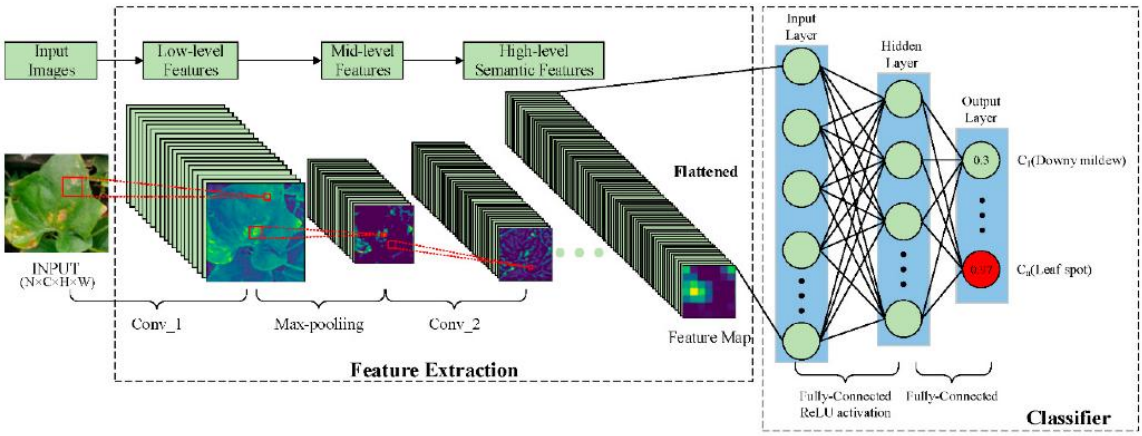
Akhirnya, kelebihan utama CNN adalah kemampuannya untuk mengekstraksi fitur secara otomatis langsung dari data mentah, menghilangkan kebutuhan untuk proses manual yang memakan waktu. Namun, kemewahan ini datang dengan biaya komputasi yang sangat tinggi, mengharuskan penggunaan perangkat keras khusus seperti GPU untuk melatih modelnya dalam waktu yang wajar.

### 5.3 Studi Kasus: CNN untuk Deteksi Penyakit Daun

CNN dapat mendeteksi penyakit pada tanaman, seperti daun sawi atau tomat, yang merupakan salah satu aplikasi menariknya. CNN memiliki
kemampuan untuk membedakan pola warna dan tekstur yang sangat halus, yang biasanya tidak terlihat oleh manusia.

**PlantVillage** adalah contoh dataset yang menyediakan ribuan gambar berlabel yang menunjukkan berbagai kondisi tanaman. CNN dapat
dilatih untuk mengenali bercak, warna yang tidak biasa, dan kerusakan tekstur daun.

**Visualisasi karakteristik CNN**: CNN dapat "melihat" berbagai tingkat informasi. Layer awal mengidentifikasi tepi dan garis, layer
tengah mengidentifikasi bagian objek, seperti batang atau daun, dan layer akhir memodelkan representasi abstrak.




Gambar 21. CNN mendeteksi Penyakit Daun

Gambar 21 menggambarkan alur kerja Convolutional Neural Network (CNN) dalam mendeteksi penyakit daun.

**Input Image**

Proses dimulai dari citra paru-paru yang dimasukkan ke dalam sistem. Citra ini masih berupa data mentah yang mengandung informasi kompleks mengenai struktur organ.

**Feature Extraction**

CNN secara otomatis mengekstraksi fitur melalui beberapa lapisan:

- Low-level features (Convolution Layer 1): menangkap pola dasar seperti tepi, garis, dan tekstur sederhana.
- Mid-level features (Setelah Max-pooling dan Convolution Layer 2): mengidentifikasi bentuk dan pola yang lebih kompleks, seperti area bercak atau lesi pada daun
- High-level semantic features: merepresentasikan informasi abstrak yang berkaitan dengan ciri khas penyakit daun

**Flattened Layer**

Hasil ekstraksi fitur berupa peta fitur multidimensi kemudian diratakan (flattening) agar dapat diproses lebih lanjut oleh lapisan fully connected.

**Classifier (Fully Connected Layers)**

Lapisan fully connected bertugas mengklasifikasikan citra berdasarkan fitur yang telah diekstraksi. Proses ini menghasilkan probabilitas diagnosis, misalnya normal atau terindikasi penyakit daun

**Output Layer**

Bagian akhir berupa label kelas yang menunjukkan kondisi pasien, didukung oleh fungsi aktivasi (misalnya Softmax atau Sigmoid) yang memberikan skor kepercayaan untuk setiap kategori.

# BAB VI
# Arsitektur Deep Learning Populer dalam Computer Vision

## Pendahuluan

Perjalanan perkembangan computer vision tidak dapat dipisahkan dari evolusi algoritma yang mendukungnya. Pada tahap awal, metode yang digunakan masih mengandalkan ekstraksi fitur secara manual, misalnya menggunakan SIFT (Scale Invariant Feature Transform) atau HOG (Histogram of Oriented Gradients). Teknik ini cukup berhasil untuk kasus sederhana seperti pendeteksian tepi atau pengenalan bentuk dasar, tetapi mulai menunjukkan keterbatasan ketika dihadapkan pada dataset yang besar, kompleks, dan beragam.

Keterbatasan utama pendekatan klasik terletak pada ketergantungan terhadap keahlian manusia dalam merancang fitur. Setiap domain aplikasi, baik itu medis, pertanian, maupun transportasi, membutuhkan desain fitur yang berbeda. Proses ini memakan waktu dan tidak selalu memberikan hasil yang optimal. Situasi tersebut mendorong lahirnya pendekatan baru yang mampu belajar langsung dari data tanpa campur tangan manusia secara berlebihan.

Di sinilah Convolutional Neural Network (CNN) memainkan peran sentral. CNN tidak hanya mengubah cara kita mengekstraksi fitur visual, tetapi juga membuka paradigma baru dalam pembelajaran mesin: sebuah sistem yang mampu end-to-end learning, dari input citra mentah hingga menghasilkan prediksi akhir. CNN belajar secara hierarkis; lapisan awal mengenali pola sederhana seperti garis dan sudut, lapisan menengah mendeteksi tekstur atau bagian objek, dan lapisan akhir memahami bentuk kompleks seperti wajah, kendaraan, atau daun tanaman.

Popularitas CNN meningkat pesat setelah kemenangan AlexNet pada kompetisi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) tahun 2012. Model ini secara drastis menurunkan tingkat kesalahan klasifikasi citra dibandingkan metode sebelumnya. Keberhasilan tersebut menjadi penanda dimulainya "era deep learning" dalam computer vision. Setelah itu, lahirlah berbagai arsitektur baru seperti VGGNet, ResNet, dan EfficientNet, yang masing-masing menyumbangkan inovasi dalam kedalaman jaringan, stabilitas pelatihan, serta efisiensi komputasi.

Penerapan CNN kini tidak lagi terbatas pada penelitian akademik, melainkan telah merambah ke berbagai bidang praktis. Dalam kesehatan, CNN digunakan untuk mendeteksi tumor otak melalui MRI. Di sektor pertanian, CNN membantu mengidentifikasi penyakit daun padi secara dini. Pada transportasi, CNN berperan dalam sistem kendaraan otonom untuk mengenali rambu lalu lintas dan pejalan kaki secara real-time. Keberhasilan tersebut menunjukkan bahwa CNN bukan sekadar model teoretis, melainkan solusi nyata yang berdampak langsung pada kehidupan manusia.

Meskipun demikian, CNN bukan tanpa kelemahan. Ketergantungannya pada dataset besar, kebutuhan perangkat keras berperforma tinggi, serta sifatnya yang sulit dijelaskan (black box) menimbulkan tantangan baru. Namun, keterbatasan ini juga menjadi titik awal bagi lahirnya inovasi lain, seperti model deteksi real-time YOLO, arsitektur Mask R-CNN untuk segmentasi, dan belakangan Vision Transformer yang menggabungkan konsep dari pemrosesan bahasa alami.

Dengan memahami CNN secara mendalammulai dari sejarah, struktur, variasi, hingga aplikasinya pembaca dapat memperoleh fondasi yang kuat untuk memahami perkembangan computer vision modern. Bab ini disusun untuk mengupas CNN tidak hanya sebagai model algoritmik, tetapi juga sebagai tonggak sejarah yang menghubungkan era feature engineering klasik dengan era representation learning yang sepenuhnya otomatis.

## 6.1 Convolutional Neural Network (CNN): Fondasi Visi Modern

### 6.1.1 Sejarah Perkembangan CNN

Perjalanan panjang Convolutional Neural Network (CNN) dimulai pada awal 1990-an melalui karya Yann LeCun dan rekan-rekannya yang memperkenalkan LeNet-5. Model ini digunakan untuk membaca digit tulisan tangan pada cek bank di Amerika Serikat. Meskipun arsitekturnya sederhana, LeNet sudah menunjukkan konsep dasar CNN: penggunaan lapisan konvolusi untuk mengekstraksi fitur, lapisan pooling untuk mereduksi dimensi, serta lapisan fully connected untuk melakukan klasifikasi. Pada masa itu, keterbatasan perangkat keras membuat CNN belum banyak digunakan secara luas.

Terobosan besar terjadi dua dekade kemudian dengan munculnya AlexNet pada tahun 2012. Model ini memenangkan kompetisi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dengan mengalahkan pesaing tradisional berbasis handcrafted features dengan margin yang sangat signifikan. AlexNet memperkenalkan beberapa inovasi, seperti penggunaan GPU untuk mempercepat pelatihan, penggunaan Rectified Linear Unit (ReLU) sebagai fungsi aktivasi untuk mengatasi masalah vanishing gradient, serta penerapan dropout untuk mengurangi overfitting. Keberhasilan AlexNet menjadi momentum dimulainya era “deep learning modern” dalam computer vision.

Setelah AlexNet, perkembangan arsitektur CNN berlangsung sangat cepat. VGGNet (2014) memperkenalkan arsitektur yang lebih dalam dengan menggunakan filter kecil berukuran 3×3, menjadikan struktur jaringan lebih konsisten dan mudah dianalisis. Namun, VGGNet memiliki kelemahan berupa jumlah parameter yang sangat besar sehingga membutuhkan memori tinggi. Untuk mengatasi masalah kedalaman jaringan yang menyebabkan vanishing gradient, ResNet (2015) menghadirkan konsep residual connection, yaitu jalur pintas yang memungkinkan gradien mengalir lebih stabil saat melatih jaringan sangat dalam. Dengan pendekatan ini, ResNet berhasil melatih jaringan dengan ratusan hingga ribuan lapisan tanpa kehilangan performa.

Perkembangan berikutnya ditandai dengan lahirnya EfficientNet (2019) yang menggunakan strategi compound scaling, yaitu pendekatan sistematis untuk menyeimbangkan kedalaman (depth), lebar (width), dan resolusi (resolution) jaringan. Hasilnya, EfficientNet mampu mencapai akurasi tinggi dengan jumlah parameter yang lebih efisien dibandingkan arsitektur sebelumnya. Model ini menjadi contoh nyata bahwa kemajuan CNN tidak hanya berfokus pada akurasi, tetapi juga efisiensi komputasi yang penting untuk implementasi pada perangkat terbatas, seperti ponsel cerdas atau sistem berbasis IoT.

Sejarah CNN menunjukkan adanya pola evolusi yang konsisten: setiap arsitektur baru hadir untuk menjawab kelemahan generasi sebelumnya. Dari LeNet yang sederhana, AlexNet yang revolusioner, VGGNet yang konsisten, ResNet yang stabil, hingga EfficientNet yang efisien,

semua menunjukkan bahwa CNN merupakan teknologi yang terus berkembang untuk menyesuaikan diri dengan kebutuhan zaman.

Tabel 6.1 Evolusi CNN dari Generasi Awal hingga Modern

| Model | Tahun | Kontribusi Utama | Kelebihan | Keterbatasan |
|---|---|---|---|---|
| LeNet-5 | 1990s | Pengenalan digit tulisan tangan | Sederhana, ringan | Hanya cocok untuk dataset kecil |
| AlexNet | 2012 | Revolusi deep learning dengan GPU, ReLU | Akurasi tinggi, pionir DL | Membutuhkan GPU, data besar |
| VGGNet | 2014 | Filter 3×3 berlapis, arsitektur konsisten | Desain sederhana, mudah dipahami | Parameter sangat banyak |
| ResNet | 2015 | Residual connection (skip connection) | Bisa sangat dalam, stabil | Lebih kompleks, komputasi berat |
| EfficientNet | 2019 | Compound scaling (depth, width, resolution) | Akurasi & efisiensi tinggi | Sulit ditrain tanpa teknik lanjutan |

Tabel 6.1 merangkum perjalanan panjang CNN dari generasi awal hingga modern. LeNet-5 menjadi pionir pada tahun 1990-an dengan aplikasi sederhana dalam pengenalan digit tulisan tangan. Keunggulan LeNet terletak pada arsitekturnya yang ringan, tetapi keterbatasannya sangat jelas: hanya dapat bekerja pada dataset kecil dan terbatas.

Kemajuan besar datang melalui AlexNet pada tahun 2012. Model ini menggabungkan kekuatan GPU, fungsi aktivasi ReLU, dan strategi dropout untuk mencegah overfitting. Hasilnya spektakuler: AlexNet mendominasi kompetisi ImageNet dengan margin besar. Namun, keberhasilan ini hanya dapat dicapai dengan dukungan perangkat keras kelas tinggi dan dataset berukuran masif, sehingga tidak selalu praktis untuk semua kalangan.

Arsitektur berikutnya, VGGNet, memperkenalkan pendekatan yang lebih konsisten dengan menggunakan filter konvolusi 3×3 berlapis. Desain ini membuat struktur jaringan mudah dipahami dan direplikasi. Akan tetapi, konsekuensinya adalah jumlah parameter yang sangat besar, sehingga model menjadi lambat dan boros memori.

Untuk mengatasi kendala jaringan yang semakin dalam, ResNet hadir dengan inovasi residual connection. Konsep ini memungkinkan pelatihan jaringan dengan ratusan lapisan tanpa mengalami masalah vanishing gradient. ResNet terbukti stabil dan akurat, tetapi kompleksitasnya membuat kebutuhan komputasi meningkat tajam.

Perjalanan berlanjut dengan EfficientNet, yang berfokus pada efisiensi. Alih-alih hanya menambah kedalaman, EfficientNet memperkenalkan compound scaling yang menyeimbangkan kedalaman, lebar, dan resolusi jaringan. Hasilnya, model ini mampu mencapai akurasi tinggi dengan jumlah parameter lebih sedikit dibandingkan generasi

sebelumnya. Namun, pelatihan EfficientNet tetap membutuhkan teknik lanjutan yang tidak selalu mudah diimplementasikan.

Secara keseluruhan, tabel ini memperlihatkan bahwa setiap generasi CNN hadir untuk menjawab kelemahan generasi sebelumnya. Dari LeNet yang sederhana, AlexNet yang revolusioner, VGGNet yang konsisten, ResNet yang stabil, hingga EfficientNet yang efisien, semuanya menandai tonggak penting dalam evolusi computer vision.

### 6.1.2 Formulasi Matematis Dasar CNN

Untuk memahami CNN secara lebih dalam, perlu ditinjau formulasi matematis dari proses konvolusi. Misalkan sebuah citra input $I$ berukuran $m \times n$, dan sebuah kernel atau filter $K$ Berukuran $p \times q$.Operasi konvolusi dapat dituliskan sebagai:

$$S(i,j) = \sum_{u=0}^{p-1} \sum_{v=0}^{q-1} I(i+u, j+v) \cdot K(u,v)$$

di mana $S(i,j)$ adalah nilai keluaran pada posisi $(i,j)$. Filter $K$ akan digeser (*sliding*) di seluruh citra input, menghasilkan feature map yang menyoroti pola tertentu, seperti tepi horizontal, tepi vertikal, atau tekstur tertentu.

Selain konvolusi, CNN juga menggunakan fungsi aktivasi untuk memberikan sifat non-linear. Fungsi yang paling populer adalah Rectified Linear Unit (ReLU) yang didefinisikan sebagai:

$$f(x) = \max(0, x)$$

Fungsi ini sederhana namun efektif dalam mempercepat pelatihan dan mengurangi masalah vanishing gradient.

### 6.1.3 Struktur Lapisan CNN

Arsitektur CNN terdiri atas beberapa komponen utama yang bekerja secara berurutan untuk mengekstraksi fitur dari citra hingga menghasilkan prediksi. Setiap lapisan memiliki fungsi spesifik yang saling melengkapi, sehingga jaringan mampu belajar secara hierarkis dari pola sederhana hingga bentuk yang kompleks.

a. Lapisan Konvolusi (Convolution Layer)

Lapisan konvolusi adalah inti dari CNN. Proses konvolusi dilakukan dengan menggeser kernel atau filter pada citra masukan untuk menghasilkan feature map. Setiap filter dirancang untuk menangkap pola tertentu, misalnya garis horizontal, vertikal, atau tekstur khusus. Secara matematis, operasi konvolusi dua dimensi dapat dituliskan sebagai:

$$S(i,j) = \sum_{u=0}^{p-1} \sum_{v=0}^{q-1} I(i+u, j+v) \cdot K(u,v)$$

b. Fungsi Aktivasi

Setelah konvolusi, hasil feature map dilewatkan ke fungsi aktivasi untuk menambahkan sifat non-linear. Tanpa aktivasi, jaringan hanya menjadi transformasi linier yang terbatas. Fungsi ReLU (Rectified Linear Unit) paling banyak digunakan karena sederhana dan efektif mengurangi masalah vanishing gradient. Pada kasus tertentu, varian lain seperti Leaky ReLU atau Sigmoid juga digunakan sesuai kebutuhan.

c. Lapisan Pooling

Pooling layer bertugas mereduksi dimensi data sehingga ukuran feature map menjadi lebih kecil, tanpa kehilangan informasi penting. Dengan pooling, CNN menjadi lebih efisien dan lebih tahan terhadap pergeseran (translation invariance). Ada dua jenis pooling yang paling umum:

Max Pooling → mengambil nilai maksimum dari area tertentu (misalnya 2×2). Teknik ini cenderung menyoroti fitur paling dominan.

Average Pooling → menghitung rata-rata nilai dalam area tertentu. Pendekatan ini lebih lembut dan mempertahankan distribusi informasi.

Tabel 6.2 Perbandingan Max Pooling dan Average Pooling

| Jenis Pooling | Mekanisme | Kelebihan | Kelemahan |
|---|---|---|---|
| Max Pooling | Mengambil nilai maksimum | Menyoroti fitur dominan, robust terhadap noise | Bisa mengabaikan informasi halus |
| Average Pooling | Mengambil nilai rata-rata | Mempertahankan distribusi informasi | Kurang efektif menyoroti fitur kuat |

d. Lapisan Fully Connected

Setelah beberapa kali konvolusi dan pooling, hasil feature map diproyeksikan ke dalam lapisan fully connected. Lapisan ini bertindak seperti jaringan syaraf tiruan tradisional yang menyatukan seluruh informasi untuk menghasilkan keputusan klasifikasi. Misalnya, dalam deteksi penyakit daun, lapisan ini akan mengeluarkan probabilitas apakah daun sehat atau terinfeksi.

e. Normalisasi

Untuk mempercepat dan menstabilkan pelatihan, CNN sering menggunakan Batch Normalization (BN). Teknik ini menormalkan distribusi data di setiap lapisan sehingga proses

propagasi gradien menjadi lebih stabil. BN juga berfungsi sebagai bentuk regularisasi, sehingga dapat meningkatkan generalisasi model.

Tabel 6.3 Komponen Penyusun CNN dan Fungsi nya

| **Komponen CNN** | **Fungsi Utama** | **Output** |
|---|---|---|
| Conv Layer | Deteksi fitur spasial lokal | Feature Maps |
| ReLU | Aktivasi non-linear | Feature Maps (tanpa nilai negatif) |
| Pooling | Reduksi dimensi (max/average pool) | Ringkasan fitur spasial |
| FC Layer | Klasifikasi | Label probabilitas |

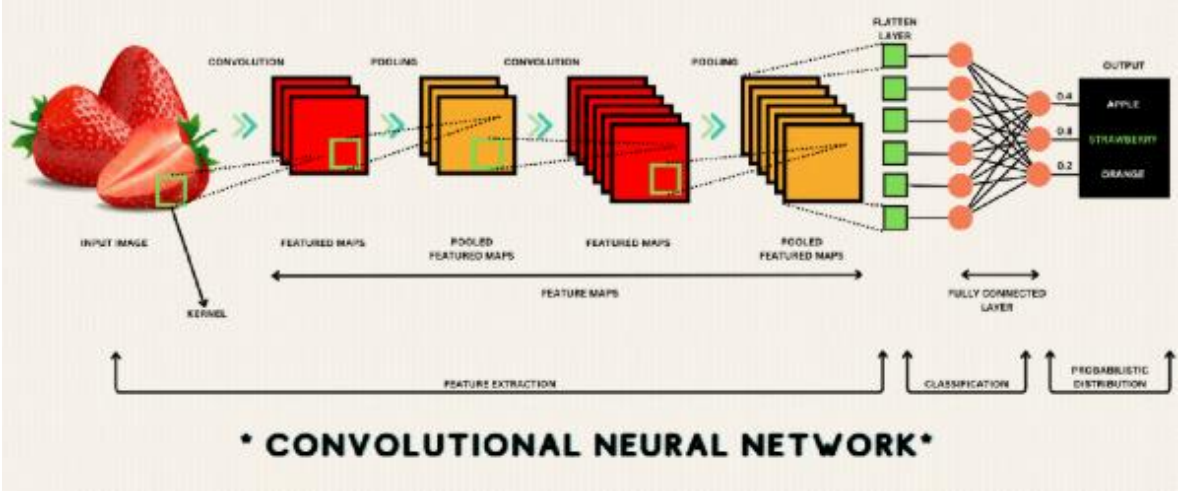
Pada tabel 6.3 Dalam arsitektur Convolutional Neural Network (CNN) setiap komponen menjalankan fungsi khusus yang saling melengkapi dalam memproses informasi visual. Proses diawali oleh Lapisan Konvolusi (Convolutional Layer) yang berperan sebagai detektor fitur spasial. Lapisan ini menggunakan sejumlah filter untuk memindai gambar secara sistematis guna mengidentifikasi pola-pola lokal seperti tepian, sudut, atau tekstur. Keluaran dari lapisan ini berupa feature maps yang memetakan lokasi dan intensitas fitur tertentu dalam gambar.

Selanjutnya, Fungsi Aktivasi ReLU menerima feature maps tersebut dan menerapkan non-linearitas dengan mengganti semua nilai negatif menjadi nol. Operasi ini tidak hanya memperkenalkan sifat non-linear yang crucial untuk memodelkan hubungan kompleks, tetapi juga meningkatkan efisiensi komputasi dan mitigasi masalah vanishing gradient

Lapisan Pooling kemudian melakukan reduksi dimensi spasial melalui operasi max-pooling atau average-pooling. Dengan mempertahankan hanya fitur dominan dari setiap wilayah pada feature maps, lapisan ini menghasilkan representasi yang lebih ringkas namun informatif, sekaligus memberikan ketahanan terhadap variasi posisi dan noise.

Tahap akhir pemrosesan dilakukan oleh Lapisan Terhubung Penuh (Fully Connected Layer) yang berfungsi sebagai klasifikator. Lapisan ini mengintegrasikan seluruh fitur yang telah diekstraksi, melakukan transformasi non-linear, dan menghasilkan vektor probabilitas yang

menunjukkan likelihood setiap kelas target. Melalui urutan transformasi ini, CNN mampu mengubah piksel mentah menjadi representasi hierarkis yang semakin abstrak dan bermakna.
Visualisasi Arsitektur CNN




Gambar 22. Alur Pendeteksi CNN

Gambar 22 mengilustrasikan proses berurutan yang dilakukan oleh sebuah Convolutional Neural Network (CNN) dalam menganalisis dan mengenali objek di dalam sebuah gambar. Alur kerja dimulai ketika sebuah gambar masukan dimasukkan ke dalam jaringan.

Pada tahap awal, gambar tersebut melalui serangkaian lapisan konvolusi dan pooling. Lapisan konvolusi bertindak seperti serangkaian filter yang menyisir setiap bagian kecil gambar untuk menemukan pola-pola dasar, seperti garis, tepian, sudut, atau tekstur. Setiap lapisan konvolusi biasanya diikuti oleh lapisan pooling, yang menyederhanakan informasi yang diperoleh dengan mempertahankan hanya fitur-fitur yang paling menonjol. Proses ini dilakukan secara berulang, di mana setiap lapisan berikutnya belajar untuk mendeteksi pola yang lebih kompleks dan abstrak (seperti bentuk mata atau hidung) yang dibangun dari pola-pola sederhana yang terdeteksi di lapisan sebelumnya.

Fitur-fitur yang telah diekstraksi dan disaring ini kemudian "diratakan" dan diteruskan ke lapisan terhubung sepenuhnya (fully connected layers). Di sinilah jaringan melakukan interpretasi akhir. Dengan menganalisis kombinasi dari semua fitur yang telah dikumpulkan, jaringan saraf ini pada akhirnya menghasilkan sebuah probabilitas atau keyakinan mengenai objek apa yang terdapat dalam gambar tersebut. Hasil akhirnya sering ditampilkan sebagai persentase keyakinan untuk setiap kategori yang mungkin (misalnya, "98% stroberi", "2% semangka").

Secara visual, gambar ini kemungkinan menunjukkan transformasi bertahap dari gambar asli yang terdiri dari piksel, menjadi peta fitur yang semakin abstrak, dan akhirnya berujung pada sebuah label keluaran. Ilustrasi semacam ini membantu pembaca memahami bagaimana CNN

membangun pemahaman secara hierarkis, dari hal yang sederhana hingga yang kompleks, untuk mencapai sebuah pengenalan visual.

Kelebihan CNN dalam Vision:

- Pemrosesan spasial lokal: CNN dapat mengenali pola yang muncul di mana pun dalam gambar.
- Parameter pembagian: CNN dapat mengurangi jumlah bobot yang perlu dilatih.
- Invarian terhadap translasi: Ini bagus untuk mengidentifikasi objek yang bergerak.

**Perkembangan Algoritma CNN**

Sejak diperkenalkan pertama kali oleh Yann LeCun pada akhir 1980-an melalui arsitektur LeNet-5, Convolutional Neural Network (CNN) telah mengalami perkembangan yang sangat pesat dan menjadi salah satu fondasi utama dalam bidang computer vision. Pada masa awal, CNN hanya mampu digunakan untuk tugas sederhana seperti pengenalan angka tulisan tangan dalam dataset MNIST. Namun, seiring bertambahnya kapasitas komputasi dan ketersediaan data berukuran besar, CNN berkembang menjadi algoritma yang sangat kuat untuk berbagai aplikasi, khususnya dalam pendeteksian dan pengenalan gambar.

Perkembangan CNN tidak bisa dilepaskan dari dua faktor penting, yaitu kemajuan perangkat keras (GPU) dan konsep arsitektur yang semakin kompleks. Kemunculan GPU memungkinkan pelatihan jaringan yang jauh lebih besar dan lebih dalam, sementara ide-ide baru dalam desain arsitektur CNN membawa efisiensi serta akurasi yang semakin baik.

Beberapa tonggak penting dalam perjalanan CNN dapat dilihat dari lahirnya arsitektur-arsitektur baru. Misalnya, AlexNet (2012) yang memenangkan kompetisi ImageNet dengan margin yang sangat besar, membuktikan keunggulan CNN dibandingkan metode tradisional berbasis hand-crafted features. Setelah itu, muncul VGGNet (2014) yang memperkenalkan ide kesederhanaan arsitektur dengan menggunakan filter berukuran kecil namun dalam jumlah lapisan yang banyak, sehingga mampu menangkap pola visual yang lebih kompleks.

Perkembangan berikutnya ditandai oleh GoogLeNet/Inception (2014) yang memperkenalkan konsep Inception module, memungkinkan jaringan memilih ukuran filter terbaik secara paralel. Lalu ResNet (2015) hadir dengan inovasi residual connection, yang menyelesaikan masalah vanishing gradient pada jaringan yang sangat dalam, sekaligus membuka jalan bagi pengembangan deep neural network berskala ratusan bahkan ribuan lapisan.

Tidak berhenti di situ, muncul pula berbagai varian CNN yang berfokus pada efisiensi komputasi, seperti MobileNet dan EfficientNet, yang dirancang agar dapat berjalan di perangkat dengan sumber daya terbatas tanpa mengorbankan kinerja secara signifikan. Arsitektur-arsitektur ini menjadi fondasi bagi pengembangan sistem real-time image detection di perangkat mobile maupun IoT.

Dengan demikian, perkembangan CNN dapat dipandang sebagai perjalanan evolusi dari algoritma sederhana menuju ekosistem yang kaya inovasi. Setiap generasi arsitektur menghadirkan terobosan yang bukan hanya meningkatkan akurasi deteksi, tetapi juga memperluas cakupan penerapan CNN pada berbagai bidang, mulai dari kesehatan, transportasi, hingga keamanan siber. Perjalanan inilah yang kemudian akan dirangkum secara lebih sistematis dalam tabel perkembangan CNN untuk pendeteksian gambar.

Tabel 6.4 Perkembangan CNN untuk Pendeteksian Gambar

| Model CNN | Tahun | Kontribusi Utama | Kelebihan | Keterbatasan |
|---|---|---|---|---|
| LeNet-5 | 1998 | Model awal CNN untuk pengenalan digit tulisan tangan (MNIST). | Sederhana, menjadi dasar arsitektur CNN modern. | Kurang kuat untuk data kompleks berskala besar. |
| AlexNet | 2012 | Pemenang ImageNet, memperkenalkan ReLU, dropout, dan penggunaan GPU. | Akurasi tinggi, mampu memproses data besar. | Ukuran model besar, risiko overfitting. |
| VGGNet | 2014 | Menggunakan arsitektur sederhana dengan banyak layer konvolusi 3x3. | Struktur mudah dipahami, hasil representasi fitur lebih baik. | Parameter sangat banyak, membutuhkan memori besar. |
| GoogLeNet (Inception) | 2014 | Memperkenalkan Inception Module untuk efisiensi komputasi. | Lebih ringan dari VGG, akurasi tinggi. | Arsitektur lebih kompleks untuk implementasi. |
| ResNet | 2015 | Memperkenalkan residual connection untuk mengatasi degradasi akurasi di jaringan dalam. | Bisa melatih jaringan sangat dalam (hingga ratusan layer). | Membutuhkan sumber daya komputasi besar. |
| DenseNet | 2017 | Setiap layer terhubung dengan semua layer berikutnya. | Meningkatkan aliran informasi dan gradien, lebih efisien parameter. | Lebih lambat dalam inferensi pada dataset besar. |
| EfficientNet | 2019 | Menggunakan pendekatan scaling (depth, width, resolution). | Efisien dan akurat dengan parameter lebih sedikit. | Kompleks dalam desain dan tuning. |
| Vision Transformer (ViT) | 2020 | Memanfaatkan arsitektur transformer untuk visi komputer. | Mampu menangkap konteks global dengan baik. | Membutuhkan dataset besar untuk pelatihan optimal. |

Menurut table 6.4 Perkembangan *Convolutional Neural Network* (CNN) dalam bidang pendeteksian gambar tidak dapat dipisahkan dari evolusi panjang arsitektur jaringan saraf tiruan. Sejak akhir 1990-an hingga saat ini, berbagai model CNN terus bermunculan dengan karakteristik dan keunggulannya masing-masing. Berikut adalah uraian mendalam dari beberapa model penting yang menandai tonggak perkembangan CNN.

1. LeNet-5 (1998)

   LeNet-5 merupakan salah satu arsitektur CNN pertama yang dikembangkan oleh Yann LeCun. Model ini dirancang khusus untuk pengenalan digit tulisan tangan, seperti dataset MNIST. Struktur LeNet-5 masih sederhana, terdiri dari beberapa lapisan konvolusi, *pooling*, dan *fully connected*. Meskipun sederhana, LeNet-5 membuktikan bahwa CNN mampu mengekstraksi fitur visual secara otomatis tanpa harus menggunakan metode ekstraksi manual. Keterbatasannya terletak pada skalabilitas: model ini kurang mampu menangani data berukuran

besar dan kompleks, tetapi tetap menjadi pondasi penting bagi perkembangan CNN modern.

2. AlexNet (2012)

   Lompatan besar terjadi dengan hadirnya AlexNet, arsitektur yang memenangkan kompetisi ImageNet pada tahun 2012. AlexNet memperkenalkan penggunaan *Rectified Linear Unit (ReLU)* sebagai fungsi aktivasi, teknik *dropout* untuk mengurangi overfitting, serta memanfaatkan GPU untuk mempercepat proses pelatihan. Inovasi ini membuat AlexNet mampu mengatasi dataset berskala besar dengan tingkat akurasi yang jauh melampaui metode sebelumnya. Namun, ukuran model yang relatif besar dan kebutuhan komputasi yang tinggi masih menjadi kendala pada saat itu.

3. VGGNet (2014)

   VGGNet hadir dengan filosofi sederhana: memperdalam jaringan menggunakan konvolusi berukuran kecil (3x3). Struktur yang konsisten membuat VGGNet lebih mudah dipahami dan diimplementasikan. Model ini berhasil mencapai performa tinggi dengan representasi fitur yang lebih mendetail. Akan tetapi, konsekuensi dari pendekatan ini adalah jumlah parameter yang sangat besar sehingga membutuhkan kapasitas memori dan komputasi yang signifikan.

4. GoogLeNet atau Inception (2014)

   Pada tahun yang sama, muncul GoogLeNet dengan *Inception Module* sebagai inovasi utama. Inception memungkinkan kombinasi berbagai ukuran filter dalam satu lapisan sehingga jaringan menjadi lebih efisien. Dibandingkan dengan VGG, GoogLeNet jauh lebih ringan dari segi jumlah parameter, tetapi tetap mampu menghasilkan akurasi tinggi. Kompleksitas arsitekturnya, bagaimanapun, membuat implementasinya relatif lebih menantang.

5. ResNet (2015)

   ResNet memperkenalkan konsep *residual connection* yang menjadi terobosan penting dalam mengatasi masalah degradasi akurasi pada jaringan yang sangat dalam. Dengan adanya jalur pintas (*skip connection*), ResNet memungkinkan informasi mengalir tanpa terhalang oleh bertambahnya lapisan. Arsitektur ini mampu mencapai kedalaman hingga ratusan lapisan, sesuatu yang sebelumnya sulit dicapai. Kelemahannya adalah kebutuhan akan sumber daya komputasi yang lebih besar, terutama pada pelatihan awal.

6. DenseNet (2017)

   DenseNet membawa ide bahwa setiap lapisan harus terhubung dengan semua lapisan berikutnya, bukan hanya lapisan terdekat. Dengan demikian, aliran informasi dan gradien menjadi lebih lancar. Pendekatan ini meningkatkan efisiensi parameter sekaligus memperkuat pembelajaran fitur. Namun, pada

dataset besar, DenseNet cenderung lebih lambat dalam proses inferensi karena banyaknya koneksi antar lapisan.

7. EfficientNet (2019)

   Efisiensi menjadi fokus utama dalam EfficientNet. Alih-alih hanya memperdalam atau memperlebar jaringan, EfficientNet memperkenalkan strategi *compound scaling* yang menyeimbangkan kedalaman (*depth*), lebar (*width*), dan resolusi (*resolution*). Hasilnya adalah model yang relatif ringan namun tetap memiliki akurasi tinggi. Tantangan yang muncul Adalah kompleksitas dalam proses perancangan dan penyetelan parameter agar scaling dapat optimal.

8. Vision Transformer (ViT) (2020)

   Meskipun bukan CNN murni, Vision Transformer (ViT) menjadi bagian penting dalam evolusi arsitektur visi komputer. ViT menggunakan pendekatan *transformer* yang sebelumnya populer dalam pemrosesan bahasa alami. Keunggulannya terletak pada kemampuan menangkap konteks global dari suatu citra, sesuatu yang lebih sulit dicapai oleh CNN tradisional. Namun, ViT membutuhkan jumlah data yang sangat besar untuk mencapai performa optimal, sehingga masih menjadi tantangan dalam implementasi praktis.

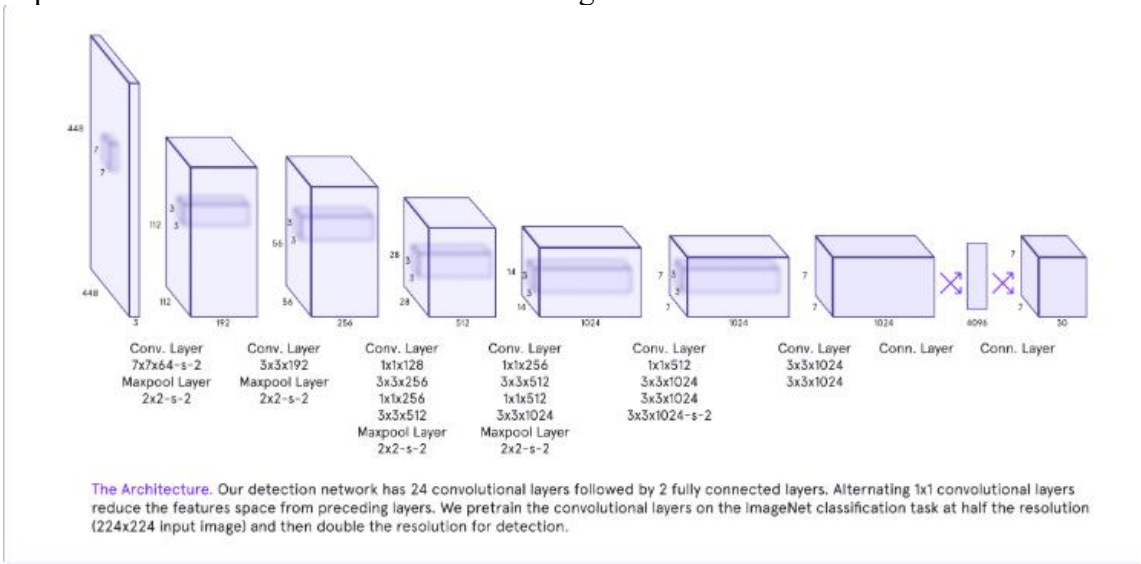
Secara keseluruhan, perkembangan CNN dalam pendeteksian gambar menunjukkan tren yang jelas: dari arsitektur sederhana menuju jaringan yang lebih dalam, efisien, dan adaptif. Setiap model tidak hanya memperbaiki keterbatasan pendahulunya, tetapi juga membuka jalan bagi aplikasi baru, mulai dari pengenalan wajah, diagnosis medis berbasis citra, hingga deteksi penyakit tanaman. Evolusi ini menegaskan bahwa CNN bukan sekadar algoritma, melainkan sebuah paradigma yang terus bertransformasi seiring dengan meningkatnya kebutuhan dan ketersediaan teknologi komputasi.

## 6.2 YOLO (You Only Look Once): Deteksi Objek Real-time

Konsep Dasar YOLO
YOLO adalah metode deteksi objek baru yang menggunakan regresi tunggal. YOLO langsung memprediksi kotak batas dan label kelas dari seluruh gambar dalam satu inferensi, bukan dua, seperti metode konvensional—region proposal lalu klasifikasi.
Sebagai contoh, YOLOv5 dan YOLOv8 telah dioptimalkan untuk kecepatan dan akurasi yang tinggi.

Struktur YOLO umum: Backbone berfungsi sebagai ekstraktor fitur, biasanya CSPDarknet atau MobileNet. Neck berfungsi sebagai struktur seperti PANet atau FPN untuk

menggabungkan informasi fitur dari berbagai skala. Head bertanggung jawab untuk memprediksi label kelas dan koordinat bounding box.




Gambar 23. Alur Pendeteksian Yolo

Gambar 23 mengilustrasikan arsitektur jaringan yang digunakan dalam YOLO (You Only Look Once), sebuah pendekatan inovatif untuk deteksi objek yang dikenal dengan kecepatan dan akurasinya. Berbeda dengan model yang menganalisis gambar dalam beberapa tahap, YOLO melakukan deteksi hanya dalam sekali lihat (single shot), membuatnya sangat efisien untuk aplikasi real-time.

Arsitektur inti YOLO terdiri dari 24 lapisan konvolusi yang disusun secara berurutan, diikuti oleh 2 lapisan terhubung penuh (fully connected layers) di bagian akhir. Lapisan konvolusi berfungsi untuk mengekstraksi fitur-fitur visual dari gambar secara hierarkis, mulai dari fitur sederhana seperti tepi dan tekstur, hingga pola yang lebih kompleks seperti bentuk dan bagian objek. Untuk mengoptimalkan proses ekstraksi fitur, lapisan konvolusi 1x1 diselang-seling di antara lapisan-lapisan lainnya. Lapisan 1x1 ini berperan dalam mengurangi dimensi ruang fitur, yang membuat komputasi menjadi lebih ringan tanpa mengorbankan kemampuan deteksi model.

Sebelum digunakan untuk tugas deteksi objek, bagian konvolusi dari jaringan ini dilatih terlebih dahulu pada tugas klasifikasi menggunakan dataset ImageNet dengan resolusi input 224x224 piksel. Proses pretraining ini memungkinkan model mempelajari representasi fitur yang umum dan berguna dari beragam objek. Setelah itu, resolusi input ditingkatkan menjadi

dua kali lipat (misalnya 448x448) untuk tugas deteksi, sehingga jaringan dapat menangkap detail yang lebih halus dan mendeteksi objek dengan lebih akurat.

Melalui desain yang efisien ini, YOLO mampu menggabungkan kecepatan dan ketepatan, menjadikannya salah satu pilihan utama dalam aplikasi deteksi objek waktu-nyata seperti sistem pengawasan autonomous driving, analisis video, dan sistem monitoring visual.

### 6.2.1 Performa YOLO

YOLO terkenal karena:

- Deteksi real-time dengan FPS tinggi;
- Presisi yang baik untuk objek umum;
- Ukuran model yang efisien untuk perangkat edge seperti Raspberry Pi atau ESP32-CAM.

Tabel 6.5 Perkembangan model YOLO

| Model | mAP (mean Average Precision) | FPS (NVIDIA RTX 2080) | Ukuran Model |
|---|---|---|---|
| YOLOv3 | 33.0 | 45 | ~237 MB |
| YOLOv4 | 43.5% (AP50) | 62 | ~244 MB |
| YOLOv5s | 36.5 | 140 | ~14.5 MB |
| YOLOv6n | 37.5 | 160 | ~10.5 MB |
| YOLOv7-tiny | 38.7 | 165 | ~12.1 MB |
| YOLOv8n | 37.9 | 180 | ~5 MB |
| YOLOv9c | ~46.8 (terbaru) | ~105 | ~75 MB |
| YOLOv10n | ~41.9 | ~210 | ~6.5 MB |

Pada tabel 6.5 Perkembangan model YOLO (You Only Look Once) dari versi ke versi menunjukkan kemajuan yang signifikan dalam menyeimbangkan aspek akurasi, kecepatan, dan efisiensi ukuran model. YOLOv3, yang menjadi fondasi perbandingan, mencatat mean Average Precision (mAP) sebesar 33.0 dengan kecepatan inferensi 45 FPS pada NVIDIA RTX 2080, meskipun dengan ukuran model yang relatif besar, yaitu sekitar 237 MB. Peningkatan drastis terlihat pada YOLOv5s, yang tidak hanya meningkatkan akurasi menjadi 36.5 mAP tetapi juga mendongkrak kecepatan hingga 140 FPS berkat optimasi arsitektur, sekaligus memampatkan ukuran model menjadi hanya 14.5 MB. Tren efisiensi ini berlanjut dengan kehadiran YOLOv8n, yang mencapai 37.9 mAP dan 180 FPS dengan ukuran model yang sangat ringkas, sekitar 5 MB.

Generasi terbaru, YOLOv9c dan YOLOv10n, mencerminkan inovasi lebih lanjut. YOLOv9c, yang dirilis pada Februari 2024, menetapkan standar baru dalam akurasi dengan mAP sekitar 46.8, meskipun dengan kebutuhan komputasi yang lebih tinggi (105 FPS) dan ukuran model sekitar 75 MB. Di sisi lain, YOLOv10n (Mei 2024) berfokus pada optimasi kecepatan dan efisiensi, mencapai 210 FPS dengan akurasi 41.9 mAP dan ukuran model 6.5 MB, menjadikannya solusi ideal untuk aplikasi real-time pada perangkat dengan sumber daya terbatas. Evolusi ini menunjukkan komitmen berkelanjutan untuk membuat deteksi objek tidak

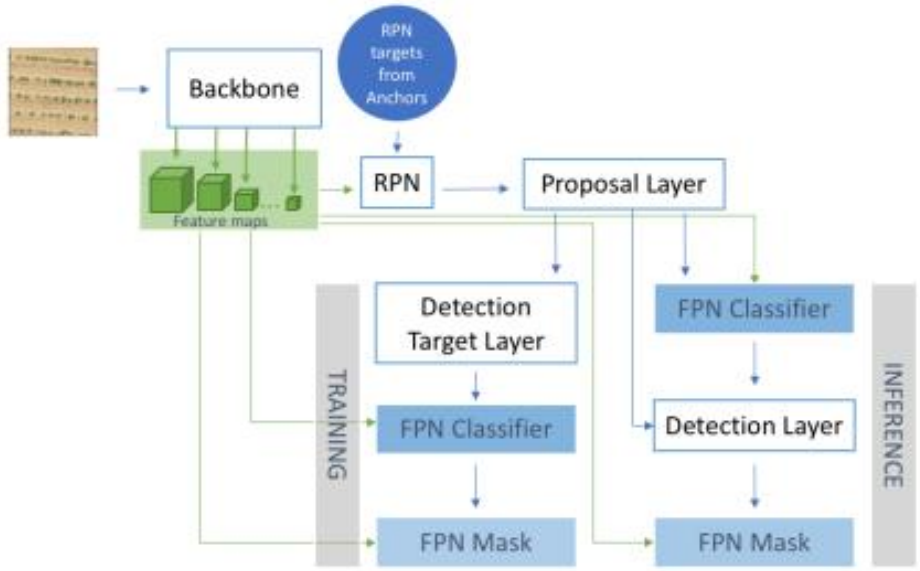
hanya lebih akurat tetapi juga lebih terjangkau dan adaptif terhadap berbagai kebutuhan komputasi.

### 6.3 Mask R-CNN: Segmentasi Objek yang Presisi
### Evolusi dari Faster R-CNN

Mask R-CNN adalah perluasan dari Faster R-CNN yang dikembangkan oleh tim penelitian AI Facebook. Mask R-CNN sangat berguna untuk
aplikasi seperti deteksi sel, analisis citra medis, dan pemrosesan gambar industri. Ini karena, dibandingkan dengan hanya mendeteksi objek dan
menghasilkan kotak pembatas, ia juga memprediksi masker segmentasi untuk setiap objek yang terdeteksi.

**Komponen Utama**

- Region Proposal Network (RPN): Mengusulkan wilayah potensial objek.
- ROI Align: Menyesuaikan area proposal ke dalam resolusi tetap.
- Branch Additional: Untuk memprediksi mask biner pada tiap pixel.




Gambar 24. Arsitektur Mask R-CNN

Gambar 24 mengilustrasikan arsitektur dari Mask R-CNN (Region-Based Convolutional Neural Network), sebuah model pionir dalam segmentasi instance yang tidak hanya mampu mendeteksi objek dalam sebuah gambar, tetapi juga menghasilkan masker pixel-pixel yang tepat yang menandai wilayah setiap objek yang terdeteksi. Arsitektur ini dapat dipahami sebagai sebuah pipeline yang elegan dan terintegrasi.

Prosesnya diawali oleh Backbone, yang biasanya terdiri dari jaringan convolutional seperti ResNet atau Feature Pyramid Network (FPN). Backbone berfungsi sebagai ekstraktor fitur multi-skala, menganalisis gambar input untuk menghasilkan representasi hierarkis yang

menangkap informasi mulai dari fitur tingkat rendah seperti tepi dan tekstur, hingga fitur tingkat tinggi seperti bentuk dan konteks objek.

Selanjutnya, Region Proposal Network (RPN) memindai peta fitur yang dihasilkan backbone untuk mengusulkan wilayah-wilayah yang mungkin mengandung objek, yang disebut region proposals. Proposal-layer kemudian menyaring dan menyempurnakan usulan wilayah ini untuk diteruskan ke tahap berikutnya.

Inti dari arsitektur ini terletak pada tiga cabang (heads) yang berjalan secara paralel untuk setiap region proposal. Cabang Classification bertugas mengidentifikasi kelas objek yang terdapat dalam wilayah tersebut (misalnya, manusia, mobil, atau anjing). Secara bersamaan, cabang Bounding Box Regression menyempurnakan koordinat kotak pembatas (bounding box) agar lebih tepat mengurung objek. Yang membedakan Mask R-CNN dari pendahulunya adalah cabang ketiga, yaitu Mask Head. Cabang ini menggunakan jaringan convolutional fully convolutional untuk memprediksi masker biner pada tingkat pixel untuk setiap objek, sehingga memberikan segmentasi yang detail dan akurat.

Dengan menggabungkan deteksi objek yang presisi melalui RPN dan segmentasi instance yang detail melalui Mask Head, arsitektur Mask R-CNN menyajikan pendekatan yang komprehensif dan powerful untuk memahami adegan visual secara mendalam, menjadikannya fondasi untuk banyak aplikasi dalam visi komputer modern. Kelebihan Mask R-CNN termasuk mendeteksi objek saling tumpang tindih dan segmentasi objek yang presisi setiap piksel. Ini juga akurat, meskipun biaya komputasi lebih tinggi daripada YOLO.

### 6.4 Perbandingan Arsitektur CNN, YOLO, dan Mask R-CNN

Tabel 6.6 perbandingan CNN Klasik,YOLO dan Mask R-CNN

| Fitur | CNN Klasik | YOLO | Mask R-CNN |
|---|---|---|---|
| Tujuan utama | Klasifikasi gambar | Deteksi objek real-time | Deteksi + segmentasi objek |
| Kecepatan | Tinggi | Sangat tinggi | Menengah |
| Akurasi lokal | Rendah | Sedang | Sangat tinggi |
| Kebutuhan resource | Rendah | Menengah | Tinggi |
| Cocok untuk embedded | Ya | Ya (YOLOv5/8) | Tidak ideal |

Tabel 6.6 membandingkan tiga pendekatan utama dalam computer vision berdasarkan karakteristik intinya, yaitu CNN Klasik, YOLO, dan Mask R-CNN. Masing-masing arsitektur

memiliki keunggulan dan trade-off yang berbeda, sehingga pemilihannya sangat bergantung pada kebutuhan spesifik aplikasi.

CNN Klasik terutama dirancang untuk tugas klasifikasi gambar, dimana tujuannya adalah mengidentifikasi objek dominan dalam suatu gambar tanpa memberikan informasi lokasi. Arsitektur ini unggul dalam kecepatan inferensi dan memiliki kebutuhan sumber daya komputasi yang rendah, sehingga sangat cocok untuk implementasi pada sistem embedded atau perangkat dengan kemampuan terbatas. Namun, kelemahan utamanya adalah akurasi lokalisasi yang rendah karena tidak menghasilkan bounding box atau segmentasi.

YOLO (You Only Look Once) merupakan arsitektur yang dikhususkan untuk deteksi objek real-time. Model ini menawarkan kecepatan inferensi sangat tinggi dengan akurasi lokalisasi pada tingkat sedang. YOLO mencapai efisiensi ini melalui pendekatan single-shot detection yang melakukan klasifikasi dan regresi bounding box secara bersamaan. Kebutuhan sumber dayanya berada pada level menengah, dan versi terbaru seperti YOLOv5 dan YOLOv8 telah dioptimalkan untuk dapat dijalankan pada perangkat embedded.

Mask R-CNN merepresentasikan pendekatan yang lebih kompleks dengan kemampuan deteksi objek sekaligus segmentasi instance tingkat pixel. Arsitektur ini menghasilkan akurasi lokalisasi yang sangat tinggi dengan kemampuan menghasilkan masker untuk setiap objek yang terdeteksi. Sebagai trade-off, model ini memiliki kecepatan inferensi menengah dan kebutuhan sumber daya komputasi yang tinggi, sehingga kurang cocok untuk aplikasi real-time atau implementasi pada perangkat embedded. Mask R-CNN lebih sesuai untuk aplikasi yang memprioritaskan presisi seperti analisis medis atau penelitian ilmiah.

Pemilihan arsitektur yang optimal harus mempertimbangkan trade-off antara kecepatan, akurasi, dan ketersediaan sumber daya komputasi. CNN Klasik sesuai untuk aplikasi klasifikasi sederhana, YOLO ideal untuk deteksi real-time, sedangkan Mask R-CNN tepat untuk aplikasi yang memerlukan presisi segmentasi tinggi.

# BAB VII
# Keterbatasan dan Faktor Kegagalan Algoritma Vision

## 7.1 Pendahuluan

Perkembangan algoritma computer vision, khususnya berbasis Convolutional Neural Network (CNN), telah membuka banyak peluang dalam bidang pendeteksian dan pengenalan gambar. Berbagai penelitian menunjukkan peningkatan akurasi yang signifikan, mulai dari pengenalan wajah, diagnosis penyakit melalui citra medis, hingga deteksi penyakit pada tanaman. Namun, di balik keberhasilan tersebut, masih terdapat sejumlah keterbatasan yang kerap menjadi penyebab kegagalan sistem dalam memberikan hasil deteksi yang konsisten dan dapat diandalkan.

Kegagalan dalam deteksi gambar bukan semata-mata disebabkan oleh kelemahan algoritma, melainkan juga dipengaruhi oleh faktor-faktor eksternal, seperti kualitas citra, kondisi lingkungan, maupun variasi objek yang tidak terduga. Misalnya, pencahayaan yang terlalu terang atau sebaliknya terlalu redup dapat mengubah tampilan objek, sehingga sistem kesulitan mengenali ciri yang seharusnya mudah dibedakan. Demikian pula, citra yang buram, berisik, atau diambil dari sudut tertentu sering kali menurunkan kinerja model secara drastis.

Selain faktor teknis, keterbatasan dataset juga menjadi tantangan besar. Model yang dilatih dengan data terbatas atau tidak beragam cenderung hanya "hapal" pola tertentu, tetapi gagal mengeneralisasi ketika dihadapkan pada kondisi nyata. Hal ini menjelaskan mengapa sistem yang bekerja baik di laboratorium belum tentu menunjukkan kinerja serupa di lapangan.
Oleh karena itu, membahas faktor kegagalan dan keterbatasan algoritma vision merupakan bagian penting dalam memahami perkembangan bidang ini. Tidak hanya untuk melihat kelemahan yang ada, tetapi juga untuk membuka ruang diskusi tentang strategi perbaikan, mulai dari desain arsitektur yang lebih adaptif, penggunaan data yang lebih representatif, hingga pengembangan metode yang mampu bekerja pada kondisi dunia nyata yang penuh ketidakpastian**.**

## 7.2 Faktor Teknis yang Mempengaruhi Kegagalan

Kinerja algoritma vision dalam mendeteksi gambar sangat dipengaruhi oleh sejumlah faktor teknis. Faktor-faktor ini sering kali bersumber dari kualitas data masukan maupun keterbatasan kemampuan model dalam mengolah variasi citra. Berikut adalah beberapa aspek utama yang dapat menyebabkan terjadinya kegagalan deteksi.

### 1. Kualitas Citra

Kualitas citra merupakan penentu utama keberhasilan sistem deteksi. Citra dengan resolusi rendah cenderung kehilangan detail penting, sehingga fitur yang seharusnya menjadi pembeda antarobjek tidak lagi terlihat jelas. Selain itu, keberadaan noise atau gangguan visual, seperti bintik acak pada citra digital, juga dapat mengaburkan informasi yang penting. Citra yang

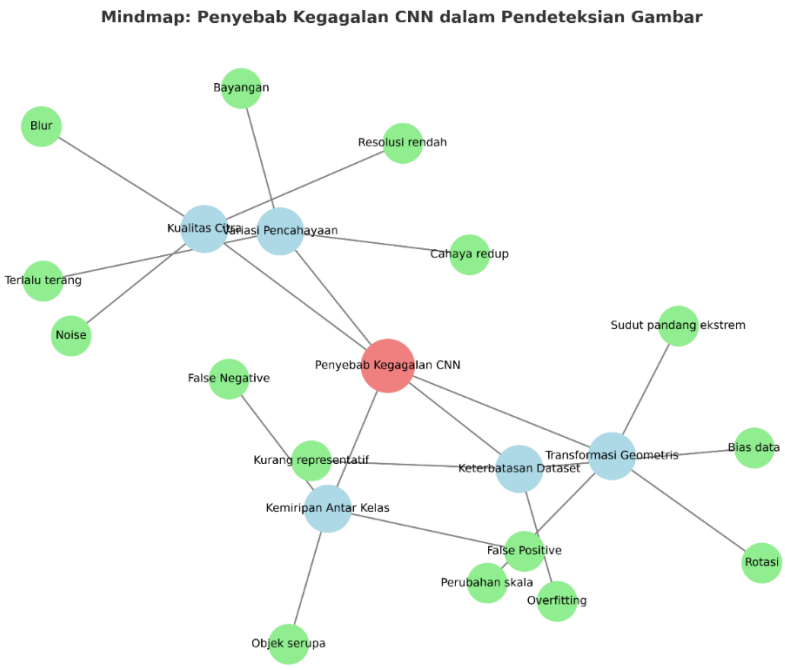
buram akibat pergerakan kamera atau objek juga menjadi penyebab umum kesalahan deteksi, karena jaringan kesulitan mengenali pola yang konsisten.

**2. Variasi Pencahayaan**
Pencahayaan yang tidak seragam dapat mengubah penampilan objek secara signifikan. Objek yang sama dapat terlihat berbeda ketika berada di bawah cahaya terang, redup, atau terkena bayangan. CNN pada umumnya sensitif terhadap perubahan ini, sehingga model yang dilatih dalam kondisi pencahayaan tertentu sering gagal ketika dihadapkan pada kondisi yang berbeda.

**3. Transformasi Geometris**
Perubahan posisi atau bentuk tampilan objek, seperti rotasi, translasi, maupun perubahan skala, juga berpengaruh pada akurasi deteksi. Misalnya, sebuah daun yang difoto dari sudut miring mungkin sulit dikenali dibandingkan dengan foto dari sudut tegak lurus. Meskipun CNN memiliki kemampuan ekstraksi fitur yang cukup kuat, perubahan ekstrem dalam sudut pandang sering kali membuat sistem salah mengenali atau bahkan gagal mendeteksi objek.

**4. Kemiripan Antar Kelas**
Kesalahan deteksi juga kerap terjadi ketika dua kelas objek memiliki kemiripan visual yang tinggi. Dalam konteks pendeteksian penyakit tanaman, misalnya, daun sehat dengan bercak alami dapat terlihat sangat mirip dengan daun yang baru mulai terinfeksi penyakit. Situasi ini sering menimbulkan false positive maupun false negative, yang pada akhirnya memengaruhi keandalan sistem secara keseluruhan.

**5. Keterbatasan Dataset**
Salah satu kendala terbesar dalam pengembangan algoritma vision adalah ketersediaan dataset yang beragam dan representatif. Dataset yang terlalu kecil atau homogen membuat model cenderung mengalami overfitting, yaitu hanya mengenali pola pada data pelatihan tanpa mampu menggeneralisasi ke data baru. Sebaliknya, dataset yang bias—misalnya hanya berisi citra dari satu kondisi tertentu—menyebabkan model gagal bekerja ketika digunakan dalam kondisi berbeda.




Gambar Mind Map Penyebab Kegagalan CNN dalam Pendeteksi Gambar

Gambar ini memvisualisasikan berbagai faktor yang dapat menyebabkan kegagalan *Convolutional Neural Network* (CNN) dalam proses pendeteksian gambar. Mind map tersebut menekankan bahwa kelemahan CNN tidak hanya bersumber dari aspek algoritmik, tetapi juga dipengaruhi oleh kondisi data, lingkungan, serta faktor eksternal lain yang sering muncul dalam praktik.

Salah satu faktor dominan adalah kualitas citra. Citra dengan resolusi rendah, pencahayaan yang tidak memadai, atau bayangan yang menutupi objek dapat mengurangi kemampuan CNN dalam mengenali pola visual. Kondisi seperti blur, noise, atau pencahayaan yang terlalu terang dan terlalu redup juga membuat fitur penting sulit diekstraksi. Hal ini menunjukkan bahwa CNN masih sangat bergantung pada kualitas masukan visual yang diberikan.

Selain itu, terdapat masalah yang berkaitan dengan keterbatasan dataset. CNN membutuhkan data latih yang besar, beragam, dan representatif. Namun, dalam kenyataannya, dataset sering kali memiliki keterbatasan, baik dari segi jumlah maupun variasi. Situasi ini dapat menimbulkan bias data, ketidakseimbangan antar kelas, serta fenomena *overfitting* ketika model hanya menghafal pola dari data latih tanpa mampu melakukan generalisasi. Kondisi ini semakin diperparah jika objek yang harus dideteksi memiliki kemiripan bentuk atau warna, sehingga meningkatkan kemungkinan *false positive* atau *false negative*.

Faktor lain yang tidak kalah penting adalah transformasi geometris. Objek yang muncul dengan rotasi, perubahan skala, atau sudut pandang yang ekstrem sering kali tidak dikenali dengan baik oleh CNN, karena model ini terbatas dalam menghadapi variasi spasial yang terlalu besar.

Meskipun terdapat teknik augmentasi data untuk mengurangi masalah tersebut, CNN tetap memiliki keterbatasan dalam menghadapi dinamika dunia nyata yang sangat kompleks. Secara keseluruhan, mind map ini menegaskan bahwa kegagalan CNN merupakan hasil dari interaksi berbagai faktor: mulai dari kualitas citra, keterbatasan dataset, hingga variasi kondisi lingkungan. Pemahaman terhadap peta penyebab kegagalan ini penting tidak hanya untuk mengkritisi kelemahan CNN, tetapi juga sebagai landasan bagi peneliti dan praktisi dalam merancang strategi perbaikan, baik melalui augmentasi data, pengembangan arsitektur baru, maupun integrasi dengan algoritma lain yang lebih adaptif.

**Tabel 7.1 Penyebab Kegagalan CNN dalam Pendeteksian Gambar dan Solusinya**

| Penyebab | Dampak pada Deteksi | Solusi yang Umum Digunakan |
|---|---|---|
| Kualitas Citra (noise, blur, resolusi rendah) | Fitur penting hilang, sistem gagal mengenali pola, hasil deteksi tidak stabil. | - Peningkatan kualitas citra (pre-processing)<br>- Filtering untuk mengurangi noise<br>- Penggunaan kamera dengan resolusi lebih tinggi |
| Variasi Pencahayaan (terlalu terang, redup, bayangan) | Model sensitif terhadap kondisi pencahayaan, objek tampak berbeda meskipun sama. | - Normalisasi pencahayaan (*histogram equalization*)<br>- Data augmentation dengan variasi pencahayaan<br>- Penggunaan sensor cahaya adaptif |
| Transformasi Geometris (rotasi, skala, sudut pandang ekstrem) | CNN gagal mengenali objek yang dipandang dari sudut berbeda atau ukuran berbeda. | - Augmentasi data (rotasi, flipping, scaling)<br>- Penggunaan arsitektur dengan *spatial transformer network* |
| Kemiripan Antar Kelas (objek mirip secara visual) | Meningkatkan *false positive* dan *false negative*, menurunkan akurasi klasifikasi. | - Penambahan fitur diskriminatif<br>- Penggunaan model dengan *attention mechanism*<br>- Labeling data lebih detail |
| Keterbatasan Dataset (terlalu kecil, bias, tidak representatif) | Model overfitting, hanya mengenali pola tertentu, gagal mengeneralisasi ke data baru. | - Data augmentation<br>- Transfer learning dari model pra-latih (pre-trained)<br>- Pengumpulan dataset yang lebih beragam |

Seperti yang terlihat pada Tabel 7.1, kegagalan CNN dalam pendeteksian gambar umumnya dipengaruhi oleh kombinasi faktor teknis yang saling berkaitan. Faktor pertama adalah kualitas citra, di mana keberadaan noise, citra buram, atau resolusi rendah dapat menyebabkan

hilangnya detail penting. Akibatnya, fitur visual yang seharusnya menjadi pembeda tidak lagi dapat diekstraksi dengan baik. Solusi yang sering diterapkan adalah melalui pre-processing, misalnya dengan teknik filtering untuk mengurangi noise, atau penggunaan kamera dengan resolusi lebih tinggi.

Faktor kedua adalah variasi pencahayaan, yang membuat objek tampak sangat berbeda hanya karena perbedaan intensitas cahaya atau adanya bayangan. Kondisi ini sering menurunkan akurasi model, terutama jika pelatihan hanya dilakukan dengan data pada pencahayaan tertentu. Untuk mengatasinya, para peneliti biasanya menggunakan normalisasi pencahayaan, seperti histogram equalization, serta menambahkan variasi pencahayaan dalam tahap data augmentation.

Selanjutnya, transformasi geometris seperti rotasi, perubahan skala, atau sudut pandang ekstrem juga menjadi penyebab kegagalan deteksi. CNN memang memiliki ketahanan tertentu terhadap variasi posisi, tetapi perubahan yang terlalu ekstrem dapat membuat sistem salah mengenali objek. Solusi yang umum dilakukan adalah memperkaya dataset dengan berbagai variasi posisi melalui teknik augmentasi data, atau memanfaatkan arsitektur yang lebih adaptif seperti spatial transformer network.

Faktor keempat adalah kemiripan antar kelas, yang kerap menimbulkan kesalahan klasifikasi. Dalam konteks penyakit tanaman misalnya, daun sehat dengan bercak alami bisa tampak mirip dengan daun yang baru terinfeksi. Situasi ini meningkatkan risiko false positive maupun false negative. Untuk mengurangi masalah ini, peneliti menambahkan fitur diskriminatif atau mengintegrasikan mekanisme attention agar model dapat lebih fokus pada area penting dari citra.

Terakhir, keterbatasan dataset menjadi salah satu tantangan terbesar. Dataset yang terlalu kecil atau bias membuat model cenderung overfitting, sehingga hanya "hapal" pola tertentu tanpa mampu mengeneralisasi ke data baru. Solusi yang banyak dipakai adalah data augmentation, penggunaan transfer learning dengan model pra-latih dari dataset besar, atau pengumpulan data baru yang lebih representatif terhadap kondisi nyata.
Dengan memahami penyebab kegagalan beserta dampaknya, pembaca tidak hanya melihat sisi keterbatasan CNN, tetapi juga memahami strategi yang dapat digunakan untuk mengatasi tantangan tersebut. Hal ini penting agar sistem pendeteksian gambar tidak hanya bekerja di lingkungan laboratorium, tetapi juga dapat diandalkan ketika diterapkan pada kondisi nyata yang penuh variasi**.**

### 7.3 Keterbatasan Algoritma dan Komputasi

Selain faktor teknis yang berasal dari kualitas citra dan kondisi lingkungan, kegagalan sistem computer vision juga dapat disebabkan oleh keterbatasan pada sisi algoritma dan kebutuhan komputasi. CNN memang telah terbukti mampu mengekstraksi fitur visual dengan baik, namun cara kerja dan struktur model ini masih menyimpan sejumlah kelemahan yang perlu dipahami.

#### 1. Kompleksitas Komputasi

Arsitektur CNN modern, terutama yang memiliki ratusan hingga ribuan lapisan seperti ResNet atau DenseNet, memerlukan sumber daya komputasi yang sangat besar. Proses pelatihan sering kali membutuhkan GPU dengan kapasitas tinggi dan waktu yang lama. Hal ini menjadi kendala utama ketika sistem harus diimplementasikan pada perangkat dengan daya terbatas, seperti edge device atau IoT. Meskipun sudah ada model yang lebih ringan seperti MobileNet dan

EfficientNet, keterbatasan daya komputasi tetap menjadi tantangan nyata dalam penggunaan algoritma vision pada aplikasi dunia nyata.

**2. Keterbatasan Interpretabilitas**
CNN sering kali dianggap sebagai black box karena sulit dijelaskan mengapa suatu prediksi dihasilkan. Meskipun model dapat memberikan hasil klasifikasi dengan akurasi tinggi, alasan di balik keputusan tersebut jarang transparan. Kurangnya interpretabilitas ini menimbulkan masalah serius pada bidang yang membutuhkan penjelasan yang jelas, misalnya dalam diagnosis medis atau sistem keamanan. Tanpa pemahaman yang memadai mengenai bagaimana model bekerja, sulit bagi pengguna atau peneliti untuk sepenuhnya mempercayai hasil deteksi.

**3. Masalah Generalisasi**
Model CNN biasanya dilatih dengan dataset tertentu, dan hasilnya sangat dipengaruhi oleh kualitas serta representasi data tersebut. Ketika model dihadapkan pada data baru dengan karakteristik berbeda dari data pelatihan, kinerjanya sering kali menurun drastis. Inilah yang dikenal sebagai masalah generalisasi. Misalnya, model yang dilatih dengan citra daun dalam kondisi pencahayaan laboratorium bisa gagal mendeteksi penyakit jika digunakan pada citra yang diambil langsung di lapangan dengan pencahayaan alami. Hal ini menunjukkan bahwa meskipun CNN kuat pada data terstruktur, ia masih kesulitan beradaptasi dengan variasi lingkungan nyata.

**4. Kebutuhan Data yang Besar**
Keberhasilan CNN sangat bergantung pada jumlah data pelatihan. Semakin dalam jaringan yang digunakan, semakin banyak pula data yang dibutuhkan untuk menghindari overfitting. Tantangan ini tidak selalu mudah diatasi, terutama di bidang yang sulit memperoleh dataset dalam jumlah besar, seperti citra medis atau data citra spesifik untuk tanaman tertentu. Akibatnya, pengembangan model yang andal sering terhambat oleh keterbatasan ketersediaan data.

**7.4 Faktor Lingkungan dan Konteks**
Selain keterbatasan teknis dan komputasi, kegagalan algoritma vision juga sering dipengaruhi oleh faktor lingkungan tempat citra diambil. Kondisi dunia nyata jauh lebih kompleks dibandingkan dengan lingkungan terkontrol di laboratorium. Variasi kontekstual seperti cuaca, pergerakan objek, atau latar belakang yang tidak terduga dapat menyebabkan penurunan akurasi deteksi secara signifikan.

**1. Variasi Lingkungan Nyata**
Kondisi alam yang dinamis menjadi salah satu tantangan utama. Misalnya, kabut, hujan, atau pencahayaan matahari yang berubah-ubah dapat membuat objek tampak berbeda dari citra dalam dataset pelatihan. Bayangan yang muncul pada waktu tertentu juga bisa menutupi sebagian fitur penting dari objek, sehingga sistem kesulitan mengenali pola. Dalam konteks pertanian, daun yang terkena embun atau debu mungkin terlihat sangat berbeda dibandingkan daun yang bersih, sehingga memicu kesalahan dalam klasifikasi.

**2. Objek Bergerak Cepat**
Dalam aplikasi tertentu, seperti pengawasan lalu lintas atau deteksi aktivitas manusia, objek sering kali bergerak cepat. Pergerakan ini menyebabkan citra menjadi buram (motion blur) sehingga detail visual sulit ditangkap. CNN yang dilatih dengan citra statis biasanya tidak

memiliki kemampuan adaptif untuk mengatasi fenomena ini, sehingga hasil deteksinya kurang andal.

**3. Gangguan Latar Belakang**
Latar belakang yang terlalu ramai atau kompleks dapat mengganggu proses deteksi. Objek yang seharusnya menjadi fokus bisa tenggelam di antara pola visual lain yang mirip. Sebagai contoh, deteksi hewan di hutan sering kali gagal karena warna dan tekstur hewan menyatu dengan lingkungan sekitarnya. Hal ini menegaskan bahwa sistem vision masih memiliki keterbatasan dalam memilah informasi utama dari distraksi visual yang berlebihan.

**4. Keterbatasan Adaptasi Domain**
Masalah lain yang sering muncul adalah ketika model digunakan di luar domain data pelatihannya. Misalnya, model yang dilatih untuk mendeteksi penyakit tanaman pada varietas tertentu sering gagal ketika diterapkan pada varietas lain yang memiliki morfologi berbeda.
Fenomena ini dikenal sebagai domain adaptation problem, yang menunjukkan bahwa CNN masih lemah dalam beradaptasi dengan konteks baru tanpa pelatihan tambahan.

## 7.5 Upaya Mengatasi Keterbatasan

Keterbatasan dan faktor kegagalan algoritma vision bukan berarti menjadi penghalang mutlak. Justru, dari berbagai kendala tersebut lahirlah beragam inovasi dan strategi yang bertujuan untuk meningkatkan kinerja model, baik dari sisi akurasi, efisiensi, maupun kemampuan generalisasi. Beberapa pendekatan yang umum digunakan dalam penelitian maupun implementasi praktis antara lain sebagai berikut.

**1. Augmentasi Data**
Salah satu cara paling sederhana namun efektif untuk mengatasi keterbatasan dataset adalah melalui data augmentation. Teknik ini memperbanyak variasi data pelatihan dengan cara memodifikasi citra asli, misalnya dengan rotasi, flipping, perubahan skala, hingga penyesuaian pencahayaan. Dengan demikian, model tidak hanya “mengingat” pola tertentu, tetapi juga belajar mengenali objek dalam berbagai kondisi.

**2. Transfer Learning**
Untuk mengatasi keterbatasan data sekaligus mempercepat pelatihan, pendekatan transfer learning sering digunakan. Teknik ini memanfaatkan model pra-latih (pre-trained model) yang telah dilatih pada dataset besar, seperti ImageNet, kemudian menyesuaikannya dengan dataset yang lebih kecil dan spesifik. Dengan cara ini, model dapat mewarisi pengetahuan umum dari data besar, lalu mengadaptasikannya pada domain baru.

**3. Pengembangan Model Ringan dan Efisien**
Kompleksitas komputasi CNN mendorong munculnya arsitektur yang lebih efisien, seperti MobileNet, SqueezeNet, dan EfficientNet. Model-model ini dirancang agar tetap mempertahankan akurasi tinggi, tetapi dengan kebutuhan memori dan komputasi yang lebih rendah, sehingga lebih sesuai untuk implementasi pada perangkat mobile atau IoT.

**4. Hybrid Model dengan Mekanisme Attention**
Kombinasi CNN dengan model lain, misalnya Transformer atau mekanisme attention, menjadi tren terkini dalam penelitian vision. Pendekatan ini memungkinkan model untuk tidak hanya mengekstraksi fitur lokal, tetapi juga memahami konteks global dalam citra. Dengan demikian,

masalah kemiripan antar kelas atau gangguan latar belakang dapat diminimalisasi, karena model lebih fokus pada area penting dari objek yang dianalisis.

**5. Explainable AI (XAI)**
Untuk mengatasi keterbatasan interpretabilitas, para peneliti mengembangkan teknik Explainable AI yang mampu memberikan penjelasan visual maupun numerik atas keputusan model. Misalnya, Grad-CAM dapat menunjukkan area citra mana yang menjadi dasar keputusan klasifikasi CNN. Dengan adanya transparansi ini, pengguna dapat lebih memahami dan memverifikasi hasil deteksi, terutama pada aplikasi kritis seperti kesehatan atau keamanan.

**6. Domain Adaptation dan Generative Models**
Untuk meningkatkan kemampuan adaptasi model terhadap variasi lingkungan, pendekatan domain adaptation mulai banyak digunakan. Model dilatih agar dapat mentransfer pengetahuan dari satu domain ke domain lain, misalnya dari citra laboratorium ke citra lapangan. Selain itu, pemanfaatan generative models seperti GAN (Generative Adversarial Network) juga populer untuk menghasilkan data sintetis yang mendekati kondisi nyata, sehingga memperkaya variasi dalam pelatihan.

## 7.6 Perbandingan Kegagalan CNN dengan Yolo

Pada bagian ini menegaskan bahwa meskipun Convolutional Neural Network (CNN) dan turunannya telah membawa lompatan besar dalam dunia computer vision, berbagai keterbatasan masih menjadi tantangan yang nyata. Faktor-faktor seperti kualitas citra, variasi pencahayaan, transformasi geometris, kemiripan antar kelas, keterbatasan dataset, hingga kompleksitas komputasi, semuanya dapat memengaruhi keandalan sistem dalam mendeteksi objek. Selain itu, pengaruh lingkungan nyata yang penuh dengan dinamika dan ketidakpastian juga memperlihatkan bahwa algoritma vision belum sepenuhnya matang untuk menghadapi semua konteks aplikasi.

Hal yang sama juga berlaku pada algoritma YOLO (You Only Look Once), yang hingga kini dikenal sebagai salah satu metode deteksi objek paling populer karena kecepatan dan efisiensinya. Meskipun mampu melakukan deteksi secara real-time dengan akurasi yang baik, YOLO tetap memiliki sejumlah keterbatasan yang perlu dicermati:

**Deteksi pada Objek Kecil**
YOLO cenderung kesulitan dalam mendeteksi objek yang berukuran sangat kecil, terutama ketika objek tersebut berada jauh dari kamera. Hal ini disebabkan oleh proses downsampling pada lapisan konvolusi yang membuat detail objek kecil sering hilang.

**Overlap dan Objek yang Saling Bertumpukan**
Pada situasi di mana banyak objek saling menutupi atau berdekatan, YOLO kadang gagal memisahkan objek dengan benar. Hal ini dapat menurunkan akurasi, terutama pada aplikasi padat objek seperti lalu lintas kendaraan atau kerumunan manusia.

**Keterbatasan pada Variasi Bentuk dan Pencahayaan**
Sama seperti CNN pada umumnya, YOLO juga sensitif terhadap variasi pencahayaan, rotasi ekstrem, maupun perubahan skala. Model dapat bekerja optimal pada kondisi data yang mirip dengan pelatihan, tetapi menurun ketika dihadapkan pada kondisi yang berbeda.

**Trade-off antara Kecepatan dan Akurasi**
YOLO dirancang dengan orientasi kecepatan. Versi awal hingga YOLOv3, misalnya, sering kali harus mengorbankan tingkat akurasi untuk mempertahankan performa real-time. Versi terbaru seperti YOLOv7 dan YOLOv8 memang lebih baik dalam menyeimbangkan keduanya, tetapi tantangan ini tetap ada.

**Keterbatasan Generalisasi**
YOLO yang dilatih dengan dataset tertentu sering kesulitan beradaptasi ke domain baru tanpa pelatihan tambahan. Misalnya, model yang dilatih pada dataset COCO mungkin tidak bekerja optimal untuk mendeteksi objek pada lingkungan industri atau pertanian tanpa proses fine-tuning.

Sejumlah strategi telah dikembangkan untuk mengatasi keterbatasan tersebut, mulai dari penggunaan arsitektur multi-skala untuk mendeteksi objek kecil, penambahan mekanisme attention untuk meningkatkan fokus model, hingga integrasi dengan transfer learning agar YOLO lebih adaptif pada domain baru.

Dengan memahami keterbatasan baik pada CNN maupun YOLO, kita dapat melihat bahwa setiap algoritma memiliki keunggulan sekaligus kelemahannya masing-masing. Justru, kesadaran akan kelemahan ini membuka jalan bagi penelitian lanjutan, seperti pengembangan model hibrida, optimalisasi komputasi, serta pemanfaatan explainable AI untuk meningkatkan transparansi sistem. Dengan demikian, arah perkembangan computer vision ke depan tidak hanya akan berfokus pada akurasi, tetapi juga pada ketahanan, efisiensi, serta kemampuan adaptasi dalam menghadapi keragaman dunia nyata.

**Tabel 7.2 Perbandingan CNN dan YOLO dalam Pendeteksian Gambar**

| Aspek | CNN | YOLO |
|---|---|---|
| Kelebihan | - Akurasi tinggi dalam klasifikasi<br>- Mampu mengekstraksi fitur bertingkat<br>- Fleksibel untuk berbagai arsitektur (ResNet, DenseNet) | - Kecepatan tinggi, real-time detection<br>- Efisien untuk aplikasi praktis (IoT, kendaraan otonom)<br>- Deteksi simultan pada banyak objek |
| Keterbatasan | - Membutuhkan dataset besar dan representatif<br>- Komputasi berat, tidak efisien di perangkat terbatas<br>- Sulit diinterpretasikan (black box) | - Trade-off antara kecepatan dan akurasi<br>- Kurang optimal mendeteksi objek kecil<br>- Membutuhkan fine-tuning untuk domain khusus |
| Faktor Kegagalan | - Sulit generalisasi pada domain baru<br>- Sensitif terhadap noise, pencahayaan, dan rotasi ekstrem<br>- Kinerja menurun pada citra berkualitas rendah | - Kesulitan mendeteksi objek bertumpukan<br>- Sensitif terhadap pencahayaan dan gangguan latar belakang<br>- Lemah dalam generalisasi ke kondisi baru |

Tabel 7.2 merangkum perbandingan antara CNN dan YOLO berdasarkan kelebihan, keterbatasan, serta faktor-faktor kegagalan yang umum ditemui. Perbandingan ini penting

karena kedua algoritma tersebut mewakili dua pendekatan yang berbeda dalam pendeteksian gambar: CNN berorientasi pada kedalaman analisis fitur, sedangkan YOLO lebih menekankan pada kecepatan deteksi secara real-time.

Dari sisi kelebihan, CNN dikenal memiliki akurasi tinggi karena kemampuannya mengekstraksi fitur bertingkat melalui lapisan konvolusi. Fleksibilitas arsitekturnya memungkinkan CNN dikembangkan dalam berbagai bentuk, seperti ResNet, DenseNet, atau EfficientNet, yang disesuaikan dengan kebutuhan spesifik. Sementara itu, YOLO unggul dalam kecepatan dan efisiensi. Dengan pendekatan *single-shot detection*, YOLO mampu mendeteksi banyak objek secara simultan, menjadikannya ideal untuk aplikasi yang menuntut respons cepat seperti kendaraan otonom atau pengawasan video.

Namun, keduanya memiliki keterbatasan yang berbeda. CNN membutuhkan dataset besar dan representatif agar dapat dilatih secara optimal. Selain itu, kompleksitas komputasi yang tinggi membuatnya sulit diimplementasikan pada perangkat dengan daya terbatas, seperti sistem IoT. CNN juga kerap dikritik karena sifatnya yang *black box*, sehingga sulit dipahami bagaimana sebuah keputusan klasifikasi dihasilkan. Sebaliknya, YOLO menghadapi keterbatasan dalam hal trade-off antara kecepatan dan akurasi. Model ini sering kali kurang optimal dalam mendeteksi objek berukuran kecil dan memerlukan *fine-tuning* tambahan jika digunakan pada domain yang berbeda dari data pelatihannya.

Pada aspek faktor kegagalan, CNN sering kali gagal dalam melakukan generalisasi ketika dihadapkan pada data baru yang berbeda dari data pelatihan. Model ini juga sensitif terhadap kondisi pencahayaan, keberadaan noise, atau perubahan sudut pandang yang ekstrem. YOLO pun menghadapi tantangan serupa, terutama ketika objek saling bertumpukan atau berada dalam latar belakang yang kompleks. Selain itu, sensitivitas terhadap variasi pencahayaan dan keterbatasan dalam beradaptasi dengan domain baru memperlihatkan bahwa YOLO juga rentan mengalami penurunan kinerja di luar skenario ideal.

Secara keseluruhan, tabel ini menegaskan bahwa CNN dan YOLO tidak dapat dipandang sebagai algoritma yang saling meniadakan, melainkan saling melengkapi. CNN memberikan kekuatan pada analisis fitur mendalam, sedangkan YOLO menghadirkan efisiensi untuk kebutuhan praktis. Kesadaran terhadap kelebihan dan keterbatasan masing-masing akan membantu peneliti maupun praktisi dalam memilih algoritma yang sesuai dengan konteks aplikasi, sekaligus mendorong pengembangan model hibrida yang mampu menggabungkan akurasi tinggi dengan efisiensi komputasi.


Gambar Perbandingan CNN vs Yolo dalam Pendeteksian Gambar

Gambar ini memperlihatkan perbandingan antara dua pendekatan populer dalam pendeteksian gambar, yaitu *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO). CNN dikenal memiliki keunggulan dalam hal akurasi dan kemampuan mengekstraksi fitur secara bertingkat. Arsitektur CNN, seperti ResNet dan DenseNet, memberikan fleksibilitas tinggi dalam mengolah berbagai jenis data citra. Namun, CNN memiliki keterbatasan yang signifikan, di antaranya kebutuhan akan dataset yang besar dan representatif, komputasi yang berat sehingga kurang efisien di perangkat terbatas, serta sifatnya yang sering dianggap sebagai *black box* karena sulit diinterpretasi. Faktor kegagalan CNN biasanya berkaitan dengan sensitivitas terhadap kualitas data, misalnya citra yang mengandung noise, pencahayaan yang buruk, atau rotasi ekstrem, sehingga model kesulitan melakukan generalisasi pada domain baru.

Di sisi lain, YOLO dikembangkan untuk menjawab kebutuhan deteksi objek dengan kecepatan tinggi. Model ini mampu melakukan *real-time detection* dan efisien untuk aplikasi praktis, seperti Internet of Things (IoT) dan kendaraan otonom. Selain itu, YOLO unggul dalam mendeteksi banyak objek secara simultan. Namun, kecepatan tersebut datang dengan kompromi, karena YOLO sering kali menghadapi *trade-off* antara kecepatan dan akurasi. Model ini juga cenderung kurang optimal dalam mendeteksi objek berukuran kecil dan memerlukan proses *fine-tuning* agar dapat bekerja maksimal pada domain tertentu. Faktor kegagalan YOLO lebih banyak ditemui pada situasi dunia nyata, seperti objek yang bertumpukan, pencahayaan yang kompleks, serta keterbatasannya dalam melakukan generalisasi pada kondisi baru yang berbeda dari data latih.

Secara keseluruhan, CNN lebih menonjol dalam hal akurasi dan kedalaman analisis fitur, sementara YOLO unggul pada aspek kecepatan dan efisiensi. Perbandingan ini menegaskan bahwa pemilihan algoritma sangat bergantung pada kebutuhan aplikasi: apakah lebih

mengutamakan akurasi tinggi untuk penelitian mendalam, atau kecepatan deteksi untuk aplikasi praktis di lapangan.

Pada bab ini menunjukkan bahwa keberhasilan algoritma vision tidak dapat dipahami hanya dari keunggulannya, melainkan juga dari keterbatasan dan faktor kegagalannya. CNN, dengan kekuatan analisis fitur bertingkat dan fleksibilitas arsitektur, menawarkan akurasi tinggi namun menuntut sumber daya besar serta menghadapi tantangan interpretabilitas. Di sisi lain, YOLO memberikan solusi praktis melalui deteksi objek real-time yang efisien, tetapi harus berhadapan dengan persoalan trade-off antara kecepatan dan akurasi, terutama pada objek kecil atau kondisi lingkungan yang kompleks.

Pemahaman mendalam terhadap kelebihan, keterbatasan, dan faktor kegagalan ini memberi gambaran yang lebih realistis tentang bagaimana algoritma vision bekerja dalam praktik. Tidak ada satu pendekatan pun yang sepenuhnya ideal; justru dari celah inilah lahir inovasi, baik berupa arsitektur baru, teknik augmentasi data, maupun integrasi dengan model lain seperti attention dan transformer.

Dengan demikian, Bab 6 tidak hanya menyoroti sisi teknis, tetapi juga menghadirkan perspektif kritis: bahwa kemajuan computer vision bergantung pada kemampuan kita mengakui keterbatasan yang ada, sekaligus merancang strategi untuk mengatasinya. Refleksi ini menjadi dasar penting bagi bab selanjutnya, yang akan membahas arah perkembangan algoritma vision dan penerapannya dalam konteks yang lebih luas.

# Bab VIII
# Dataset & Preprocessing

## 8.1 Pentingnya Dataset dalam Pengembangan Algoritma Vision

Dalam perkembangan ilmu computer vision, data berperan sebagai fondasi yang menentukan kualitas sistem yang dibangun. Algoritma, meskipun dirancang dengan arsitektur kompleks dan inovatif, tidak akan mampu menampilkan kinerja optimal tanpa dukungan dataset yang kaya, representatif, dan relevan. Oleh karena itu, dataset dapat dipandang sebagai "sumber pengetahuan empiris" yang membentuk pola pemahaman model terhadap realitas visual.

Kekuatan utama sebuah model vision terletak pada kemampuannya melakukan generalisasi, yaitu memahami pola baru yang tidak pernah muncul pada data pelatihan. Generalisasi inilah yang membedakan model yang berguna dalam praktik dengan sekadar model teoretis. Namun, kemampuan generalisasi tersebut sangat dipengaruhi oleh variasi data yang digunakan. Dataset yang homogen dan bias hanya akan melahirkan model yang unggul pada situasi terbatas tetapi gagal ketika dihadapkan pada variasi dunia nyata. Sebaliknya, dataset yang beragam dan kaya konteks akan memperluas ruang representasi model sehingga lebih robust.

Dengan demikian, diskursus tentang dataset bukanlah isu teknis semata. Ia menyangkut dimensi metodologis, epistemologis, bahkan etis, karena pada akhirnya dataset menentukan bagaimana algoritma "melihat" dan "memahami" dunia**.**

## 8.2 Sumber Dataset Publik dalam Computer Vision

Kemajuan riset computer vision tidak dapat dilepaskan dari kehadiran dataset publik berskala besar yang memungkinkan peneliti melakukan evaluasi dan pengembangan model secara terbuka. Salah satu yang paling monumental adalah **ImageNet**, sebuah dataset dengan lebih dari 14 juta gambar yang tersebar ke dalam 20 ribu kategori. Dataset ini tidak hanya menjadi repositori citra, melainkan juga mendorong lahirnya kompetisi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) yang sejak 2010 hingga 2017 menjadi barometer utama perkembangan algoritma vision. Keberhasilan arsitektur AlexNet dalam kompetisi tahun 2012, misalnya, dianggap sebagai tonggak sejarah munculnya deep learning modern. Dataset ini dapat diakses melalui laman resmi http://www.image-net.org, dengan lisensi yang umumnya mengizinkan penggunaan untuk kepentingan akademis dan riset non-komersial.

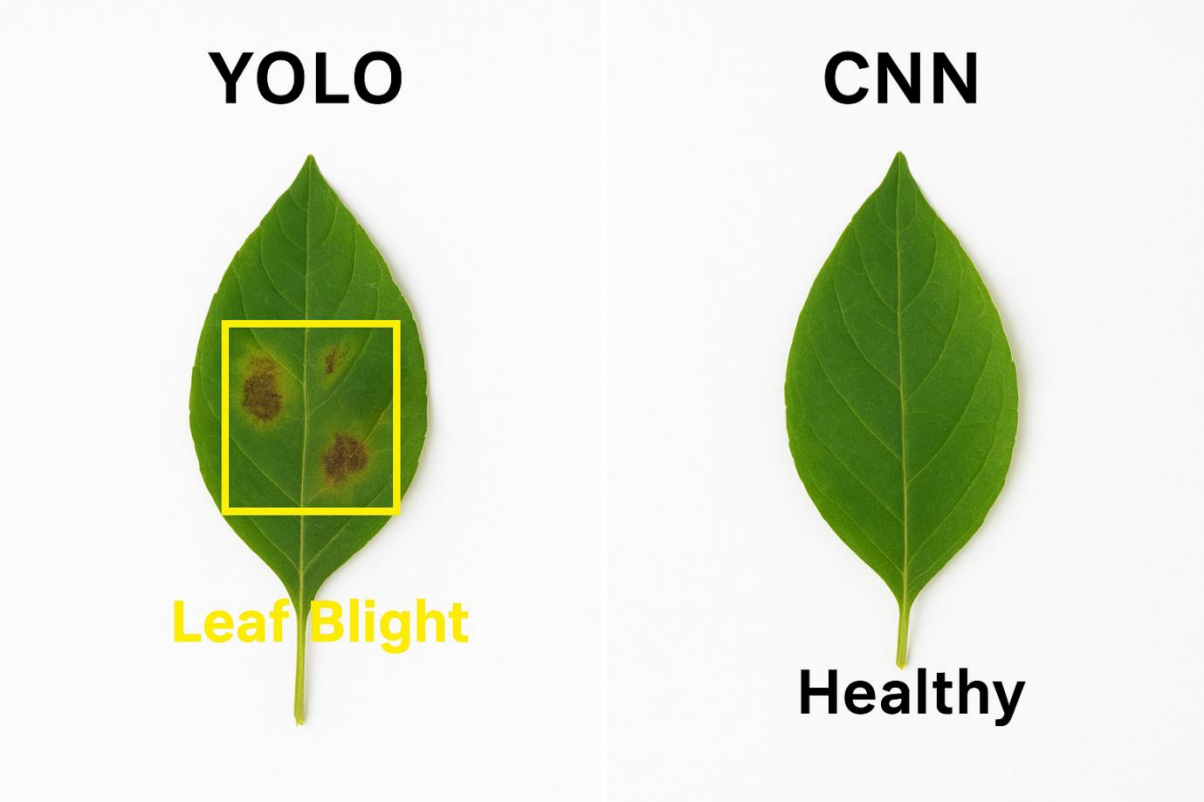
Selain ImageNet, dataset lain yang sangat berpengaruh adalah **COCO** (Common Objects in Context). Berbeda dari ImageNet yang berfokus pada klasifikasi gambar, COCO menyediakan sekitar 330 ribu gambar dengan 80 kelas objek dan menekankan pada pemahaman konteks visual. Dataset ini dirancang untuk mendukung berbagai tugas seperti deteksi objek, segmentasi instance, dan image captioning. Karena kompleksitasnya, COCO telah menjadi tolok ukur penting dalam lahirnya arsitektur-arsitektur mutakhir seperti Faster R-CNN, Mask R-CNN, dan keluarga YOLO. Dataset COCO dapat diunduh secara terbuka melalui situs

https://cocodataset.org dengan lisensi Creative Commons Attribution 4.0, sehingga penggunaannya relatif fleksibel baik untuk riset akademis maupun aplikasi praktis.

Sebelum COCO mendominasi, **Pascal VOC** merupakan dataset utama yang dipakai luas oleh komunitas vision. Dataset ini berisi sekitar 11 ribu gambar dengan anotasi berupa bounding box dan segmentasi, yang sejak 2005 hingga 2012 menjadi rujukan kompetisi internasional Visual Object Classes Challenge. Meskipun ukurannya lebih kecil dibandingkan COCO, Pascal VOC tetap penting sebagai standar awal dalam pengembangan metode deteksi dan segmentasi. Dataset ini tersedia melalui https://robots.ox.ac.uk/ dan dapat digunakan bebas untuk riset non-komersial. Untuk pembelajaran dasar, terdapat pula MNIST dan Fashion-MNIST yang hingga kini masih digunakan secara luas dalam pengajaran maupun eksperimen awal. MNIST memuat 70 ribu gambar digit tulisan tangan hitam putih, masing-masing berukuran 28x28 piksel, yang ideal untuk tugas klasifikasi sederhana. Dataset ini dapat diunduh langsung dari laman di https://docs.ultralytics.com/datasets/classify/fashion-mnist/#labels dan selain itu dataset juga dapat di donlote pada halaman tensorflow dimana di dalam halaman itu banyak terdapat juga dataset lain nya sehingga cocok untuk belajar algortima vision https://www.tensorflow.org/datasets/catalog/mnist. Sebagai alternatif yang lebih menantang, Zalando merilis Fashion-MNIST dengan jumlah data yang sama, namun berisi citra pakaian seperti sepatu, kaos, dan tas. Dataset ini bisa diakses melalui repositori resmi di https://github.com/zalandoresearch/fashion-mnist dengan lisensi MIT yang sangat terbuka.

Selain dataset umum tersebut, bidang-bidang khusus seperti medis dan pertanian juga memiliki dataset publik yang kini menjadi acuan penelitian. Dalam ranah medis, misalnya, ChestX-ray14 menyediakan lebih dari 112 ribu citra X-ray dada dari sekitar 30 ribu pasien yang dikembangkan oleh National Institutes of Health (NIH), dan dapat diunduh melalui https://nihcc.app.box.com/v/ChestXray-NIHCC. Untuk deteksi kanker kulit, komunitas internasional menyediakan ISIC Archive yang dapat diakses di https://www.isic-archive.com , serta HAM10000 yang berisi 10 ribu citra lesi kulit dan tersedia di platform Kaggle pada https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham1000.

Sementara itu, untuk kebutuhan di sektor pertanian, salah satu dataset yang populer adalah PlantVillage, yang memuat lebih dari 54 ribu citra daun tanaman sehat maupun yang terinfeksi penyakit. Dataset ini dapat diakses secara terbuka di Kaggle melalui https://www.kaggle.com/datasets/emmarex/plantdisease
. Untuk tanaman padi yang menjadi komoditas vital di Asia, tersedia pula Rice Disease Detection Dataset yang berisi sekitar 6 ribu citra daun padi dengan berbagai jenis penyakit, dapat diunduh melalui https://www.kaggle.com/datasets/minhhuy2810/rice-disease-detection-dataset
.
Keberagaman dataset publik tersebut memperlihatkan bahwa computer vision kini telah menjangkau berbagai ranah kehidupan, mulai dari kesehatan, pertanian, keamanan, hingga industri. Penyediaan link resmi untuk mengunduh dataset memungkinkan pembaca tidak hanya memahami teori, tetapi juga langsung mencoba eksperimen nyata menggunakan data standar internasional. Dengan demikian, pembelajaran computer vision akan lebih aplikatif dan berbasis pada praktik nyata yang sesuai dengan kebutuhan riset modern.




Gambar . Contoh pendeteksi Yolo dan CNN

Gambar ini menampilkan perbedaan prinsipil antara dua paradigma dalam computer vision, yakni YOLO (You Only Look Once) dan CNN (Convolutional Neural Network), dalam menganalisis citra daun untuk mendeteksi gejala penyakit.

Pada sisi kiri, ditunjukkan bagaimana YOLO bekerja dengan paradigma deteksi berbasis lokal. Model ini tidak hanya melakukan klasifikasi, tetapi juga localization, yaitu menentukan posisi spasial dari objek atau gejala yang diamati. Hal tersebut divisualisasikan melalui bounding box berwarna kuning yang mengelilingi area bercak penyakit. YOLO menggunakan pendekatan end-to-end yang membagi citra ke dalam grid, lalu secara simultan memprediksi kelas objek dan koordinat lokasinya. Keunggulan metode ini terletak pada kemampuannya mendeteksi lebih dari satu objek dalam satu citra, bahkan ketika gejala penyakit hanya muncul sebagian kecil pada permukaan daun. Informasi spasial yang dihasilkan sangat penting dalam aplikasi pertanian cerdas, karena dapat membantu pemetaan intensitas serangan penyakit, monitoring pertumbuhan infeksi, serta perencanaan tindakan pengendalian yang lebih presisi.

Sementara itu, pada sisi kanan, CNN ditampilkan dalam fungsi klasifikasi global. CNN bekerja dengan mengekstraksi fitur visual dari keseluruhan permukaan daun, kemudian menempatkan citra tersebut ke dalam salah satu kelas yang telah dipelajari, misalnya Healthy atau Leaf Blight. Informasi yang dihasilkan bersifat biner atau kategorikal tanpa disertai lokasi spesifik gejala. Pendekatan ini relatif lebih sederhana dan banyak digunakan sebagai tahap awal dalam pengembangan model pengenalan citra, terutama ketika fokus penelitian hanya pada presence detection (apakah daun sehat atau sakit). Namun, keterbatasannya jelas: CNN tidak dapat

memberikan informasi spasial yang diperlukan untuk pemetaan lebih lanjut, sehingga hasil klasifikasi masih bersifat umum.

Perbandingan ini memperlihatkan bahwa kedua pendekatan memiliki peran komplementer. CNN unggul dalam kesederhanaan, efisiensi komputasi, dan kecepatan dalam tugas klasifikasi tunggal, menjadikannya cocok untuk aplikasi edukatif maupun baseline penelitian. Sebaliknya, YOLO lebih unggul dalam situasi nyata di lapangan yang membutuhkan keakuratan spasial, misalnya untuk sistem monitoring tanaman berbasis drone, sensor kamera lapangan, atau aplikasi deteksi penyakit secara real-time di perangkat seluler. Dengan demikian, pemilihan antara CNN dan YOLO sebaiknya mempertimbangkan kebutuhan aplikasi: apakah cukup dengan mengetahui status kesehatan daun secara umum, atau diperlukan pula informasi detail tentang lokasi gejala yang terdeteksi.

Dalam konteks perkembangan riset terkini, banyak studi mengombinasikan kekuatan kedua pendekatan ini. CNN digunakan sebagai feature extractor awal, sementara YOLO dimanfaatkan untuk pelokalan detail. Integrasi semacam ini memungkinkan sistem vision dalam pertanian menjadi lebih adaptif, akurat, dan aplikatif, sehingga mampu menjawab .tantangan nyata yang dihadapi petani, khususnya dalam mendeteksi dan mengendalikan penyakit tanaman secara dini.

### 8.3 Augmentasi Data: Strategi Mengatasi Keterbatasan

Meskipun dataset publik tersedia, banyak bidang spesifik (seperti medis atau pertanian tropis) masih menghadapi keterbatasan jumlah data. Untuk menjawab masalah ini, para peneliti menggunakan teknik data augmentation, yaitu strategi memperluas variasi data dengan memodifikasi citra yang ada. Augmentasi bukan sekadar menambah jumlah sampel, melainkan berfungsi untuk meningkatkan ketahanan model terhadap variasi kondisi nyata.
Transformasi geometris seperti rotasi, flipping, translasi, scaling, dan shearing membantu model lebih adaptif terhadap perubahan posisi dan orientasi objek. Transformasi fotometris berupa penyesuaian kontras, kecerahan, saturasi, atau hue meniru variasi pencahayaan.
Penambahan Gaussian noise atau salt-and-pepper noise mengajarkan model mengenali objek meskipun terganggu kualitas visual. Dalam perkembangannya, teknik modern seperti CutMix dan MixUp yang menggabungkan dua citra berbeda terbukti meningkatkan generalisasi model. Pada domain tertentu, seperti medis atau industri, bahkan digunakan synthetic data generation. melalui Generative Adversarial Networks (GANs) atau diffusion models untuk menciptakan citra sintetis yang tetap realistis.

### 8.4 Normalisasi dan Reduksi Noise: Menyiapkan Data untuk Algoritma

Tahapan preprocessing memegang peranan penting dalam pipeline computer vision. Ia berfungsi sebagai proses penyelarasan antara data mentah dengan kebutuhan algoritma.
Preprocessing yang dilakukan dengan tepat dapat meningkatkan stabilitas pelatihan, mempercepat konvergensi, serta memperbaiki akurasi.
Strategi normalisasi piksel, misalnya, menjaga agar rentang nilai piksel konsisten pada skala tertentu ([0,1] atau [-1,1]), sehingga mempercepat proses optimasi. Histogram equalization digunakan untuk meningkatkan kontras citra, terutama pada gambar medis atau citra satelit dengan distribusi intensitas yang tidak merata. Filtering dengan Gaussian atau median filter membantu mengurangi noise, sementara autoencoder denoising menawarkan pendekatan berbasis deep learning. Akhirnya, resizing dan cropping diperlukan untuk menyesuaikan

dimensi input model sekaligus memfokuskan perhatian algoritma pada area penting dalam citra.

**Taksonomi Image Intention**

Dalam penelitian dan aplikasi computer vision, konsep image intention dapat dipetakan ke dalam beberapa kategori utama berdasarkan tingkat granularitas informasi yang diinginkan serta sifat tugas yang harus diselesaikan. Setiap kategori memiliki kebutuhan anotasi, strategi pra-pemrosesan, dan metrik evaluasi yang berbeda, sehingga pemahaman yang jelas mengenai taksonomi ini menjadi kunci dalam merancang pipeline yang sesuai.

Intention yang paling sederhana berada pada level global atau image-level, di mana sistem hanya diminta mengidentifikasi kelas utama dari sebuah citra. Contoh tipikal adalah klasifikasi gambar tunggal atau multi-label, seperti menentukan apakah sebuah daun tergolong sehat atau terinfeksi penyakit tertentu. Anotasi untuk tugas ini relatif sederhana, cukup berupa satu atau beberapa label per gambar. Pra-pemrosesan biasanya mencakup resize ke ukuran standar model (misalnya 224×224 piksel), normalisasi nilai piksel, serta augmentasi dasar seperti flipping, cropping, atau color jitter. Evaluasi dilakukan dengan metrik akurasi, top-k accuracy, atau F1-score. Dengan kata lain, intention ini berusaha menjawab pertanyaan: “Apa kelas utama pada citra ini?”

Naik satu tingkat, terdapat lokalisasi dan deteksi objek. Pada tugas ini, model tidak hanya harus mengenali kelas, tetapi juga menentukan posisi objek melalui bounding box. Contoh yang umum adalah algoritma deteksi seperti YOLO atau Faster R-CNN. Anotasi yang diperlukan lebih kompleks, yakni koordinat kotak beserta label kelas objek. Proses pra-pemrosesan biasanya melibatkan resize dengan letterboxing untuk menjaga rasio aspek citra, serta augmentasi yang mempertahankan konsistensi koordinat, seperti mosaic atau translation. Kinerja dievaluasi menggunakan metrik mean Average Precision (mAP) dengan ambang Intersection over Union (IoU). Intention ini pada dasarnya menjawab pertanyaan: “Apa objek yang ada, dan di mana lokasinya?”

Lebih detail lagi, muncul intention pada tingkat piksel atau segmentasi. Segmentasi semantik mengklasifikasikan setiap piksel ke dalam kelas tertentu, sedangkan segmentasi instansi menambahkan informasi individual untuk tiap objek, dan panoptic segmentasi menggabungkan keduanya. Anotasi dalam bentuk mask biner, poligon, atau representasi run-length encoding (RLE) diperlukan. Pra-pemrosesan mencakup augmentasi geometris yang konsisten antara gambar dan mask, normalisasi warna, serta high-resolution crops untuk menjaga detail. Evaluasi dilakukan dengan metrik Intersection over Union (IoU), mean IoU, Dice coefficient, atau average precision pada level mask. Intention ini bertujuan menjawab: “Untuk setiap piksel, kelas apa yang dimiliki?”

Selain itu, terdapat intention keypoint detection dan pose estimation, yang berfokus pada pengenalan titik-titik penting pada tubuh atau wajah, seperti sendi manusia atau landmark wajah. Anotasi berupa koordinat titik (x, y) beserta visibilitasnya, terkadang dalam bentuk heatmap. Pra-pemrosesan biasanya mencakup normalisasi posisi, alignment, serta augmentasi yang mempertahankan konsistensi geometri. Evaluasi dilakukan dengan metrik seperti Percentage of Correct Keypoints (PCK) atau Object Keypoint Similarity (OKS). Dengan

demikian, intention ini menjawab pertanyaan: “Di mana titik-titik anatomis atau landmark berada?”

Pada ranah yang berhubungan dengan pemahaman spasial, terdapat estimasi kedalaman dan geometri. Tugas ini mencakup prediksi peta kedalaman, rekonstruksi 3D, atau estimasi normal permukaan. Anotasi biasanya berupa peta kedalaman atau data point cloud dari sensor. Pra-pemrosesan mencakup kalibrasi radiometrik dan normalisasi kanal, bahkan kadang windowing pada area tertentu. Evaluasi kinerja menggunakan RMSE, error relatif absolut (Abs Rel), atau δ-thresholds. Intention ini berupaya menjawab pertanyaan: “Seberapa jauh atau bagaimana bentuk tiga dimensi dari benda di dalam citra?”

Pada data video, muncul intention temporal, seperti pelacakan multi-objek (MOT), pengenalan aksi, atau estimasi aliran optik. Anotasi yang digunakan lebih kompleks, mencakup bounding box dengan ID unik per frame untuk tracking, label aksi per rangkaian frame, atau peta aliran optik yang padat. Pra-pemrosesan biasanya melibatkan frame sampling, augmentasi temporal, atau stabilisasi citra. Evaluasi dilakukan dengan metrik MOTA dan IDF1 untuk tracking, mAP untuk pengenalan aksi, serta End-Point Error (EPE) untuk optical flow. Intention ini menjawab pertanyaan: “Bagaimana objek bergerak, dan tindakan apa yang terjadi dari waktu ke waktu?”

Lebih jauh lagi, terdapat pemahaman multimodal dan tingkat tinggi, yang menggabungkan citra dengan bahasa atau relasi antar-objek. Contoh tugas meliputi image captioning (menghasilkan deskripsi teks), visual question answering (menjawab pertanyaan berbasis gambar), dan scene graph generation (membangun relasi antar-objek). Anotasi berupa teks, pasangan pertanyaan-jawaban, atau relasi objek. Pra-pemrosesan mengintegrasikan proposal wilayah visual dengan tokenisasi teks untuk alignment multimodal. Evaluasi dilakukan menggunakan metrik linguistik seperti BLEU, METEOR, CIDEr, atau akurasi jawaban. Intention ini bertujuan menjawab: “Bagaimana menjelaskan isi gambar dalam bahasa alami atau menjawab pertanyaan berdasarkan konteks visual?”

Akhirnya, terdapat intention domain-spesifik yang muncul dalam konteks aplikasi khusus, seperti OCR untuk pengenalan teks, deteksi lesi medis, klasifikasi tutupan lahan pada citra satelit, atau person re-identification. Pada kategori ini, anotasi dan pra-pemrosesan sangat bergantung pada domain, misalnya deskewing untuk OCR, stain normalization pada citra histopatologi, atau georeferensi untuk citra penginderaan jauh. Evaluasi pun bervariasi, mulai dari Character Error Rate (CER) atau Word Error Rate (WER) pada OCR, hingga sensitivitas-spesifisitas pada medis, atau IoU untuk penginderaan jauh.

Dengan demikian, taksonomi image intention menunjukkan keragaman kebutuhan dalam computer vision. Setiap jenis intention memiliki logika anotasi, pra-pemrosesan, dan evaluasi yang unik, sekaligus menjawab pertanyaan berbeda tentang citra: apakah itu mengenai kelas global, posisi objek, detail piksel, titik anatomis, struktur spasial, dinamika temporal, pemahaman multimodal, atau kebutuhan domain tertentu. Memahami spektrum intention ini sangat penting agar rancangan sistem vision benar-benar sesuai dengan tujuan akhir yang ingin dicapai.

**Implikasi Image Intention terhadap Desain Pipeline**

Penetapan image intention sejak awal berimplikasi langsung terhadap bagaimana sebuah pipeline computer vision harus dirancang, mulai dari anotasi, pra-pemrosesan, pemilihan model, hingga evaluasi. Setiap jenis intention membawa konsekuensi teknis yang berbeda, sehingga pemahaman yang jelas mengenai implikasi ini menjadi kunci agar sistem yang dibangun benar-benar relevan dengan kebutuhan aplikasi.

Aspek pertama yang terdampak adalah anotasi. Bentuk anotasi sepenuhnya ditentukan oleh jenis intention yang diusung. Pada klasifikasi, anotasi cukup berupa label tunggal atau multi-label per citra. Pada deteksi objek, anotasi harus berupa bounding box dengan koordinat dan kelas, sebagaimana pada dataset COCO atau YOLO. Segmentasi membutuhkan mask poligon atau representasi piksel penuh, sedangkan pose estimation memerlukan anotasi titik koordinat atau heatmap. Untuk tugas kedalaman, anotasi biasanya berupa peta kedalaman yang padat (dense map). Kualitas anotasi sangat menentukan hasil akhir; kesalahan kecil pada segmentasi atau pose estimation jauh lebih berbahaya dibandingkan pada klasifikasi, karena dapat merusak representasi spasial yang menjadi inti dari intention tersebut.

Tahap berikutnya adalah pra-pemrosesan dan augmentasi. Pada deteksi objek, pra-pemrosesan harus menjaga rasio aspek citra, biasanya dengan teknik letterboxing, sementara augmentasi harus memperhitungkan ulang koordinat bounding box agar tetap konsisten, misalnya melalui mosaic augmentation atau random crop with box clipping. Pada segmentasi, transformasi augmentasi harus diterapkan seragam baik pada citra maupun mask, dan sering kali diperlukan high-resolution crop untuk mempertahankan detail. Pada tugas kedalaman atau aplikasi medis, pra-pemrosesan dapat mencakup kalibrasi radiometrik, peningkatan kontras adaptif (CLAHE), atau stain normalization pada citra histopatologi. Sedangkan pada OCR, teknik seperti deskewing, binarisasi, dan operasi morfologi sering digunakan untuk meningkatkan keterbacaan teks.

Image intention juga memengaruhi arsitektur model dan fungsi loss. Sebuah tugas klasifikasi cukup menggunakan classifier head dengan cross-entropy loss. Pada deteksi, model memerlukan detection head dengan kombinasi loss klasifikasi dan regresi posisi, misalnya IoU loss atau focal loss. Segmentasi membutuhkan decoder yang menghasilkan mask, dengan loss seperti Dice atau IoU loss. Untuk regresi kedalaman, digunakan loss berbasis L1/L2. Dalam konteks multi-task learning, di mana satu sistem melayani lebih dari satu intention sekaligus, desain pipeline harus mengakomodasi shared backbone dengan beberapa head khusus, serta mekanisme penyeimbang loss agar semua tugas dapat dipelajari secara optimal.

Selain itu, pemilihan metrik evaluasi juga sangat dipengaruhi oleh intention. Akurasi cukup untuk klasifikasi sederhana, tetapi tidak relevan untuk segmentasi yang membutuhkan mIoU atau Dice coefficient. Deteksi objek dinilai dengan mAP pada berbagai ambang IoU, sementara pose estimation menggunakan metrik Percentage of Correct Keypoints (PCK). Bahkan dalam evaluasi, penting untuk merancang skenario uji yang sesuai dengan intention aplikasi nyata, misalnya menguji deteksi objek pada kondisi occlusion atau objek kecil apabila aplikasi lapangan memang menuntut hal tersebut.

Implikasi lain muncul pada deployment constraints. Untuk intention real-time seperti YOLO pada sistem kendaraan otonom atau pengawasan video, kompromi antara akurasi dan latensi menjadi tak terelakkan. Pemilihan model ringan, serta optimisasi melalui teknik quantization

dan pruning, menjadi solusi agar model dapat berjalan pada perangkat terbatas. Sebaliknya, pada aplikasi medis dengan intention presisi tinggi, yang lebih penting adalah reprodusibilitas, interpretabilitas, dan jejak audit yang jelas, karena hasil model akan memengaruhi keputusan klinis yang kritis.

Akhirnya, image intention juga membawa implikasi etika dan privasi. Intention yang berkaitan dengan pengenalan wajah atau pengawasan publik menimbulkan pertanyaan serius mengenai bias, keadilan, dan persetujuan (consent). Oleh karena itu, desain pipeline harus mencakup pertimbangan etis, seperti audit fairness, perlindungan data, dan pembatasan penggunaan sesuai konteks sosial.

Dengan demikian, dapat disimpulkan bahwa image intention bukan hanya menentukan keluaran teknis, tetapi juga membentuk keseluruhan desain pipeline, mulai dari format data hingga aspek etis implementasi. Memahami implikasi ini memastikan bahwa teknologi vision tidak hanya akurat, tetapi juga relevan, dapat dipercaya, dan bertanggung jawab.

**Rekomendasi praktis untuk perancangan dataset & pipeline berdasarkan intention**

1) Tentukan intention dari awal — tuliskan requirement fungsional (apa keluaran yang harus dihasilkan) sehingga seluruh aktivitas anotasi dan preprocessing memiliki tujuan yang jelas.
2) Rancang pedoman anotasi terperinci — definisikan kelas, aturan bounding box (berapa banyak objek tumpang tindih yang diizinkan), definisi batas mask, format file. Lakukan pilot anotasi untuk mengukur inter-annotator agreement.
3) Sesuaikan augmentasi dengan intention — hindari augmentasi yang merusak sinyal yang relevan (mis. aggressive color jitter untuk deteksi cancer stain).
4) Sediakan negative / hard negative examples — untuk deteksi/ocr/medical agar model tidak overfit pada kondisi ideal.
5) Pilih metrik yang benar — jangan gunakan hanya accuracy ketika intention adalah localization (pakai mAP/IoU).
6) Pertimbangkan deployment constraints — jika intention butuh real-time, validasi latency dan throughput pada perangkat target.
7) Perhatikan etika & bias — jika intention menyangkut orang (face, crowd analytics), lakukan audit fairness dan privacy impact assessment.

## 8.5 Pra-pemrosesan untuk YOLO: Deteksi Objek Real-Time

YOLO (You Only Look Once) dirancang untuk melakukan deteksi objek secara cepat dan akurat dalam satu kali inferensi. Berbeda dengan model klasifikasi yang hanya memberikan label global untuk sebuah citra, YOLO memerlukan citra yang dipersiapkan sedemikian rupa agar model mampu mengenali objek, menentukan posisi spasialnya, serta memberikan label yang sesuai.

Tahap pertama adalah penyeragaman resolusi citra. YOLO memerlukan ukuran input tetap, umumnya 416×416 atau 640×640 piksel. Untuk menjaga proporsi objek, citra asli biasanya di-resize menggunakan teknik letterboxing sehingga tidak terjadi distorsi bentuk. Tahap berikutnya adalah normalisasi nilai piksel ke rentang [0–1]. Normalisasi ini bukan hanya

memudahkan perhitungan gradien dalam proses pelatihan, tetapi juga menstabilkan distribusi intensitas antar-citra.

Tahap khusus yang menjadi pembeda YOLO adalah pemberian anotasi bounding box. Setiap citra dalam dataset harus dilengkapi koordinat objek yang berada di dalamnya, dinyatakan dalam format relatif (x_center, y_center, width, height) terhadap ukuran citra. Anotasi ini berfungsi sebagai ground truth yang memungkinkan model tidak hanya mengenali kelas objek, tetapi juga mempelajari letak spasialnya. Akurasi dan konsistensi anotasi sangat menentukan kualitas deteksi yang dihasilkan.

Selanjutnya dilakukan augmentasi data. Teknik ini meliputi rotasi, flipping, perubahan kontras, hingga strategi khusus seperti mosaic augmentation yang menggabungkan empat citra ke dalam satu bingkai. Augmentasi ini memperkaya variasi dataset sehingga model lebih tangguh menghadapi kondisi nyata yang penuh variasi latar belakang dan pencahayaan.

Dari perspektif image intention, pra-pemrosesan YOLO bertujuan memaksimalkan kemampuan sistem untuk menjawab pertanyaan "Apa objek yang ada di dalam citra, dan di mana posisinya?". Oleh karena itu, setiap tahap resize, normalisasi, augmentasi, dan anotasidirancang untuk menjaga keterbacaan objek dan meningkatkan kemampuan model dalam mengaitkan label dengan posisi visual.

### 8.6 Pra-pemrosesan untuk CNN: Klasifikasi dan Ekstraksi Fitur

Convolutional Neural Networks (CNN) pada dasarnya digunakan untuk mengenali pola visual dalam citra dan mengklasifikasikannya ke dalam kategori tertentu. Agar CNN dapat bekerja dengan baik, citra harus diproses terlebih dahulu sehingga informasi yang diberikan lebih seragam dan representatif.

Tahap pertama adalah resize citra ke ukuran standar sesuai arsitektur yang digunakan. Misalnya, AlexNet menggunakan 227×227 piksel, sementara ResNet dan VGG menggunakan 224×224 piksel. Ukuran ini bukan angka sembarangan, melainkan hasil optimisasi agar jaringan dapat menangkap pola lokal sekaligus menjaga efisiensi komputasi.

Selanjutnya adalah normalisasi nilai piksel. CNN bekerja lebih stabil bila nilai piksel berada pada rentang [0–1] atau dinormalisasi menggunakan z-score. Normalisasi ini membuat distribusi input lebih seimbang, sehingga proses pembelajaran tidak bias terhadap fitur tertentu. Dalam banyak kasus, mean subtraction (pengurangan nilai rata-rata dataset dari setiap piksel) juga digunakan agar distribusi data lebih terpusat.

Tahap berikutnya adalah augmentasi data untuk memperluas variasi input. Augmentasi yang umum pada CNN meliputi rotasi, flipping horizontal, random cropping, dan manipulasi warna (color jitter). Dengan cara ini, CNN belajar menjadi lebih robust terhadap variasi pose, pencahayaan, maupun latar belakang.

Selain itu, dalam beberapa kasus dilakukan transformasi ruang warna. CNN biasanya menggunakan RGB, tetapi untuk aplikasi tertentu—misalnya pengenalan tekstur atau citra medis—grayscale atau HSV lebih sesuai. Transformasi ini membantu CNN fokus pada informasi yang relevan dengan konteks aplikasi.

Dari perspektif image intention, pra-pemrosesan CNN ditujukan untuk menjawab pertanyaan "Apa isi utama citra ini?". Berbeda dengan YOLO yang berfokus pada "apa" dan "di mana", CNN lebih menekankan pada pengenalan kelas global citra. Oleh karena itu, pra-pemrosesan CNN diarahkan untuk menghasilkan input yang konsisten, bebas gangguan, dan representatif terhadap kelas yang diinginkan.

Tabel 8.x Perbandingan Pra-pemrosesan YOLO dan CNN

| Aspek | YOLO (Deteksi Objek Real-Time) | CNN (Klasifikasi & Ekstraksi Fitur) |
|---|---|---|
| Tujuan | Deteksi objek dengan koordinat *bounding box* | Klasifikasi citra atau ekstraksi fitur hierarkis |
| Ukuran Input | Tetap (416×416 atau 640×640), menggunakan *letterboxing* | Tetap (224×224 atau 227×227), *resize* langsung |
| Normalisasi | Skala piksel ke [0–1] | Skala piksel ke [0–1] atau standarisasi (z-score) |
| Augmentasi | Rotasi, flipping, perubahan kontras/brightness, mosaic | Rotasi, flipping, cropping, *color jitter* |
| Transformasi Warna | Umumnya RGB, augmentasi fotometris untuk robustnes | Bisa RGB, grayscale, atau HSV sesuai aplikasi |
| Anotasi | Wajib: *bounding box* dengan format YOLO | Tidak diperlukan, cukup label kelas |
| Keluaran Awal Model | Prediksi kelas + posisi objek (x, y, w, h) | Prediksi kelas citra secara keseluruhan |
| Fokus Pra-pemrosesan | Menjaga proporsi objek & menyiapkan anotasi | Menstabilkan distribusi input & memperluas variasi data |

Tabel 8.x memberikan gambaran komprehensif mengenai perbedaan tahapan pra-pemrosesan antara YOLO dan CNN. Kedua algoritma ini memang sama-sama menggunakan citra digital sebagai input, namun memiliki kebutuhan yang berbeda sesuai dengan tujuan masing-masing.

Dari sisi tujuan utama, YOLO dirancang untuk mendeteksi objek sekaligus menentukan posisinya melalui koordinat bounding box. CNN, sebaliknya, berfokus pada klasifikasi citra secara keseluruhan atau ekstraksi fitur hierarkis tanpa memperhatikan lokasi objek secara spesifik. Perbedaan tujuan ini kemudian memengaruhi keseluruhan desain pipeline pra-pemrosesan. Pada ukuran input, YOLO membutuhkan citra dengan resolusi tetap, umumnya 416×416 atau 640×640 piksel. Proses letterboxing digunakan agar proporsi objek tetap terjaga. CNN juga memerlukan ukuran input standar, misalnya 224×224 atau 227×227 piksel, tetapi proses resize dilakukan secara langsung karena yang dikejar adalah konsistensi ukuran, bukan pelestarian aspek spasial objek.

Tahap normalisasi nilai piksel juga memperlihatkan perbedaan kecil. YOLO umumnya menggunakan skala [0–1] untuk menyederhanakan rentang nilai, sedangkan CNN lebih fleksibel. Selain skala [0–1], CNN sering menggunakan normalisasi berbasis distribusi (z-score) atau mean subtraction, sehingga nilai rata-rata dataset terpusat pada nol. Pada augmentasi data, YOLO dan CNN sama-sama memanfaatkan transformasi geometris dan fotometris untuk memperluas variasi dataset. Namun, YOLO memiliki teknik khas yaitu mosaic augmentation, yang menggabungkan empat citra berbeda menjadi satu. Teknik ini terbukti meningkatkan kemampuan YOLO dalam mengenali objek pada latar belakang yang kompleks. CNN lebih mengandalkan augmentasi klasik seperti rotasi, flipping, cropping, dan variasi warna (color jitter) untuk mengatasi variasi pose, pencahayaan, maupun noise.

Aspek transformasi warna memperlihatkan fleksibilitas CNN. Jika YOLO hampir selalu menggunakan ruang warna RGB dengan sedikit manipulasi fotometris, CNN sering kali mengubah citra ke grayscale atau HSV sesuai dengan kebutuhan aplikasi. Hal ini menunjukkan bahwa CNN lebih adaptif dalam hal representasi warna, terutama pada aplikasi medis atau tekstur. Perbedaan paling fundamental terlihat pada anotasi. YOLO menuntut adanya anotasi bounding box yang detail untuk setiap objek dalam citra. Informasi ini krusial agar model dapat belajar mengenali lokasi sekaligus kelas objek. CNN tidak memerlukan anotasi spasial; cukup label kelas global citra sudah memadai.

Akhirnya, perbedaan pra-pemrosesan ini berhubungan dengan keluaran awal model. YOLO menghasilkan prediksi kelas beserta koordinat posisi objek, sedangkan CNN hanya memberikan prediksi kelas citra secara keseluruhan. Dengan demikian, fokus pra-pemrosesan YOLO adalah menjaga proporsi objek dan menyiapkan anotasi, sementara fokus CNN adalah menstabilkan distribusi input agar klasifikasi lebih konsisten.

### 8.7 Image Intention dalam Computer Vision

Salah satu konsep fundamental yang sering diabaikan dalam perancangan sistem computer vision adalah image intention, yakni tujuan atau niat utama dari pemrosesan sebuah citra. Image intention pada dasarnya menjawab pertanyaan sederhana: “informasi apa yang sebenarnya ingin kita keluarkan dari gambar ini?”. Pertanyaan ini tampak sederhana, tetapi memiliki implikasi yang sangat luas karena menentukan hampir seluruh keputusan teknis, mulai dari cara dataset dibuat dan dianotasi, bagaimana citra dipra-proses, arsitektur model yang dipilih, hingga metrik evaluasi yang digunakan untuk menilai kinerja sistem.

Secara umum, terdapat beberapa jenis image intention yang dapat dibedakan berdasarkan tingkat granularitas informasi yang diinginkan. Intention yang paling sederhana adalah klasifikasi global pada tingkat citra. Dalam pendekatan ini, model diminta memberikan satu atau beberapa label yang menggambarkan isi utama gambar. Misalnya, sebuah foto daun jagung akan diberi label “sehat” atau “terinfeksi penyakit X”. Kebutuhan anotasi pada level ini relatif sederhana, hanya berupa label kelas tanpa detail spasial. Pra-pemrosesan pun berfokus pada penyeragaman ukuran citra, normalisasi, serta augmentasi sederhana seperti rotasi atau flipping.

Jenis berikutnya adalah lokalisasi dan deteksi objek, di mana sistem tidak hanya dituntut menjawab “apa” objek dalam gambar, tetapi juga “di mana” objek tersebut berada. Pada intention ini, setiap citra perlu dilengkapi anotasi berupa bounding box yang menandai posisi spasial objek. Model seperti YOLO dan Faster R-CNN dirancang khusus untuk memenuhi kebutuhan ini. Konsekuensinya, pipeline pra-pemrosesan juga lebih kompleks, mencakup

resize dengan letterboxing agar proporsi objek terjaga, serta augmentasi yang disesuaikan dengan anotasi seperti mosaic augmentation. Evaluasi kinerja pun tidak lagi menggunakan akurasi sederhana, tetapi metrik khusus seperti mean Average Precision (mAP) dengan ambang Intersection over Union (IoU).

Jika intention diperluas lebih jauh ke tingkat piksel, maka muncullah tugas segmentasi citra. Dalam kasus ini, setiap piksel harus dilabeli sesuai dengan kelas tertentu, baik itu segmentasi semantik (kelas per piksel) maupun segmentasi instansi (mask terpisah untuk tiap objek). Intention ini jauh lebih menuntut karena kualitas anotasi harus sangat tinggi dan konsisten. Pra-pemrosesan pada segmen ini mencakup transformasi yang berlaku seragam untuk gambar dan mask, serta cropping resolusi tinggi untuk menjaga detail spasial. Evaluasi pun menggunakan metrik khusus seperti mean IoU atau koefisien Dice.

Selain segmentasi, terdapat pula intention yang berfokus pada deteksi titik kunci (keypoints) atau estimasi pose. Misalnya, pendeteksian titik sendi pada tubuh manusia untuk sistem olahraga cerdas, atau landmark wajah untuk aplikasi biometrik. Intention ini menuntut anotasi koordinat titik yang presisi, serta augmentasi yang menjaga hubungan geometri antar-titik. Kinerja model biasanya diukur menggunakan metrik seperti Percentage of Correct Keypoints (PCK).

Pada ranah yang lebih lanjut, muncul pula intention lain seperti estimasi kedalaman (depth estimation), tracking objek dalam video, hingga pemahaman multimodal yang menggabungkan citra dan bahasa, seperti image captioning atau visual question answering. Masing-masing intention tersebut membutuhkan format anotasi yang unik, pra-pemrosesan khusus, serta arsitektur model yang berbeda.

Menariknya, dalam banyak aplikasi nyata, sebuah sistem sering kali memiliki lebih dari satu image intention. Kendaraan otonom, misalnya, tidak hanya membutuhkan deteksi objek (kendaraan, pejalan kaki, rambu), tetapi juga segmentasi jalur jalan, estimasi kedalaman untuk mengukur jarak, serta pelacakan temporal untuk memprediksi pergerakan objek. Dalam konteks ini, sistem harus dirancang sebagai multi-task learning dengan backbone bersama yang memiliki beberapa head keluaran sesuai masing-masing intention. Tantangan utamanya adalah menyeimbangkan bobot pelatihan dan memilih metrik evaluasi yang tepat untuk setiap keluaran.

Dari perspektif konseptual, image intention memiliki implikasi etis dan praktis yang signifikan. Untuk aplikasi medis, misalnya, intention yang berfokus pada segmentasi tumor atau deteksi lesi harus mempertimbangkan konsistensi anotasi oleh ahli, transparansi keputusan, dan akuntabilitas model. Sementara itu, pada aplikasi komersial seperti pengawasan publik berbasis deteksi wajah, intention menuntut pertimbangan serius mengenai privasi, keadilan, dan potensi bias. Dengan kata lain, image intention bukan hanya aspek teknis, tetapi juga mencerminkan nilai-nilai sosial yang melatarbelakangi penggunaan teknologi.

Secara keseluruhan, memahami dan mendefinisikan image intention sejak awal merupakan langkah penting dalam setiap penelitian maupun aplikasi computer vision. Intention menjadi kompas yang mengarahkan desain dataset, pra-pemrosesan, pemilihan algoritma, serta evaluasi hasil. Dengan penetapan intention yang jelas, sistem vision tidak hanya dapat mencapai performa teknis yang tinggi, tetapi juga relevan, adil, dan bermanfaat bagi kebutuhan dunia nyata

Tabel 8.x Jenis-Jenis Image Intention dalam Computer Vision

| Jenis Intention | Keluaran yang Diharapkan | Anotasi yang Dibutuhkan | Pra-pemrosesan Khas | Metrik Evaluasi Umum |
|---|---|---|---|---|
| **Klasifikasi Global** | Label kelas citra (single/multi-label) | Label per citra | Resize ke ukuran standar, normalisasi, augmentasi (flip, crop, color jitter) | Accuracy, Top-k accuracy, F1-score |
| **Deteksi Objek** | Label kelas + koordinat *bounding box* | Bbox (x, y, w, h) + kelas | Resize dengan letterbox, normalisasi, augmentasi box-aware (mosaic, scaling) | mAP (IoU), Recall, Precision |
| **Segmentasi Semantik** | Label kelas per piksel | Mask kelas (per pixel) | Transformasi seragam gambar & mask, high-res crop, augmentasi konsisten | IoU/mIoU, Dice coefficient |
| **Segmentasi Instansi** | Mask per objek + label kelas | Poligon/mask per instansi | Sama dengan semantic, ditambah instance balancing | AP_mask, AP_box, mIoU |
| **Keypoint/Pose** | Lokasi koordinat (x, y) tiap landmark | Titik koordinat + visibilitas | Normalisasi pose, augmentasi geometri (rotasi, scaling) | PCK (Percentage of Correct Keypoints), OKS |
| **Estimasi Kedalaman** | Peta kedalaman (depth map) | Peta kedalaman per piksel | Radiometric calibration, normalize range, cropping | RMSE, Abs Rel Error, δ-thresholds |
| **Tracking Objek** | Trajektori + ID objek per frame | Bbox + ID objek tiap frame | Frame sampling, temporal augmentation, resize | MOTA, IDF1, MOTP |
| **Pemahaman Multimodal** | Deskripsi teks (caption) atau jawaban (VQA) | Caption, pasangan Q-A | Ekstraksi fitur region, tokenisasi teks, alignment multimodal | BLEU, METEOR, CIDEr, VQA Accuracy |

Tabel 8.x merangkum berbagai jenis image intention dalam computer vision, yaitu tujuan utama dari pemrosesan citra yang menentukan bentuk keluaran, kebutuhan anotasi, teknik pra-pemrosesan, serta metrik evaluasi yang sesuai. Ringkasan ini membantu memahami bahwa meskipun semua sistem vision berangkat dari data citra yang sama, perbedaan tujuan akan menghasilkan pipeline teknis yang berbeda.

Jenis paling dasar adalah klasifikasi global, di mana model hanya dituntut memberikan label kelas untuk sebuah citra secara keseluruhan. Tugas ini relatif sederhana karena anotasi cukup berupa satu label per gambar, dan pra-pemrosesan berfokus pada penyeragaman ukuran, normalisasi, serta augmentasi sederhana. Evaluasi kinerja biasanya dilakukan dengan metrik akurasi atau F1-score.

Pada deteksi objek, intention bergeser menjadi pengenalan sekaligus pelokalan objek. Selain label kelas, model juga harus mengeluarkan koordinat spasial berupa bounding box. Oleh karena itu, anotasi lebih kompleks, dan pra-pemrosesan harus mempertahankan proporsi objek, misalnya dengan letterboxing. Evaluasi dilakukan menggunakan metrik khusus seperti mean Average Precision (mAP) dengan berbagai ambang IoU.

Jika kebutuhan ditingkatkan ke level piksel, muncullah segmentasi citra. Segmentasi semantik memberikan label kelas untuk setiap piksel, sedangkan segmentasi instansi menambahkan pembeda antar-objek dari kelas yang sama. Karena kompleksitasnya, anotasi segmentasi berupa mask atau poligon membutuhkan ketelitian tinggi, dan augmentasi harus diterapkan konsisten baik pada gambar maupun mask. Evaluasi biasanya menggunakan IoU, mIoU, atau koefisien Dice.

Jenis intention lain adalah deteksi titik kunci (keypoint) dan estimasi pose, yang berfokus pada posisi landmark seperti sendi tubuh atau titik wajah. Anotasi berupa koordinat titik, dan pra-pemrosesan menekankan pada pelestarian struktur geometris. Keberhasilan biasanya diukur dengan PCK atau OKS.

Pada ranah yang lebih mendalam, terdapat estimasi kedalaman, yang bertujuan menghasilkan peta jarak tiap piksel dari kamera. Anotasi berupa peta kedalaman atau data sensor tambahan, dan pra-pemrosesan sering mencakup kalibrasi radiometrik. Evaluasi dilakukan menggunakan RMSE, error relatif absolut, atau δ-thresholds.

Untuk aplikasi video, intention berkembang menjadi tracking objek, di mana model harus menjaga konsistensi identitas objek dari satu frame ke frame berikutnya. Hal ini memerlukan anotasi bounding box beserta ID objek di setiap frame. Evaluasi dilakukan menggunakan metrik khusus seperti MOTA dan IDF1.

Akhirnya, pada tingkat yang lebih tinggi, muncul pemahaman multimodal, di mana citra tidak hanya diproses untuk menghasilkan keluaran visual, tetapi juga untuk dihubungkan dengan representasi bahasa. Contohnya adalah image captioning atau visual question answering (VQA). Pada kasus ini, anotasi berupa kalimat deskriptif atau pasangan pertanyaan-jawaban, dan evaluasi menggunakan metrik linguistik seperti BLEU, METEOR, atau CIDEr.

Secara keseluruhan, tabel ini menunjukkan bahwa setiap jenis image intention membawa konsekuensi teknis yang berbeda: mulai dari format anotasi, strategi pra-pemrosesan, arsitektur model, hingga cara evaluasi performa. Memahami perbedaan ini sangat penting bagi peneliti maupun praktisi agar pipeline yang dibangun sesuai dengan kebutuhan nyata dan tujuan akhir aplikasi.

## 8.8 Dampak Preprocessing terhadap Kinerja Model

Banyak penelitian menunjukkan bahwa preprocessing dapat memberikan peningkatan kinerja yang signifikan. Pada deteksi penyakit daun padi, normalisasi warna dan histogram equalization terbukti meningkatkan akurasi klasifikasi hingga 25%. Dalam analisis radiologi, teknik denoising meningkatkan sensitivitas model dalam mendeteksi kelainan kecil yang sebelumnya tersembunyi. Pada sistem pengenalan wajah, augmentasi rotasi dan flipping memungkinkan model tetap akurat meskipun wajah dipotret dari sudut yang berbeda. Dengan demikian, preprocessing bukanlah tahap tambahan, melainkan komponen integral yang menentukan keberhasilan keseluruhan sistem vision.

## 8.9 Tantangan dan Isu Etis dalam Pengelolaan Dataset

Selain aspek teknis, isu konseptual dan etis perlu mendapat perhatian. Banyak dataset populer misalnya bias terhadap data negara maju, sehingga model yang dilatih tidak selalu akurat ketika diterapkan di negara berkembang. Pada domain medis, privasi pasien harus dijaga dengan ketat sesuai regulasi seperti GDPR. Masalah data imbalance juga sering muncul, di mana jumlah sampel untuk kelas tertentu jauh lebih sedikit dibanding kelas lainnya, menyebabkan model condong pada kelas mayoritas. Terakhir, aspek legal dan lisensi juga penting, karena tidak semua dataset bebas digunakan secara komersial.

# BAB . IX
# Evaluasi dan Benchmarking Algoritma dalam Computer Vision

## 9.1 Pendahuluan

Dalam penelitian computer vision, algoritma tidak hanya diukur dari kecanggihan arsitekturnya atau kompleksitas matematis yang melandasinya, melainkan juga dari sejauh mana ia mampu memberikan kinerja yang terukur, konsisten, dan dapat dibandingkan secara obyektif. Evaluasi merupakan aspek krusial, sebab tanpa pengukuran yang jelas, sulit bagi peneliti untuk mengetahui apakah suatu model benar-benar unggul atau sekadar tampak baik pada kasus terbatas.

Lebih jauh, evaluasi berfungsi sebagai mekanisme kontrol dalam dunia ilmiah. Melalui evaluasi, hasil penelitian dapat diuji ulang, diverifikasi, bahkan ditantang oleh komunitas riset. Oleh karena itu, benchmarking—yakni praktik membandingkan algoritma pada dataset dan metrik yang telah disepakati bersama—menjadi pilar penting dalam memastikan transparansi serta kemajuan kolektif bidang computer vision. Benchmark tidak hanya berperan sebagai tolok ukur kuantitatif, tetapi juga sebagai arena kompetisi intelektual, tempat berbagai pendekatan diuji dalam kondisi yang sama sehingga keunggulan atau kelemahannya dapat terlihat secara nyata**.**

## 9.2 Prinsip Dasar Evaluasi Algoritma

Evaluasi algoritma dalam computer vision bukan sekadar persoalan mengukur performa melalui angka, melainkan juga sebuah proses ilmiah yang berakar pada prinsip obyektivitas, reprodusibilitas, dan relevansi. Ketiga prinsip ini membentuk kerangka epistemologis yang membedakan penelitian ilmiah dengan sekadar eksperimen teknis.

### 9.2.1 Evaluasi sebagai Proses Ilmiah

Dalam tradisi ilmu komputer, evaluasi tidak bisa dilepaskan dari pertanyaan mendasar: apakah suatu algoritma benar-benar lebih baik daripada yang lain, dan dalam konteks apa? Pertanyaan ini menegaskan bahwa evaluasi bukanlah entitas yang berdiri sendiri, tetapi bagian dari dialektika riset, di mana klaim inovasi harus diuji secara terbuka dengan kriteria yang dapat diverifikasi.

Sejak Thomas Kuhn memperkenalkan konsep paradigm shift, jelas bahwa perkembangan ilmu tidak hanya dipengaruhi oleh ide baru, tetapi juga oleh mekanisme evaluasi yang membuat ide tersebut diterima atau ditolak komunitas. Dalam computer vision, mekanisme itu diwujudkan melalui benchmarking, publikasi hasil eksperimen, serta adopsi standar metrik tertentu.

Dengan demikian, evaluasi dapat dipandang sebagai “bahasa bersama” yang memungkinkan komunitas vision menilai validitas klaim secara obyektif. Tanpa evaluasi yang terukur, inovasi algoritmik hanya akan menjadi narasi retoris tanpa pijakan ilmiah.

### 9.2.2 Dimensi Obyektivitas dan Reprodusibilitas

Dua aspek mendasar dalam evaluasi adalah obyektivitas dan reprodusibilitas.

Obyektivitas berarti hasil evaluasi tidak boleh bergantung pada preferensi pribadi peneliti, melainkan pada ukuran yang terstandarisasi. Misalnya, perbandingan kinerja YOLOv8 dan Faster R-CNN harus menggunakan metrik yang sama (misalnya mAP pada dataset COCO) agar kesimpulan bersifat adil.

Reprodusibilitas mengandaikan bahwa hasil eksperimen dapat diulang oleh peneliti lain dengan hasil yang sebanding. Reprodusibilitas bukan hanya soal teknis (misalnya penggunaan random seed yang sama), melainkan juga menyangkut keterbukaan kode, dokumentasi metodologi, serta ketersediaan dataset. Tanpa hal ini, hasil riset hanya menjadi klaim yang tidak dapat diverifikasi.

Bagi mahasiswa doktoral, pemahaman terhadap kedua dimensi ini penting bukan hanya untuk membaca literatur, tetapi juga untuk menyusun penelitian yang kokoh secara metodologis.

### 9.2.3 Relevansi Kontekstual dalam Evaluasi

Prinsip penting berikutnya adalah relevansi aplikasi. Sebuah model tidak dapat dinilai baik atau buruk tanpa mempertimbangkan konteks penggunaannya.

Sebagai contoh, model deteksi penyakit daun dengan akurasi 95% mungkin dianggap sangat baik di laboratorium. Namun, jika model tersebut membutuhkan perangkat GPU besar dan tidak dapat dijalankan di perangkat edge seperti ESP32-CAM, maka kegunaannya di lapangan pertanian sangat terbatas. Sebaliknya, model dengan akurasi lebih rendah tetapi dapat berjalan di perangkat sederhana dengan konsumsi energi minim bisa jadi lebih relevan secara praktis.

Hal ini menegaskan bahwa evaluasi bukan hanya soal metrik numerik, tetapi juga kesesuaian dengan kebutuhan domain aplikasi. Dalam penelitian doktoral, mahasiswa dituntut untuk mampu menjembatani antara evaluasi kuantitatif dan analisis kualitatif terhadap konteks nyata.

### 9.2.4 Evaluasi sebagai Aktivitas Multi-Dimensi

Evaluasi algoritma dalam computer vision harus dilihat sebagai aktivitas multi-dimensi:

- Dimensi teknis: akurasi, kecepatan inferensi, efisiensi memori.
- Dimensi praktis: kompatibilitas dengan perangkat keras, skalabilitas, dan kemampuan real-time.
- Dimensi etis: keberpihakan, potensi bias, serta dampak sosial.
- Dimensi ekologis: konsumsi energi dan keberlanjutan.

### 9.3 Metrik Evaluasi (Precision, Recall, F1-Score, mAP, IoU)

Metrik evaluasi adalah fondasi kuantitatif yang digunakan untuk menilai kinerja algoritma computer vision. Tanpa metrik yang jelas, perbandingan antar algoritma akan bersifat subjektif dan tidak dapat dipertanggungjawabkan secara ilmiah.

- Precision
  Precision mengukur proporsi prediksi positif yang benar-benar relevan. Dalam konteks deteksi penyakit daun, precision tinggi berarti sebagian besar daun yang terdeteksi sebagai sakit memang benar sakit. Precision penting ketika false positive harus ditekan, misalnya pada sistem medis di mana salah diagnosis dapat menimbulkan kecemasan atau biaya tambahan.

- Recall (Sensitivity)
  Recall mengukur sejauh mana model berhasil menangkap semua kasus positif. Recall tinggi berarti hampir semua daun sakit berhasil terdeteksi. Metrik ini sangat penting pada domain medis atau pertanian, di mana melewatkan kasus penyakit (false negative) dapat berakibat fatal.

- F1-Score
  F1-Score adalah rata-rata harmonis antara precision dan recall. Metrik ini menjadi penting ketika dataset bersifat tidak seimbang, misalnya kasus penyakit yang jarang terjadi dibanding daun sehat. Dengan F1-score, kita memperoleh gambaran yang lebih adil tentang kinerja model dalam menyeimbangkan sensitivitas dan ketepatan.

- Intersection over Union (IoU)
  IoU digunakan untuk mengukur kesesuaian bounding box prediksi dengan ground truth. IoU bernilai 1 bila kotak prediksi menumpuk sempurna, dan bernilai 0 bila tidak ada tumpang tindih. IoU adalah standar utama dalam deteksi objek, karena ia mengukur seberapa tepat model mampu memetakan lokasi objek.

- Mean Average Precision (mAP)
  mAP adalah metrik gabungan yang menghitung rata-rata presisi pada berbagai tingkat recall dan threshold IoU. Pada dataset COCO, mAP dihitung dengan IoU bervariasi dari 0,5 hingga 0,95, sehingga benar-benar menuntut model untuk tidak hanya tepat mendeteksi objek besar, tetapi juga objek kecil atau parsial. Oleh karena itu, mAP sering dianggap sebagai “mata uang” utama dalam kompetisi deteksi objek.

## 9.4 Benchmark Dataset Internasional (COCO, ImageNet, Pascal VOC)

Benchmark dataset berfungsi sebagai "lapangan uji bersama" bagi komunitas riset. Dengan dataset standar, perbandingan antar algoritma menjadi adil dan reprodusibel.

### COCO (Common Objects in Context)

- Memuat lebih dari 330.000 citra dengan 80 kategori objek.
- Anotasi tidak hanya berbentuk bounding box, tetapi juga segmentasi instance dan keypoints.
- Menjadi standar utama untuk tugas deteksi objek dan segmentasi modern.

- Digunakan untuk mengevaluasi YOLO, Faster R-CNN, dan Vision Transformer dalam kondisi kompleks, misalnya objek kecil, tumpang tindih, atau latar belakang ramai.

### ImageNet

Dataset dengan jutaan citra beranotasi, mencakup lebih dari 20.000 kelas.

- Kompetisi tahunan ILSVRC (ImageNet Large Scale Visual Recognition Challenge) mendorong lahirnya arsitektur revolusioner: AlexNet (2012), VGG (2014), ResNet (2015).
- Lebih banyak digunakan untuk klasifikasi citra, tetapi juga menjadi dasar pretraining bagi model vision modern.

### Pascal VOC

- Dataset bersejarah dengan 20 kategori objek.
- Lebih kecil dibanding COCO, tetapi berperan penting dalam membentuk metodologi evaluasi, terutama dalam memperkenalkan mAP sebagai metrik standar.
- Masih digunakan sebagai benchmark ringan untuk pengujian awal model baru.

  Dataset ini memiliki peran berbeda: ImageNet sebagai pretraining dataset, Pascal VOC sebagai benchmark dasar, dan COCO sebagai tolok ukur utama untuk algoritma deteksi dan segmentasi kontemporer.

## 9.5 Studi Perbandingan Kinerja Model Vision

Studi perbandingan bertujuan memahami keunggulan dan kelemahan relatif antar model.

- **CNN Klasik (Faster R-CNN, ResNet)**
  - Kuat dalam akurasi dan detail, terutama pada objek berukuran menengah hingga besar.
  - Namun, lambat dalam inferensi dan membutuhkan perangkat keras kuat.

- **YOLO dan Keluarganya**

    - Menekankan kecepatan inferensi real-time.
    - Cocok untuk aplikasi praktis seperti pengawasan video, kendaraan otonom, atau deteksi penyakit di lapangan.
    - Evolusi YOLOv1 hingga YOLOv8 menunjukkan pergeseran dari single-shot detection sederhana ke arsitektur modular dengan optimisasi multi-skenario.

- **Vision Transformer (ViT, Swin Transformer)**

    - Unggul dalam menangkap hubungan spasial global.
    - Mencapai performa superlatif pada dataset besar seperti ImageNet-21k.
    - Namun, membutuhkan data sangat besar dan biaya pelatihan tinggi, sehingga sulit diterapkan pada aplikasi dengan dataset terbatas.

Studi perbandingan ini menegaskan bahwa tidak ada "satu model terbaik untuk semua". Pemilihan algoritma selalu terkait erat dengan konteks aplikasi: YOLO untuk kecepatan, Faster R-CNN untuk presisi, dan ViT untuk skala besar.

### 9.6 Keterbatasan Benchmarking

Walaupun penting, benchmarking juga memiliki keterbatasan yang perlu dikritisi:

- Overfitting terhadap Benchmark
  Banyak model "dioptimalkan" untuk memperoleh skor tinggi di dataset tertentu, tetapi gagal beradaptasi pada data nyata. Hal ini menciptakan ilusi kemajuan tanpa manfaat praktis.

- Bias Representasi
  Dataset benchmark internasional seperti COCO dan ImageNet sering bias ke konteks negara maju. Objek dan latar belakang tropis atau pedesaan sering tidak terwakili, sehingga model kurang andal di lapangan Asia atau Afrika.

- Dimensi Evaluasi yang Terbatas
  Metrik seperti mAP atau accuracy tidak memperhitungkan aspek efisiensi energi, latensi komputasi, atau keadilan algoritmik. Dalam era edge computing dan AI berkelanjutan, dimensi ini menjadi sangat relevan.

- Isu Etika dan Privasi
  Beberapa dataset melibatkan wajah atau data medis tanpa selalu mempertimbangkan privasi individu. Evaluasi yang hanya mengejar angka tanpa mempertimbangkan etika dapat menimbulkan risiko sosial.

Dengan demikian, benchmarking harus dipandang sebagai alat bantu, bukan tujuan akhir. Evaluasi yang ideal adalah kombinasi antara skor kuantitatif, analisis kualitatif, dan kesadaran etis.

Tabel 9.1 Perbandingan Kinerja Model Vision pada COCO Benchmark

| Model | mAP @ IoU 0.5:0.95 | mAP @ IoU 0.5 | FPS (Kecepatan) | Ukuran Model | Catatan Khas |
|---|---|---|---|---|---|
| Faster R-CNN (ResNet-50) | 37.9 | 59.1 | ~7 FPS | ~170 MB | Akurasi tinggi, tetapi lambat, lebih cocok untuk server dengan GPU besar. |
| YOLOv5 (Large) | 47.0 | 66.2 | ~45 FPS | ~90 MB | Seimbang antara kecepatan dan akurasi, banyak digunakan di aplikasi industri. |
| YOLOv8 (Large) | 53.0 | 71.5 | ~60 FPS | ~130 MB | Sangat baik untuk real-time, dengan peningkatan stabilitas dan akurasi. |
| Swin Transformer (Base) | 51.9 | 70.1 | ~10 FPS | ~284 MB | Unggul pada akurasi global, namun mahal secara komputasi. |
| DETR (Transformer-based) | 44.9 | 63.8 | ~28 FPS | ~160 MB | Memperkenalkan paradigma baru, unggul dalam deteksi kompleks, tetapi lebih lambat dibanding YOLO. |

Interpretasi hasil pada table 8.1 menunjukkan dengan jelas pareto trade-off antara akurasi dan kecepatan inferensi. Model-model keluarga YOLO—khususnya varian yang lebih mutakhir—tampak menempati posisi menarik pada kurva efisiensi: mAP yang kompetitif sekaligus FPS tinggi, sehingga relevan untuk skenario real-time (misalnya inspeksi lapang, pengawasan video, atau edge analytics). Sebaliknya, arsitektur dua-tahap seperti Faster R-CNN, atau pendekatan Transformer murni, cenderung mendorong akurasi ke titik yang lebih tinggi—terutama pada objek yang rumit atau tumpang tindih—namun membayar harga dalam bentuk latensi dan beban komputasi yang lebih besar. Dengan kata lain, skor tertinggi di metrik akurasi tidak otomatis "terbaik" bila syarat operasional menuntut respons sub-100 ms.

Dimensi kedua yang perlu dibaca dari tabel adalah kapasitas model dan efisiensi sumber daya. Ukuran file bobot (model size) dan jejak memori menjadi indikator praktis tentang di mana

model dapat dideploy. Varian besar Swin/DETR umumnya menghendaki GPU berdaya besar agar mencapai throughput layak, sehingga cocok untuk back-office inference atau riset skala besar; sementara itu, keluarga YOLO (terutama varian nano/small yang sering menjadi "saudara" dari varian Large di ekosistemnya) lebih mudah dioptimalkan dengan quantization, structured pruning, atau knowledge distillation untuk dijalankan pada perangkat edge. Penting dicatat, teknik kompresi tersebut hampir selalu memunculkan degradasi akurasi yang tidak linier terutama pada deteksi objek kecil sehingga keputusan optimasi harus disandarkan pada metrik yang relevan dengan kebutuhan akhir, bukan angka mAP global semata.

Aspek ketiga menyangkut konteks aplikasi dan prioritas metrik. Pada domain yang risk-averse seperti medis atau keselamatanvrecall dan sensitivitas terhadap objek kecil sering kali lebih diprioritaskan dibanding headline mAP. Ini berarti model yang tampak "lebih baik" secara mAP agregat bisa kurang sesuai bila ia rentan melewatkan kasus langka tetapi kritis (false negative). Sebaliknya, pada aplikasi industri berbiaya tinggi di false positive (misalnya quality control), precision menjadi penentu. Tabel kinerja perlu dibaca bersama kurva precision recall per kelas dan per skala (small/medium/large) agar keputusan model tidak bias oleh rata-rata global yang menyamarkan variasi penting antarkelas dan antarukuran objek.

Keempat, perbandingan FPS pada tabel harus dipahami sebagai parameter terikat: angka kecepatan sangat sensitif terhadap hardware, batch size, resolusi input, serta runtime (TensorRT, ONNX Runtime, OpenVINO, dsb.). Mengganti resolusi dari 640→1280 dapat menggeser FPS secara drastis sekaligus menaikkan mAP selektif (terutama pada objek kecil). Karena itu, interpretasi yang ilmiah menuntut kesetaraan protokol evaluasi: arsitektur diuji pada perangkat, runtime, dan resolusi yang sama, disertai reporting konsumsi daya/latensi p-95 untuk merepresentasikan performa dunia nyata.

Kelima, model berbasis Transformer umumnya menunjukkan keunggulan relasi global dan ketahanan terhadap variasi konteks, yang tercermin pada peningkatan mAP di skenario padat objek atau tata ruang kompleks. Namun keuntungan ini sering dikaitkan dengan kebutuhan pretraining berskala besar dan regularization cermat agar tidak tersandung okesimputverfitting pada benchmark. Sementara itu, detektor satu-tahap seperti YOLO unggul pada stabilitas operasional waktu muat cepat, jalur inferensi sederhana, dan post-processing (NMS) yang dapat dituning untuk menggeser operating point sesuai target precision/recall aplikasi.

Terakhir, interpretasi yang matang harus memasukkan ketidakpastian dan kalibrasi skor. Dua model dengan mAP serupa bisa memiliki perilaku probabilitas yang berbeda: satu terkalibrasi baik (skor 0,8 benar-benar berarti peluang ≈80%), sementara yang lain over-confident. Untuk sistem keputusan otomatis, kalibrasi memengaruhi thresholding, beban human-in-the-loop, dan biaya salah keputusan. Karenanya, selain mAP/IoU/FPS, praktik evaluasi tingkat **yang lebih tinggi** semestinya menambahkan audit reliabilitas (ECE/Brier score), robustness terhadap domain shift (uji lintas cuaca, sensor, atau geografis), serta biaya komputasi/energi per prediksi. Dengan membaca tabel melalui lensa-lensa ini, pemilihan model berubah dari sekadar "mencari angka tertinggi" menjadi rekayasa keputusan ilmiah yang menyeimbangkan akurasi, risiko, dan keterbatasan dunia nyata.

# BAB X.

# Implementasi Praktis & Tools dalam Computer Vision

## 10.1 Framework Populer (OpenCV, TensorFlow, PyTorch, YOLO)

Perkembangan riset computer vision tidak dapat dilepaskan dari keberadaan framework dan pustaka perangkat lunak yang membentuk ekosistem penelitian. Kehadiran OpenCV, TensorFlow, PyTorch, dan keluarga YOLO bukan sekadar alat bantu teknis, melainkan representasi dari paradigma ilmiah yang berbeda dalam memahami, merancang, dan menguji algoritma vision.

OpenCV dapat dipandang sebagai warisan dari era vision klasik. Ia menyediakan serangkaian fungsi dasar yang memungkinkan manipulasi citra, ekstraksi fitur, dan transformasi geometris. Fungsi-fungsi ini mencerminkan paradigma awal vision yang menekankan pendekatan analitis-matematis. Meskipun kedalaman fungsionalnya tidak setara dengan framework deep learning, OpenCV tetap relevan dalam konteks pre-processing dan integrasi sistem real-time.

TensorFlow menghadirkan ekosistem yang luas, dengan dukungan dari Google dan integrasi kuat dengan infrastruktur cloud. Filosofinya adalah stabilitas dan skala industri: dari prototipe hingga produksi masif. TensorFlow menekankan determinisme, optimisasi perangkat keras, dan kompatibilitas jangka panjang. Kehadirannya menjembatani dunia riset dengan kebutuhan enterprise-level deployment.

PyTorch, sebaliknya, merepresentasikan pergeseran paradigma ke arah eksplorasi ilmiah yang lebih fleksibel. Dengan dynamic computation graph, PyTorch memberikan kebebasan kepada peneliti untuk melakukan eksperimen tanpa terjebak dalam rigiditas arsitektural. Inilah sebabnya PyTorch mendominasi publikasi akademik mutakhir. Namun, transformasinya menuju dunia industri melalui ONNX dan TorchServe memperlihatkan dialektika antara riset fundamental dan adopsi praktis.

Sementara itu, YOLO adalah bukti nyata bagaimana algoritma dapat berevolusi menjadi ekosistem mandiri. Lahir sebagai solusi deteksi real-time, YOLO kini berkembang menjadi keluarga model dengan ragam arsitektur, pre-trained weights, dan pipeline siap pakai. Filosofinya bukan hanya akurasi, tetapi juga accessibility—bagaimana menjembatani teknologi mutakhir agar dapat digunakan bahkan oleh praktisi non-akademik.

Dengan demikian, pemilihan framework bukan sekadar persoalan teknis, melainkan refleksi dari orientasi riset: apakah ingin berfokus pada stabilitas industri, fleksibilitas akademis, atau aksesibilitas praktis.

## 10.2 Pipeline Deployment (Edge, Mobile, Cloud)

Tahap deployment menegaskan bahwa sebuah algoritma vision tidak berhenti sebagai entitas matematis di atas kertas, melainkan harus hidup dalam ekosistem perangkat nyata. Pipeline deployment dapat dipetakan ke dalam tiga domain utama: edge, mobile, dan cloud, masing-masing dengan tantangan epistemologis dan praktisnya sendiri.

Edge deployment menekankan prinsip kedekatan dengan sumber data. Model dijalankan pada perangkat kecil seperti kamera IoT, drone, atau sensor pertanian. Keuntungannya adalah latensi rendah dan kedaulatan data, dua hal yang semakin penting dalam era privasi digital. Namun, edge deployment menuntut kompromi: model harus diadaptasi agar efisien secara komputasi, meski itu berarti kehilangan sebagian akurasi.

Mobile deployment beroperasi di ruang transisi antara edge dan cloud. Ponsel pintar menjadi laboratorium miniatur, di mana algoritma canggih dijalankan untuk keperluan sehari-hari: augmented reality, diagnosis kesehatan sederhana, atau aplikasi edukasi. Tantangan utamanya adalah fragmentasi ekosistem perangkat (iOS vs Android) serta keterbatasan daya. Di sinilah quantization dan integrasi API tingkat sistem (CoreML, ML Kit) menjadi elemen penentu.

Cloud deployment memegang peran sentral dalam skenario skala besar. Model vision dapat dilatih dengan dataset raksasa menggunakan GPU atau TPU, lalu disajikan dalam bentuk API. Paradigma ini menawarkan elastisitas skala dan kekuatan komputasi, tetapi menghadirkan isu filosofis tentang ketergantungan: apakah sistem vision yang kita bangun benar-benar mandiri, ataukah menjadi entitas bergantung pada ekosistem korporasi besar?

Pipeline implementasi modern cenderung mengadopsi pendekatan hibrida: pre-filtering dilakukan di edge, inferensi ringan di mobile, sementara analisis lanjutan dan arsip historis dikelola di cloud. Inilah bentuk dialektika teknis antara efisiensi, privasi, dan skalabilitas.

## 10.3 Optimasi Model (Quantization, Pruning, Distillation)

Optimasi model merupakan ranah di mana ilmu bertemu dengan rekayasa. Ia lahir dari kesadaran bahwa algoritma vision modern sering kali berukuran terlalu besar untuk dunia nyata. Tiga pendekatan dominan adalah quantization, pruning, dan distillation, yang masing-masing merepresentasikan filosofi berbeda dalam menata ulang model.

Quantization mengubah representasi numerik dari presisi tinggi (float32) menjadi presisi rendah (int8 atau bahkan int4). Secara matematis, ini berarti melakukan aproksimasi fungsi dengan granularitas lebih kasar. Meski demikian, banyak penelitian menunjukkan bahwa jaringan saraf memiliki redundansi representasional yang tinggi, sehingga degradasi akurasi akibat quantization sering kali minimal.

Pruning berangkat dari prinsip biologis: otak manusia tidak selalu menggunakan semua sinapsis secara simultan. Dalam jaringan saraf buatan, banyak bobot yang kontribusinya marjinal terhadap prediksi. Dengan menghapus bobot atau filter yang lemah, kita memperoleh

model yang lebih ramping tanpa kehilangan performa signifikan. Secara filosofis, pruning adalah seni memilah esensi dari redundansi.

Knowledge Distillation menambahkan dimensi epistemologis: pengetahuan tidak hanya tertanam dalam parameter besar, tetapi juga dapat ditransfer ke representasi yang lebih kecil. Model besar bertindak sebagai guru (teacher) yang menurunkan "intuisinya" kepada murid (student). Teknik ini memperlihatkan bahwa pembelajaran mesin tidak hanya soal arsitektur, tetapi juga tentang aliran informasi lintas representasi.

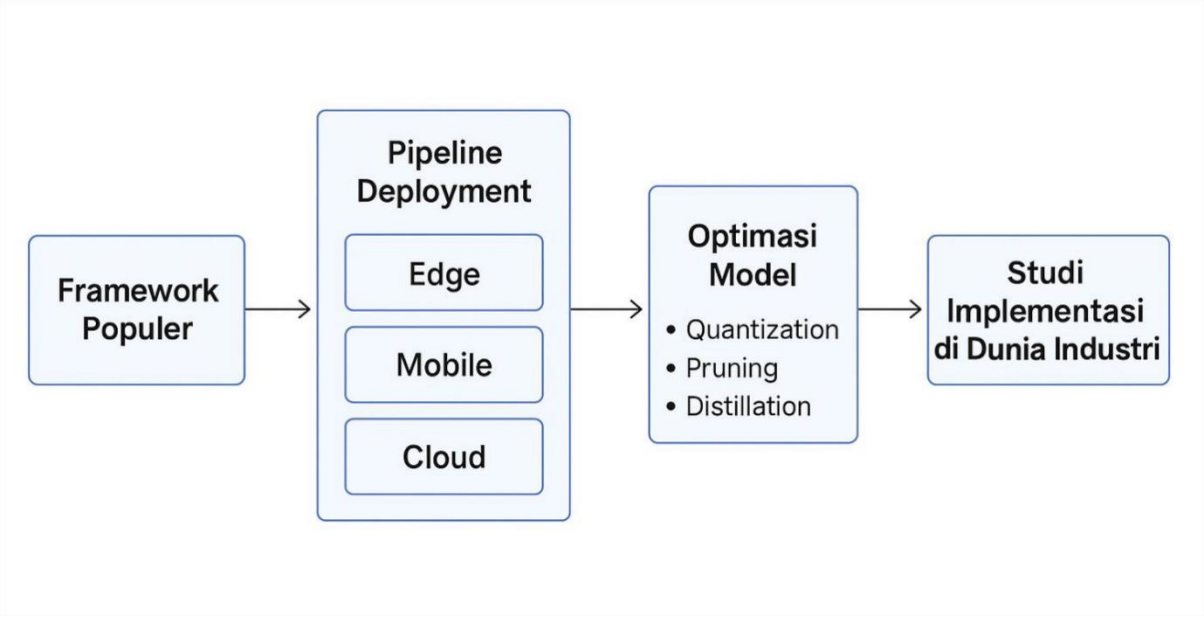
Ketiga strategi ini, jika dipahami bukan sekadar teknis, menunjukkan bahwa optimasi model adalah bagian dari refleksi tentang efisiensi, keberlanjutan, dan estetika ilmiah: bagaimana merancang sistem yang tidak hanya kuat, tetapi juga elegan dalam keterbatasannya.

### 10.4 Studi Implementasi di Dunia Industri

Implementasi di industri menegaskan bagaimana teori bertransformasi menjadi praktik. Setiap sektor menghadirkan realitas berbeda yang menguji batas klaim algoritmik.

1. Pertanian Presisi: Penggunaan YOLO di drone atau kamera edge untuk mendeteksi penyakit tanaman menunjukkan bagaimana kecepatan inferensi lebih penting daripada akurasi absolut. Tantangannya bukan sekadar mendeteksi, tetapi juga memberikan informasi yang dapat ditindaklanjuti petani secara langsung.
2. Otomotif dan Kendaraan Otonom: Di sini, trade-off antara recall dan precision menjadi isu keselamatan nyawa. Model Transformer atau hybrid CNN-Transformer digunakan untuk menangkap konteks global jalanan, tetapi deployment harus dioptimasi untuk latensi rendah di dalam kendaraan.
3. Kesehatan: Analisis citra medis memerlukan akurasi hampir sempurna, karena kesalahan kecil dapat berakibat fatal. Namun, isu privasi menuntut deployment lokal (on-premise) alih-alih cloud publik. Di sinilah distillation dan pruning memungkinkan model berukuran besar dijalankan di rumah sakit dengan infrastruktur terbatas.
4. Ritel dan Smart City: Kamera pengawasan menggunakan model deteksi untuk mengidentifikasi perilaku abnormal. Namun, keberhasilan tidak hanya ditentukan oleh mAP, melainkan juga kemampuan sistem bertahan terhadap kondisi lingkungan (hujan, pencahayaan rendah). Evaluasi di sini lebih menekankan aspek robustnes daripada sekadar akurasi laboratorium.

Studi-studi ini memperlihatkan bahwa implementasi computer vision tidak dapat dipisahkan dari konteks sosial, ekonomi, dan etis. Model yang unggul di benchmark internasional belum tentu relevan di lapangan, dan sebaliknya, inovasi kecil dalam optimasi deployment bisa membawa dampak besar bagi masyarakat.




Gambar Implementasi Praktis & Tools dalam Computer Vision.

Gambar Diagram tersebut menggambarkan sebuah peta jalan konseptual yang menuntun pembaca dari tahap pemilihan framework hingga aplikasi nyata di industri. Setiap blok utama merepresentasikan lapisan yang berbeda dalam siklus hidup implementasi computer vision, sementara cabang-cabang di dalamnya memperlihatkan detail teknis dan pilihan strategis yang tersedia.

Pada bagian awal, Framework Populer ditempatkan sebagai fondasi. Framework ini bukan sekadar perangkat teknis, melainkan medium epistemologis yang menentukan cara peneliti berinteraksi dengan data dan merancang algoritma. OpenCV, misalnya, melambangkan pendekatan klasik yang berbasis pada transformasi citra dan ekstraksi fitur manual. TensorFlow, dengan dukungan infrastruktur cloud dan TPU, menawarkan jalan menuju skala industri yang besar. PyTorch menghadirkan fleksibilitas riset melalui dynamic graph, yang memungkinkan eksperimen cepat dan adaptif. Sementara itu, keluarga YOLO menjadi simbol integrasi antara riset mutakhir dan kebutuhan praktis, karena ia menghadirkan deteksi real-time dengan pipeline end-to-end yang mudah digunakan.

Setelah fondasi ditetapkan, diagram bergerak menuju Pipeline Deployment, yang merefleksikan tantangan implementasi di dunia nyata. Di sinilah keputusan teknis harus diseimbangkan dengan keterbatasan perangkat dan konteks aplikasi. Pada perangkat edge, efisiensi dan privasi menjadi prioritas utama, sebagaimana terlihat pada pemakaian kamera IoT atau drone untuk deteksi lapangan. Mobile deployment berada di antara edge dan cloud, menekankan fleksibilitas dan keterjangkauan, dengan dukungan API sistem operasi seperti CoreML atau ML Kit. Cloud deployment, dengan daya komputasi masifnya, memungkinkan analisis berskala besar dan integrasi data multimodal, meski harus menghadapi isu privasi dan biaya bandwidth. Pendekatan hibrida kemudian lahir sebagai kompromi, di mana pre-filtering dilakukan di edge sementara analisis mendalam dikerjakan di cloud.

Bagian ketiga, Optimasi Model, merepresentasikan refleksi tentang bagaimana teori dapat dipertemukan dengan keterbatasan praktis. Quantization adalah seni mengubah representasi numerik untuk mencapai kecepatan dan efisiensi tanpa kehilangan performa signifikan. Pruning mengajarkan kita bahwa tidak semua parameter membawa nilai ilmiah yang sama; banyak bobot bersifat redundan dan dapat dihilangkan tanpa mengorbankan akurasi. Knowledge distillation, lebih dari sekadar teknik kompresi, mengilustrasikan proses epistemologis di mana model besar sebagai "guru" mentransfer pengetahuannya kepada model lebih kecil, sehingga terjadi kesinambungan antarrepresentasi. Pada tataran frontier, pendekatan Neural Architecture Search (NAS) memperluas horizon dengan merancang arsitektur yang dioptimalkan secara otomatis untuk konteks tertentu.

Akhirnya, diagram bermuara pada Studi Implementasi di Dunia Industri, yang memperlihatkan bagaimana teori dan teknik bertransformasi menjadi solusi nyata. Di sektor pertanian presisi, YOLO digunakan untuk mendeteksi penyakit daun secara real-time melalui drone, memberikan keputusan cepat bagi petani. Dalam otomotif, deteksi objek jalanan menuntut keseimbangan antara recall tinggi dan latensi rendah, karena kesalahan sekecil apa pun dapat berimplikasi pada keselamatan. Di bidang kesehatan, Vision Transformer digunakan dalam analisis MRI, tetapi model perlu dioptimasi agar dapat berjalan pada server lokal demi menjaga privasi pasien. Ritel dan smart city memperlihatkan sisi sosial teknologi vision, mulai dari analisis perilaku konsumen hingga deteksi kerumunan pada ruang publik.

Implementasi algoritma computer vision bukanlah sekadar tahapan teknis yang mengikuti pelatihan model. Ia adalah fase di mana abstraksi matematis, arsitektur jaringan, dan teori pembelajaran mesin diuji dalam dunia nyata yang penuh keterbatasan sumber daya, dinamika lingkungan, dan tuntutan operasional. Karena itu, bab ini berfokus pada dimensi praktis yang menjembatani antara riset konseptual dan penerapan nyata.

Untuk memahami kompleksitas implementasi, penting bagi kita untuk melihatnya bukan hanya sebagai kumpulan prosedur teknis, tetapi sebagai suatu ekosistem multidimensi. Ekosistem ini mencakup pilihan framework, pipeline deployment, strategi optimasi model, hingga studi kasus penerapan di industri. Diagram yang ditampilkan dalam bab ini berfungsi sebagai peta jalan konseptual yang menuntun pembaca melewati empat lapisan utama:

1. Framework Populer – tahap awal berupa pemilihan pustaka dan kerangka kerja yang akan menentukan arah riset dan pengembangan. OpenCV, TensorFlow, PyTorch, dan YOLO mewakili keragaman paradigma, mulai dari vision klasik, ekosistem industri berskala besar, fleksibilitas akademis, hingga solusi deteksi real-time yang siap pakai.
2. Pipeline Deployment – representasi bagaimana model dibawa keluar dari ruang laboratorium menuju perangkat nyata. Edge deployment menekankan latensi rendah dan privasi data, mobile deployment membuka jalan bagi aplikasi sehari-hari, cloud deployment menawarkan skala besar, sementara pendekatan hibrida menggabungkan keunggulan masing-masing.

3. Optimasi Model – refleksi tentang bagaimana teori bertransformasi agar sesuai dengan batasan komputasi. Quantization, pruning, dan knowledge distillation menegaskan bahwa efisiensi adalah sama pentingnya dengan akurasi. Bahkan, frontier baru seperti Neural Architecture Search memperlihatkan bahwa desain arsitektur pun dapat diotomatisasi untuk mencapai keseimbangan optimal.
4. Studi Implementasi di Dunia Industri – tahap akhir yang memperlihatkan bagaimana algoritma vision berdampak nyata dalam sektor pertanian, kesehatan, otomotif, ritel, dan smart city. Bagian ini menunjukkan bahwa metrik laboratorium (mAP, IoU, F1-score) hanya bermakna jika mampu memberikan solusi yang relevan dengan kebutuhan sosial-ekonomi.

Diagram konseptual yang menyertai bab ini memperlihatkan bagaimana keempat lapisan tersebut terhubung secara hierarkis dan saling memengaruhi. Pemilihan framework menentukan fleksibilitas riset dan kesiapan deployment. Pipeline deployment memengaruhi strategi optimasi model, karena keterbatasan perangkat menuntut adaptasi arsitektur. Hasil dari ketiga lapisan tersebut kemudian diuji dalam realitas industri, yang pada gilirannya memberi umpan balik ke tahap riset dan pengembangan.

Dengan demikian, bab ini tidak hanya akan menjelaskan prosedur implementasi, tetapi juga mengajak pembaca untuk merefleksikan aspek filosofis dari teknologi vision: bagaimana algoritma yang lahir dari ruang abstraksi matematika akhirnya berinteraksi dengan dunia nyata yang penuh dinamika, risiko, dan nilai sosial. Inilah yang membedakan implementasi praktis sebagai sekadar “penggunaan teknologi” dengan implementasi sebagai proses ilmiah yang reflektif.

# BAB XI
# Integrasi Vision dan IoT: Aplikasi Cerdas dalam Dunia Nyata

## 11.1 Konsep Dasar Penggabungan Vision dan Internet of Things (IoT)

**Pendahuluan**

Teknologi tidak lagi bekerja secara silo di era digital saat ini. Computer Vision dan Internet of Things (IoT) adalah dua contoh integrasi teknologi yang sangat menjanjikan. Yang pertama memungkinkan
perangkat fisik untuk berkomunikasi dan saling terhubung, sedangkan yang kedua memungkinkan sistem untuk "melihat" dan menafsirkan
dunia visual. Aplikasi yang tidak hanya dapat mendeteksi atau mengenali objek, tetapi juga dapat mengambil keputusan secara
otomatis dan mengirimkan data ke cloud, perangkat pengguna, atau sistem lainnya timbul ketika keduanya digabungkan.

Definisi dan hubungannya:
Vision computer (CV): bagian kecerdasan buatan yang memungkinkan komputer memahami dan menafsirkan data visual, seperti gambar atau video.
Internet of Things (IoT): jaringan perangkat yang terhubung ke internet, mengumpulkan, dan berbagi data.


Gambar 25 Ilustrasi Sistem IoT dan Computer Vision

Pada gambar 25 gambaran kolaborasi antara computer vision dan IoT digunakan, pada kombinasi ini sering di pakai untuk sistem absensi, sistem mobil cerdas dan bahkan mesin pendingin cerdas berbasis IoT

## 11.2 Arsitektur Sistem Vision-IoT

Pada gambar 26 Sistem integrasi vision dan Internet of Things terdiri dari beberapa komponen utama. Arsitektur umumnya dapat dilihat di bawah ini:

1. Perangkat Penginderaan (Lapisan Sensor) Komponen ini berfungsi untuk menangkap gambar. Biasanya terdiri dari:
   - Kamera digital biasa
   - Kamera termal • Kamera inframerah
   - ESP32-CAM (kamera kecil yang terhubung ke internet)
2. Lapisan Prosesing dan Analisis (Edge atau Cloud Layer) :
   Data dari sensor dapat diproses langsung di perangkat (edge computing) atau dikirim ke cloud untuk diproses.
3. Lapisan Network:
   Protokol seperti MQTT, HTTP, atau WebSocket digunakan untuk mengirimkan data dari sensor ke server.
4. Lapisan Aplikasi dan Tampilan :
   Semua data yang telah diproses dapat ditampilkan di dashboard, dikirim sebagai peringatan, atau digunakan untuk membuat keputusan.


Gambar 26 arsitektur sistem Vision-IoT umum

## 11.3 Studi Kasus: Deteksi Penyakit Daun dengan ESP32-CAM dan YOLO

Latar Belakang: Penyakit daun pada tanaman membahayakan hasil panen dalam sektor pertanian, khususnya di negara tropis. Deteksi manual membutuhkan waktu dan keahlian khusus. Oleh karena itu, sistem otomatis dan terintegrasi diperlukan. Sistem ini harus dapat mendeteksi penyakit secara visual dan mengirimkan laporan ke petani atau sistem pusat secara real-time.

Tabel 10.1 Rangkaian Sistem IoT dan Komputer Vision

| **Komponen** | **Fungsi** |
|---|---|
| ESP32-CAM | Mengambil gambar daun |
| YOLOv5 (di edge/cloud) | Mendeteksi dan mengklasifikasi penyakit |
| MQTT / HTTP | Mengirim data hasil deteksi ke server |
| Dashboard (Grafana/Node-RED) | Menampilkan hasil secara visual |

Tabel 6.1 menguraikan komponen-komponen utama dalam sebuah sistem deteksi penyakit daun beserta fungsi spesifik masing-masing komponen. Sistem ini dirancang untuk bekerja secara terintegrasi, menggabungkan perangkat keras, kecerdasan buatan, dan antarmuka visual.

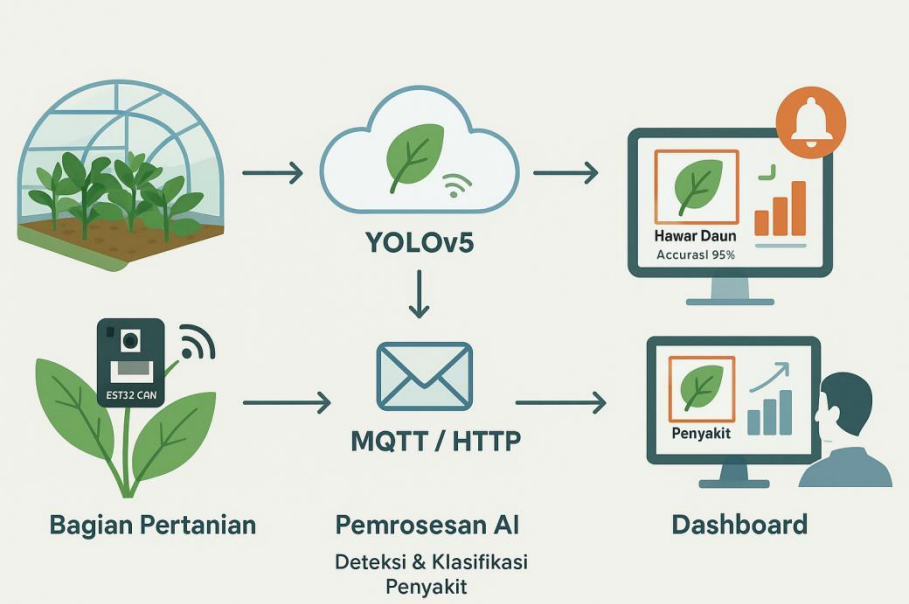
**ESP32-CAM** berperan sebagai edge device yang bertugas melakukan akuisisi data visual. Modul ini berfungsi untuk mengambil gambar daun secara on-site menggunakan sensor kamera terintegrasi. Kemampuan komputasi dan konektivitas nirkabel yang dimilikinya memungkinkan pengambilan gambar dilakukan secara mandiri di lapangan.

**YOLOv5** merupakan inti kecerdasan buatan dalam sistem ini, yang dapat diimplementasikan secara edge computing atau cloud computing. Algoritma ini bertanggung jawab untuk mendeteksi dan mengklasifikasikan penyakit pada daun yang telah diambil gambarnya.
Melalui teknik deep learning, model ini mampu mengidentifikasi pola-pola visual spesifik yang mengindikasikan jenis penyakit tertentu dengan akurasi tinggi.

**MQTT/HTTP** berfungsi sebagai protokol komunikasi data yang mengirimkan hasil deteksi ke server. MQTT dipilih untuk skenario yang memerlukan efisiensi bandwidth dan komunikasi real-time, sementara HTTP digunakan untuk transfer data yang lebih konvensional. Protokol ini memastikan kelancaran transmisi data dari perangkat akuisisi ke sistem pemrosesan pusat.

**Dashboard** berperan sebagai antarmuka visualisasi data yang menampilkan hasil deteksi secara visual. Dashboard ini menyajikan informasi dalam bentuk grafik, metrik, dan notifikasi yang mudah dipahami, memungkinkan pengguna untuk memantau kondisi tanaman secara real-time dan mengambil keputusan berdasarkan data yang terukur.

Integrasi keempat komponen ini menciptakan sebuah sistem yang tidak hanya mampu mendeteksi penyakit daun secara otomatis, tetapi juga menyediakan platform monitoring yang komprehensif bagi petani atau peneliti pertanian.



Gambar 27 Sistem terintegrasi dari pengambilan gambar hingga notifikasi petani

Pada Gamar 27 Sistem deteksi penyakit daun berbasis Internet of Things (IoT) dan kecerdasan buatan (AI) dirancang untuk memberikan solusi praktis dalam pemantauan kesehatan tanaman secara real-time. Alur kerjanya dapat dibagi ke dalam empat tahapan utama, yaitu pengambilan data di lapangan, pemrosesan berbasis AI, transmisi data, dan visualisasi hasil melalui dashboard.

**1. Bagian Pertanian (Lapangan)**
Pada tahap awal, sistem bekerja langsung di area pertanian atau rumah kaca. Sebuah perangkat ESP32-CAM dipasang di dekat tanaman untuk mengambil gambar daun. Kamera ini berfungsi sebagai sensor visual yang merekam kondisi aktual tanaman, baik secara berkala maupun berdasarkan pemicu tertentu. Pemanfaatan perangkat kecil seperti ESP32-CAM dipilih karena sifatnya yang hemat daya, murah, serta mudah diintegrasikan dengan jaringan nirkabel. Hasil rekaman ini merupakan data mentah yang kemudian dikirimkan ke tahap berikutnya untuk dianalisis.

**2. Pemrosesan AI**
Setelah gambar diperoleh, data dikirim menuju server atau cloud untuk diproses menggunakan model deteksi berbasis deep learning, misalnya YOLOv5. Model ini dipilih karena mampu melakukan deteksi objek dengan cepat dan akurat, termasuk mengidentifikasi gejala penyakit pada daun. Proses ini melibatkan dua tahapan penting:

**Deteksi**, yaitu mengidentifikasi area pada daun yang menunjukkan tanda penyakit.

**Klasifikasi,** yaitu menentukan jenis penyakit berdasarkan pola visual yang terdeteksi, misalnya "Hawar Daun". Dengan pendekatan ini, sistem dapat memberikan hasil diagnosis otomatis tanpa memerlukan pemeriksaan manual yang seringkali memakan waktu.

**3. Transmisi Data**
Setelah diperoleh hasil klasifikasi, informasi tersebut tidak berhenti di server saja. Data kemudian ditransmisikan melalui protokol komunikasi IoT, seperti MQTT atau HTTP.

Pemilihan protokol ini memungkinkan data berpindah secara cepat, ringan, dan aman dari lapangan menuju pusat kendali. Tahap ini berfungsi sebagai jembatan agar hasil deteksi bisa diakses oleh pengguna, baik peneliti maupun petani.

**4. Dashboard Visualisasi**

Tahap terakhir adalah penyajian hasil dalam bentuk visual yang mudah dipahami. Hasil analisis ditampilkan melalui dashboard berbasis Grafana atau Node-RED. Pada dashboard, pengguna dapat melihat:

Gambar daun yang dianalisis, lengkap dengan bounding box dan label penyakit.

Persentase akurasi deteksi, misalnya "Hawar Daun – Akurasi 95%".

Grafik tren penyakit yang menggambarkan perkembangan dari waktu ke waktu. Notifikasi peringatan (alert) jika penyakit terdeteksi, sehingga petani dapat segera melakukan tindakan pencegahan atau pengendalian. Selain itu, dashboard ini dapat diakses melalui perangkat komputer maupun tablet, sehingga fleksibel digunakan di lapangan maupun di pusat penelitian. Kehadiran fitur notifikasi sangat membantu dalam pengambilan keputusan yang lebih cepat dan tepat.

Output Sistem

- Nama penyakit daun
- Probabilitas klasifikasi
- Tanggal dan waktu deteksi
- Foto daun yang terdeteksi penyakit
- Lokasi (jika dilengkapi GPS)

**10.4 Tantangan dalam Integrasi Vision dan IoT**

**Dalam Project Integrasi ini mempunyai beberapa tantangan**

9. Perangkat kamera dan mikroprosesor kecil seperti ESP32 memiliki batas daya proses dan memori. Oleh karena itu, model deep learning harus dioptimalkan
10. dengan menggunakan metode seperti quantization (mengurangi ukuran model), pruning (menghapus node yang tidak penting), dan transfer pembelajaran ke
11. YOLOv5n atau MobileNet solusi bisa memakai Raseberry Pi namu harga lebih mahal
12. Koneksi Jaringan: Beberapa area pertanian atau manufaktur memiliki koneksi internet yang terbatas. Akibatnya, protokol ringan seperti MQTT atau penyimpanan lokal sementara diperlukan.
13. Akurasi deteksi dapat berkurang karena pencahayaan, latar belakang yang kompleks, atau objek tumpang tindih. Dibutuhkan dataset representatif dan peningkatan data.

## 11.5 Keuntungan Integrasi Computer Vision dan IoT

Tabel 10.2 Aspek dan Dampak Kolaborasi Computer Vision dan IoT

| Aspek | Dampak / Manfaat |
|---|---|
| Efisiensi Waktu | Menghemat waktu inspeksi manual |
| Real-time Monitoring | Deteksi cepat dan tanggap darurat |
| Otomatisasi Proses | Tidak bergantung pada operator manusia |
| Skalabilitas | Dapat diterapkan ke banyak perangkat atau lahan |
| Dokumentasi Data | Seluruh hasil deteksi dapat disimpan untuk audit |

Pada Table 6.2 Kolaborasi antara teknologi Computer Vision dan Internet of Things (IoT) telah menciptakan paradigma baru dalam efisiensi dan efektivitas sistem pemantauan modern, khususnya dalam konteks pertanian, industri, dan manajemen lingkungan. Integrasi ini tidak hanya meningkatkan kemampuan sistem secara teknis, tetapi juga membawa dampak operasional yang signifikan.

Dalam hal efisiensi waktu, sistem yang menggabungkan sensor IoT dan algoritma computer vision mampu mengotomasi proses inspeksi yang sebelumnya bergantung pada tenaga manusia. Hal ini secara drastis mengurangi waktu yang diperlukan untuk pemantauan rutin, memungkinkan sumber daya manusia dialihkan ke tugas-tugas yang lebih strategis.

Kemampuan pemantauan waktu-nyata (real-time monitoring) menjadi tulang punggung sistem responsif. Sensor IoT terus-menerus mengumpulkan data visual dan lingkungan, sementara algoritma computer vision menganalisis data tersebut secara instan untuk mendeteksi anomali atau kondisi darurat. Kombinasi ini memungkinkan intervensi cepat yang dapat mencegah kerugian lebih besar, seperti dalam kasus deteksi dini penyakit tanaman atau kegagalan peralatan.

Tingkat otomatisasi yang tinggi mengurangi ketergantungan pada keahlian operator manusia. Sistem dapat beroperasi secara mandiri, membuat keputusan berbasis data yang konsisten dan terhindar dari subjektivitas atau kelelahan manusia yang dapat mempengaruhi akurasi. Dari perspektif skalabilitas, solusi ini dapat dengan mudah diimplementasikan pada banyak perangkat atau lahan tanpa penambahan biaya operasional yang signifikan. Satu algoritma dapat di-deploy pada ratusan perangkat IoT, memungkinkan pemantauan terpadu pada area yang sangat luas. Terakhir, kolaborasi ini memungkinkan dokumentasi data yang komprehensif. Setiap hasil deteksi, beserta konteks waktunya, dapat disimpan secara

terstruktur untuk keperluan audit, analisis tren, atau bahkan sebagai bahan pelatihan untuk meningkatkan kinerja model AI di masa depan.

## 11.6 Aplikasi Lain Integrasi Vision-IoT

Tabel 10.3 Bidang Kolaborasi ComputerVision dan IoT

| Bidang | Contoh Aplikasi |
|---|---|
| Pertanian | Pemantauan pertumbuhan tanaman, hama, irigasi |
| Industri | Quality control visual pada jalur produksi |
| Keamanan | CCTV pintar, deteksi orang mencurigakan |
| Kesehatan | Telemedicine berbasis gambar, pemantauan luka |
| Lingkungan | Deteksi kebakaran hutan dini, pantau polusi visual |

Pada tabel 10.3 Integrasi antara Computer Vision dan Internet of Things (IoT) telah membuka peluang inovasi yang transformative di berbagai sektor, memungkinkan sistem yang tidak hanya cerdas tetapi juga saling terhubung dan responsif. Kolaborasi ini memanfaatkan kekuatan pengolahan citra digital dengan jaringan sensor fisik yang ekstensif, menciptakan solusi yang berdampak signifikan secara praktis dan ekonomis.

Di bidang Pertanian, teknologi ini merevolusi praktik agrikultur tradisional melalui aplikasi seperti pemantauan pertumbuhan tanaman berbasis sensor visual, deteksi dini serangan hama melalui analisis citra daun, dan sistem irigasi presisi yang diaktifkan oleh data visual dan lingkungan. Implementasi ini mendukung pertanian presisi yang berkelanjutan dengan mengoptimalkan penggunaan sumber daya seperti air dan pupuk.

Sektor Industri memanfaatkan integrasi ini untuk meningkatkan jaminan kualitas melalui sistem inspeksi visual otomatis pada jalur produksi. Kamera yang terhubung dengan jaringan IoT dapat mendeteksi cacat produk secara real-time, mengukur dimensi dengan akurasi sub-milimeter, dan memantau kondisi mesin secara terus-menerus, sehingga mengurangi kesalahan manusia dan meningkatkan efisiensi produksi.

Dalam bidang Keamanan, teknologi ini mengubah sistem pengawasan konvensional menjadi solusi keamanan proaktif melalui CCTV pintar yang mampu mengenali pola perilaku mencurigakan, mendeteksi pelanggaran perimeter, dan mengidentifikasi ancaman potensial secara otomatis tanpa memerlukan pemantauan manusia yang terus-menerus.

Aplikasi di sektor Kesehatan termasuk sistem telemedicine yang memungkinkan diagnosis berbasis gambar jarak jauh, pemantauan perkembangan luka kronis melalui analisis citra serial, dan bantuan diagnosis medis dengan analisis citra radiologi yang terintegrasi dengan sistem informasi kesehatan.

Terakhir, di bidang Lingkungan, teknologi ini berkontribusi pada sistem peringatan dini kebakaran hutan melalui deteksi asap otomatis, pemantauan kualitas udara berdasarkan analisis visibilitas atmosfer, dan tracking perubahan iklim melalui penginderaan visual jangka panjang yang terintegrasi dengan sensor lingkungan IoT.

Kesimpulan

Untuk mewujudkan sistem cerdas, integrasi antara visi komputer dan Internet of Things adalah kebutuhan nyata. Perpaduan ini memungkinkan sistem untuk "melihat" dan "berpikir" sebelum menyampaikan data secara real-time dalam industri pertanian dan manufaktur. Dalam bab ini, pendekatan sistemik dan terdistribusi sangat penting untuk membangun aplikasi berbasis visi kontemporer yang efektif dan bermanfaat di dunia nyata.

# BAB XII

# Studi Kasus Aplikatif Computer Vision

## Pendahuluan

Perkembangan pesat dalam bidang *computer vision* telah menempatkan teknologi ini tidak lagi sekadar sebagai kajian teoretis, melainkan sebagai instrumen strategis yang secara nyata mengubah berbagai sektor kehidupan manusia. Jika pada dekade 1980–1990-an computer vision lebih banyak dibicarakan dalam ruang akademik sebagai cabang dari kecerdasan buatan (AI) yang berfokus pada pemrosesan dan pemahaman citra, maka kini ia hadir dalam bentuk aplikasi yang menyentuh langsung kebutuhan dasar masyarakat: ketersediaan pangan, layanan kesehatan, keamanan publik, transportasi cerdas, hingga kenyamanan berbelanja di perkotaan modern.

Pergeseran dari konsep ke implementasi ini tidak terjadi secara kebetulan. Terdapat tiga faktor utama yang mendorongnya. Pertama, ketersediaan perangkat keras yang semakin terjangkau, mulai dari kamera resolusi tinggi hingga *embedded system* seperti ESP32-CAM dan NVIDIA Jetson Nano. Kedua, kematangan algoritma vision, terutama setelah revolusi *deep learning* yang menghadirkan arsitektur seperti CNN, YOLO, dan Vision Transformer, yang memungkinkan sistem memperoleh tingkat akurasi deteksi maupun klasifikasi yang sangat tinggi. Ketiga, ekosistem data yang semakin luas dan mudah diakses, baik berupa dataset terbuka maupun hasil akuisisi lapangan melalui sensor dan drone.

Dalam kerangka pendidikan tinggi, terutama bagi mahasiswa strata satu (S1) dan pascasarjana, memahami computer vision tidak cukup hanya dari sisi algoritmik. Mahasiswa perlu melihat bagaimana teori yang dipelajari di ruang kuliah mampu bertransformasi menjadi solusi nyata yang berkontribusi terhadap keberlanjutan sosial, ekonomi, dan lingkungan. Pendekatan berbasis studi kasus menjadi penting karena menyajikan contoh konkret, memperlihatkan keterkaitan lintas disiplin, serta memberikan inspirasi untuk penelitian dan inovasi berikutnya. Bab ini secara khusus akan menguraikan lima sektor utama penerapan computer vision: pertanian, kesehatan, transportasi, keamanan, dan ritel serta smart city. Setiap sektor dipilih karena mewakili kebutuhan fundamental masyarakat sekaligus memperlihatkan spektrum luas pemanfaatan teknologi ini, dari penyediaan pangan hingga pembangunan kota cerdas. Melalui uraian yang mendalam, diharapkan pembaca dapat menangkap gambaran utuh mengenai bagaimana computer vision telah menjadi teknologi kunci dalam menjawab tantangan era digital, sekaligus membuka peluang penelitian baru yang relevan dengan konteks lokal maupun global.

## 12.1 Computer Vision di Bidang Pertanian

Pertanian modern sedang beralih dari praktik yang banyak bergantung pada pengalaman dan kebiasaan menjadi pendekatan yang lebih terukur: pertanian presisi. Alih paradigma ini memadukan pengukuran lapang yang sistematis, analisis data, dan pengambilan keputusan berbasis bukti. Dalam kerangka tersebut, computer vision berfungsi sebagai indera visual yang menangkap informasi spasial dan spektral dari daun, kanopi, buah, dan lanskap, lalu menyajikannya dalam bentuk yang dapat diinterpretasikan oleh model atau manusia. Keunggulan utama teknologi ini adalah kemampuannya untuk bekerja dalam skala luas (mis. pemindaian menggunakan drone), memberikan umpan balik dengan latensi rendah (mendekati

waktu-nyata pada skenario edge), dan menyediakan penilaian yang lebih objektif dibanding observasi manusia. Namun, untuk benar-benar berguna di lapang, penerapan vision haruslah dirancang mulai dari perumusan tugas yang jelas hingga rencana operasional yang mempertimbangkan keterbatasan infrastruktur dan ekonomi lokal.

### 12.1.1 Taksonomi Masalah dan Formulasi Tugas

Agar solusi yang dikembangkan tidak berhenti sebagai prototip laboratorium, masalah harus dipetakan dan diformulasikan secara eksplisit. Dalam praktik, tugas-tugas computer vision di pertanian umumnya dapat dikelompokkan menjadi empat kategori utama. Pertama, diagnosis dan pemantauan kesehatan tanaman—di sini tugasnya meliputi klasifikasi tingkat daun (mis. sehat vs terinfeksi jenis penyakit tertentu), deteksi objek (mengidentifikasi bercak atau lesi pada kanopi ketika frame memuat banyak daun), serta segmentasi area terserang untuk mengukur keparahan. Kedua, estimasi variabel agronomis, seperti regresi untuk memprediksi biomassa, penghitungan leaf area index, atau estimation yield berbasis indeks vegetasi; tugas ini menuntut integrasi fitur tekstur, warna, dan spektral serta kadang pemodelan temporal. Ketiga, pengendalian gulma serta pemupukan/irigasi presisi, yaitu mendeteksi gulma antar-barisan, memetakan area defisiensi nutrisi, atau mengidentifikasi stres kekeringan; hasilnya menjadi input bagi aktuator otomatis. Keempat, sortasi dan grading pascapanen, meliputi penilaian mutu buah (ukuran, warna, cacat permukaan) dan klasifikasi tingkat kematangan pada jalur produksi. Formulasi tugas yang cermat (mis. klasifikasi multi-kelas vs segmentasi) menentukan pemilihan sensor, arsitektur model, metrik evaluasi, serta strategi penerapan di lapangan—oleh karena itu fase perumusan tidak boleh diremehkan.

### 12.1.2 Akuisisi Data: Sensor, Protokol, dan Kualitas Label

Kualitas keluaran model sangat dipengaruhi oleh kualitas input; oleh karena itu desain prosedur akuisisi data adalah fondasi. Dua rezim akuisisi dominan adalah proximal sensing (kamera dekat objek seperti smartphone, kamera stasioner atau ESP32-CAM) dan remote sensing (UAV/drone, kadang satelit untuk skala besar). Pemilihan modalitas sensor harus selaras dengan tujuan: citra RGB memadai untuk gejala visual kasat mata dan biaya rendah; sensor multispektral (termasuk band NIR) memungkinkan deteksi perubahan fisiologis sebelum manifestasi visual; sensor termal membantu mengidentifikasi stres air; sedangkan hiperspektral meskipun mahal membuka kemungkinan karakterisasi spektral untuk membedakan penyebab stres.

Standar pengambilan citra perlu didokumentasikan dengan ketat: jarak dan sudut pengambilan, waktu pengambilan (jam pada hari yang konsisten), papan kalibrasi warna untuk koreksi radiometrik, serta metadata seperti fase fenologi, varietas, dan perlakuan lahan. Untuk UAV, koreksi radiometrik dan georeferensi (RTK/PPK) sangat penting agar nilai spektral dan koordinat spasial dapat direproduksi lintas misi. Pada tahap pelabelan, keterlibatan ahli agronomi diperlukan: pedoman anotasi yang eksplisit (mis. skala persentase area terinfeksi) dan prosedur adjudikasi ketika ada perbedaan penilai meningkatkan reliabilitas ground truth. Ukur pula kesepakatan antar-penilai (mis. Cohen’s kappa) untuk mengkuantifikasi kualitas label. Ketidakseimbangan kelas (mis. penyakit langka) harus ditangani sejak desain dataset melalui pengambilan bertarget, oversampling yang rasional, atau strategi augmentasi yang realistis (perubahan pencahayaan, blur, flip, random crop, mixup/mosaic dalam konteks deteksi

objek). Untuk data multispektral, normalisasi antar-saluran pasca-koreksi radiasi diperlukan agar nilai spektral dapat dibandingkan antar sesi.

### 12.1.3 Desain Model: Dari CNN ke Transformer dan Fusi Modalitas

Pemilihan arsitektur model harus pragmatis: cocokkan kapasitas model dengan ketersediaan data, kompleksitas tugas, dan sumber daya operasional. Untuk klasifikasi daun, arsitektur seperti ResNet atau EfficientNet sering menjadi baseline yang kuat; bila terjadi ketidakseimbangan kelas, gunakan loss yang mengatasi hal tersebut (mis. focal loss) atau sampling berbobot. Kalibrasi probabilitas (mis. temperature scaling) penting ketika keputusan model berimplikasi pada rekomendasi agronomis. Untuk deteksi gejala pada kanopi masalah yang sering melibatkan objek kecil detektor berbasis satu tahap seperti YOLOv5/v8 unggul pada latensi dan throughput; namun untuk pengukuran keparahan yang presisi, segmentasi instance (Mask R-CNN) atau model semantik seperti U-Net/DeepLab lebih tepat karena memberikan peta area yang terinfeksi.

Model temporal atau hibrida diperlukan bila input adalah deret waktu (mis. seri citra UAV): kombinasi fitur CNN dengan LSTM, 1D-CNN, atau transformer temporal dapat memodelkan dinamika pertumbuhan dan menghasilkan prediksi yield. Untuk data multispektral/hiperspektral, pertimbangkan strategi fusi—fusi awal (stacking saluran lalu 3D-CNN) berguna untuk menangkap interaksi spektral-spasial; fusi akhir (menggabungkan skor per-modalitas) memberikan modularitas dan memungkinkan fallback saat satu modalitas hilang. Pada konteks edge, model ringan (MobileNetV3, ShuffleNet, varian YOLO-Nano) yang dioptimasi melalui kuantisasi INT8, pruning terstruktur, dan knowledge distillation menjadi solusi agar inferensi dapat berjalan di Jetson Nano, Raspberry Pi, atau micro-accelerators; pada perangkat sangat terbatas seperti ESP32-CAM umumnya hanya dilakukan akuisisi dan praproses, sementara inferensi dikirim ke gateway lokal.

### 12.1.4 Metrik Evaluasi dan Protokol Validasi

Evaluasi yang benar-benar merefleksikan kesiapan lapang menuntut lebih dari sekadar akurasi rata-rata. Untuk klasifikasi gunakan macro F1, balanced accuracy, dan kurva Precision-Recall (PR) karena ROC dapat menyesatkan pada dataset sangat tak seimbang. Selain itu ukur calibration model (Expected Calibration Error) agar probabilitas keluaran dapat dipakai sebagai ukuran risiko. Untuk deteksi dan segmentasi, laporan harus mencakup mAP@[.5:.95], IoU/Dice per-kelas, serta kinerja terperinci per-ukuran objek (S/M/L) karena lesi sering berukuran sangat kecil dan mudah terlewat. Pada regresi hasil panen gunakan RMSE/MAE serta interval ketidakpastian (confidence/prediction intervals). Penting pula melakukan validasi silang lintas-konteks—mis. leave-one-field-out atau season-out—untuk menilai ketahanan model terhadap pergeseran domain (varietas, musim, teknik budidaya). Uji robustnes dengan stress test (blur, bayangan, occlusion) serta deteksi OOD (out-of-distribution) untuk meminimalkan kesalahan saat model bertemu kondisi yang tidak terlihat selama pelatihan.

### 12.1.5 Dari Laboratorium ke Lahan: Arsitektur Sistem dan MLOps

Keberhasilan adopsi bergantung pada ekosistem operasional, bukan hanya model akurat. Arsitektur sistem yang direkomendasikan meliputi: (1) akuisisi (kamera/drone) → (2) edge gateway untuk praproses dan kompresi → (3) server lokal/cloud untuk inferensi dan agregasi → (4) dashboard rekomendasi dengan peta dan skor keparahan → (5) mekanisme umpan balik

oleh penyuluh/petani yang memasok label baru. Praktik MLOps pada konteks pertanian harus mencakup versi data & anotasi (mis. DVC atau sistem serupa), pelacakan eksperimen, pipeline continuous training yang memanfaatkan active learning dari koreksi manusia, dan penjadwalan retraining per musim untuk mengatasi data drift. Karena konektivitas di daerah rural sering terputus, desain harus tahan gangguan: batch upload, metadata ringkas, inferensi offline bila perlu, dan sinkronisasi periodik. Untuk membangun kepercayaan, tampilkan visualisasi perhatian (mis. Grad-CAM) dan ukur ketidakpastian keluaran (ensembles, MC dropout); saat ketidakpastian tinggi, sistem harus mengeluarkan rekomendasi fail-safe (mis. rujuk ke penyuluh).

### 12.1.6 Studi Kasus Representatif

Beberapa studi representatif menonjol karena tantangan yang mereka wakili. Pada padi (Oryza sativa), pipeline tipikal memulai dari pengambilan citra daun/kanopi menggunakan smartphone atau UAV, koreksi warna, kemudian deteksi/segmentasi gejala dengan YOLO/U-Net. Sumber kesalahan khas termasuk defisiensi nutrisi yang menyerupai gejala penyakit serta artefak bayangan; mitigasinya adalah anotasi multi-ahli, hard example mining, dan penggunaan data multispektral untuk membedakan stres abiotik dan biotik. Pada kelapa sawit, tantangan adalah kanopi rapat dan objek kecil (tandan buah) tertutup pelepah; solusi praktis melibatkan deteksi multiskala dengan tiling citra resolusi tinggi ditambah post-processing untuk menghindari double-count. Pada sayuran hortikultura (cabai, tomat), kombinasi fitur tekstur (GLCM) dan CNN memperbaiki ketahanan terhadap variasi pencahayaan untuk deteksi early blight, sementara dalam proses sortasi pascapanen UNet yang dikombinasikan dengan filtering morfologis memberi hasil grading yang stabil. Robot penyiang di lahan padi/tebu menuntut latensi inferensi sangat rendah (<50 ms) dan efisiensi energi, sehingga kuantisasi dan pruning menjadi keharusan pada model edge.

### 12.1.7 Ekonomi, Adopsi, dan Tata Kelola Data

Teknologi hanya menghasilkan dampak bila ekonomis dan dapat diterima pengguna. Analisis biaya-manfaat harus memperhitungkan CAPEX (kamera, drone, edge compute, lisensi) dan OPEX (anotasi, pemeliharaan model, pelatihan pengguna). Benefit nyata diukur melalui pengurangan pemakaian pestisida, peningkatan yield, dan efisiensi tenaga kerja—angka-angka ini harus dihitung per musim untuk menilai ROI:

$$\mathrm{ROI} = \frac{\mathrm{Benefit} - \mathrm{Biaya}}{\mathrm{Biaya}}$$

misal, jika penghematan dan peningkatan hasil musim itu menghasilkan manfaat finansial $3.000 dan total biaya $1.000, maka ROI = 2 (200%). Model adopsi yang realistis seringkali berupa layanan bersama melalui koperasi atau mitra—mengurangi beban CAPEX per petani. Dari perspektif tata kelola, kepemilikan data harus jelas: data mentah idealnya dimiliki petani, sedangkan agregasi yang dianonimkan dapat dipakai untuk pengembangan model lebih lanjut; prinsip FAIR (Findable, Accessible, Interoperable, Reusable) dan persetujuan terinformasi

perlu diaplikasikan. Audit kinerja per-subkelompok (varietas, daerah) dan transparansi rekomendasi (alas keputusan visual) penting untuk keadilan dan adopsi.

### 12.1.8 Rekomendasi Praktis bagi Peneliti dan Praktisi

Untuk meningkatkan peluang sukses, beberapa langkah praktis direkomendasikan: (1) bangun dataset terbuka yang merepresentasikan varietas lokal serta kondisi musim hujan-kemarau lengkap dengan metadata fenologi; (2) laporkan kinerja model per-musim dan per-lahan, bukan sekadar rata-rata global; (3) siapkan workflow re-label dan active learning untuk mengatasi contoh sulit yang muncul di lapang; (4) gunakan baseline model ringan sebagai floor kinerja—barulah evaluasi peningkatan biaya-manfaat saat memakai model besar; (5) desain antarmuka yang menampilkan rekomendasi beserta tingkat ketidakpastian dan peta perhatian sehingga penyuluh dapat mengaudit hasil; (6) ukur indikator keberlanjutan (mis. kg pestisida/ha atau $m^3$ air/ha) untuk menilai dampak lingkungan selain ekonomi. Dengan menggabungkan perencanaan teknis yang matang, praktek MLOps yang adaptif, dan skema adopsi ekonomi yang inklusif, computer vision dapat menjadi alat yang efektif untuk mewujudkan pertanian yang lebih produktif, efisien, dan berkelanjutan.

## 12.2 Computer Vision di Bidang Kesehatan

Bidang kesehatan merupakan salah satu sektor yang paling cepat mengadopsi teknologi computer vision. Alasan utamanya jelas: data visual sudah lama menjadi bagian integral dari praktik medis, mulai dari foto hasil rontgen, CT-scan, MRI, hingga citra mikroskopis sel. Tantangan dalam kesehatan modern adalah volume data yang sangat besar dan kebutuhan diagnosis yang cepat serta akurat. Tenaga medis, terutama radiolog dan patolog, harus menafsirkan ribuan citra setiap harinya, sesuatu yang rentan terhadap kelelahan, bias subjektif, dan variasi interpretasi antarindividu. Di sinilah computer vision berperan sebagai alat bantu klinis yang mampu meningkatkan konsistensi, mempercepat analisis, serta menyediakan lapisan keamanan tambahan bagi pengambilan keputusan medis.

### 12.2.1 Analisis Citra Radiologi

Radiologi adalah cabang kedokteran yang paling banyak menghasilkan data visual. Setiap rumah sakit besar dapat menghasilkan ribuan citra X-ray, CT, atau MRI dalam sehari. Analisis manual oleh radiolog membutuhkan waktu lama, padahal keputusan klinis sering kali harus diambil dengan segera. Dengan memanfaatkan CNN dan arsitektur modern lainnya, sistem computer vision dapat dilatih untuk mendeteksi pola halus yang bahkan sulit dikenali mata manusia.

Sebagai contoh, penelitian pada deteksi nodul paru di citra CT menunjukkan bahwa model CNN dapat mencapai sensitivitas yang setara dengan radiolog berpengalaman. Pada kasus kanker payudara, algoritma deteksi berbasis deep learning mampu menemukan kelainan mikrokalsifikasi dengan tingkat akurasi yang tinggi, sehingga membantu mengurangi kasus false negative yang sangat berisiko. Meskipun tidak dimaksudkan untuk menggantikan

radiolog, sistem ini dapat bertindak sebagai pembaca kedua (second reader) yang memberikan konfirmasi tambahan atau menyoroti area mencurigakan untuk dianalisis lebih lanjut.

### 12.2.2 Diagnostik Penyakit Kulit dan Aplikasi Mobile

Selain radiologi, computer vision juga dimanfaatkan dalam diagnostik dermatologi. Kulit manusia menyajikan berbagai tanda klinis yang dapat didokumentasikan dengan kamera sederhana, termasuk kamera ponsel. Aplikasi berbasis deep learning yang dilatih menggunakan dataset besar seperti ISIC (International Skin Imaging Collaboration) kini dapat memberikan prediksi probabilitas apakah sebuah lesi kulit bersifat jinak atau ganas.

Kelebihan dari pendekatan ini adalah aksesibilitas. Pasien di daerah terpencil yang jauh dari fasilitas kesehatan spesialis dapat memperoleh deteksi awal hanya dengan mengambil foto melalui aplikasi. Walaupun hasilnya belum dapat menggantikan pemeriksaan klinis, aplikasi ini membuka peluang deteksi dini dan rujukan lebih cepat, yang sangat penting dalam kasus kanker kulit seperti melanoma.

### 12.2.3 Analisis Histopatologi dan Mikroskopis

Bidang lain yang juga berkembang pesat adalah analisis citra histopatologi, yaitu pemeriksaan jaringan yang dilakukan melalui mikroskop. Slide digital dengan resolusi sangat tinggi dapat mencapai ukuran gigapiksel, sehingga analisis manual sangat memakan waktu. Computer vision membantu patolog dalam menyoroti area dengan kepadatan sel abnormal, mitosis yang tinggi, atau infiltrasi jaringan yang mencurigakan.

Model berbasis CNN maupun Vision Transformer (ViT) digunakan untuk mengidentifikasi pola pada citra histologi kanker. Dengan bantuan algoritma, patolog dapat lebih fokus pada bagian jaringan yang paling relevan, sehingga meningkatkan efisiensi kerja sekaligus mengurangi risiko terlewatnya area penting.

### 12.2.4 Monitoring Pasien dan Telemedicine

Tidak hanya untuk diagnosis berbasis citra medis, computer vision juga digunakan dalam pemantauan pasien. Kamera di ruang perawatan dapat mendeteksi perubahan kondisi pasien secara real-time, misalnya mendeteksi gerakan jatuh pada lansia, memantau pola pernapasan bayi, atau menilai ekspresi wajah untuk mengukur tingkat nyeri. Dalam konteks telemedicine, kemampuan ini memungkinkan tenaga medis melakukan pemantauan jarak jauh, yang sangat penting terutama di masa pandemi atau pada daerah dengan keterbatasan tenaga kesehatan.

Selain itu, pengembangan sistem computer vision juga diarahkan pada pengukuran tanda vital non-invasif. Misalnya, algoritma dapat menghitung denyut jantung berdasarkan variasi warna kulit dari video wajah pasien, atau memperkirakan laju pernapasan dari gerakan dada. Inovasi semacam ini membuka peluang monitoring yang lebih nyaman, murah, dan minim risiko.

### 12.2.5 Tantangan Implementasi Klinis

Walaupun manfaatnya jelas, penerapan computer vision di bidang kesehatan tidak lepas dari tantangan. Pertama, standar regulasi yang sangat ketat dalam dunia medis menuntut validasi menyeluruh sebelum sistem dapat dipakai secara luas. Model tidak hanya harus akurat, tetapi

juga harus dapat dipercaya dan menjelaskan dasar pengambilan keputusannya (explainable AI). Kedua, keterbatasan dataset yang representatif menjadi masalah klasik. Banyak dataset medis bersifat privat dan tidak mudah dibagikan karena isu kerahasiaan pasien. Hal ini sering kali membuat model yang dilatih pada data dari satu rumah sakit tidak bekerja baik ketika diterapkan pada populasi berbeda.

Ketiga, aspek etika juga harus diperhatikan. Pasien berhak tahu bagaimana data citra mereka digunakan, siapa yang mengaksesnya, dan sejauh mana keputusan algoritma mempengaruhi terapi yang akan diberikan. Oleh karena itu, kolaborasi erat antara peneliti komputer, dokter, regulator, dan pasien menjadi kunci agar teknologi ini benar-benar bermanfaat tanpa menimbulkan masalah baru.

## 12.3 Computer Vision di Bidang Transportasi

Transportasi modern berada di persimpangan antara kebutuhan mobilitas yang semakin tinggi dan tuntutan keselamatan serta efisiensi. Kemacetan lalu lintas, tingginya angka kecelakaan, serta kebutuhan energi yang besar menjadikan sektor ini sebagai salah satu bidang prioritas untuk inovasi teknologi. Computer vision hadir sebagai salah satu komponen penting dalam membangun sistem transportasi cerdas (intelligent transportation system/ITS) maupun kendaraan otonom. Teknologi ini memungkinkan mesin untuk "melihat" jalan raya, mengenali objek yang relevan, dan mengambil keputusan dengan kecepatan yang sulit dicapai manusia.

### 12.3.1 Kendaraan Otonom

Kendaraan otonom adalah contoh paling nyata dari integrasi computer vision dalam transportasi. Sebuah mobil tanpa pengemudi mengandalkan beragam sensor, termasuk kamera, radar, dan LiDAR, untuk membangun persepsi menyeluruh terhadap lingkungannya. Dari perspektif computer vision, kamera berfungsi sebagai indera utama karena mampu menangkap detail visual secara murah dan fleksibel. Algoritma deteksi objek seperti YOLO digunakan untuk mengenali kendaraan lain, pejalan kaki, maupun rambu lalu lintas dalam waktu nyata.

Lebih dari sekadar deteksi, sistem ini juga memerlukan segmentasi jalan untuk mengenali marka, memprediksi arah lajur, serta mengantisipasi skenario kompleks seperti persimpangan atau jalan dua arah dengan arus padat. Kombinasi Simultaneous Localization and Mapping (SLAM) dengan jaringan saraf mendalam memungkinkan kendaraan mengetahui posisinya secara presisi dalam peta tiga dimensi. Uji coba oleh perusahaan seperti Tesla, Waymo, atau Baidu menunjukkan bahwa tanpa kemampuan vision yang andal, kendaraan otonom tidak mungkin berfungsi dengan aman.

### 12.3.2 Sistem Transportasi Cerdas (ITS)

Tidak semua inovasi vision diarahkan pada kendaraan otonom. Di banyak kota besar, intelligent transportation system dikembangkan untuk mengurangi kemacetan dan meningkatkan keselamatan pengguna jalan. Kamera lalu lintas yang dipasang di persimpangan atau sepanjang jalur utama tidak hanya berfungsi merekam, tetapi juga dilengkapi dengan

algoritma analisis. Sistem mampu mendeteksi jumlah kendaraan, memantau kepadatan lalu lintas, bahkan mengidentifikasi kecelakaan dalam hitungan detik.

Dengan pendekatan ini, pengaturan lampu lalu lintas dapat disesuaikan secara dinamis sesuai kepadatan nyata, bukan berdasarkan jadwal statis. Beberapa kota juga mengintegrasikan teknologi ini dengan pusat kontrol transportasi sehingga respon terhadap kecelakaan dapat lebih cepat, termasuk pengiriman ambulans atau pemadam kebakaran. Keunggulannya terletak pada kemampuan sistem mengambil keputusan adaptif, yang pada akhirnya mengurangi waktu perjalanan sekaligus meningkatkan keselamatan.

### 12.3.3 Analisis Pola Perjalanan

Computer vision juga digunakan untuk menganalisis pola perjalanan masyarakat. Dengan memproses data dari kamera publik, sistem dapat memperkirakan arus pergerakan kendaraan pada jam-jam tertentu, mendeteksi perubahan perilaku pengguna jalan, atau memprediksi titik-titik rawan kemacetan. Hasil analisis ini tidak hanya berguna bagi perencana transportasi kota, tetapi juga dapat dimanfaatkan oleh perusahaan transportasi daring untuk mengoptimalkan rute kendaraan dan mengurangi waktu tunggu penumpang.

Analisis berbasis vision memungkinkan perencanaan transportasi publik yang lebih responsif terhadap kebutuhan masyarakat. Misalnya, jika sistem mendeteksi bahwa rute tertentu mengalami lonjakan pengguna pada jam tertentu, operator dapat menambah armada bus atau kereta di jalur tersebut. Dengan demikian, kebijakan transportasi tidak lagi didasarkan pada perkiraan, melainkan bukti visual yang objektif.

### 12.3.4 Tantangan Implementasi

Walaupun potensinya besar, implementasi computer vision dalam transportasi tidak bebas dari kendala. Tantangan utama terletak pada kondisi lingkungan yang sangat dinamis: pencahayaan berubah-ubah, cuaca ekstrem, jalan yang tergenang air, atau objek tak terduga di jalan raya dapat mengganggu akurasi sistem. Model yang bekerja baik di laboratorium belum tentu tangguh ketika diterapkan di jalan dengan kompleksitas tinggi.

Selain itu, aspek regulasi juga menjadi perhatian. Untuk kendaraan otonom, kejelasan tanggung jawab hukum jika terjadi kecelakaan masih menjadi perdebatan. Apakah kesalahan ditanggung produsen kendaraan, pengembang algoritma, atau pengguna? Di sisi lain, penggunaan kamera publik untuk ITS menimbulkan isu privasi, karena citra yang diambil sering kali juga merekam identitas pengguna jalan. Oleh sebab itu, keberhasilan implementasi teknologi vision dalam transportasi menuntut tidak hanya kematangan teknis, tetapi juga kerangka regulasi dan etika yang jelas.

## 12.4 Computer Vision di Bidang Keamanan dan Forensik

Keamanan dan forensik merupakan bidang yang sejak lama bergantung pada bukti visual. Kamera pengawas, foto, dan video menjadi sumber utama untuk memantau aktivitas manusia sekaligus merekonstruksi kejadian kriminal. Namun, jumlah data visual yang semakin masif membuat analisis manual tidak lagi memadai. Computer vision menawarkan cara baru untuk

mengekstraksi informasi dari citra dan video, sehingga memungkinkan deteksi ancaman secara cepat, analisis forensik yang lebih teliti, dan pengambilan keputusan yang berbasis data.

### 12.4.1 Pengawasan Berbasis Kamera Cerdas

Di kota-kota besar, kamera pengawas (closed-circuit television/CCTV) tersebar hampir di setiap sudut jalan. Tantangan utama bukan lagi ketersediaan data, melainkan kemampuan untuk memprosesnya secara real-time. Sistem computer vision kini mampu mendeteksi perilaku mencurigakan, mengenali pergerakan abnormal, atau mengidentifikasi individu berdasarkan penampilan visual.

Teknik seperti deteksi objek (misalnya dengan YOLO atau Faster R-CNN) memungkinkan sistem secara otomatis melacak kendaraan yang melanggar aturan lalu lintas atau mendeteksi orang yang memasuki area terbatas. Lebih jauh lagi, analisis perilaku berbasis action recognition dapat digunakan untuk mengenali pola kekerasan, pencurian, atau perkelahian dalam kerumunan. Hal ini memberikan keuntungan signifikan bagi aparat keamanan, karena sistem dapat bertindak sebagai "penjaga digital" yang tidak pernah lelah.

### 12.4.2 Pengenalan Wajah dan Identifikasi Individu

Salah satu aplikasi paling dikenal dari computer vision di bidang keamanan adalah pengenalan wajah (face recognition). Dengan memanfaatkan jaringan saraf konvolusional atau arsitektur lebih baru seperti ArcFace, sistem dapat mengenali identitas individu dengan tingkat akurasi tinggi. Teknologi ini digunakan di berbagai bandara internasional untuk mempercepat proses imigrasi, serta dalam sistem keamanan gedung untuk membatasi akses hanya kepada orang yang berwenang.

Di sisi lain, penggunaan pengenalan wajah menimbulkan perdebatan serius terkait privasi. Ada kekhawatiran bahwa teknologi ini dapat digunakan secara berlebihan oleh otoritas atau pihak swasta untuk mengawasi warga tanpa persetujuan. Oleh sebab itu, meskipun secara teknis sangat bermanfaat, implementasi pengenalan wajah harus selalu disertai kerangka hukum dan etika yang jelas.

### 12.4.3 Analisis Forensik Digital

Dalam ranah forensik, computer vision membantu penegak hukum menganalisis bukti visual dari TKP. Misalnya, citra dari kamera CCTV dapat ditingkatkan kualitasnya menggunakan teknik super-resolution, sehingga wajah atau nomor kendaraan menjadi lebih jelas. Selain itu, algoritma video forensics digunakan untuk mendeteksi manipulasi digital, seperti deepfake atau rekayasa citra, yang kian sering muncul dalam kasus kriminal maupun penyebaran disinformasi.

Computer vision juga dimanfaatkan dalam rekonstruksi kejadian. Dengan menganalisis arah gerakan orang atau kendaraan dalam video, penyidik dapat menyusun kembali urutan peristiwa dengan tingkat akurasi yang lebih tinggi. Dalam beberapa kasus, informasi spasial dari citra 3D bahkan digunakan untuk memperkirakan jalur peluru atau lokasi pasti pelaku dalam sebuah insiden.

### 12.4.4 Deteksi Ancaman dan Keamanan Siber Visual

Aspek keamanan tidak hanya terbatas pada ruang fisik, tetapi juga merambah ke dunia digital. Computer vision kini digunakan untuk memantau konten berbahaya di platform daring, seperti penyebaran gambar kekerasan, pornografi anak, atau konten terorisme. Sistem otomatis dapat memfilter dan menandai konten semacam itu dengan cepat, sehingga mempercepat proses moderasi yang sebelumnya sangat mengandalkan laporan manual.

Selain itu, bidang forensik siber juga mulai memanfaatkan analisis citra untuk melacak pola serangan, misalnya mendeteksi phishing website yang meniru tampilan visual situs resmi. Dengan membandingkan kesamaan tata letak, warna, dan ikon, sistem vision dapat mengenali situs palsu lebih efektif dibandingkan metode berbasis teks semata.

### 12.4.5 Tantangan Etika dan Hukum

Walaupun computer vision membuka peluang besar dalam keamanan dan forensik, penerapannya tidak lepas dari tantangan serius. Masalah privasi menjadi isu utama, karena pengawasan visual yang masif berpotensi mengurangi kebebasan individu. Selain itu, bias dalam algoritma pengenalan wajah masih menjadi perhatian. Beberapa penelitian menunjukkan bahwa akurasi sistem sering kali lebih rendah pada kelompok etnis atau gender tertentu, yang dapat memicu ketidakadilan hukum.

Di sisi lain, penggunaan bukti visual hasil analisis algoritma dalam pengadilan juga menimbulkan pertanyaan: sejauh mana hakim dan juri dapat mempercayai hasil dari sistem otomatis? Apakah algoritma yang digunakan transparan dan dapat diaudit? Pertanyaan-pertanyaan ini menegaskan bahwa penerapan computer vision dalam keamanan harus dibarengi dengan tata kelola yang ketat, audit berkala, serta keterlibatan multi-pihak dalam pengawasan.

## 12.5 Computer Vision di Bidang Lingkungan dan Konservasi

Isu lingkungan dan konservasi semakin mendesak di tengah perubahan iklim, deforestasi, dan hilangnya keanekaragaman hayati. Pemantauan ekosistem tradisional sering bergantung pada survei manual, yang memerlukan waktu, biaya, dan tenaga besar. Pendekatan ini tidak mampu menjawab tantangan skala global, di mana perubahan ekologi dapat terjadi sangat cepat. Computer vision menghadirkan paradigma baru: dengan memanfaatkan citra satelit, drone, kamera jebak (camera trap), maupun sensor bawah laut, sistem mampu mengidentifikasi pola perubahan lingkungan secara objektif, cepat, dan berkelanjutan.

### 12.5.1 Pemantauan Ekosistem dan Tutupan Lahan

Pemantauan tutupan lahan adalah langkah krusial dalam mengukur dampak deforestasi dan degradasi lingkungan. Citra satelit resolusi tinggi, ketika dianalisis dengan teknik segmentasi berbasis deep learning, dapat mengklasifikasikan area hutan, perkebunan, lahan pertanian, maupun wilayah terbangun. Dengan algoritma seperti U-Net atau DeepLab, perubahan tutupan

lahan dari waktu ke waktu dapat dideteksi secara otomatis, sehingga memungkinkan pemerintah dan lembaga konservasi memetakan titik-titik kritis deforestasi.

Sebagai contoh, organisasi internasional memanfaatkan data satelit bersama algoritma computer vision untuk mengidentifikasi aktivitas penebangan liar di kawasan hutan Amazon hanya dalam hitungan hari. Tanpa pendekatan berbasis citra, kegiatan seperti ini bisa tidak terdeteksi hingga berbulan-bulan, yang berarti kerusakan ekosistem sudah terlanjur meluas.

### 12.5.2 Konservasi Satwa Liar

Konservasi satwa liar menjadi bidang lain yang mendapat manfaat signifikan. Kamera jebak yang dipasang di hutan, padang rumput, atau kawasan lindung menghasilkan jutaan gambar hewan setiap tahunnya. Analisis manual terhadap data sebesar ini hampir tidak mungkin dilakukan. Dengan computer vision, identifikasi spesies dapat dilakukan secara otomatis, bahkan hingga tingkat individu dalam beberapa kasus.

Teknik deep metric learning misalnya, telah digunakan untuk mengenali pola unik pada tubuh hewan, seperti belang zebra atau tutul macan. Dengan cara ini, peneliti dapat melacak pergerakan individu tanpa perlu penandaan fisik yang invasif. Selain itu, sistem deteksi berbasis YOLO digunakan untuk menghitung populasi satwa, sehingga memberikan data real-time tentang dinamika populasi yang penting bagi kebijakan konservasi.

### 12.5.3 Pemantauan Laut dan Perairan

Tidak hanya di darat, computer vision juga memainkan peran penting dalam pemantauan lingkungan laut. Kamera bawah air digunakan untuk memantau kondisi terumbu karang, mendeteksi spesies ikan, serta mengukur tingkat kekeruhan air. Dengan bantuan algoritma segmentasi, tingkat kerusakan karang akibat pemutihan dapat diukur lebih cepat dan akurat dibandingkan pengamatan manual penyelam.

Di sisi lain, citra satelit digunakan untuk mendeteksi tumpahan minyak atau pencemaran laut lainnya. Analisis tekstur dan warna air laut membantu mengidentifikasi area yang terkena dampak, sehingga memudahkan respons cepat dari otoritas terkait. Inovasi ini sangat penting karena kerusakan ekosistem laut sering kali sulit dipulihkan jika tidak ditangani sejak dini.

### 12.5.4 Peran dalam Mitigasi Perubahan Iklim

Computer vision juga berkontribusi pada upaya mitigasi perubahan iklim. Salah satu aplikasi utama adalah penghitungan biomassa dan stok karbon melalui citra udara atau satelit. Model berbasis deep learning dapat mengestimasi volume pohon dan vegetasi, sehingga memungkinkan perhitungan karbon tersimpan dalam suatu wilayah hutan. Data ini penting untuk mendukung skema perdagangan karbon serta evaluasi kebijakan lingkungan global.

Selain itu, sistem vision membantu memantau dampak bencana iklim seperti banjir, kebakaran hutan, atau badai tropis. Citra satelit sebelum dan sesudah bencana dianalisis untuk mengukur skala kerusakan, sehingga mendukung pengambilan keputusan cepat dalam penanganan

darurat. Dengan demikian, computer vision tidak hanya berfungsi dalam konservasi jangka panjang, tetapi juga dalam mitigasi bencana lingkungan yang semakin sering terjadi.

### 12.5.5 Tantangan dan Etika Konservasi Digital

Walaupun potensinya besar, penerapan computer vision dalam konservasi menghadapi sejumlah tantangan. Pertama, keterbatasan infrastruktur di daerah terpencil sering menghambat pengumpulan data secara konsisten. Kedua, bias dalam data dapat memengaruhi hasil analisis, misalnya kamera jebak yang lebih sering menangkap hewan berukuran besar dibandingkan yang kecil.

Aspek etika juga tidak kalah penting. Pemantauan berbasis kamera kadang berisiko mengganggu habitat alami atau menimbulkan pertanyaan tentang privasi masyarakat lokal yang terekam dalam proses konservasi. Oleh karena itu, penggunaan teknologi ini harus disertai prinsip-prinsip konservasi yang berkelanjutan dan berkeadilan.

## 12.6 Computer Vision di Bidang Industri dan Manufaktur

Industri dan manufaktur modern bergerak menuju era Industry 4.0, di mana otomasi, data, dan kecerdasan buatan menjadi pilar utama. Dalam konteks ini, computer vision berfungsi sebagai "mata digital" yang memungkinkan mesin mengamati, memahami, dan mengambil keputusan berdasarkan kondisi nyata di lini produksi. Keunggulan utama teknologi ini adalah kemampuannya melakukan inspeksi visual secara cepat, konsisten, dan tanpa lelah—sesuatu yang sulit ditandingi oleh tenaga manusia. Hal ini menjadikannya komponen vital dalam menjaga kualitas produk, meningkatkan efisiensi proses, serta mengurangi risiko kesalahan produksi.

### 12.6.1 Inspeksi Kualitas Produk

Salah satu aplikasi paling dominan dari computer vision di industri adalah inspeksi kualitas produk. Proses ini sebelumnya dilakukan oleh operator manusia yang memeriksa cacat pada permukaan, warna, atau bentuk produk. Namun, pendekatan manual rentan terhadap kelelahan, subjektivitas, dan keterbatasan kecepatan. Dengan sistem vision, kamera beresolusi tinggi dipasang di lini produksi untuk menangkap citra produk, kemudian algoritma deteksi citra memverifikasi kesesuaian dengan standar mutu.

Sebagai contoh, dalam industri semikonduktor, computer vision digunakan untuk mendeteksi retakan mikroskopis pada wafer silikon. Di industri makanan dan minuman, sistem mampu mengidentifikasi cacat pada botol, kemasan, atau kontaminasi benda asing. Dengan algoritma segmentasi, bahkan kerusakan kecil sekalipun dapat terdeteksi secara konsisten, sehingga mencegah produk cacat sampai ke konsumen.

### 12.6.2 Otomatisasi Proses Produksi

Selain inspeksi kualitas, computer vision juga mendukung otomatisasi proses produksi. Robot industri yang dilengkapi kamera mampu mengenali posisi komponen dengan presisi tinggi, sehingga proses perakitan dapat dilakukan tanpa kesalahan. Dalam industri otomotif, misalnya,

kamera digunakan untuk memastikan baut terpasang dengan benar, kabel tidak terpuntir, dan cat pada bodi mobil merata.

Integrasi vision dengan sistem robotik tidak hanya meningkatkan akurasi, tetapi juga fleksibilitas. Jika sebelumnya robot hanya dapat menangani tugas repetitif yang sama, kini robot vision-enabled mampu beradaptasi dengan variasi produk atau perubahan konfigurasi jalur produksi. Hal ini memungkinkan pabrik beroperasi dengan model mass customization, yaitu produksi skala besar tetapi tetap bisa disesuaikan dengan kebutuhan pelanggan.

### 12.6.3 Pemeliharaan Prediktif

Computer vision juga digunakan dalam pemeliharaan prediktif (predictive maintenance). Kamera termal atau sensor visual lainnya dipasang pada mesin untuk memantau kondisi operasional secara terus-menerus. Algoritma analisis citra kemudian mendeteksi tanda-tanda keausan, retakan, atau panas berlebih yang berpotensi menyebabkan kerusakan.

Dengan pendekatan ini, perawatan dapat dilakukan sebelum terjadi kerusakan total, sehingga mengurangi downtime produksi yang biasanya sangat merugikan. Misalnya, pada industri energi, kamera termal digunakan untuk mendeteksi panas abnormal pada turbin atau generator. Data visual yang diolah oleh model deep learning memberikan peringatan dini, sehingga tim teknis dapat melakukan tindakan pencegahan.

### 12.6.4 Manajemen Logistik dan Gudang

Dalam rantai pasok, computer vision membantu mengoptimalkan manajemen logistik dan gudang. Kamera dipasang di pintu masuk gudang untuk mengenali kode pada paket, mengukur dimensi barang, atau memverifikasi label pengiriman. Teknologi ini menggantikan proses manual yang lambat dan rentan kesalahan.

Di sisi lain, robot otonom di gudang menggunakan computer vision untuk bernavigasi, menghindari rintangan, serta mengambil barang dari rak. Perusahaan e-commerce global telah memanfaatkan sistem ini untuk mempercepat proses pemenuhan pesanan. Dengan analisis visual, waktu pencarian barang dapat dikurangi drastis, sekaligus mengurangi ketergantungan pada tenaga kerja manusia di area berisiko tinggi.

### 12.6.5 Tantangan Implementasi di Industri

Meskipun manfaat computer vision di industri jelas, implementasinya juga menghadapi beberapa tantangan. Pertama, variasi kondisi pencahayaan, debu, atau getaran di lini produksi dapat mengganggu kualitas citra. Kedua, model yang dilatih pada dataset terbatas sering kali gagal ketika dihadapkan pada variasi produk baru. Oleh karena itu, strategi continuous learning dan domain adaptation menjadi penting untuk menjaga performa sistem.

Selain aspek teknis, tantangan lainnya adalah integrasi dengan sistem produksi yang sudah ada. Banyak pabrik masih menggunakan mesin lama yang tidak dirancang untuk otomasi modern.

Untuk itu, dibutuhkan solusi vision yang modular, ringan, dan mampu bekerja bersama sistem lama tanpa memerlukan investasi besar.

### 12.6.6 Dampak Ekonomi dan Sosial

Penerapan computer vision di industri memberikan dampak ekonomi yang signifikan. Efisiensi produksi meningkat, biaya tenaga kerja berkurang, dan risiko produk cacat menurun. Namun, dari sisi sosial, muncul kekhawatiran mengenai berkurangnya lapangan kerja bagi operator manusia. Hal ini menuntut adanya strategi reskilling dan upskilling, agar tenaga kerja tetap relevan di era industri berbasis otomasi.

## 12.7 Computer Vision di Bidang Pendidikan

Pendidikan adalah sektor yang terus berevolusi mengikuti perkembangan teknologi. Di era digital, pembelajaran tidak lagi terbatas pada interaksi tatap muka di ruang kelas, melainkan diperluas melalui platform daring, simulasi interaktif, serta integrasi kecerdasan buatan. Dalam kerangka ini, computer vision hadir sebagai teknologi pendukung yang mampu meningkatkan kualitas pembelajaran, memperkaya pengalaman belajar, sekaligus membantu pendidik memahami dinamika siswa dengan lebih mendalam.

### 12.7.1 Analisis Interaksi Siswa di Kelas

Salah satu aplikasi computer vision dalam pendidikan adalah pemantauan interaksi siswa di kelas. Kamera yang terpasang dapat merekam ekspresi wajah, gerakan tubuh, maupun tingkat perhatian siswa selama proses pembelajaran. Dengan bantuan algoritma facial expression recognition atau pose estimation, sistem mampu menilai apakah siswa sedang fokus, bingung, atau kehilangan minat.

Data ini memberi masukan berharga bagi guru untuk menyesuaikan metode pengajaran secara real-time. Misalnya, jika banyak siswa menunjukkan ekspresi kebingungan, guru dapat mengulang materi atau menggunakan pendekatan yang lebih sederhana. Dengan cara ini, computer vision tidak dimaksudkan menggantikan peran guru, melainkan sebagai alat bantu refleksi yang meningkatkan efektivitas pembelajaran.

### 12.7.2 Evaluasi Berbasis Gesture dan Aktivitas

Dalam pembelajaran praktikum, terutama di bidang sains atau teknik, computer vision digunakan untuk mengevaluasi keterampilan motorik siswa. Sistem dapat mengenali apakah prosedur laboratorium dilakukan dengan langkah yang benar, atau apakah siswa melakukan kesalahan yang berpotensi berbahaya.

Sebagai contoh, pada pelatihan kedokteran, simulasi operasi berbasis VR yang dipadukan dengan computer vision memungkinkan penilaian objektif terhadap keterampilan bedah mahasiswa. Sistem tidak hanya menilai hasil akhir, tetapi juga ketepatan gerakan tangan,

kecepatan, dan koordinasi selama proses berlangsung. Dengan demikian, evaluasi menjadi lebih komprehensif dibandingkan penilaian manual semata.

### 12.7.3 Pembelajaran Inklusif dan Aksesibilitas

Computer vision juga memainkan peran penting dalam pendidikan inklusif. Bagi siswa dengan kebutuhan khusus, teknologi ini dapat membantu menciptakan lingkungan belajar yang lebih ramah. Misalnya, sistem pengenalan bahasa isyarat berbasis vision memungkinkan siswa tunarungu berkomunikasi lebih mudah dengan guru maupun teman sekelas.

Selain itu, teknologi text-to-speech berbasis OCR (Optical Character Recognition) membantu siswa tunanetra membaca buku cetak dengan cara mengonversi teks menjadi suara. Dengan integrasi computer vision, hambatan fisik yang selama ini membatasi akses pendidikan dapat dikurangi secara signifikan, membuka peluang belajar yang lebih setara.

### 12.7.4 Ujian dan Pengawasan Akademik

Dalam konteks ujian, computer vision digunakan untuk mendukung sistem e-proctoring atau pengawasan ujian daring. Kamera laptop siswa dapat mendeteksi perilaku mencurigakan, seperti sering menoleh ke samping, berbicara dengan orang lain, atau mencoba membuka materi tambahan. Sistem ini membantu menjaga integritas akademik, terutama dalam skala besar ketika pengawasan manual tidak memungkinkan.

Walau menimbulkan perdebatan terkait privasi, teknologi ini menjadi solusi praktis di era pembelajaran jarak jauh. Untuk meminimalisasi resistensi, beberapa institusi mulai menerapkan model hybrid, di mana computer vision digunakan sebagai lapisan tambahan, sementara keputusan akhir tetap berada di tangan pengawas manusia.

### 12.7.5 Tantangan Implementasi di Pendidikan

Implementasi computer vision dalam pendidikan juga menghadapi sejumlah tantangan. Pertama, keterbatasan infrastruktur teknologi di sekolah-sekolah, khususnya di daerah terpencil, membuat adopsi tidak merata. Kedua, isu privasi dan etika muncul ketika data visual siswa direkam dan dianalisis. Perlindungan data harus menjadi prioritas, agar penggunaan teknologi tidak menimbulkan risiko baru bagi keamanan siswa.

Selain itu, efektivitas sistem sangat bergantung pada kualitas algoritma. Misalnya, pengenalan ekspresi wajah bisa bias terhadap ras atau kelompok tertentu, sehingga menimbulkan interpretasi yang tidak adil. Oleh karena itu, pengembangan teknologi ini harus selalu disertai evaluasi kritis serta keterlibatan pendidik, siswa, dan masyarakat luas.

## 12.8 Computer Vision di Bidang Perdagangan dan Retail

Perdagangan dan retail merupakan salah satu sektor yang paling cepat mengadopsi teknologi computer vision. Persaingan yang semakin ketat menuntut inovasi dalam meningkatkan pengalaman pelanggan, mengoptimalkan rantai pasok, serta menjaga efisiensi operasional. Jika pada masa lalu analisis perilaku konsumen hanya mengandalkan survei dan observasi manual, kini kamera dan algoritma vision dapat memberikan wawasan yang lebih kaya, objektif, dan

real-time. Dengan demikian, computer vision menjadi komponen penting dalam transformasi digital retail modern.

### 12.8.1 Analisis Perilaku Konsumen

Salah satu penerapan utama adalah analisis perilaku konsumen di dalam toko. Kamera yang dipasang di area strategis dapat merekam pola pergerakan pelanggan, durasi interaksi dengan produk tertentu, hingga ekspresi wajah yang menunjukkan ketertarikan atau kebingungan. Data ini diproses menggunakan algoritma pose estimation atau facial expression recognition untuk menghasilkan wawasan mengenai preferensi konsumen.

Sebagai contoh, jika banyak pelanggan menghabiskan waktu di area tertentu tetapi tingkat pembelian rendah, manajer toko dapat mengevaluasi ulang strategi penempatan produk atau harga. Dengan demikian, keputusan bisnis tidak lagi berdasarkan intuisi semata, melainkan bukti visual yang terukur.

### 12.8.2 Sistem Kasir Otomatis

Computer vision juga memungkinkan terwujudnya sistem kasir tanpa kasir (cashierless store). Melalui kombinasi kamera, sensor, dan algoritma deteksi objek, sistem dapat mengenali produk yang diambil pelanggan dan secara otomatis menambahkan ke keranjang belanja virtual. Pembayaran dilakukan secara digital ketika pelanggan keluar toko, tanpa perlu antre di kasir.

Konsep ini telah diimplementasikan oleh beberapa perusahaan besar, dan terbukti mampu mengurangi waktu tunggu sekaligus meningkatkan kenyamanan pelanggan. Tantangan teknis utamanya adalah akurasi deteksi dalam kondisi toko yang padat serta integrasi dengan sistem pembayaran yang aman. Namun, seiring kemajuan model vision dan edge computing, sistem ini semakin realistis untuk diadopsi secara luas.

### 12.8.3 Manajemen Stok dan Rantai Pasok

Selain berfokus pada pelanggan, computer vision juga mendukung manajemen stok dan rantai pasok. Kamera yang dipasang di gudang atau rak toko dapat memantau ketersediaan produk secara otomatis. Algoritma deteksi objek mengidentifikasi produk yang hampir habis, sehingga staf dapat segera melakukan pengisian ulang.

Lebih jauh lagi, integrasi dengan sistem rantai pasok memungkinkan prediksi kebutuhan berdasarkan pola permintaan yang terlihat. Misalnya, jika kamera mendeteksi lonjakan permintaan pada produk tertentu, sistem dapat menyesuaikan pesanan ke pemasok sebelum stok benar-benar habis. Pendekatan ini mengurangi risiko out of stock yang sering menurunkan kepuasan pelanggan.

### 12.8.4 Keamanan Toko dan Pencegahan Kehilangan

Retail juga menghadapi tantangan serius berupa kehilangan barang akibat pencurian. Computer vision digunakan untuk mendeteksi perilaku mencurigakan di dalam toko, seperti seseorang

yang sering menoleh ke arah kamera atau berusaha menyembunyikan barang. Dengan teknik anomaly detection, sistem dapat memberi peringatan dini kepada petugas keamanan.

Selain itu, integrasi dengan sistem pengawasan otomatis membantu mencegah kesalahan transaksi di kasir. Kamera dapat memverifikasi apakah barang yang dibawa keluar toko sesuai dengan yang dibayar, sehingga mengurangi potensi kerugian akibat kecurangan.

### 12.8.5 Tantangan Implementasi

Walaupun membawa manfaat besar, adopsi computer vision di retail juga menghadapi sejumlah kendala. Investasi awal yang tinggi untuk perangkat keras dan integrasi sistem menjadi salah satu hambatan, terutama bagi toko berskala kecil. Selain itu, isu privasi konsumen menjadi perhatian utama. Pelanggan mungkin merasa tidak nyaman jika perilaku mereka direkam dan dianalisis secara detail.

Untuk mengatasi hal ini, beberapa perusahaan mulai menerapkan kebijakan transparansi, seperti pemberitahuan eksplisit mengenai penggunaan kamera dan penyimpanan data. Di sisi teknis, tantangan lainnya adalah variasi pencahayaan, keramaian toko, serta keberagaman produk yang memerlukan model vision dengan generalisasi tinggi.

## 12.9 Computer Vision di Bidang Smart City dan Infrastruktur Perkotaan

Konsep smart city lahir dari kebutuhan untuk mengelola kompleksitas kehidupan perkotaan dengan lebih efisien, aman, dan berkelanjutan. Pertumbuhan populasi yang pesat di kota-kota besar membawa tantangan besar dalam transportasi, energi, lingkungan, hingga layanan publik. Untuk itu, dibutuhkan sistem cerdas yang mampu memantau kondisi kota secara real-time, menganalisis data, dan memberikan rekomendasi keputusan berbasis bukti. Di sinilah computer vision memainkan peran strategis sebagai “mata” kota pintar, yang menangkap dinamika perkotaan melalui kamera dan sensor visual, lalu mengubahnya menjadi informasi yang bermanfaat bagi pemerintah maupun warga.

### 12.9.1 Manajemen Lalu Lintas dan Mobilitas

Salah satu penerapan utama computer vision dalam smart city adalah pengelolaan lalu lintas. Kamera jalan raya yang terintegrasi dengan algoritma deteksi objek dapat menghitung jumlah kendaraan, mengidentifikasi pelanggaran lalu lintas, serta menilai kepadatan di berbagai titik kota. Data ini digunakan untuk mengatur sinyal lampu lalu lintas secara adaptif, sehingga mengurangi kemacetan dan waktu tempuh.

Lebih jauh lagi, sistem vision dapat mendukung layanan transportasi publik dengan memprediksi lonjakan penumpang pada jam tertentu. Misalnya, kamera di halte bus atau stasiun kereta dapat memperkirakan jumlah penumpang yang menunggu, sehingga operator transportasi dapat menyesuaikan armada secara dinamis. Dengan demikian, computer vision berkontribusi langsung pada peningkatan mobilitas perkotaan.

### 12.9.2 Pemantauan Keamanan Publik

Smart city juga memanfaatkan computer vision untuk meningkatkan keamanan publik. Kamera yang tersebar di ruang kota dapat mendeteksi aktivitas mencurigakan, kerumunan yang

berpotensi menimbulkan kerusuhan, atau insiden darurat seperti kebakaran dan kecelakaan. Dengan analisis perilaku berbasis video, aparat keamanan dapat memperoleh peringatan dini sebelum situasi berkembang menjadi masalah serius.

Sebagai contoh, teknologi pengenalan wajah digunakan di beberapa kota untuk mengidentifikasi orang yang masuk daftar pencarian. Walaupun masih menuai kontroversi, sistem ini telah terbukti membantu aparat dalam mempercepat penangkapan pelaku kriminal. Dengan tata kelola etis yang baik, computer vision berpotensi meningkatkan rasa aman warga tanpa mengorbankan kebebasan sipil.

### 12.9.3 Manajemen Energi dan Infrastruktur

Computer vision juga berperan dalam pemeliharaan infrastruktur kota. Kamera drone digunakan untuk memeriksa kondisi jembatan, gedung, atau jaringan listrik, mendeteksi retakan kecil atau korosi yang berpotensi membahayakan. Dengan analisis otomatis, pemerintah kota dapat melakukan pemeliharaan prediktif sebelum kerusakan menjadi parah.

Di bidang energi, computer vision membantu mengoptimalkan penggunaan listrik. Misalnya, kamera yang terhubung ke sistem lampu jalan dapat mendeteksi tingkat keramaian di suatu area dan menyesuaikan intensitas pencahayaan. Pendekatan ini tidak hanya meningkatkan efisiensi energi, tetapi juga mengurangi emisi karbon kota.

### 12.9.4 Layanan Publik dan Partisipasi Warga

Konsep smart city menekankan keterlibatan warga sebagai pengguna sekaligus produsen data. Computer vision dapat digunakan dalam aplikasi layanan publik, misalnya mendeteksi tumpukan sampah di jalan untuk segera ditangani petugas kebersihan, atau memantau kondisi taman kota agar tetap terjaga.

Lebih jauh, integrasi dengan aplikasi mobile memungkinkan warga mengirim foto kondisi infrastruktur rusak (jalan berlubang, lampu mati) yang kemudian dianalisis secara otomatis untuk menentukan prioritas perbaikan. Dengan demikian, computer vision tidak hanya memperkuat peran pemerintah kota, tetapi juga memperluas partisipasi aktif masyarakat dalam menjaga lingkungannya.

### 12.9.5 Tantangan Implementasi Smart City

Meskipun potensinya besar, penerapan computer vision dalam smart city menghadapi tantangan serius. Isu privasi menjadi yang paling menonjol, karena kamera kota merekam jutaan wajah dan aktivitas warga setiap harinya. Tanpa regulasi yang ketat, risiko penyalahgunaan data sangat tinggi. Selain itu, biaya infrastruktur yang besar dan kebutuhan integrasi dengan sistem lama sering menjadi hambatan bagi kota di negara berkembang.

Di sisi teknis, heterogenitas data visual dari berbagai kamera dan sensor memerlukan standar interoperabilitas yang jelas. Jika tidak, sistem akan terfragmentasi dan sulit mencapai efisiensi optimal. Oleh karena itu, keberhasilan smart city tidak hanya ditentukan oleh kecanggihan

teknologi vision, tetapi juga oleh kebijakan tata kelola data, transparansi penggunaan, dan partisipasi publik yang inklusif.

## 12.10 Computer Vision di Bidang Pertahanan dan Militer

Pertahanan dan militer merupakan salah satu bidang yang paling intensif dalam pemanfaatan teknologi mutakhir. Kompleksitas medan operasi, kebutuhan pengambilan keputusan cepat, serta risiko tinggi bagi personel menuntut sistem yang mampu memberikan informasi akurat dalam waktu singkat. Dalam konteks ini, computer vision menjadi teknologi kunci yang memperluas kapabilitas intelijen, pengawasan, dan operasi militer modern. Dengan kamera, drone, maupun satelit, computer vision berfungsi sebagai "indra tambahan" yang mampu memperbesar jangkauan, meningkatkan ketelitian, dan mempercepat analisis situasi.

### 12.10.1 Pengintaian dan Pengawasan Medan

Salah satu aplikasi utama computer vision dalam pertahanan adalah pengintaian (reconnaissance) dan pengawasan (surveillance). Drone yang dilengkapi kamera resolusi tinggi dapat memantau pergerakan pasukan musuh, mendeteksi kendaraan militer, atau mengidentifikasi perubahan posisi strategis di medan perang. Algoritma deteksi objek berbasis YOLO atau Faster R-CNN memungkinkan sistem mengenali target dengan cepat bahkan dalam kondisi kompleks seperti kabut asap, malam hari, atau area padat vegetasi.

Lebih dari itu, analisis citra satelit menggunakan deep learning dapat mendeteksi pembangunan infrastruktur militer baru, pergerakan kapal di pelabuhan, atau aktivitas tidak biasa di wilayah tertentu. Dengan demikian, computer vision mempercepat siklus intelijen dari pengumpulan data hingga penyampaian laporan strategis.

### 12.10.2 Navigasi Otonom Kendaraan Militer

Kendaraan otonom bukan hanya tren di sektor sipil, tetapi juga menjadi prioritas di dunia militer. Tank, kendaraan logistik, atau robot penjinak bom dapat dipandu dengan sistem computer vision untuk bernavigasi secara mandiri di medan berbahaya. Segmentasi jalan, deteksi rintangan, serta terrain classification memungkinkan kendaraan beroperasi tanpa operator langsung, sehingga mengurangi risiko bagi prajurit.

Selain kendaraan darat, konsep serupa diterapkan pada kapal laut dan pesawat tanpa awak (unmanned aerial vehicles). Dengan computer vision, sistem navigasi dapat berfungsi secara adaptif, menyesuaikan jalur dengan kondisi medan sekaligus menghindari deteksi musuh.

### 12.10.3 Identifikasi Target dan Sistem Persenjataan

Computer vision juga digunakan dalam sistem persenjataan pintar. Kamera yang terintegrasi dengan algoritma deteksi memungkinkan senjata mengunci target secara otomatis dengan akurasi tinggi. Misalnya, sistem pertahanan udara menggunakan vision untuk mendeteksi dan melacak rudal atau pesawat musuh yang bergerak cepat.

Namun, penerapan ini menimbulkan perdebatan etis, terutama terkait dengan konsep lethal autonomous weapon systems (LAWS), yaitu senjata yang dapat mengambil keputusan menyerang tanpa campur tangan manusia. Di satu sisi, sistem ini dapat meningkatkan

efektivitas militer dan mengurangi korban di pihak sendiri. Di sisi lain, risiko kesalahan deteksi dan implikasi moralnya masih menjadi diskusi panjang di forum internasional.

### 12.10.4 Simulasi Pelatihan dan Analisis Pasca Operasi

Computer vision juga mendukung pelatihan militer melalui simulasi berbasis realitas virtual (VR) atau realitas tertambah (AR). Dengan teknologi ini, prajurit dapat berlatih dalam lingkungan simulasi yang realistis, mencakup pergerakan lawan, medan bervariasi, hingga penggunaan peralatan. Vision berperan dalam menghasilkan interaksi yang natural, seperti pelacakan gerakan tubuh atau ekspresi wajah prajurit selama latihan.

Selain pelatihan, analisis pasca-operasi juga terbantu oleh computer vision. Rekaman video dari drone atau kamera helm prajurit dapat diproses untuk menilai efektivitas strategi, mengidentifikasi kesalahan taktis, dan menyusun rekomendasi perbaikan. Dengan demikian, setiap operasi menjadi sumber pembelajaran yang lebih sistematis.

### 12.10.5 Tantangan dan Etika Penggunaan

Walaupun manfaatnya signifikan, penggunaan computer vision dalam militer menimbulkan tantangan besar. Dari sisi teknis, model harus tangguh terhadap kondisi ekstrem seperti debu, hujan, asap, atau sinyal komunikasi yang terganggu. Kesalahan kecil dalam identifikasi target bisa berakibat fatal, baik secara militer maupun politik.

Dari sisi etika, perdebatan tentang otonomi sistem persenjataan masih berlangsung. Banyak pihak menekankan pentingnya human-in-the-loop, yaitu memastikan manusia tetap menjadi pengambil keputusan terakhir dalam penggunaan kekuatan mematikan. Selain itu, risiko penyalahgunaan teknologi untuk pengawasan massal atau represi politik juga menjadi perhatian serius.

## 12.11 Computer Vision di Bidang Seni, Budaya, dan Kreativitas Digital

Seni dan budaya selalu menjadi ruang eksplorasi bagi teknologi baru. Jika pada masa lalu teknologi hanya berfungsi sebagai alat bantu produksi, kini dengan hadirnya computer vision, proses kreatif mengalami transformasi yang jauh lebih mendalam. Vision bukan hanya membantu seniman menciptakan karya baru, tetapi juga memungkinkan digitalisasi, preservasi, hingga interaksi budaya dalam bentuk yang lebih interaktif dan imersif. Dengan kata lain, computer vision memperluas batas kreativitas sekaligus menjaga warisan budaya agar tetap relevan di era digital.

### 12.11.1 Digitalisasi dan Preservasi Warisan Budaya

Museum dan lembaga kebudayaan di seluruh dunia menghadapi tantangan besar dalam melestarikan artefak yang rentan terhadap kerusakan. Computer vision digunakan untuk mendukung proses digitalisasi artefak, lukisan, maupun manuskrip kuno dengan detail tinggi. Teknik image enhancement membantu mengungkap tulisan atau ukiran yang sudah memudar,

sementara 3D reconstruction memungkinkan artefak dipelajari dalam bentuk digital tiga dimensi tanpa risiko kerusakan fisik.

Sebagai contoh, naskah kuno yang rapuh dapat dipindai dengan kamera multispektral, kemudian algoritma vision mengekstrak teks yang tidak terlihat dengan mata telanjang. Teknologi ini membuka akses baru bagi peneliti maupun masyarakat luas, sekaligus memperpanjang umur pengetahuan yang terkandung dalam warisan budaya.

### 12.11.2 Karya Seni Generatif dan Interaktif

Computer vision juga menjadi bagian dari seni generatif, di mana algoritma digunakan untuk menciptakan pola, gambar, atau animasi yang bersifat otonom. Dengan mengombinasikan data visual nyata dan model vision, seniman dapat menghasilkan karya yang responsif terhadap lingkungan atau interaksi penonton.

Sebagai contoh, instalasi seni interaktif menggunakan kamera untuk mendeteksi gerakan tubuh pengunjung, lalu mengubah proyeksi visual sesuai dengan interaksi tersebut. Hasilnya adalah pengalaman seni yang unik bagi setiap individu, karena karya terus berubah mengikuti interaksi penonton. Pendekatan ini tidak hanya memperkaya medium seni, tetapi juga mengaburkan batas antara pencipta dan penikmat karya.

### 12.11.3 Restorasi Digital Karya Seni

Selain menciptakan karya baru, computer vision juga berperan penting dalam restorasi karya seni. Lukisan yang rusak akibat waktu, kelembaban, atau bencana dapat diperbaiki secara digital dengan bantuan algoritma inpainting dan style transfer. Teknologi ini tidak menggantikan restorasi fisik, tetapi memberikan gambaran hipotetis mengenai kondisi asli karya.

Dengan pendekatan ini, seniman dan kurator memiliki panduan yang lebih akurat untuk mengambil keputusan dalam proses konservasi. Lebih jauh lagi, versi digital karya yang direstorasi dapat diakses publik, sehingga pengetahuan dan apresiasi budaya tidak terhalang oleh kondisi fisik artefak.

### 12.11.4 Kreativitas di Era Media Sosial

Dalam ranah budaya populer, computer vision menjadi tulang punggung berbagai aplikasi media sosial. Filter wajah, augmented reality stickers, hingga efek visual dinamis pada platform seperti Instagram atau TikTok semuanya berbasis vision. Teknologi facial landmark detection

memungkinkan efek ditempelkan dengan akurat pada wajah pengguna, menciptakan pengalaman interaktif yang menyenangkan dan personal.

Fenomena ini menunjukkan bagaimana computer vision telah meresap ke dalam praktik budaya sehari-hari. Seni tidak lagi terbatas pada galeri atau museum, melainkan hadir dalam bentuk ekspresi digital yang diproduksi massal dan dikonsumsi secara global.

### 12.11.5 Tantangan Etika dan Otentisitas

Meskipun penuh peluang, penggunaan computer vision di bidang seni dan budaya juga menimbulkan pertanyaan kritis. Pertama, isu otentisitas karya digital menjadi semakin kompleks. Dengan kemampuan menghasilkan karya melalui algoritma, batas antara karya "asli" dan "buatan mesin" menjadi kabur. Hal ini menantang konsep tradisional tentang kepemilikan dan nilai seni.

Kedua, muncul risiko penyalahgunaan teknologi, seperti pembuatan deepfake yang merusak reputasi individu atau manipulasi gambar sejarah untuk tujuan tertentu. Oleh karena itu, pemanfaatan computer vision dalam seni dan budaya harus dibarengi dengan kerangka etika yang jelas, yang menekankan transparansi, kejujuran artistik, serta penghargaan terhadap hak cipta.

## 12.12 Computer Vision di Bidang Olahraga dan Kesehatan Kebugaran

Olahraga dan kebugaran merupakan bidang yang sangat dipengaruhi oleh data visual, mulai dari analisis gerakan atlet hingga pemantauan kebugaran individu sehari-hari. Computer vision menghadirkan cara baru untuk memahami performa tubuh manusia melalui analisis citra dan video, yang sebelumnya hanya bisa dilakukan dengan peralatan laboratorium mahal. Dengan kamera sederhana, algoritma vision mampu menilai postur, menghitung kecepatan, bahkan memprediksi risiko cedera. Teknologi ini membuka peluang luas, baik untuk olahraga profesional maupun kebugaran masyarakat umum.

### 12.12.1 Analisis Performa Atlet

Dalam olahraga profesional, setiap detail gerakan atlet dapat menentukan hasil pertandingan. Computer vision digunakan untuk menganalisis teknik gerakan, kecepatan, sudut tubuh, dan pola koordinasi. Misalnya, dalam sepak bola, sistem vision membantu pelatih mengevaluasi pergerakan pemain di lapangan, termasuk kecepatan sprint, posisi strategis, dan pola passing.

Di cabang olahraga individu seperti atletik atau renang, kamera berkecepatan tinggi dikombinasikan dengan algoritma pose estimation untuk menilai efisiensi gerakan. Data ini menjadi dasar bagi pelatih untuk memberikan koreksi teknik yang lebih presisi dibandingkan observasi manual. Dengan demikian, computer vision berperan sebagai "asisten pelatih digital" yang mendukung peningkatan performa atlet.

### 12.12.2 Deteksi Cedera dan Pencegahan Risiko

Selain meningkatkan performa, computer vision juga digunakan untuk mencegah cedera. Sistem dapat menganalisis postur tubuh atlet selama latihan untuk mendeteksi pola gerakan

yang berisiko, seperti pendaratan lutut yang tidak stabil pada pemain basket atau teknik angkat beban yang salah di gym.

Dengan menggunakan algoritma analisis biomekanika berbasis citra, potensi cedera dapat diidentifikasi lebih awal. Misalnya, pose estimation 3D mampu menghitung sudut persendian dan mendeteksi asimetri gerakan yang menjadi indikator ketidakseimbangan otot. Pendekatan ini membantu atlet melakukan koreksi sebelum masalah berkembang menjadi cedera serius.

### 12.12.3 Aplikasi Kebugaran untuk Masyarakat Umum

Tidak hanya di level profesional, computer vision juga hadir dalam aplikasi kebugaran sehari-hari. Banyak aplikasi mobile kini memanfaatkan kamera ponsel untuk memberikan umpan balik pada latihan, seperti yoga, pilates, atau latihan kekuatan. Algoritma skeleton tracking dapat menilai apakah posisi tubuh sudah benar, lalu memberikan koreksi secara langsung melalui layar ponsel.

Selain itu, sistem vision juga digunakan untuk menghitung repetisi latihan secara otomatis, misalnya jumlah squat, push-up, atau jumping jack. Hal ini membuat latihan mandiri lebih terstruktur dan memotivasi pengguna untuk mencapai target kebugaran.

### 12.12.4 Analisis Pertandingan dan Hiburan

Dalam industri olahraga, computer vision tidak hanya membantu atlet, tetapi juga memperkaya pengalaman penonton. Teknologi player tracking digunakan dalam siaran langsung untuk menampilkan data kecepatan lari, jarak tempuh, atau heatmap pergerakan pemain. Data visual ini meningkatkan interaktivitas penonton sekaligus membuka peluang analisis mendalam bagi penggemar dan analis pertandingan.

Di sisi lain, sistem deteksi bola berbasis vision membantu wasit dalam mengambil keputusan penting, seperti menentukan apakah bola sudah melewati garis gawang atau keluar lapangan. Dengan demikian, computer vision juga berperan dalam meningkatkan keadilan dan transparansi dalam pertandingan.

### 12.12.5 Tantangan Implementasi

Meskipun potensinya besar, implementasi computer vision dalam olahraga dan kebugaran menghadapi tantangan teknis. Variasi pencahayaan, kecepatan gerakan, serta kondisi lapangan yang dinamis dapat memengaruhi akurasi sistem. Di tingkat aplikasi konsumen, keterbatasan kamera ponsel juga membatasi detail yang dapat ditangkap.

Selain itu, isu privasi tetap menjadi perhatian. Data visual tubuh individu termasuk informasi sensitif, sehingga harus dikelola dengan hati-hati agar tidak disalahgunakan. Oleh karena itu, integrasi computer vision dalam olahraga dan kebugaran harus selalu disertai kebijakan perlindungan data yang ketat.

# BAB XIII

## Arah Masa Depan Computer Vision: Tren, Integrasi, dan Tantangan

Perjalanan panjang computer vision telah membawa teknologi ini dari sekadar algoritma deteksi tepi sederhana hingga menjadi tulang punggung berbagai aplikasi cerdas yang kita temui dalam kehidupan sehari-hari. Dalam bab-bab sebelumnya, kita telah melihat bagaimana vision diterapkan dalam pertanian, kesehatan, transportasi, keamanan, hingga seni dan budaya. Aplikasi tersebut menggambarkan betapa luasnya spektrum dampak teknologi ini. Namun, jika perjalanan ini kita anggap sebagai lintasan evolusi, maka apa yang sudah dicapai saat ini baru merupakan fondasi awal. Bab ini akan memandu kita menengok ke masa depan computer vision, sekaligus merefleksikan tren, integrasi teknologi, tantangan etika, serta arah penelitian yang akan membentuk wajah teknologi visual di dekade mendatang.

Salah satu hal mendasar yang membedakan era computer vision mendatang adalah skala dan konteks penerapannya. Jika sebelumnya vision banyak berfokus pada tugas spesifik—misalnya deteksi objek di citra medis atau klasifikasi penyakit daun—maka masa depan akan ditandai oleh kebutuhan sistem vision yang lebih adaptif, multimodal, efisien, dan bertanggung jawab secara sosial. Hal ini dipengaruhi oleh tiga faktor besar:

Pertama, ketersediaan perangkat keras yang semakin miniatur namun kuat, seperti prosesor edge dan kamera cerdas yang dapat bekerja tanpa ketergantungan penuh pada cloud. Kedua, kemajuan model AI multimodal, di mana pengolahan visual tidak lagi berdiri sendiri, melainkan terhubung dengan bahasa, suara, maupun pengetahuan simbolis. Ketiga, dorongan etika dan regulasi, karena masyarakat kini semakin kritis terhadap implikasi teknologi AI, khususnya dalam isu privasi, bias, dan transparansi algoritma.

Dengan kerangka ini, Bab XIII akan membahas beberapa tema utama. Pertama, pergeseran menuju Edge AI dan strategi hemat energi dalam menjalankan model vision. Kedua, lahirnya Vision-Language Models (VLM) dan sistem multimodal yang membuka kemungkinan pemahaman visual lebih kaya. Ketiga, kebutuhan akan Explainable AI (XAI) agar model tidak sekadar akurat, tetapi juga dapat dipahami dan diaudit. Keempat, tantangan privasi dan solusi federated learning dalam konteks data visual. Kelima, integrasi vision dengan teknologi frontier lain seperti IoT, blockchain, AR/VR, dan big data. Keenam, refleksi mengenai etika, tata kelola, dan regulasi yang mendampingi perkembangan ini. Dan terakhir, arah penelitian ke depan: bagaimana computer vision dapat berkembang hingga 2030 dengan tetap selaras dengan kebutuhan manusia dan keberlanjutan planet.

Pendekatan yang digunakan dalam bab ini bersifat analitis sekaligus spekulatif. Artinya, kita akan mengulas hasil riset mutakhir yang telah dipublikasikan, sekaligus membaca kecenderungan masa depan berdasarkan garis besar perkembangan teknologi. Dengan demikian, bab ini tidak hanya relevan bagi peneliti, tetapi juga bagi praktisi, pembuat kebijakan, dan mahasiswa yang ingin memahami lanskap computer vision masa depan secara komprehensif.

## 13.1 Edge AI dan Komputasi Hemat Energi

Salah satu tren besar dalam perkembangan computer vision masa depan adalah pergeseran dari pemrosesan berbasis pusat data (cloud computing) menuju pemrosesan langsung di perangkat lokal atau tepi jaringan, yang dikenal dengan istilah edge AI. Pergeseran ini bukan sekadar perubahan teknis, tetapi transformasi paradigma dalam cara sistem visual dirancang, dijalankan, dan diintegrasikan ke dalam kehidupan sehari-hari.

Selama bertahun-tahun, sebagian besar aplikasi computer vision bergantung pada infrastruktur cloud. Model deep learning yang kompleks membutuhkan daya komputasi besar, sehingga citra atau video dikirim ke server pusat untuk dianalisis sebelum hasilnya dikirim kembali ke perangkat pengguna. Meskipun pendekatan ini efektif untuk penelitian dan prototipe, dalam praktiknya ia menghadapi keterbatasan serius: latensi yang tinggi, ketergantungan pada koneksi internet stabil, risiko privasi karena data visual dikirim ke pihak ketiga, serta konsumsi energi yang signifikan di pusat data.

Edge AI hadir sebagai solusi yang mencoba mengatasi masalah-masalah tersebut. Alih-alih mengirim data ke cloud, analisis dilakukan langsung di perangkat—baik itu kamera cerdas, ponsel, drone, maupun sensor IoT. Pendekatan ini menawarkan beberapa keunggulan utama. Pertama, latensi rendah, karena hasil analisis dapat diperoleh hampir seketika tanpa harus menunggu pengiriman data ke server pusat. Hal ini krusial dalam aplikasi yang menuntut respons cepat, seperti kendaraan otonom, robot industri, atau sistem keamanan real-time. Kedua, privasi lebih terjaga, sebab data visual sensitif tidak perlu meninggalkan perangkat. Ketiga, efisiensi energi dan biaya, karena mengurangi kebutuhan transmisi data berukuran besar yang biasanya membebani jaringan dan pusat data.

Namun, untuk mewujudkan visi ini, dibutuhkan strategi khusus dalam merancang model computer vision yang ringan namun tetap akurat. Model-model besar seperti ResNet-152 atau Vision Transformer skala penuh sulit dijalankan di perangkat dengan keterbatasan memori dan daya. Oleh karena itu, riset terkini banyak difokuskan pada pengembangan arsitektur efisien seperti MobileNet, ShuffleNet, EfficientNet-Lite, hingga varian YOLO-Nano dan YOLOv8-n. Model-model ini mengorbankan sebagian kecil akurasi untuk mendapatkan kinerja real-time dengan konsumsi daya rendah.

Selain arsitektur efisien, teknik optimisasi model juga memegang peranan penting. Quantization misalnya, mengubah representasi bobot model dari presisi 32-bit menjadi 8-bit atau bahkan lebih rendah, tanpa kehilangan akurasi signifikan. Pruning digunakan untuk memangkas neuron atau saluran konvolusi yang tidak terlalu berkontribusi, sehingga memperkecil ukuran model. Teknik lain adalah knowledge distillation, di mana model besar (teacher model) mentransfer pengetahuan ke model kecil (student model), menghasilkan model ringkas yang tetap kompetitif.

Implementasi edge AI tidak hanya soal perangkat keras, tetapi juga arsitektur sistem. Dalam banyak kasus, sistem hybrid diperlukan, di mana sebagian analisis dilakukan di perangkat tepi, sementara tugas yang lebih kompleks tetap dikirim ke cloud. Pendekatan ini dikenal sebagai split computing atau edge-cloud collaboration. Misalnya, kamera pengawas dapat mendeteksi objek dasar seperti manusia atau kendaraan di perangkat lokal, tetapi untuk analisis forensik

yang lebih detail, data tetap dikirim ke server pusat. Dengan cara ini, keseimbangan antara efisiensi dan ketelitian tetap terjaga.

Kaitannya dengan keberlanjutan juga tidak bisa diabaikan. Pusat data global diperkirakan menyumbang konsumsi energi setara dengan negara-negara industri menengah. Dengan semakin banyak aplikasi computer vision, kebutuhan energi bisa meningkat drastis. Edge AI menawarkan jalur menuju green AI, yaitu pengembangan kecerdasan buatan yang ramah lingkungan. Dengan menurunkan ketergantungan pada pusat data dan mengoptimalkan konsumsi energi di perangkat lokal, sistem vision masa depan dapat lebih berkelanjutan.

Selain itu, edge AI memiliki relevansi besar untuk negara berkembang, termasuk Indonesia. Keterbatasan infrastruktur internet sering menjadi hambatan dalam implementasi sistem vision berbasis cloud. Dengan edge AI, aplikasi seperti deteksi penyakit tanaman, monitoring lalu lintas, atau sistem kesehatan berbasis kamera dapat berjalan secara lokal tanpa memerlukan koneksi konstan ke server. Hal ini membuka peluang pemerataan akses teknologi AI, bahkan di daerah terpencil.

Meski begitu, tantangan tetap ada. Pertama, keterbatasan perangkat keras murah berarti kompromi antara akurasi dan kecepatan. Kedua, pemeliharaan model di edge lebih sulit, karena update harus dilakukan ke banyak perangkat sekaligus, bukan satu server pusat. Inilah mengapa riset tentang federated learning (yang akan dibahas di 13.4) menjadi penting, karena memungkinkan model diperbarui secara kolektif tanpa perlu mengumpulkan semua data ke pusat.

Dengan demikian, edge AI dan komputasi hemat energi bukan sekadar tren teknis, tetapi fondasi bagi masa depan computer vision yang lebih cepat, aman, inklusif, dan berkelanjutan. Ia memungkinkan visi menjadi bagian dari kehidupan sehari-hari, tidak hanya di laboratorium atau perusahaan besar, melainkan juga di tangan petani, guru, dokter, dan masyarakat luas.

## 13.2 Vision-Language Models (VLM) dan Multimodal AI

Salah satu lonjakan besar dalam perkembangan artificial intelligence saat ini adalah lahirnya sistem multimodal, yaitu model yang mampu memahami dan mengolah lebih dari satu jenis data sekaligus—misalnya teks, gambar, suara, dan bahkan sensor lain. Dalam konteks computer vision, integrasi ini melahirkan apa yang disebut sebagai Vision-Language Models (VLM), yaitu model yang mampu menghubungkan informasi visual dengan bahasa alami.

Perkembangan ini didorong oleh kebutuhan mendasar: dunia nyata tidak pernah hadir dalam satu modalitas tunggal. Manusia, misalnya, memahami sebuah adegan bukan hanya dengan melihat, tetapi juga dengan menamai objek, menafsirkan konteks, serta mengaitkannya dengan pengetahuan linguistik. Jika computer vision klasik hanya fokus pada “melihat”, maka VLM memungkinkan mesin untuk “melihat dan berbicara”—sebuah lompatan yang membuka pintu ke aplikasi jauh lebih kaya.

### 13.2.1 Evolusi dari Vision Klasik ke Multimodal

Pada tahap awal, sistem vision bekerja secara terpisah dari pemrosesan bahasa. Model deteksi objek hanya menghasilkan label seperti “kucing” atau “mobil” tanpa kemampuan menjelaskan

konteksnya. Namun, dengan munculnya model multimodal, komputer tidak hanya memberi label, tetapi juga dapat menghasilkan deskripsi naratif yang menyerupai penjelasan manusia.

Salah satu tonggak penting adalah pengembangan CLIP (Contrastive Language-Image Pretraining) oleh OpenAI, yang melatih model untuk memahami hubungan antara teks dan gambar dalam skala miliaran pasangan data. CLIP mampu melakukan tugas zero-shot learning, misalnya mengenali objek baru hanya berdasarkan deskripsi teks tanpa harus dilatih secara khusus pada dataset tersebut. Pendekatan ini merevolusi cara sistem vision dipelajari, karena tidak lagi terbatas pada dataset teranotasi yang sempit.

### 13.2.2 Kemampuan Inti VLM

Ada tiga kemampuan utama yang menjadikan VLM berbeda dengan vision tradisional. Pertama, image captioning, yaitu menghasilkan deskripsi bahasa alami dari sebuah gambar. Misalnya, sebuah foto dapat dijelaskan dengan kalimat “Seorang anak sedang berlari di pantai sambil membawa layang-layang.” Kedua, visual question answering (VQA), di mana sistem dapat menjawab pertanyaan berbasis citra, seperti “Berapa jumlah kursi di ruangan ini?” atau “Apakah orang dalam foto sedang tersenyum?”. Ketiga, cross-modal retrieval, yaitu pencarian silang antar modalitas, misalnya mencari gambar berdasarkan deskripsi teks atau sebaliknya.

Kemampuan ini memperluas peran vision dari sekadar pengenalan pola menjadi alat komunikasi. VLM tidak hanya melihat objek, tetapi juga memahami makna di balik visual tersebut dalam konteks bahasa.

### 13.2.3 Aplikasi Nyata VLM dan Multimodal AI

Integrasi vision dan bahasa memiliki implikasi luas di berbagai sektor. Dalam dunia medis, misalnya, model multimodal dapat membaca citra radiologi dan menghasilkan laporan awal yang menyerupai catatan dokter, sehingga menghemat waktu tenaga medis. Di bidang pendidikan, sistem dapat menganalisis video praktikum dan memberikan penjelasan naratif bagi siswa.

Dalam e-commerce, VLM memungkinkan pencarian produk berbasis deskripsi bahasa alami, seperti “sepatu lari warna biru dengan sol putih”, yang kemudian dipadankan secara otomatis dengan gambar produk yang relevan. Sementara di bidang aksesibilitas, VLM membantu penyandang tunanetra dengan menghasilkan deskripsi audio dari lingkungan sekitar berdasarkan kamera ponsel.

Lebih jauh lagi, VLM menjadi fondasi bagi chatbot multimodal seperti GPT-4V atau Gemini, yang memungkinkan interaksi manusia-mesin semakin alami. Pengguna dapat mengunggah gambar, lalu menanyakan informasi tentang gambar tersebut, atau meminta sistem melakukan analisis mendalam yang melibatkan teks dan visual sekaligus.

### 13.2.4 Tantangan Teknis dan Etika

Meski potensinya besar, pengembangan VLM menghadapi tantangan serius. Pertama, kompleksitas data multimodal sangat tinggi. Model harus mampu menyelaraskan representasi visual dan linguistik yang secara alamiah sangat berbeda. Ketidakseimbangan kualitas data—

misalnya gambar berkualitas rendah dengan deskripsi ambigu—dapat menurunkan performa sistem.

Kedua, isu bias dan representasi. Karena model multimodal dilatih pada data dalam jumlah besar yang diambil dari internet, ia berpotensi menyerap bias sosial, budaya, atau gender yang ada pada teks dan gambar tersebut. Hal ini dapat menyebabkan sistem memberikan deskripsi diskriminatif atau jawaban yang menyesatkan.

Ketiga, tantangan interpretabilitas. Semakin kompleks sebuah VLM, semakin sulit menjelaskan mengapa sistem menghasilkan deskripsi atau jawaban tertentu. Padahal, dalam konteks medis atau hukum, keterjelasan keputusan algoritma sangat penting.

### 13.2.5 Masa Depan Multimodal AI

Masa depan computer vision hampir pasti bergerak ke arah multimodal. Penelitian kini difokuskan pada model yang tidak hanya menghubungkan gambar dan teks, tetapi juga suara, sensor 3D, hingga data biologis. Konsep foundation model yang serbaguna akan menjadi standar baru, di mana satu model besar dapat digunakan untuk berbagai tugas lintas modalitas dengan sedikit penyesuaian.

Selain itu, arah penelitian juga mulai menekankan pada efisiensi energi dan inklusivitas. Bagaimana membangun VLM yang cukup ringan untuk berjalan di perangkat tepi? Bagaimana memastikan representasi budaya yang adil dalam model multimodal global? Pertanyaan-pertanyaan ini akan membentuk arah riset dalam dekade mendatang.

Dengan kata lain, Vision-Language Models bukan hanya fase sementara, melainkan sebuah transformasi mendasar dalam cara kita membayangkan computer vision. Jika sebelumnya mesin “melihat”, maka masa depan adalah mesin yang “melihat, memahami, dan berbicara”—membuka ruang kolaborasi baru antara manusia dan kecerdasan buatan.

## 13.3 Explainable AI (XAI) untuk Computer Vision

Salah satu tantangan terbesar dalam perkembangan computer vision modern adalah menjembatani jurang antara akurasi teknis dan kepercayaan manusia. Model deep learning seperti Convolutional Neural Networks (CNN) atau Vision Transformers (ViT) mampu mencapai performa luar biasa pada berbagai tugas—dari klasifikasi citra medis hingga deteksi objek real-time. Namun, di balik keunggulan tersebut, muncul satu pertanyaan mendasar: mengapa model mengambil keputusan tertentu?

Pertanyaan ini bukan sekadar akademis, melainkan juga etis dan praktis. Dalam konteks medis, misalnya, seorang dokter tidak cukup hanya menerima hasil “gambaran X-ray ini positif pneumonia”. Ia membutuhkan penjelasan: area paru mana yang menunjukkan indikasi penyakit, apa dasar visual yang digunakan model, dan seberapa besar tingkat keyakinannya. Tanpa transparansi, kepercayaan pengguna terhadap AI akan rapuh, bahkan berpotensi

menimbulkan resistensi atau kesalahan fatal. Inilah konteks kelahiran bidang Explainable AI (XAI).

### 13.3.1 Mengapa XAI Diperlukan?

Ada beberapa alasan mendasar mengapa keterjelasan (explainability) sangat penting. Pertama, keamanan dan keselamatan. Dalam aplikasi kritis seperti kendaraan otonom, sistem vision yang tidak dapat dijelaskan bisa berbahaya. Kesalahan deteksi lampu lalu lintas atau pejalan kaki dapat berakibat fatal jika tidak bisa ditelusuri penyebabnya.

Kedua, kepatuhan regulasi. Sejumlah yurisdiksi, termasuk Uni Eropa melalui General Data Protection Regulation (GDPR), menekankan hak individu untuk mendapatkan penjelasan dari keputusan otomatis yang memengaruhi mereka. Hal ini berarti sistem vision yang digunakan untuk identifikasi wajah atau verifikasi identitas harus dapat dipertanggungjawabkan.

Ketiga, etika dan keadilan sosial. Model vision yang "black box" berisiko memperkuat bias, misalnya diskriminasi rasial dalam sistem pengenalan wajah. Dengan XAI, bias tersebut dapat diidentifikasi dan diintervensi sebelum berdampak luas.

### 13.3.2 Pendekatan XAI dalam Computer Vision

Terdapat berbagai pendekatan yang digunakan untuk membuat model vision lebih dapat dijelaskan. Secara umum, metode XAI dapat dibagi menjadi dua kategori besar: post-hoc explanation dan intrinsically interpretable models.

a) **Post-hoc explanation:** teknik ini diterapkan setelah model selesai dilatih. Contoh paling populer adalah saliency maps atau heatmaps yang menyoroti bagian citra yang paling berkontribusi terhadap prediksi model. Metode seperti Grad-CAM (Gradient-weighted Class Activation Mapping) mampu menunjukkan area spesifik pada gambar yang menjadi dasar keputusan.

b) **Intrinsically interpretable models:** pendekatan ini berusaha membuat model itu sendiri transparan sejak awal. Contohnya adalah model berbasis aturan (rule-based), prototype learning, atau model hibrida yang menggabungkan deep learning dengan pengetahuan simbolis. Meski lebih mudah dijelaskan, model ini biasanya kurang kompetitif dalam akurasi dibandingkan deep learning murni.

Selain itu, berkembang pula teknik counterfactual explanation, yang menjawab pertanyaan "apa yang harus diubah agar prediksi berbeda?" Misalnya, jika sebuah sistem vision menilai wajah seseorang tidak lolos verifikasi, counterfactual explanation dapat menunjukkan fitur wajah mana yang paling berpengaruh terhadap keputusan tersebut.

### 13.3.3 Studi Kasus XAI dalam Vision

Dalam bidang medis, penggunaan Grad-CAM pada CNN untuk analisis citra radiologi telah membantu dokter memvalidasi apakah sistem benar-benar fokus pada area relevan, misalnya nodul kecil pada paru-paru. Di bidang pertanian, visualisasi saliency map memungkinkan

peneliti melihat apakah model deteksi penyakit tanaman memang memperhatikan bercak daun, atau justru terganggu oleh latar belakang tanah.

Di sektor keamanan, XAI digunakan untuk memvalidasi sistem pengenalan wajah, memastikan bahwa keputusan model tidak didasarkan pada faktor yang salah, seperti pencahayaan atau latar belakang, melainkan benar-benar pada fitur wajah individu.

### 13.3.4 Tantangan dan Batasan XAI

Meskipun menjanjikan, XAI menghadapi sejumlah keterbatasan. Pertama, kompleksitas model. Deep learning dengan miliaran parameter sulit direduksi menjadi penjelasan sederhana tanpa kehilangan detail penting. Penjelasan visual seperti saliency map sering kali terlalu abstrak bagi pengguna awam.

Kedua, trade-off antara akurasi dan interpretabilitas. Model yang mudah dijelaskan cenderung kurang akurat, sedangkan model yang akurat sering kali sulit dijelaskan. Tantangan ini mengharuskan peneliti mencari titik keseimbangan.

Ketiga, risiko over-interpretation. Penjelasan yang dihasilkan XAI terkadang hanya representasi perkiraan, bukan alasan sejati dari model. Jika pengguna tidak memahami keterbatasan ini, penjelasan bisa menimbulkan rasa percaya palsu.

### 13.3.5 Masa Depan XAI untuk Vision

Ke depan, XAI diperkirakan akan menjadi komponen standar dalam setiap sistem computer vision, terutama di sektor yang berhubungan langsung dengan manusia. Penelitian mulai bergerak ke arah interactive explainability, di mana pengguna dapat berinteraksi dengan sistem untuk meminta penjelasan sesuai kebutuhan. Selain itu, integrasi XAI dengan regulasi AI global akan memperkuat standar transparansi dan akuntabilitas.

Lebih jauh, muncul pula gagasan tentang “causal XAI”, yaitu penjelasan berbasis hubungan sebab-akibat, bukan sekadar korelasi. Pendekatan ini diyakini lebih dekat dengan cara manusia memahami dunia visual.

Pada akhirnya, XAI bukan hanya soal menjelaskan model, tetapi juga membangun jembatan kepercayaan antara manusia dan mesin. Dengan keterjelasan, sistem vision dapat diterima lebih luas, digunakan dengan lebih aman, dan dipandang bukan sebagai kotak hitam misterius, melainkan sebagai alat yang dapat diaudit, dikritisi, dan disempurnakan.

## 13.4 Federated Learning dan Privasi Data Visual

Salah satu isu paling sensitif dalam perkembangan computer vision adalah privasi data visual. Berbeda dengan data numerik atau teks, citra dan video sering kali memuat informasi pribadi yang sangat mudah mengidentifikasi individu atau lingkungan tertentu. Foto wajah, rekaman CCTV, atau citra medis bukan sekadar data teknis, tetapi juga representasi identitas yang melekat pada pemiliknya. Dalam konteks inilah Federated Learning (FL) muncul sebagai

pendekatan baru yang berusaha menyeimbangkan kebutuhan pelatihan model AI dengan perlindungan privasi pengguna.

### 13.4.1 Keterbatasan Paradigma Terpusat

Tradisionalnya, model vision dilatih dengan cara mengumpulkan data dari berbagai sumber, lalu menyimpannya di satu server pusat. Strategi ini menimbulkan dua masalah utama. Pertama, risiko kebocoran privasi: data yang dipindahkan ke pusat penyimpanan rentan terhadap serangan atau penyalahgunaan. Kedua, kendala hukum dan etika: regulasi seperti GDPR di Eropa atau HIPAA di Amerika Serikat secara tegas membatasi distribusi data medis atau biometrik tanpa izin eksplisit.

Di banyak negara berkembang, termasuk Indonesia, masalah serupa muncul dalam skala berbeda: transfer data dalam jumlah besar sering terhambat oleh keterbatasan infrastruktur jaringan. Artinya, pendekatan terpusat bukan hanya berisiko secara etis, tetapi juga tidak praktis secara teknis.

### 13.4.2 Prinsip Federated Learning

Federated Learning menawarkan paradigma terbalik. Alih-alih memindahkan data ke pusat, FL memungkinkan model dikirim ke perangkat tepi (edge devices) untuk dilatih secara lokal. Setelah pelatihan selesai, hanya parameter model (misalnya bobot neural network) yang dikirim kembali ke server pusat, lalu digabungkan dengan parameter dari perangkat lain. Dengan demikian, data mentah tidak pernah meninggalkan perangkat pengguna.

Ilustrasi sederhananya dapat dilihat pada aplikasi ponsel. Misalnya, aplikasi keyboard prediktif yang menggunakan FL tidak perlu mengirim seluruh isi pesan pengguna ke server. Sebaliknya, aplikasi melatih model lokal di ponsel berdasarkan pola pengetikan, lalu hanya mengirim pembaruan parameter model. Pendekatan serupa dapat diterapkan dalam computer vision, baik untuk citra medis, data kendaraan, maupun rekaman CCTV.

### 13.4.3 Keunggulan FL dalam Vision

Ada beberapa keuntungan utama FL untuk computer vision. Pertama, privasi lebih terlindungi, sebab data visual tidak pernah meninggalkan perangkat asal. Kedua, efisiensi bandwidth, karena yang ditransfer hanya parameter model yang jauh lebih kecil dibandingkan citra atau video mentah. Ketiga, adaptasi kontekstual, sebab model dilatih pada distribusi data nyata dari masing-masing perangkat atau lokasi. Misalnya, sistem deteksi penyakit tanaman dapat belajar langsung dari foto yang diambil petani di lahan mereka, sehingga model lebih relevan terhadap kondisi lokal.

### 13.4.4 Tantangan Teknis FL

Meskipun menjanjikan, implementasi FL pada vision menghadapi sejumlah kendala. Salah satunya adalah heterogenitas data. Citra dari perangkat berbeda sering kali memiliki kualitas,

resolusi, atau sudut pengambilan yang sangat bervariasi. Hal ini menyebabkan non-iid problem (non-independent and identically distributed), yang membuat pelatihan model lebih sulit.

Tantangan kedua adalah keterbatasan sumber daya perangkat tepi. Tidak semua smartphone atau kamera cerdas memiliki kapasitas komputasi memadai untuk melatih model deep learning, sehingga diperlukan arsitektur model yang efisien atau strategi split learning, di mana sebagian pelatihan tetap dilakukan di cloud.

Ketiga, keamanan parameter model. Meskipun data mentah tidak ditransfer, parameter yang dikirim masih berpotensi diserang dengan teknik model inversion attack, yaitu upaya merekonstruksi data asli dari bobot model. Untuk mengatasi hal ini, FL sering dikombinasikan dengan differential privacy atau secure aggregation untuk menambahkan lapisan perlindungan.

### 13.4.5 Studi Kasus FL dalam Vision

Beberapa studi mutakhir telah menunjukkan potensi besar FL dalam computer vision. Dalam bidang medis, FL digunakan untuk melatih model deteksi tumor otak dari MRI dengan data yang tersebar di berbagai rumah sakit, tanpa harus melanggar privasi pasien. Di bidang transportasi, FL memungkinkan kendaraan otonom belajar dari pengalaman kolektif banyak mobil, sehingga setiap kendaraan menjadi lebih pintar tanpa harus berbagi rekaman jalanan mentah.

Di sektor pertanian, FL dapat digunakan untuk membangun model deteksi penyakit tanaman berdasarkan data yang dikumpulkan langsung dari petani di berbagai wilayah. Dengan cara ini, model menjadi representatif terhadap keragaman varietas dan kondisi geografis tanpa memaksa petani mengunggah seluruh data visual ke server pusat.

### 13.4.6 Masa Depan FL untuk Vision

Arah penelitian FL ke depan akan menekankan pada efisiensi komunikasi, robustness terhadap data non-iid, serta integrasi dengan teknologi edge AI. Salah satu tren menarik adalah federated multitask learning, di mana perangkat tidak hanya melatih model global, tetapi juga menyesuaikan model lokal sesuai kebutuhan spesifik.

Selain itu, FL juga akan semakin dipadukan dengan pendekatan green AI, dengan tujuan mengurangi konsumsi energi dalam proses pelatihan terdistribusi. Kombinasi ini akan sangat relevan ketika computer vision diterapkan pada miliaran perangkat IoT di seluruh dunia.

Dengan demikian, Federated Learning dapat dipandang sebagai jalan tengah yang strategis: ia menjaga privasi dan keamanan data visual, sekaligus memungkinkan akselerasi penelitian vision melalui kolaborasi skala besar. Jika cloud mewakili sentralisasi dan edge mewakili desentralisasi, maka FL adalah jembatan yang berusaha menggabungkan keduanya secara harmonis.

## 13.5 Integrasi Vision dengan Teknologi Lain (IoT, Blockchain, AR/VR, Big Data)

Perkembangan computer vision tidak berlangsung dalam ruang hampa. Justru, kekuatan sesungguhnya muncul ketika vision diintegrasikan dengan teknologi frontier lain yang berkembang paralel, seperti Internet of Things (IoT), blockchain, Augmented/Virtual Reality

(AR/VR), serta analitik big data. Integrasi ini memungkinkan lahirnya sistem yang lebih cerdas, terdistribusi, aman, sekaligus aplikatif dalam berbagai sektor kehidupan.

### 13.5.1 Vision + IoT: Menuju Sistem Cyber-Physical

IoT telah melahirkan miliaran perangkat sensor yang tersebar di dunia nyata—mulai dari kamera, sensor lingkungan, hingga perangkat rumah pintar. Integrasi dengan computer vision menjadikan IoT tidak hanya "merasakan" data numerik, tetapi juga "melihat" dunia sekitar.

Contoh nyata terlihat dalam pertanian presisi, di mana kamera yang tertanam pada drone atau sensor lapangan digunakan untuk memantau kesehatan tanaman secara visual, lalu mengirim data ke sistem IoT untuk dipadukan dengan informasi suhu, kelembaban, atau kondisi tanah. Hasil integrasi ini adalah pemetaan kondisi lahan yang lebih akurat, yang mendukung keputusan irigasi atau pemupukan presisi.

Dalam transportasi cerdas, kamera lalu lintas yang terhubung dengan IoT memungkinkan deteksi real-time kemacetan, pelanggaran lalu lintas, atau kecelakaan. Data visual ini, ketika digabungkan dengan sensor GPS dan sistem navigasi, dapat mengoptimalkan aliran lalu lintas kota.

Namun, integrasi vision dan IoT juga menimbulkan tantangan: beban komputasi yang tinggi, latensi jaringan, serta kebutuhan akan standar interoperabilitas. Oleh karena itu, pendekatan edge computing sering dipadukan agar sebagian analisis dilakukan langsung di perangkat IoT.

### 13.5.2 Vision + Blockchain: Transparansi dan Ketelusuran Data

Salah satu masalah besar dalam sistem vision adalah kepercayaan pada data. Bagaimana memastikan bahwa citra medis tidak dimanipulasi? Bagaimana menjamin rekaman CCTV benar-benar autentik? Di sinilah blockchain berperan.

Blockchain, dengan sifatnya yang terdistribusi dan tidak dapat diubah, dapat digunakan untuk mencatat metadata setiap citra atau video. Misalnya, setiap kali sebuah gambar diambil oleh kamera IoT, hash uniknya dicatat dalam blockchain, sehingga setiap perubahan di kemudian hari dapat dilacak. Hal ini penting dalam konteks forensik digital maupun supply chain monitoring, misalnya untuk memastikan produk pangan atau farmasi yang diawasi dengan vision benar-benar berasal dari sumber yang sah.

Lebih jauh, blockchain juga membuka jalan bagi data marketplace yang adil. Pemilik data visual, seperti petani atau rumah sakit, dapat berbagi data mereka ke peneliti atau perusahaan

dengan jaminan imbalan yang transparan, karena setiap transaksi tercatat dalam buku besar terdistribusi.

Tantangan integrasi ini terletak pada skalabilitas. Blockchain tradisional sering dianggap lambat untuk volume data besar. Solusinya adalah menggabungkan blockchain dengan penyimpanan off-chain atau layer-2 solutions untuk menjaga efisiensi.

### 13.5.3 Vision + AR/VR: Interaksi Manusia-Mesin yang Imersif

AR dan VR adalah medium baru yang mengandalkan persepsi visual untuk menciptakan pengalaman imersif. Integrasi computer vision dengan AR/VR memungkinkan interaksi manusia-mesin yang lebih alami dan adaptif.

Dalam pendidikan, misalnya, AR dapat memanfaatkan vision untuk mengenali objek nyata, lalu menambahkan lapisan informasi digital di atasnya. Seorang siswa biologi yang mengarahkan ponsel ke daun dapat melihat overlay digital tentang struktur sel dan proses fotosintesis.

Di industri, AR yang digabungkan dengan vision digunakan untuk maintenance assistance. Pekerja lapangan dapat mengenakan kacamata AR yang mengenali mesin secara visual, lalu menampilkan instruksi perbaikan secara real-time.

Sementara dalam hiburan, VR berbasis vision memungkinkan sistem melacak gerakan tubuh atau ekspresi wajah pemain untuk menghasilkan pengalaman lebih realistis. Tantangan utamanya adalah kebutuhan latensi sangat rendah (<20 ms) agar pengalaman imersif tidak menimbulkan ketidaknyamanan.

### 13.5.4 Vision + Big Data: Analitik Visual Skala Besar

Volume data visual yang dihasilkan dunia modern sangat masif. Diperkirakan setiap hari lebih dari 3 miliar gambar diunggah hanya di media sosial. Tanpa analitik big data, informasi berharga dari citra ini akan terbuang percuma.

Integrasi computer vision dengan big data memungkinkan analisis pola visual dalam skala luas. Contohnya, dalam epidemiologi digital, analisis jutaan foto dan video publik dapat membantu memantau penyebaran penyakit atau dampak bencana. Dalam perdagangan ritel, analisis data kamera dari ribuan toko dapat mengungkap pola perilaku konsumen dan preferensi produk.

Teknik distributed training dan parallel computing menjadi krusial untuk memproses dataset visual besar. Framework seperti TensorFlow Distributed atau PyTorch Lightning memungkinkan pelatihan model vision raksasa di kluster komputasi big data.

Namun, integrasi ini menghadapi dilema antara skala dan privasi. Analisis big data visual sering bersinggungan dengan hak individu, sehingga pendekatan etis dan legal menjadi kunci keberlanjutan.

### 13.5.5 Refleksi: Ekosistem Vision yang Terhubung

Integrasi vision dengan IoT, blockchain, AR/VR, dan big data pada dasarnya sedang membentuk ekosistem cyber-physical-socio-technical yang semakin kompleks. Di satu sisi, teknologi ini menjanjikan efisiensi, transparansi, dan pengalaman imersif yang belum pernah ada sebelumnya. Di sisi lain, ia menuntut standar baru dalam keamanan, privasi, interoperabilitas, dan keberlanjutan.

Dengan perspektif ini, masa depan computer vision bukan hanya soal algoritma yang lebih akurat, tetapi juga soal bagaimana vision hidup berdampingan dengan teknologi frontier lain, saling melengkapi, dan bersama-sama membentuk infrastruktur digital yang lebih cerdas dan adil.

# BAB XIV

## Perbandingan Framework & Tools dalam Computer Vision

### Pendahuluan

Dalam perkembangan computer vision, tidak hanya algoritma yang menentukan keberhasilan implementasi, tetapi juga framework dan tools yang mendasarinya. Framework adalah jembatan antara teori dan praktik; ia menyediakan pustaka, modul, serta API yang memungkinkan peneliti maupun praktisi membangun model vision secara lebih cepat, efisien, dan terstandarisasi. Tanpa framework yang matang, pengembangan algoritma vision akan sangat lambat, karena setiap peneliti harus membangun sistem dari nol.

Sejak dekade 2010-an, deep learning merevolusi bidang vision dan mendorong lahirnya berbagai framework besar seperti TensorFlow, PyTorch, dan MXNet. Di sisi lain, framework klasik seperti OpenCV tetap relevan karena menyediakan fungsi dasar pengolahan citra yang ringan dan fleksibel. Pada saat yang sama, ekosistem algoritma vision juga berkembang pesat dengan munculnya keluarga YOLO, Vision Transformer (ViT), DETR, hingga model generatif seperti Segment Anything Model (SAM).

Namun, keberagaman framework dan tools ini justru menghadirkan tantangan: framework mana yang paling sesuai untuk suatu kebutuhan tertentu? Seorang peneliti medis yang ingin mendeteksi tumor dari citra MRI mungkin membutuhkan stabilitas dan dukungan komunitas luas. Seorang pengembang IoT yang bekerja dengan kamera kecil di lapangan justru lebih membutuhkan framework ringan dan deployable di perangkat edge. Sementara itu, praktisi industri mengutamakan kecepatan pengembangan, dokumentasi yang lengkap, serta integrasi dengan pipeline MLOps.

Bab XIV ini hadir untuk memberikan perspektif komparatif atas ekosistem framework dan tools dalam computer vision. Fokusnya bukan sekadar membandingkan spesifikasi teknis, tetapi juga menilai aspek praktis seperti:

- Kemudahan penggunaan: Seberapa mudah framework diadopsi oleh pemula maupun peneliti lanjutan?
- Kinerja dan efisiensi: Bagaimana performa framework dalam melatih dan menguji model vision skala besar?
- Ekosistem dan komunitas: Seberapa luas dukungan pustaka, modul pihak ketiga, serta dokumentasi?
- Portabilitas dan deployment: Apakah framework dapat dijalankan di cloud, edge, hingga perangkat IoT?
- Kesesuaian dengan domain aplikasi: Framework apa yang lebih cocok untuk penelitian medis, industri manufaktur, atau aplikasi mobile?

Tujuan utama bab ini adalah membantu pembaca baik mahasiswa, peneliti, maupun praktisi untuk memahami lanskap tools vision, sehingga mampu memilih framework yang tepat sesuai konteks riset dan aplikasinya. Dengan begitu, computer vision tidak hanya berkembang dari sisi teori, tetapi juga menemukan jalannya menuju implementasi nyata yang efektif dan berkelanjutan.

## 14.1 OpenCV vs TensorFlow vs PyTorch

Ketika membicarakan computer vision, tiga framework yang hampir selalu muncul dalam diskusi adalah OpenCV, TensorFlow, dan PyTorch. Masing-masing memiliki sejarah, filosofi, dan ekosistem yang berbeda, yang menjadikannya unggul di konteks tertentu namun kurang sesuai di konteks lainnya. Pemahaman atas kelebihan dan keterbatasan tiap framework sangat penting, agar peneliti dan praktisi tidak hanya mengikuti tren, tetapi mampu memilih alat yang paling relevan dengan kebutuhan riset atau aplikasi mereka.

### 14.1.1 OpenCV: Pilar Klasik Vision

OpenCV (Open Source Computer Vision Library) merupakan salah satu pustaka tertua dan paling luas digunakan dalam pengolahan citra. Diluncurkan pertama kali pada tahun 2000, OpenCV dirancang untuk menyediakan fungsi dasar pengolahan gambar yang ringan, efisien, dan mudah diintegrasikan.

Keunggulan utama OpenCV terletak pada kelengkapan fungsi klasik: deteksi tepi, segmentasi sederhana, transformasi geometris, ekstraksi fitur (SIFT, SURF, ORB), hingga modul untuk pengolahan video real-time. Karena ditulis dalam C++ dengan binding Python, OpenCV dikenal sangat cepat dalam menangani operasi berbasis piksel.

Namun, OpenCV bukan framework deep learning. Ia lebih cocok diposisikan sebagai “toolbox” yang melengkapi framework modern. Dalam banyak aplikasi, OpenCV digunakan untuk preprocessing (misalnya normalisasi warna, cropping, augmentasi) sebelum data masuk ke model berbasis TensorFlow atau PyTorch. Dalam konteks industri, OpenCV sering dipakai untuk aplikasi embedded atau real-time vision pada perangkat terbatas karena jejak memorinya yang kecil.

### 14.1.2 TensorFlow: Standar Industri dan Produksi

TensorFlow, dikembangkan oleh Google Brain pada 2015, menjadi framework deep learning yang paling banyak diadopsi di dunia industri. Desain awalnya menekankan pada portabilitas dan skalabilitas, sehingga model yang dibangun dengan TensorFlow dapat dengan mudah dijalankan di berbagai platform: server cloud, perangkat edge, bahkan browser melalui TensorFlow.js.

Salah satu keunggulan TensorFlow adalah dukungan ekosistem yang luas. Dengan Keras API, pengembangan model vision menjadi lebih mudah diakses, bahkan oleh pemula. Untuk kebutuhan tingkat lanjut, TensorFlow mendukung pelatihan terdistribusi di kluster GPU/TPU

dengan efisiensi tinggi. Selain itu, integrasi dengan TensorFlow Lite memungkinkan deployment model ke perangkat mobile dan IoT.

Meski begitu, TensorFlow sempat dikritik karena kompleksitas sintaks pada versi awal. Banyak peneliti menganggap PyTorch lebih "alami" digunakan untuk eksperimen akademik. Namun, dengan TensorFlow 2.x, framework ini beralih ke mode eager execution yang lebih intuitif, sehingga kesenjangan dengan PyTorch semakin kecil.

Dalam konteks computer vision, TensorFlow banyak digunakan pada proyek industri berskala besar, misalnya sistem pengenalan wajah komersial, klasifikasi citra medis, atau analitik video di pusat perbelanjaan. Keunggulannya terletak pada kestabilan jangka panjang, dokumentasi yang matang, serta dukungan penuh untuk pipeline produksi.

### 14.1.3 PyTorch: Favorit Akademisi dan Peneliti

PyTorch, yang dikembangkan oleh Facebook AI Research (FAIR) pada 2016, dengan cepat menjadi framework favorit di komunitas akademik. Filosofi PyTorch adalah fleksibilitas dan kejelasan, dengan pendekatan eager execution sejak awal, yang membuat kode lebih mirip Python murni dan mudah di-debug.

Bagi peneliti, PyTorch menawarkan kebebasan untuk membangun arsitektur model vision yang eksperimental tanpa terikat terlalu ketat pada struktur framework. Hal ini menjadikannya populer dalam pengembangan arsitektur mutakhir seperti Vision Transformer, GANs, maupun model multimodal.

Ekosistem PyTorch juga terus berkembang. Torchvision menyediakan dataset standar (CIFAR, ImageNet, COCO) serta model pra-latih (ResNet, Faster R-CNN, YOLO). Selain itu, PyTorch mendukung integrasi dengan framework riset lain seperti Hugging Face Transformers untuk pengolahan multimodal.

Kelemahan PyTorch pada awalnya adalah keterbatasan dukungan untuk deployment skala industri. Namun, sejak hadirnya TorchScript dan ONNX (Open Neural Network Exchange), PyTorch kini lebih mudah diintegrasikan ke pipeline produksi.

### 14.1.4 Perbandingan Kekuatan dan Kelemahan

Jika dibandingkan secara garis besar, masing-masing framework menempati posisi unik:

- OpenCV: unggul dalam pemrosesan citra klasik dan aplikasi real-time ringan, tetapi tidak ditujukan untuk deep learning skala besar.
- TensorFlow: kuat dalam ekosistem industri, deployment, dan produksi jangka panjang, dengan dukungan penuh Google.
- PyTorch: unggul dalam riset akademik, eksperimen model baru, serta komunitas peneliti yang sangat aktif.

Dalam praktik, banyak proyek vision modern justru menggabungkan ketiganya. Misalnya, preprocessing citra menggunakan OpenCV, pelatihan model dengan PyTorch, lalu deployment ke perangkat mobile menggunakan TensorFlow Lite atau ONNX.

### 14.1.5 Implikasi untuk Peneliti dan Praktisi

Bagi mahasiswa atau peneliti awal, pemilihan framework sebaiknya mempertimbangkan learning curve dan kebutuhan riset. PyTorch biasanya lebih ramah untuk eksperimen, sementara TensorFlow lebih tepat jika hasil riset diarahkan ke implementasi industri. OpenCV tetap menjadi keterampilan dasar yang wajib dikuasai, karena hampir semua pipeline vision membutuhkan pengolahan citra tingkat rendah.

Bagi praktisi industri, keputusan biasanya didorong oleh faktor skalabilitas, dukungan jangka panjang, dan integrasi dengan infrastruktur yang ada. TensorFlow lebih aman untuk proyek yang membutuhkan dukungan enterprise, sementara PyTorch kini mulai diadopsi luas karena fleksibilitas dan kompatibilitasnya dengan ONNX.

Dengan kata lain, tidak ada framework tunggal yang "terbaik" untuk semua kasus. Pilihan terbaik adalah kombinasi yang cerdas, sesuai kebutuhan spesifik riset maupun aplikasi.

Tabel 14.1 Perbandingan Framework

| Aspek | OpenCV | TensorFlow | PyTorch |
|---|---|---|---|
| Fokus Utama | Pemrosesan citra klasik & video real-time | Deep learning skala industri, deployment, cloud | Riset akademik, eksperimen model baru, fleksibilitas |
| Bahasa Pemrograman | C++ (inti), Python, Java, MATLAB binding | Python (API utama), C++, Java, Go, JavaScript | Python (utama), C++ (backend) |
| Kemudahan Penggunaan | Mudah untuk fungsi dasar, dokumentasi luas | Keras API memudahkan pemula; versi awal rumit | Sangat intuitif, mirip Python murni, mudah debugging |
| Komunitas & Ekosistem | Komunitas lama, stabil, library ringan | Komunitas besar, dukungan industri Google | Komunitas riset aktif, integrasi Hugging Face |
| Model Pra-Latih | Terbatas (fitur klasik: SIFT, ORB, dll) | Sangat banyak (ResNet, Inception, EfficientNet) | Banyak (ResNet, Faster R-CNN, YOLO, ViT, dll) |
| Deployment | Embedded system, aplikasi desktop, mobile | TensorFlow Lite (mobile/IoT), TensorFlow.js, TPU | TorchScript, ONNX, integrasi ke C++/mobile |
| Keunggulan | Ringan, cepat untuk operasi piksel, real-time | Stabil, enterprise-ready, portabel, dokumentasi matang | Fleksibel, cepat untuk eksperimen, adaptif terhadap model baru |

| Aspek | OpenCV | TensorFlow | PyTorch |
|---|---|---|---|
| Keterbatasan | Tidak dirancang untuk deep learning | Versi awal kompleks; lebih “kaku” dari PyTorch | Deployment industri lebih baru, awalnya terbatas |
| Konteks Ideal | Preprocessing, edge vision, prototyping cepat | Produksi skala besar, cloud, aplikasi enterprise | Penelitian, pengembangan model baru, eksperimen akademik |

Tabel 14.1 memberikan gambaran ringkas mengenai posisi tiga framework besar dalam ekosistem computer vision. Namun, untuk memahami makna dari perbedaan ini, perlu dijelaskan lebih mendalam setiap aspek yang ditampilkan.

Pertama, dari sisi fokus utama, OpenCV menempati peran sebagai “toolbox klasik” untuk pemrosesan citra. Ia unggul dalam operasi dasar seperti deteksi tepi, segmentasi sederhana, atau transformasi geometris. TensorFlow, di sisi lain, didesain sejak awal untuk skala industri, dengan penekanan pada deep learning dan kemampuan deployment. PyTorch lebih menonjol di ranah penelitian akademik karena fleksibilitas dan kemudahan dalam membangun arsitektur model baru. Perbedaan fokus ini penting: ia menentukan apakah framework lebih cocok dipakai di laboratorium penelitian, industri berskala besar, atau perangkat embedded dengan keterbatasan sumber daya.

Kedua, bahasa pemrograman yang didukung turut memengaruhi adopsi. OpenCV berakar pada C++ sehingga sangat cepat dalam operasi tingkat rendah, tetapi penggunaannya lebih mudah diakses melalui binding Python. TensorFlow dan PyTorch keduanya berorientasi Python, bahasa yang menjadi standar de facto dalam riset AI, meskipun tetap memiliki dukungan backend C++ untuk performa tinggi. Adanya dukungan lintas bahasa, seperti TensorFlow.js atau OpenCV Java, memperluas ekosistem, khususnya untuk pengembangan web dan mobile.

Aspek kemudahan penggunaan menunjukkan dinamika evolusi framework. OpenCV relatif sederhana untuk fungsi dasar, tetapi membutuhkan pemahaman algoritmik jika digunakan untuk tugas kompleks. TensorFlow pada versi awal dikenal rumit, namun setelah hadirnya Keras API dan eager execution di TensorFlow 2.x, kurva pembelajarannya jauh lebih ramah. PyTorch sejak awal dirancang intuitif: sintaksisnya menyerupai Python murni, sehingga memudahkan debugging dan eksperimen. Tidak mengherankan bila PyTorch menjadi pilihan utama peneliti yang ingin membangun arsitektur baru dengan cepat.

Komunitas dan ekosistem menjadi indikator penting dari keberlanjutan framework. OpenCV memiliki komunitas lama dengan pustaka ringan yang stabil. TensorFlow didukung oleh Google, yang memastikan keberlanjutan jangka panjang, dokumentasi matang, serta dukungan enterprise. PyTorch, walaupun lahir lebih muda, berkembang sangat cepat berkat komunitas riset aktif. Kolaborasinya dengan ekosistem Hugging Face semakin memperkuat posisinya dalam pengembangan model mutakhir, termasuk Vision Transformer dan model multimodal.

Jika ditinjau dari ketersediaan model pra-latih, TensorFlow dan PyTorch unggul jauh dibandingkan OpenCV. Keduanya menawarkan model state-of-the-art yang dapat langsung digunakan atau di-fine-tune sesuai kebutuhan. PyTorch, dengan torchvision, menyediakan

banyak arsitektur populer seperti ResNet, Faster R-CNN, dan bahkan ViT. TensorFlow dengan TensorFlow Hub mempermudah penggunaan model-model besar untuk produksi. OpenCV, sebaliknya, tetap lebih terbatas pada fitur klasik seperti SIFT atau ORB, meskipun dapat memanggil model deep learning eksternal.

Dari segi deployment, TensorFlow memiliki keunggulan signifikan dengan varian TensorFlow Lite dan TensorFlow.js, yang memungkinkan model dijalankan di perangkat mobile maupun browser. PyTorch awalnya tertinggal, tetapi dengan TorchScript dan dukungan ONNX, kini ia mampu mengejar ketertinggalan di bidang ini. OpenCV lebih menonjol untuk aplikasi desktop, mobile ringan, atau perangkat embedded karena footprint memorinya kecil.

Perbedaan-perbedaan ini bermuara pada konteks ideal penggunaan. OpenCV sangat sesuai untuk preprocessing atau aplikasi vision ringan di edge devices. TensorFlow menjadi pilihan aman untuk sistem enterprise yang membutuhkan stabilitas dan portabilitas. PyTorch lebih unggul untuk eksplorasi akademik, riset, dan prototyping model baru yang cepat berubah.

Pada akhirnya, tabel ini tidak dimaksudkan untuk menilai siapa "pemenang mutlak", tetapi menunjukkan bahwa setiap framework memiliki niche masing-masing. Dalam praktik, integrasi ketiganya justru sering menjadi strategi optimal: OpenCV untuk preprocessing, PyTorch untuk eksperimen dan pelatihan, lalu TensorFlow atau ONNX untuk deployment. Sinergi ini mencerminkan kenyataan bahwa ekosistem computer vision lebih efektif jika dilihat sebagai jaringan saling melengkapi, bukan kompetisi tunggal.

## 14.2 Ekosistem YOLO dan Varian Terbaru

Di antara berbagai model object detection dalam computer vision, You Only Look Once (YOLO) menempati posisi unik sebagai salah satu arsitektur paling berpengaruh. Sejak diperkenalkan oleh Joseph Redmon dan timnya pada 2016, YOLO merevolusi deteksi objek dengan pendekatan real-time detection berbasis single shot. Filosofi YOLO sederhana namun radikal: alih-alih memproses citra melalui beberapa tahap region proposal seperti R-CNN, YOLO melakukan prediksi kelas dan bounding box secara langsung dalam satu kali inferensi.

Keunggulan YOLO terletak pada kombinasi kecepatan dan akurasi. Model ini mampu mendeteksi objek dalam video real-time, menjadikannya relevan untuk aplikasi praktis seperti sistem keamanan, kendaraan otonom, hingga deteksi penyakit tanaman. Dalam satu dekade terakhir, YOLO berkembang melalui berbagai versi, baik resmi maupun turunan komunitas, membentuk ekosistem yang sangat dinamis.

### 14.2.1 Evolusi YOLO

Evolusi YOLO dapat dipahami sebagai perjalanan dari model yang relatif sederhana menuju ekosistem yang kompleks dan beragam. YOLOv1 hingga YOLOv3 berfokus pada peningkatan akurasi sambil menjaga kecepatan. YOLOv4 memperkenalkan banyak teknik bag of freebies untuk memperkuat performa, sementara YOLOv5 yang lahir dari komunitas Ultralytics

menjadi titik balik dalam adopsi luas. Versi selanjutnya, YOLOv7 dan YOLOv8, semakin menekankan efisiensi, modularitas, dan kemudahan integrasi.

### 14.2.2 Perbandingan Evolusi YOLO

Tabel 14.2 Perbandingan Evolusi YOLO

| Versi YOLO | Tahun | Ciri Utama | Kelebihan | Keterbatasan |
|---|---|---|---|---|
| YOLOv1 | 2016 | Single CNN, prediksi langsung bounding box dan kelas | Sangat cepat, konsep baru | Akurasi rendah pada objek kecil |
| YOLOv2 (YOLO9000) | 2017 | Penambahan batch norm, anchor boxes, multi-scale training | Lebih akurat, mendukung >9000 kelas | Masih kesulitan pada deteksi multi-objek kompleks |
| YOLOv3 | 2018 | Darknet-53 backbone, deteksi multi-scale | Seimbang kecepatan-akurasi, sangat populer | Ukuran model relatif besar |
| YOLOv4 | 2020 | CSPDarknet53, *bag of freebies & specials* (Mosaic, CIoU, Mish activation) | Performa SOTA, open-source | Kompleksitas lebih tinggi |
| YOLOv5 | 2020 | Dikembangkan Ultralytics (PyTorch), mudah dipakai, modular | Populer, dokumentasi baik, dukungan komunitas luas | Tidak “resmi” dari penulis asli |
| YOLOv6 | 2022 | Optimalisasi industri (Meituan), fokus efisiensi produksi | Sangat cepat, cocok edge AI | Dukungan komunitas lebih terbatas |
| YOLOv7 | 2022 | Joint training, Extended E-ELAN, performa tinggi | SOTA di banyak benchmark, efisien | Relatif kompleks, kebutuhan hardware tinggi |
| YOLOv8 | 2023 | Dikembangkan Ultralytics, unified task (deteksi, segmentasi, klasifikasi) | Modular, mudah digunakan, mendukung multi-task | Masih relatif baru, ekosistem sedang berkembang |

Tabel 14.2 merangkum perjalanan YOLO dari versi pertama hingga YOLOv8. Namun untuk memahami konteks perkembangan ini, perlu dijelaskan lebih detail bagaimana setiap versi membawa lompatan inovasi sekaligus menghadapi keterbatasannya.

**YOLOv1 (2016)** adalah tonggak awal yang memperkenalkan ide radikal bahwa deteksi objek dapat dilakukan dalam satu kali inferensi (single shot). Pendekatan ini menghasilkan kecepatan sangat tinggi dibandingkan metode region proposal yang dominan saat itu. Namun, YOLOv1 masih menghadapi kelemahan mendasar: akurasi rendah terutama pada objek kecil atau yang saling berdekatan. Meski demikian, versi ini menandai perubahan paradigma bahwa deteksi objek bisa bersifat real-time.

**YOLOv2 (2017),** juga dikenal sebagai YOLO9000, memperbaiki keterbatasan tersebut dengan memperkenalkan anchor boxes dan pelatihan multi-skala. Tambahan ini membuat model lebih adaptif terhadap variasi ukuran objek. Lebih jauh, YOLOv2 mampu mengenali lebih dari 9000 kelas objek berkat pelatihan gabungan dengan data klasifikasi. Inovasi ini menjadikan YOLO semakin relevan, meskipun masalah deteksi objek kompleks tetap belum sepenuhnya teratasi.

**YOLOv3 (2018)** membawa perbaikan signifikan dengan backbone baru, Darknet-53, yang lebih dalam dan efisien. Pendekatan deteksi multi-skala semakin meningkatkan akurasi, sehingga YOLOv3 dikenal sebagai versi yang sangat seimbang antara kecepatan dan performa. Tidak heran bila hingga beberapa tahun kemudian, YOLOv3 tetap menjadi baseline populer di banyak penelitian. Kekurangannya adalah ukuran model yang relatif besar, sehingga kurang cocok untuk perangkat dengan keterbatasan memori.

**YOLOv4 (2020)** lahir dari komunitas open-source yang dipimpin Alexey Bochkovskiy. Versi ini menggabungkan berbagai teknik mutakhir, seperti Mosaic augmentation, CIoU loss, dan aktivasi Mish. Hasilnya, YOLOv4 mampu mencapai performa state-of-the-art pada benchmark deteksi objek, sekaligus mempertahankan efisiensi komputasi. Akan tetapi, kompleksitas tambahan membuat kurva pembelajaran lebih curam bagi pengguna pemula.

**YOLOv5 (2020)** menjadi titik balik yang menarik. Dikembangkan oleh Ultralytics dengan basis PyTorch, YOLOv5 tidak berasal dari penulis asli YOLO, namun justru menjadi varian paling populer. Keunggulannya terletak pada kemudahan penggunaan, dokumentasi lengkap, serta modularitas kode. Dukungan komunitas yang masif menjadikan YOLOv5 pilihan utama praktisi industri dan peneliti yang membutuhkan solusi cepat. Meski statusnya "tidak resmi", YOLOv5 berhasil mendominasi adopsi global.

**YOLOv6 (2022)** dikembangkan oleh perusahaan Meituan, dengan fokus pada efisiensi untuk aplikasi produksi skala besar. Versi ini mengoptimalkan performa untuk kebutuhan industri, terutama dalam sistem edge AI. Meski sangat cepat, ekosistem komunitasnya lebih terbatas dibandingkan YOLOv5, sehingga lebih sering digunakan dalam konteks internal perusahaan.

**YOLOv7 (2022)** memperkenalkan pendekatan baru dalam pelatihan gabungan (joint training) dan arsitektur Extended E-ELAN. Hasilnya, YOLOv7 mampu melampaui performa banyak model deteksi lain dalam benchmark publik. Dengan efisiensi tinggi, YOLOv7 dipandang sebagai salah satu puncak pencapaian dalam evolusi YOLO. Namun, kompleksitas

arsitekturnya menuntut hardware yang lebih kuat, sehingga adopsinya lebih banyak di lingkungan riset atau industri dengan sumber daya besar.

**YOLOv8 (2023)** kembali dikembangkan Ultralytics, dan kali ini lebih ambisius: ia dirancang sebagai framework unified yang tidak hanya untuk deteksi objek, tetapi juga segmentasi dan klasifikasi. Modularitas tinggi membuatnya mudah digunakan oleh berbagai tingkat pengguna, dari pemula hingga ahli. Meskipun masih relatif baru, YOLOv8 mulai dipandang sebagai standar baru ekosistem YOLO karena kemudahan integrasi dan fleksibilitas multi-task.

Melalui perjalanan delapan versi ini, terlihat jelas bahwa YOLO bukan sekadar model tunggal, melainkan sebuah ekosistem yang berevolusi mengikuti kebutuhan komunitas dan industri . Setiap versi membawa kompromi antara kecepatan, akurasi, dan kompleksitas, tetapi filosofi dasarnya tetap sama: menghadirkan deteksi objek yang cepat dan praktis untuk aplikasi nyata.

### 14.2.3 Keunggulan Ekosistem YOLO

Ada beberapa alasan mengapa YOLO menjadi ekosistem yang sangat dominan dalam object detection:

- **Real-time performance** YOLO sejak awal dirancang untuk kecepatan, menjadikannya relevan untuk aplikasi praktis yang membutuhkan inferensi instan.
- **Sederhana dan modular** Implementasi YOLO relatif mudah dipahami dibandingkan model deteksi lain seperti Faster R-CNN, sehingga cepat diadopsi komunitas.
- **Komunitas aktif** Sejak YOLOv5, komunitas open-source berperan besar dalam mengembangkan fitur baru, dokumentasi, dan integrasi ke berbagai platform.
- **Portabilitas** Varian YOLO dapat dijalankan di server GPU, perangkat edge seperti Jetson Nano, hingga mikrokontroler ringan, menjadikannya fleksibel lintas platform.

### 14.2.4 Keterbatasan dan Kritik

Meskipun unggul, YOLO tidak bebas dari kritik. Pada versi awal, model kesulitan mendeteksi objek kecil atau yang sangat berdekatan. Selain itu, perkembangan YOLO yang terfragmentasi (misalnya YOLOv4 dikembangkan Alexey Bochkovskiy, sementara YOLOv5 dikelola Ultralytics) menimbulkan kebingungan mengenai "versi resmi". Untuk riset akademik, fragmentasi ini menuntut kehati-hatian dalam membandingkan hasil eksperimen.

### 14.2.5 Implikasi untuk Riset dan Industri

Bagi peneliti, ekosistem YOLO menawarkan playground yang luas untuk menguji ide baru dalam deteksi objek, segmentasi, atau multi-task learning. Bagi praktisi industri, YOLO memberikan jalan pintas menuju implementasi deteksi real-time dengan dokumentasi yang mudah diikuti. Ke depan, YOLO kemungkinan akan terus berevolusi, bukan hanya sebagai arsitektur deteksi objek, tetapi juga sebagai platform multimodal yang mengintegrasikan visi dengan tugas lain seperti tracking dan pose estimation.

## 14.3 Vision Transformer (ViT, DETR, SAM)

Perkembangan deep learning dalam computer vision awalnya didominasi oleh Convolutional Neural Networks (CNN). Namun, dalam beberapa tahun terakhir, paradigma baru mulai

muncul dengan diperkenalkannya Vision Transformer (ViT) dan model turunannya seperti DETR (Detection Transformer) serta Segment Anything Model (SAM). Perubahan ini bukan sekadar inovasi teknis, melainkan pergeseran paradigma dalam cara sistem visual memahami citra.

Transformers, yang awalnya dikembangkan untuk pemrosesan bahasa alami (NLP), terbukti mampu menangkap hubungan global dalam data secara lebih baik dibandingkan pendekatan konvolusi yang cenderung fokus pada konteks lokal. Ketika arsitektur ini diadaptasi ke computer vision, lahirlah ViT sebagai pionir, yang kemudian diikuti oleh varian dan model turunannya.

### 14.3.1 Vision Transformer (ViT)

ViT diperkenalkan oleh tim Google pada tahun 2020. Prinsip dasarnya adalah memecah citra menjadi potongan kecil (patches) yang kemudian diperlakukan layaknya token dalam NLP. Dengan mekanisme self-attention, model dapat memahami hubungan antar-patch secara global, bukan hanya dalam lingkup lokal seperti CNN.

Keunggulan utama ViT adalah kemampuannya menangkap konteks global dengan lebih baik, sehingga efektif untuk tugas-tugas yang membutuhkan pemahaman struktur keseluruhan gambar. Namun, ViT membutuhkan dataset yang sangat besar (misalnya ImageNet-21k atau JFT-300M) agar dapat dilatih secara efektif. Hal ini membuatnya kurang praktis untuk domain dengan data terbatas, seperti medis atau pertanian, kecuali digabung dengan teknik transfer learning.

### 14.3.2 DETR (Detection Transformer)

Sementara ViT fokus pada klasifikasi citra, DETR yang diperkenalkan oleh tim Facebook AI pada 2020 memperluas konsep Transformer untuk deteksi objek. Alih-alih menggunakan region proposal seperti Faster R-CNN atau anchor boxes seperti YOLO, DETR memformulasikan deteksi objek sebagai tugas set prediction. Dengan mekanisme self-attention dan object queries, DETR langsung memprediksi bounding box dan label kelas tanpa komponen tambahan.

Kelebihan DETR adalah desainnya yang elegan dan sederhana, menghapus banyak tahap yang sebelumnya rumit dalam pipeline deteksi objek. Namun, kelemahannya terletak pada kebutuhan pelatihan yang panjang (konvergensi lambat) serta performa yang kurang optimal pada objek kecil, meskipun varian-varian baru (misalnya Deformable DETR) mulai mengatasi keterbatasan ini.

### 14.3.3 Segment Anything Model (SAM)

SAM, yang dirilis oleh Meta AI pada 2023, membawa computer vision ke babak baru: foundation model untuk segmentasi. Model ini dilatih dengan dataset skala masif (SA-1B dengan lebih dari 1 miliar mask), menjadikannya mampu melakukan segmentasi pada hampir semua objek, bahkan yang tidak pernah terlihat sebelumnya.

Keunggulan SAM adalah sifatnya general-purpose. Pengguna cukup memberikan prompt berupa klik, kotak, atau teks, dan SAM akan menghasilkan segmentasi akurat. Hal ini membuat

SAM bukan hanya alat riset, tetapi juga platform yang bisa digunakan di berbagai domain, mulai dari medis, geospasial, hingga seni digital.

Keterbatasan SAM lebih banyak terkait kebutuhan sumber daya komputasi yang besar, serta fakta bahwa model ini cenderung terlalu umum. Untuk aplikasi domain-spesifik (misalnya deteksi penyakit daun), hasil SAM sering kali perlu disesuaikan dengan fine-tuning tambahan.

### 14.3.4 Perbandingan ViT, DETR, dan SAM

Tabel 14.3 Perbandingan ViT, DETR, dan SAM

| Model | Tahun | Fokus Utama | Kelebihan | Keterbatasan | Aplikasi Ideal |
|---|---|---|---|---|---|
| ViT | 2020 | Klasifikasi citra | Menangkap konteks global; sederhana secara arsitektur | Butuh dataset besar; kurang efisien di data kecil | Klasifikasi skala besar, transfer learning |
| DETR | 2020 | Deteksi objek | Pipeline sederhana; menghapus anchor boxes | Latihan lambat; objek kecil sulit dikenali | Deteksi objek umum, aplikasi industri |
| SAM | 2023 | Segmentasi universal | Foundation model; mendukung prompt interaktif | Butuh resource besar; generalisasi perlu fine-tune | Segmentasi medis, geospasial, kreatif |

Tabel 14.3 menyajikan ringkasan mengenai tiga model vision berbasis Transformer yang menandai perubahan besar dalam lanskap computer vision: ViT, DETR, dan SAM. Masing-masing model tidak hanya menghadirkan pendekatan baru, tetapi juga mewakili arah evolusi penelitian yang berbeda: klasifikasi, deteksi, dan segmentasi universal.

Vision Transformer (ViT) yang lahir pada 2020 menjadi pionir penggunaan arsitektur Transformer di domain vision. Seperti terlihat pada tabel, fokus utama ViT adalah klasifikasi citra. Kelebihannya terletak pada kemampuannya menangkap konteks global—karena mekanisme self-attention memungkinkan hubungan antarbagian gambar dianalisis secara menyeluruh, bukan sekadar lokal seperti pada CNN. Keunggulan ini menjadikan ViT unggul pada dataset skala besar, misalnya ImageNet-21k. Namun, sebagaimana ditunjukkan di kolom keterbatasan, ViT sangat bergantung pada ketersediaan data masif. Jika dataset terbatas, performanya cenderung di bawah CNN. Oleh karena itu, ViT lebih ideal digunakan pada skenario klasifikasi berskala besar atau melalui pendekatan transfer learning.

DETR (Detection Transformer), juga dari tahun 2020, membawa konsep Transformer lebih jauh ke ranah deteksi objek. Tabel menegaskan bahwa keunggulan DETR adalah pipeline yang sederhana: ia menghilangkan kebutuhan anchor boxes atau region proposal yang selama ini menjadi komponen rumit dalam deteksi. Model ini memformulasikan deteksi sebagai masalah set prediction, sehingga arsitekturnya lebih elegan dan mudah dipahami. Akan tetapi, keterbatasannya cukup nyata: proses pelatihan lambat dan performa kurang optimal untuk objek kecil. Oleh sebab itu, DETR lebih sesuai untuk deteksi objek umum dalam aplikasi

industri yang tidak terlalu menuntut sensitivitas tinggi pada detail kecil, meski varian Deformable DETR sudah mulai mengatasi masalah ini.

Segment Anything Model (SAM), yang diperkenalkan pada 2023, adalah model segmentasi universal. Sebagaimana ditunjukkan di tabel, keunggulan utama SAM adalah sifatnya sebagai foundation model: dilatih pada dataset skala masif (lebih dari satu miliar mask), SAM mampu melakukan segmentasi pada hampir semua objek, bahkan yang belum pernah ditemui sebelumnya. Dengan sistem prompt interaktif, pengguna dapat meminta segmentasi hanya dengan klik, kotak, atau teks. Namun, keterbatasan SAM juga signifikan—ia membutuhkan sumber daya komputasi besar dan sifatnya yang terlalu umum membuat hasilnya kadang perlu fine-tuning tambahan agar sesuai dengan domain tertentu (misalnya medis atau pertanian). Meski begitu, SAM sangat potensial untuk aplikasi lintas sektor, dari kedokteran hingga geospasial dan seni kreatif.

Secara keseluruhan, tabel ini menggambarkan trilogi Transformer dalam vision: ViT mewakili klasifikasi global, DETR membawa kesederhanaan dalam deteksi, dan SAM memperkenalkan era segmentasi universal. Masing-masing memiliki keunggulan dan keterbatasan, sehingga pilihan terbaik bergantung pada konteks penggunaan. Yang lebih penting, keberadaan ketiga model ini menandai bahwa computer vision kini bergerak menuju paradigma foundation model—model besar yang mampu beradaptasi ke berbagai tugas tanpa harus dilatih ulang dari nol.

### 14.3.5 Implikasi Paradigma Baru

Kemunculan ViT, DETR, dan SAM menandai transisi computer vision menuju era foundation models dan multimodal AI. Jika CNN adalah fondasi generasi sebelumnya, maka Transformer menjadi fondasi generasi berikutnya. Model-model ini menunjukkan bahwa pendekatan yang awalnya dominan di NLP kini mulai mendefinisi ulang lanskap vision.

Bagi peneliti, paradigma ini membuka ruang eksplorasi baru: bagaimana mengadaptasi Transformer untuk data terbatas, bagaimana menggabungkan vision dengan bahasa, atau bagaimana membangun model general-purpose yang tetap efisien. Bagi praktisi, model seperti SAM menjadi peluang besar untuk mempercepat pengembangan aplikasi tanpa harus melatih model dari nol.

Namun, tantangan juga muncul. Resource komputasi yang besar bisa menimbulkan kesenjangan antara institusi besar dan peneliti individu. Oleh karena itu, penelitian ke depan juga harus menekankan efisiensi dan demokratisasi penggunaan foundation model vision

## 14.4 Edge AI & TinyML dalam Vision

Selama dua dekade terakhir, computer vision berkembang pesat berkat ketersediaan GPU dan cloud computing. Namun, tidak semua aplikasi memungkinkan akses ke cloud dengan latensi rendah dan sumber daya besar. Banyak skenario justru menuntut pemrosesan langsung di perangkat ujung (edge devices), seperti kamera, drone, robot pertanian, atau sensor IoT. Dari sinilah lahir konsep Edge AI—pemrosesan kecerdasan buatan langsung di perangkat ujung—

dan TinyML, yaitu implementasi machine learning dalam perangkat dengan daya dan memori yang sangat terbatas.

Kedua pendekatan ini berperan penting dalam memperluas cakupan aplikasi vision ke lapangan nyata. Edge AI menjembatani kebutuhan real-time processing dengan keterbatasan konektivitas, sementara TinyML memungkinkan model vision dijalankan pada mikrokontroler dengan konsumsi daya miliwatt.

### 14.4.1 Konsep Dasar Edge AI

Edge AI merujuk pada pemrosesan inferensi model machine learning di perangkat edge, tanpa selalu bergantung pada server pusat. Dalam konteks vision, hal ini berarti gambar atau video dianalisis langsung di perangkat kamera, drone, atau gateway lokal.

Keunggulan utama Edge AI adalah latensi rendah dan privasi lebih terjaga, karena data tidak perlu dikirim penuh ke cloud. Namun, tantangannya adalah keterbatasan daya komputasi. Oleh karena itu, desain arsitektur model harus menekankan efisiensi, misalnya dengan model pruning, kuantisasi, atau distilasi.

### 14.4.2 TinyML: Vision dalam Mikrokontroler

TinyML membawa ide lebih jauh: menjalankan model vision pada mikrokontroler berdaya sangat rendah, seperti ESP32, STM32, atau Arduino. Meski kemampuan komputasi terbatas, TinyML tetap relevan untuk tugas-tugas sederhana, misalnya deteksi gerakan, pengenalan pola daun, atau klasifikasi sederhana di lapangan pertanian.

TinyML memanfaatkan model ringan seperti MobileNet, SqueezeNet, atau varian YOLO-Nano. Tantangan utamanya adalah bagaimana menyeimbangkan ukuran model dengan akurasi, karena semakin ringan model, semakin besar risiko kehilangan informasi penting.

### 14.4.3 Perbandingan Edge AI dan TinyML

Tabel 14.4 Perbandingan Edge AI dan TinyML

| Aspek | Edge AI | TinyML |
|---|---|---|
| Perangkat | Jetson Nano, Raspberry Pi, Google Coral | ESP32, STM32, Arduino, microcontrollers |
| Kapasitas Komputasi | GPU/TPU kecil, RAM ratusan MB – beberapa GB | RAM sangat terbatas (KB–MB) |
| Tugas Vision | Deteksi objek real-time, segmentasi ringan | Klasifikasi sederhana, deteksi peristiwa |
| Konsumsi Daya | Watt-level (relatif hemat) | Miliwatt-level (sangat hemat) |

| Aspek | Edge AI | TinyML |
|---|---|---|
| Kelebihan | Latensi rendah, dukungan model kompleks | Ultra-low power, biaya sangat murah |
| Keterbatasan | Tetap butuh daya lebih tinggi dari sensor IoT | Akurasi terbatas, sulit untuk tugas kompleks |

Tabel 14.4 menggambarkan perbedaan mendasar sekaligus hubungan komplementer antara Edge AI dan TinyML dalam konteks computer vision. Kedua pendekatan ini sering kali disebut beriringan, tetapi sebenarnya melayani kebutuhan yang berbeda dengan karakteristik perangkat keras dan target aplikasi yang khas.

Pada aspek perangkat, Edge AI biasanya dijalankan pada platform dengan dukungan GPU atau akselerator khusus seperti Jetson Nano, Raspberry Pi, atau Google Coral. Perangkat ini masih relatif kecil, tetapi memiliki kekuatan komputasi yang cukup untuk memproses model vision kompleks seperti YOLO atau U-Net. Sebaliknya, TinyML berjalan pada mikrokontroler berdaya sangat rendah seperti ESP32, STM32, atau Arduino. Perangkat ini tidak memiliki GPU dan memori sangat terbatas, sehingga hanya mampu menjalankan model ringan.

Perbedaan berikutnya terlihat pada kapasitas komputasi. Edge AI bekerja dengan RAM ratusan megabyte hingga beberapa gigabyte, sehingga mampu menampung model deteksi dan segmentasi tingkat lanjut. TinyML, di sisi lain, biasanya hanya memiliki RAM dalam skala kilobyte hingga megabyte, sehingga model yang dijalankan harus sangat ringkas, sering kali hasil pruning atau quantization ekstrem.

Dari sisi tugas vision, Edge AI cocok untuk aplikasi real-time yang membutuhkan deteksi objek, segmentasi ringan, atau bahkan penghitungan jumlah objek pada video. TinyML lebih difokuskan pada klasifikasi sederhana, misalnya membedakan daun sehat atau sakit berdasarkan warna dominan, atau mendeteksi adanya gerakan pada sensor kamera kecil.

Konsumsi daya juga menjadi pembeda penting. Edge AI umumnya membutuhkan daya listrik pada tingkat watt, cukup hemat dibanding server cloud, tetapi tetap memerlukan suplai yang stabil. TinyML, sebaliknya, beroperasi pada tingkat miliwatt, memungkinkan perangkat bekerja lama dengan baterai kecil atau bahkan energi terbarukan seperti panel surya mini.

Kelebihan masing-masing pendekatan pun berbeda. Edge AI unggul dalam kemampuannya menjalankan model vision yang relatif kompleks dengan latensi rendah, sehingga dapat memberikan umpan balik instan di lapangan. TinyML unggul dalam efisiensi energi dan biaya: perangkat murah, konsumsi daya sangat rendah, serta mudah dipasang di lokasi terpencil.

Namun, keduanya juga memiliki keterbatasan. Edge AI, meskipun efisien, tetap tidak cocok untuk perangkat IoT ultra-ringan karena masih membutuhkan daya lebih tinggi dan komponen lebih mahal. TinyML, di sisi lain, terbatas pada akurasi dan kompleksitas tugas; semakin ringan

modelnya, semakin besar kemungkinan kehilangan detail penting, sehingga tidak dapat diandalkan untuk aplikasi vision yang memerlukan presisi tinggi.

Dari penjelasan ini, jelas bahwa Edge AI dan TinyML bukanlah pesaing, melainkan pelengkap. Edge AI mengisi ruang aplikasi vision yang memerlukan inferensi kompleks di lapangan, sedangkan TinyML memungkinkan pemantauan skala besar dengan biaya rendah dan daya sangat hemat. Keduanya, bila digabungkan, dapat membentuk ekosistem vision yang tangguh: Edge AI sebagai otak lokal yang kuat, dan TinyML sebagai mata-mata ringan yang tersebar di berbagai titik lingkungan.

### 14.4.4 Aplikasi Nyata Edge AI & TinyML dalam Vision

- **Pertanian Presisi** Drone dengan Edge AI dapat mendeteksi gulma atau penyakit daun secara real-time. Sementara itu, sensor berbasis TinyML bisa dipasang di lahan untuk mendeteksi kondisi visual sederhana seperti warna daun atau keberadaan hama.

- **Keamanan dan Pengawasan** Kamera edge dapat melakukan deteksi wajah tanpa perlu mengirim data ke cloud, menjaga privasi. TinyML dapat mendeteksi gerakan pada sensor pintu atau jendela.

- **Industri Manufaktur** Edge AI digunakan untuk quality control produk dengan deteksi cacat permukaan. TinyML dipakai untuk mendeteksi pola getaran mesin melalui sensor visual atau kombinasi vision dengan sensor lain.

- **Kesehatan** Kamera edge dapat digunakan untuk skrining medis sederhana. TinyML, dengan sensor kamera mini, dapat memonitor pola tidur atau gerakan pasien di rumah.

### 14.4.5 Implikasi Penelitian dan Masa Depan

Edge AI dan TinyML membuka ruang penelitian baru dalam kompresi model, efisiensi energi, dan integrasi vision dengan IoT. Ke depan, kombinasi keduanya bisa menghasilkan sistem pertanian cerdas yang hemat energi, rumah sakit digital dengan sensor otonom, hingga kota pintar yang lebih ramah privasi.

Meski demikian, ada tantangan yang perlu diantisipasi: keterbatasan memori, masalah kompatibilitas perangkat keras, serta kebutuhan untuk menjaga keamanan data di jaringan edge. Hal-hal ini menuntut pendekatan interdisipliner, menggabungkan ilmu komputer, elektronika, dan keamanan siber.

## 14.5 Studi Perbandingan Performa Framework

Salah satu tantangan utama dalam computer vision bukan hanya merancang model, tetapi juga memilih framework yang tepat untuk implementasi. Framework berfungsi sebagai fondasi perangkat lunak yang menyediakan pustaka, API, dan optimisasi sehingga peneliti maupun praktisi dapat lebih mudah membangun dan menguji model.

Framework vision berkembang seiring evolusi deep learning. OpenCV sebagai pionir banyak dipakai untuk operasi dasar citra, TensorFlow dan PyTorch mendominasi riset dan produksi

model deep learning, sementara framework khusus seperti YOLO dan Detectron2 muncul untuk mempercepat aplikasi deteksi dan segmentasi.

Membandingkan framework bukan sekadar soal kecepatan, tetapi juga mencakup akurasi model, efisiensi komputasi, kemudahan penggunaan, fleksibilitas, dan dukungan ekosistem.

### 14.5.1 Dimensi Perbandingan Framework

1. Kinerja Model (Performance) – sejauh mana framework memungkinkan implementasi model dengan akurasi tinggi dan kecepatan inferensi baik, khususnya pada GPU maupun edge device.
2. Kemudahan Penggunaan (Usability) – kualitas dokumentasi, ketersediaan pre-trained models, serta komunitas pengguna yang mendukung.
3. Ekosistem dan Integrasi – dukungan framework terhadap pipeline lengkap, mulai dari akuisisi data, pelatihan, hingga deployment di edge maupun cloud.
4. Efisiensi Komputasi – kemampuan framework dalam mengoptimalkan penggunaan memori dan daya, terutama penting dalam konteks edge AI dan TinyML.
5. Kesesuaian Domain Aplikasi – apakah framework lebih cocok untuk penelitian akademik, prototipe industri, atau aplikasi produksi skala besar.

### 14.5.2 Perbandingan Framework Vision

Tabel 14.5Perbandingan Framework Vision

| Framework | Kelebihan | Keterbatasan | Aplikasi Ideal |
|---|---|---|---|
| OpenCV | Ringan, kaya fungsi dasar citra, mudah integrasi dengan C++/Python | Terbatas untuk *deep learning* modern | Preprocessing, pengolahan citra dasar |
| TensorFlow | Dukungan produksi kuat, kompatibel dengan TPU, ekosistem luas (TF Lite, TF Serving) | Kurva belajar curam, sintaks relatif kompleks | Produksi skala besar, aplikasi cloud |
| PyTorch | Sintaks intuitif, populer di riset, ekosistem kuat (TorchVision, PyTorch Lightning) | Deployment lebih rumit dibanding TensorFlow (meski membaik) | Penelitian akademik, prototipe cepat |
| YOLO Framework (Ultralytics) | Mudah digunakan, fokus pada deteksi, dukungan komunitas luas | Terbatas pada tugas spesifik (deteksi/segmentasi) | Deteksi objek real-time di industri |

| Framework | Kelebihan | Keterbatasan | Aplikasi Ideal |
|---|---|---|---|
| Detectron2 | Modular, mendukung berbagai arsitektur deteksi/segmentasi, benchmark kuat | Membutuhkan resource besar, setup relatif kompleks | Benchmark akademik, penelitian SOTA |
| ONNX Runtime | Portabilitas tinggi, optimisasi lintas platform, mendukung konversi model | Tergantung framework asal (TF/PyTorch) | Deployment lintas perangkat |

Tabel 14.5 menyajikan enam framework vision utama yang banyak digunakan dalam riset dan aplikasi industri: OpenCV, TensorFlow, PyTorch, YOLO (Ultralytics), Detectron2, dan ONNX Runtime. Setiap framework memiliki karakteristik unik, yang tercermin dari kelebihan, keterbatasan, serta konteks aplikasi idealnya.

OpenCV menempati posisi sebagai toolkit fundamental dalam pengolahan citra digital. Dengan pustaka yang ringan dan kaya fungsi dasar, OpenCV sangat cocok untuk tahap awal pipeline vision seperti preprocessing, augmentasi, atau ekstraksi fitur klasik. Keunggulannya adalah kemudahan integrasi dengan C++ maupun Python. Namun, sebagaimana tercatat di tabel, OpenCV tidak dikembangkan untuk mendukung deep learning modern secara penuh, sehingga fungsinya kini lebih sering sebagai pelengkap framework lain.

TensorFlow, dikembangkan oleh Google, adalah framework deep learning yang kuat dengan dukungan produksi kelas industri. Ekosistemnya luas, mencakup TensorFlow Lite untuk perangkat edge, TensorFlow Serving untuk deployment server, serta dukungan TPU di Google Cloud. Hal ini menjadikan TensorFlow sangat relevan untuk implementasi skala besar. Akan tetapi, kurva belajar TensorFlow relatif curam karena sintaksnya kompleks, sehingga pengguna baru sering kesulitan di tahap awal.

PyTorch menjadi favorit di kalangan peneliti akademik. Sintaksnya yang intuitif dan menyerupai Python murni membuat proses eksperimen lebih fleksibel dan cepat. PyTorch memiliki ekosistem TorchVision dan PyTorch Lightning yang mendukung berbagai model pre-trained, sehingga mempercepat riset dan prototipe. Keterbatasannya, seperti tercatat di tabel, terletak pada aspek deployment yang semula lebih rumit dibandingkan TensorFlow. Meski begitu, perkembangan ONNX dan TorchServe kini mulai mengatasi kendala tersebut.

YOLO Framework (Ultralytics) menonjol karena sifatnya yang domain-spesifik, yakni deteksi dan segmentasi objek. Framework ini dirancang dengan antarmuka yang sederhana, sehingga pengguna dapat melatih model deteksi dengan sedikit baris kode. Dukungan komunitas yang luas membuatnya populer di industri yang membutuhkan deteksi real-time, seperti pertanian presisi, keamanan, dan robotika. Namun, karena fokus pada tugas tertentu, YOLO tidak sefleksibel TensorFlow atau PyTorch dalam mendukung berbagai jenis arsitektur vision.

Detectron2, dikembangkan oleh Facebook AI Research, merupakan framework riset modular untuk deteksi dan segmentasi. Keunggulannya adalah dukungan terhadap berbagai arsitektur state-of-the-art (misalnya Faster R-CNN, Mask R-CNN, RetinaNet) serta benchmark kuat.

Detectron2 sering dipakai dalam penelitian yang bertujuan menghasilkan kinerja tertinggi di dataset publik. Namun, sebagaimana tercatat di tabel, framework ini membutuhkan sumber daya besar dan pengaturan awal yang relatif kompleks, sehingga lebih cocok untuk lingkungan riset daripada deployment praktis.

ONNX Runtime berbeda dari framework lain karena berfokus pada tahap deployment. Ia berfungsi sebagai jembatan yang memungkinkan model dari TensorFlow maupun PyTorch dijalankan secara portabel di berbagai perangkat dengan optimisasi performa. Keunggulannya terletak pada fleksibilitas lintas platform, sehingga ideal untuk industri yang ingin mengintegrasikan model ke dalam sistem produksi heterogen. Namun, keterbatasannya adalah ONNX tidak menyediakan fasilitas pelatihan model; ia hanya bekerja pada model yang sudah dilatih sebelumnya.

Dari perbandingan ini dapat dilihat bahwa tidak ada framework yang unggul mutlak dalam semua aspek. OpenCV unggul pada pengolahan citra dasar, PyTorch memimpin di ranah riset, TensorFlow lebih matang dalam produksi, YOLO sangat efektif untuk deteksi real-time, Detectron2 unggul dalam benchmark akademik, dan ONNX Runtime menjadi solusi portabilitas. Dengan demikian, pemilihan framework harus disesuaikan dengan tujuan: apakah untuk riset, prototipe cepat, aplikasi industri, atau deployment lintas perangkat.

### 14.5.3 Penjelasan Perbandingan Framework

Dari tabel, OpenCV menonjol sebagai framework klasik yang hingga kini tetap relevan. Meski tidak dirancang khusus untuk deep learning, OpenCV sangat penting untuk tahap awal seperti preprocessing, augmentasi, atau pengolahan citra dasar.

**TensorFlow dan PyTorch** adalah dua raksasa deep learning. TensorFlow unggul dalam produksi dan deployment berkat ekosistem luas (TF Lite untuk perangkat mobile/edge, TF Serving untuk deployment server, serta dukungan TPU di Google Cloud). Namun, sintaks TensorFlow sering dianggap lebih sulit dipahami pemula. Sebaliknya, PyTorch populer di kalangan akademik karena sintaksnya yang intuitif dan mirip Python murni. Ekosistem TorchVision menyediakan pre-trained models yang mudah digunakan untuk penelitian.

**YOLO Framework (Ultralytics)** berbeda karena bersifat domain-spesifik: fokus pada deteksi dan segmentasi real-time. Keunggulannya adalah kemudahan pemakaian—pengguna dapat melatih dan menguji model deteksi hanya dengan beberapa baris kode. Framework ini menjadi favorit di industri pertanian presisi, keamanan, dan robotika.

**Detectron2**, dikembangkan oleh Facebook AI Research, dirancang lebih modular dan mendukung berbagai arsitektur state-of-the-art (Faster R-CNN, Mask R-CNN, RetinaNet). Framework ini sangat berguna untuk riset akademik karena menyediakan benchmark yang kuat. Namun, kebutuhan komputasinya tinggi sehingga kurang ramah untuk perangkat edge.

**ONNX Runtime** bukan framework pelatihan, melainkan platform untuk deployment. Keunggulannya adalah portabilitas: model dari TensorFlow atau PyTorch dapat diekspor ke ONNX dan dijalankan di berbagai perangkat dengan optimisasi performa. Hal ini membuat

ONNX Runtime sangat ideal untuk aplikasi lintas platform, dari server cloud hingga mikrokontroler.

### 14.5.4 Implikasi bagi Peneliti dan Praktisi

Bagi peneliti, pilihan framework menentukan kecepatan iterasi riset. PyTorch sering menjadi pilihan utama untuk eksperimen karena fleksibilitas dan komunitas aktif. TensorFlow tetap relevan untuk riset yang diarahkan ke implementasi produksi.

Bagi praktisi industri, YOLO dan ONNX Runtime menawarkan jalan pintas untuk membangun aplikasi vision siap pakai. YOLO menyederhanakan deteksi objek real-time, sementara ONNX Runtime memungkinkan integrasi model yang sebelumnya dikembangkan di berbagai framework.

Kesimpulannya, tidak ada satu framework yang unggul mutlak. Pilihan terbaik tergantung konteks: OpenCV untuk preprocessing, PyTorch untuk riset, TensorFlow untuk produksi, YOLO untuk aplikasi deteksi cepat, Detectron2 untuk benchmark, dan ONNX Runtime untuk deployment lintas platform. Kombinasi beberapa framework sering kali menjadi strategi optimal dalam proyek computer vision nyata.

# Bab XV
# Etika, Keamanan, dan Privasi dalam Computer Vision

## 15.1 Isu Etika dalam Computer Vision

Perkembangan pesat computer vision membawa dampak yang melampaui ranah teknis. Algoritma vision kini menjadi elemen penting dalam sistem medis, keamanan publik, transportasi otonom, hingga interaksi sehari-hari melalui perangkat pintar. Namun, di balik capaian tersebut, muncul pertanyaan mendasar: apakah teknologi vision yang kita bangun selalu netral dan adil? Pertanyaan ini membuka diskusi etika yang tidak dapat diabaikan, terutama dalam konteks akademik tingkat lanjut.

Bias Data dan Diskriminasi Algoritmik

Salah satu isu etika yang paling menonjol adalah bias dalam data. Model vision belajar dari dataset, dan kualitas pembelajaran sangat bergantung pada representativitas data. Jika dataset didominasi oleh citra dari kelompok tertentu—misalnya mayoritas wajah dari ras kulit putih—maka model cenderung berkinerja buruk ketika menghadapi wajah dari kelompok lain. Studi yang dilakukan MIT Media Lab menunjukkan bahwa sistem pengenalan wajah komersial memiliki tingkat kesalahan yang jauh lebih tinggi pada wajah perempuan berkulit gelap dibandingkan laki-laki berkulit terang. Hal ini menegaskan bahwa bias dalam dataset dapat menghasilkan diskriminasi algoritmik. Dalam konteks medis, bias serupa dapat menyebabkan diagnosa yang salah pada kelompok etnis tertentu karena kurangnya data pelatihan yang representatif. Isu ini bukan hanya persoalan teknis, tetapi juga etis. Model yang bias berpotensi memperkuat ketidakadilan sosial yang sudah ada. Dengan demikian, peneliti memiliki tanggung jawab moral untuk memastikan dataset yang digunakan benar-benar mencerminkan keberagaman populasi.

### Transparansi dan Tantangan Black Box

Model vision modern, terutama yang berbasis deep learning, sering digambarkan sebagai kotak hitam (black box). Meskipun akurasinya tinggi, sulit menjelaskan mengapa model membuat keputusan tertentu. Dalam aplikasi sensitif seperti kesehatan, hal ini menimbulkan dilema etis. Misalnya, jika sebuah sistem deteksi kanker payudara berbasis CNN menyatakan bahwa suatu pasien positif kanker, dokter dan pasien berhak mengetahui dasar keputusan tersebut. Namun, jika model hanya memberikan prediksi tanpa penjelasan, kepercayaan publik akan terganggu. Kebutuhan akan transparansi algoritma kemudian muncul. Penelitian tentang interpretabilitas, seperti Grad-CAM atau saliency map, merupakan upaya awal untuk membuka isi kotak hitam ini. Secara etis, transparansi bukan sekadar kemewahan, melainkan prasyarat agar teknologi vision dapat dipertanggungjawabkan.

### Implikasi Sosial dan Potensi Penyalahgunaan

Selain bias dan keterbatasan transparansi, computer vision juga menghadirkan risiko sosial yang serius. Salah satunya adalah penggunaan sistem pengenalan wajah untuk pengawasan massal (mass surveillance). Teknologi yang semula dikembangkan untuk keamanan dapat berubah menjadi instrumen kontrol yang mengancam privasi dan kebebasan sipil.

Fenomena deepfake juga menjadi contoh nyata. Dengan mengombinasikan vision dan generative models, kini siapa pun dapat memproduksi video palsu yang sangat realistis. Hal ini memunculkan risiko besar terhadap kepercayaan publik, politik, dan bahkan keamanan nasional. Dalam konteks ini, muncul pertanyaan etis yang lebih luas: apakah peneliti vision bertanggung jawab atas penyalahgunaan hasil penelitiannya? Sebagian berpendapat bahwa teknologi bersifat netral dan hanya penggunaannya yang menentukan. Namun, perspektif lain menegaskan bahwa sejak tahap desain, peneliti sudah membuat keputusan yang memengaruhi arah penggunaan teknologi. Dengan kata lain, etika harus hadir sejak awal proses penelitian, bukan hanya setelah produk digunakan.

## 15.2 Keamanan Model dan Data dalam Computer Vision

Isu keamanan dalam computer vision tidak hanya menyangkut perlindungan data citra, tetapi juga mencakup integritas model, kerentanan terhadap serangan, serta reliabilitas sistem ketika berhadapan dengan kondisi dunia nyata. Kompleksitas ini muncul karena algoritma vision modern bekerja pada skala besar, dengan parameter jutaan hingga miliaran, serta sering beroperasi di lingkungan yang tidak sepenuhnya dapat dikendalikan.

Oleh sebab itu, pembahasan keamanan model dan data menjadi elemen krusial. Ia bukan sekadar diskusi teknis tentang enkripsi atau akses kontrol, tetapi juga refleksi etis dan epistemologis mengenai bagaimana pengetahuan visual diproduksi, disebarkan, dan dipertahankan dalam ekosistem digital yang rentan.

### Serangan Adversarial: Tantangan Terhadap Keandalan Model

Salah satu kerentanan paling serius adalah serangan adversarial. Dengan menambahkan gangguan kecil yang hampir tidak terlihat pada sebuah citra, penyerang dapat membuat model deep learning salah klasifikasi secara drastis. Misalnya, sebuah gambar rambu “STOP” yang diberi pola perturbation dapat dikenali model kendaraan otonom sebagai “Speed Limit 60”.

Dampaknya jelas: keselamatan publik bisa terancam hanya karena manipulasi visual yang hampir tidak terlihat oleh mata manusia. Fenomena ini menunjukkan bahwa meskipun model deep learning mencapai akurasi tinggi dalam kondisi laboratorium, ia tetap rapuh terhadap eksploitasi terarah. Upaya mitigasi serangan adversarial mencakup adversarial training, deteksi input abnormal, hingga penggunaan arsitektur yang lebih robust. Namun, tidak ada solusi yang benar-benar kebal. Hal ini menimbulkan pertanyaan mendalam: sejauh mana kita dapat mempercayai keputusan algoritma vision ketika dunia nyata penuh dengan potensi serangan yang halus namun berbahaya?

### Model Inversion dan Kebocoran Data

Selain serangan pada input, terdapat risiko model inversion dan data leakage. Melalui teknik tertentu, penyerang dapat merekonstruksi kembali informasi sensitif dari parameter model. Sebagai contoh, sebuah model pengenalan wajah yang dilatih dengan dataset medis berpotensi “membocorkan” citra pasien jika terjadi inversi model. Masalah ini menggarisbawahi dilema antara keterbukaan ilmiah dan keamanan. Di satu sisi, publikasi model terbuka (open model) mempercepat kolaborasi dan inovasi. Namun di sisi lain, semakin banyak parameter dan bobot model tersedia, semakin besar pula peluang penyalahgunaan. Strategi mitigasi mencakup penggunaan differential privacy, enkripsi model, serta pendekatan federated learning, di mana data tidak pernah keluar dari perangkat pengguna. Pendekatan ini menyeimbangkan kebutuhan riset terbuka dengan perlindungan privasi individu.

### Robustness dan Reliability dalam Dunia Nyata

Keamanan model juga terkait dengan robustness—kemampuan model untuk tetap berfungsi dalam kondisi yang berbeda dari data pelatihan. Sistem vision sering diuji dalam lingkungan terkendali, tetapi ketika diimplementasikan di dunia nyata, ia harus menghadapi variasi cahaya, cuaca, sensor, atau bahkan kerusakan perangkat keras.

Sebagai contoh, kamera CCTV di ruang publik mungkin berfungsi optimal pada siang hari, tetapi performa model menurun drastis ketika malam tiba atau saat hujan lebat. Hal serupa berlaku dalam sektor medis, di mana kualitas citra MRI dapat bervariasi antar rumah sakit atau mesin. Isu ini menunjukkan bahwa keamanan tidak hanya menyangkut protection from attack, tetapi juga resilience terhadap ketidakpastian operasional. Model vision yang aman adalah model yang tetap dapat diandalkan dalam menghadapi gangguan yang tidak terduga, baik teknis maupun lingkungan.

## 15.3 Privasi dalam Computer Vision

Di antara berbagai isu etis dan keamanan, privasi menempati posisi yang sangat sentral dalam diskursus computer vision. Berbeda dengan teks atau data numerik, citra dan video menyimpan informasi yang sangat personal—wajah, tubuh, lokasi, bahkan pola perilaku individu. Oleh karena itu, implementasi teknologi vision selalu berhadapan dengan dilema antara kebutuhan fungsional dan perlindungan hak privasi manusia.

### Face Recognition dan Identitas Digital

Pengenalan wajah (face recognition) adalah salah satu aplikasi vision yang paling banyak digunakan sekaligus paling kontroversial. Di satu sisi, teknologi ini memberikan manfaat nyata, seperti verifikasi identitas pada perangkat pintar atau keamanan akses gedung. Namun, di sisi lain, ia juga menjadi instrumen pengawasan massal yang berpotensi mengikis kebebasan sipil. Beberapa kota di Amerika Serikat telah melarang penggunaan sistem pengenalan wajah oleh aparat kepolisian karena risiko pelanggaran privasi dan diskriminasi. Kasus serupa terjadi di Tiongkok, di mana teknologi ini digunakan untuk memantau populasi dalam skala luas. Fenomena ini memperlihatkan bahwa face recognition tidak hanya soal teknis akurasi, tetapi juga representasi kekuasaan dan kontrol sosial. Lebih jauh, wajah manusia bukan sekadar "fitur visual". Ia adalah simbol identitas dan martabat. Ketika sebuah sistem vision menyimpan atau memproses data wajah tanpa izin, maka yang terancam bukan hanya privasi, tetapi juga hak dasar manusia atas identitasnya.

### Regulasi dan Standar Internasional

Privasi dalam computer vision juga harus dipahami dalam kerangka hukum dan regulasi. Di Eropa, General Data Protection Regulation (GDPR) menetapkan aturan ketat terkait pengumpulan, penyimpanan, dan pemrosesan data pribadi, termasuk data citra. Setiap organisasi yang melanggar dapat dikenai denda besar, bahkan jika pelanggaran terjadi secara tidak sengaja. Di Amerika Serikat, regulasi lebih terfragmentasi, dengan beberapa negara bagian seperti California menerapkan California Consumer Privacy Act (CCPA). Sementara di Indonesia, UU ITE dan regulasi terkait data pribadi masih berkembang menuju standar internasional.

Implikasinya bagi peneliti vision adalah jelas: setiap eksperimen harus memperhatikan aspek legal dan regulasi di wilayahnya. Sebuah sistem yang akurat secara teknis namun melanggar regulasi privasi pada akhirnya tidak dapat diadopsi secara luas.

### Privacy-Preserving Vision

Sebagai respons terhadap tantangan privasi, muncul paradigma baru yaitu privacy-preserving vision. Pendekatan ini berusaha menyeimbangkan kebutuhan analisis visual dengan perlindungan data individu. Beberapa teknik yang relevan antara lain:

1. Differential Privacy: menambahkan noise terkontrol pada data atau hasil prediksi agar informasi individu tidak dapat diidentifikasi secara spesifik.
2. Federated Learning: memungkinkan pelatihan model dilakukan langsung pada perangkat pengguna, sehingga data mentah tidak pernah dikirim ke server pusat.
3. Homomorphic Encryption: memungkinkan pemrosesan data dalam keadaan terenkripsi, sehingga pihak ketiga tidak pernah melihat data asli.

Pendekatan-pendekatan ini menegaskan bahwa privasi tidak harus dikorbankan demi kemajuan teknologi. Justru, inovasi vision masa depan perlu dibangun di atas fondasi bahwa perlindungan data adalah hak asasi, bukan sekadar opsi tambahan.

### Dilema Praktis dalam Privasi

Meskipun berbagai teknik dan regulasi telah hadir, dilema privasi tetap muncul dalam praktik. Misalnya, dalam riset kesehatan, citra pasien (MRI, X-ray, foto dermatoskopi) sangat berharga untuk pelatihan model, tetapi sekaligus sangat sensitif. Bagaimana cara menyeimbangkan kebutuhan ilmiah untuk berbagi dataset dengan kewajiban menjaga kerahasiaan pasien?

Demikian pula dalam sektor pertanian cerdas, foto lahan pertanian bisa dianggap data biasa, tetapi bagi petani kecil, data tersebut dapat mencerminkan pola produksi dan tingkat kerentanan ekonomi mereka. Privasi, dalam konteks ini, tidak hanya tentang individu, tetapi juga komunitas dan kelompok sosial.

## 15.4 Framework Etis untuk Riset Vision

Membangun algoritma computer vision bukan hanya soal akurasi dan efisiensi, tetapi juga bagaimana hasil penelitian tersebut memberi manfaat sekaligus menghindarkan kerugian sosial. Dalam konteks ini, diperlukan framework etis yang dapat menjadi panduan praktis bagi peneliti, terutama di tingkat doktoral, untuk menavigasi kompleksitas riset vision yang kian berhubungan dengan kehidupan manusia sehari-hari.

Framework ini berfungsi sebagai kompas moral dan metodologis, memastikan bahwa setiap tahapan—mulai dari pengumpulan data, pelatihan model, evaluasi, hingga deployment—dilakukan dengan mempertimbangkan aspek keadilan, akuntabilitas, transparansi, dan tanggung jawab sosial.

**Prinsip FATE: Fairness, Accountability, Transparency, Ethics**

Salah satu pendekatan yang sering dirujuk adalah kerangka FATE (Fairness, Accountability, Transparency, Ethics). Kerangka ini menekankan empat pilar utama:

1. Fairness (Keadilan)
   Model vision harus dilatih pada data yang beragam dan representatif agar tidak memunculkan bias diskriminatif. Misalnya, sistem diagnosis kulit harus mampu mendeteksi kelainan pada berbagai warna kulit, bukan hanya kulit terang.
2. Accountability (Akuntabilitas)
   Peneliti dan institusi yang mengembangkan algoritma vision harus bertanggung jawab atas dampak yang ditimbulkan. Ini mencakup mekanisme audit, publikasi metodologi yang terbuka, serta pengakuan terhadap keterbatasan model.
3. Transparency (Transparansi)
   Masyarakat dan pemangku kepentingan berhak mengetahui bagaimana model bekerja. Interpretabilitas, dokumentasi dataset, dan keterbukaan arsitektur menjadi syarat penting agar hasil riset dapat dipercaya.
4. Ethics (Etika)
   Aspek ini menegaskan bahwa riset vision harus mempertimbangkan konsekuensi jangka panjang terhadap manusia dan lingkungan. Penggunaan teknologi untuk tujuan yang merugikan, meskipun mungkin menguntungkan secara teknis, harus dihindari.

**Checklist Etis untuk Peneliti Vision**

Agar prinsip FATE dapat diterapkan secara praktis, peneliti perlu memiliki checklist etis yang dapat dijadikan pedoman dalam setiap proyek:

1. Dataset: Apakah dataset yang digunakan sudah mencerminkan keragaman populasi? Apakah terdapat risiko pelanggaran privasi?
2. Model: Apakah model diuji pada berbagai skenario ekstrem (cahaya rendah, noise tinggi, data tidak seimbang)?
3. Deployment: Apakah ada potensi penyalahgunaan model ketika diimplementasikan di luar konteks akademik?
4. Komunikasi: Apakah keterbatasan model dijelaskan secara terbuka dalam publikasi atau dokumentasi?

Checklist ini tidak dimaksudkan sebagai dokumen birokratis, melainkan sebagai sarana refleksi bagi peneliti agar tidak terjebak dalam euforia teknis tanpa mempertimbangkan implikasi sosial.

**Studi Kasus: Penerapan Framework Etis**

Untuk memperjelas, mari kita pertimbangkan dua studi kasus:

Face Recognition di Ruang Publik

- Fairness: Apakah model bekerja adil pada semua etnis?
- Accountability: Siapa yang bertanggung jawab jika terjadi salah tangkap akibat kesalahan identifikasi?
- Transparency: Apakah masyarakat diberi tahu tentang penggunaan teknologi ini?
- Ethics: Apakah manfaat keamanan lebih besar daripada potensi pelanggaran privasi?

### Deteksi Penyakit Tanaman

- Fairness: Apakah dataset mencakup berbagai varietas tanaman dan kondisi geografis?
- Accountability: Bagaimana jika petani salah mengambil keputusan karena kesalahan deteksi
- Transparency: Apakah hasil model dapat dijelaskan kepada petani dengan bahasa sederhana?
- Ethics: Apakah teknologi ini membantu memberdayakan petani kecil atau justru membuat mereka bergantung pada pihak tertentu?

Studi kasus ini menunjukkan bahwa framework etis tidak hanya berlaku pada isu-isu sensitif seperti biometrik, tetapi juga pada bidang lain yang terlihat lebih netral, seperti pertanian.

## Kolaborasi Lintas Disiplin

Etika dalam computer vision tidak dapat dipikul sendirian oleh ilmuwan komputer. Dibutuhkan kolaborasi lintas disiplin dengan ahli hukum, filsuf, sosiolog, hingga pembuat kebijakan. Dengan demikian, penelitian vision tidak hanya unggul secara teknis, tetapi juga selaras dengan norma sosial, hukum, dan budaya yang berlaku.

# BAB XVI
# Tantangan dan Arah Masa Depan dalam Computer Vision

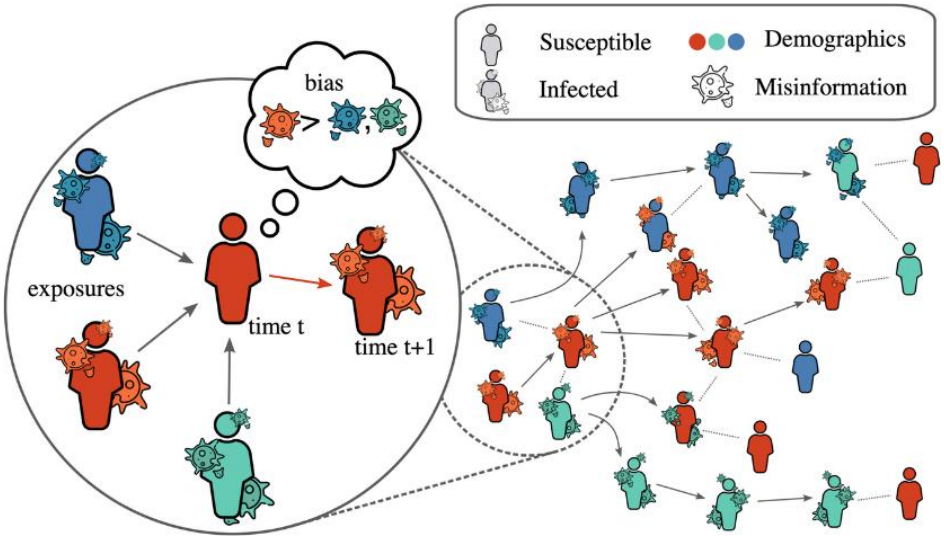
## 16.1 Tantangan Terkini dalam Pengembangan Algoritma Vision

Computer Vision (CV) telah menciptakan dalam berbagai bidang septi kesehatan, transportasi, pertanian, dan keamanan, tantangan besar masih harus dihadapi dalam pengembangan algoritma vision yang lebih tangguh dan berkelanjuan. Tantangan ini hadir tidak hanya dalam aspek teknis; namun juga menyangkut aspek etis, privasi, dan sosial yang kian kompleks.

**a. Keterbatasan Data dan Masalah Bias**

Algoritma deep learning, termasuk dalam bidang vision, membutuhkan data yang besar. Kualitas dan distribusi yang digunakan sering kali tidak merepresentasikan seluruh populasi atau lingkungan global secara adil. Hal ini menimbuka algoritmik.

Dataset Contohnya pengenalan wajah yang mayoritas berasal dari populasi kulit putih atau lingkungan urban menyebabkan penurunan akurasi ketika sistem digunakan di negara-negara berkembang. Penelitian oleh MIT Media Lab menunjukkan model pengenalan wajah komersial memiliki tingkat kesalahan lebih dari 30% untuk wajah perempuan berkulit, dibandingkan dengan hanya 1% untuk laki-liaki kulit putih




Gambar 28.Visualisasi: Akurasi Model Dampak Bias Data

Gambar 28 menampilkan sebuah model visualisasi yang menjelaskan bagaimana bias data dapat memengaruhi akurasi suatu sistem, khususnya dalam konteks penyebaran informasi maupun misinformasi. Dalam ilustrasi ini, individu digambarkan dengan warna dan simbol berbeda untuk merepresentasikan status mereka, seperti rentan (susceptible), terinfeksi (infected), faktor demografis, hingga keterpaparan terhadap misinformasi. Alur pergerakan dari waktu ke waktu menunjukkan bagaimana individu yang awalnya netral atau rentan dapat berubah status setelah terpapar informasi tertentu.

Komponen utama yang ditunjukkan adalah adanya bias pada tahap pengambilan atau pemrosesan data. Bias ini dapat muncul karena distribusi data yang tidak seimbang, representasi demografis yang timpang, atau pola paparan informasi yang tidak merata. Ketika

bias tersebut masuk dalam proses analisis model, dampaknya adalah interpretasi yang menyimpang dari kondisi sebenarnya. Hal ini terlihat dari perbedaan keadaan pada waktu t dan t+1, di mana sebaran individu yang terpengaruh menjadi lebih luas, tidak hanya karena paparan informasi tetapi juga akibat bias yang memperkuat proses penyebaran.

Visualisasi ini juga menegaskan bahwa akurasi model tidak semata-mata ditentukan oleh algoritma, melainkan sangat bergantung pada kualitas data yang digunakan. Jika data awal sudah dipengaruhi oleh bias, maka hasil model cenderung memberikan kesimpulan yang kurang tepat atau bahkan menyesatkan. Dengan kata lain, model yang tampak memiliki tingkat akurasi tinggi sekalipun, sebenarnya bisa saja gagal merepresentasikan realitas apabila bias data tidak dikendalikan

**b. Generalisasi dan Overfitting**
Model Computer Vision sering kali mengalami dengan terhadap data pelatihan dan gagal beradaptasi dengan kondisi baru septi pencahayaan rendah, rotasi, atau noise. Tantangan ini singat penting terutama dalam aplikasi lapangan septi pertanian presisi dan kendaraan otonom. Mengatasi hal ini, pendekatan septi data augmentation, transfer learning, dan penggunaan data sintetis mulai diperkenalkan. Namun, tetap diperlukan mendalam untuk men Benjamin kemampuan generalisasi yang konstens.

**c. Kompleksitas dan Konsumsi Energi**
Algoritma modern seperti Vision Transformer (ViT) dan model multi-modal memerluktu komputasi tinggi. Proses pelatihan model besar konsumsi energi setara ribuan jam listrik rumah tangga.

Tabel 16.1. Konsumsi Energi Beberapa Model Vision

| Model | Dataset | Energi Training (kWh) |
|---|---|---|
| ResNet-50 | ImageNet | 256 |
| EfficientNet-B7 | ImageNet | 624 |
| Vision Transformer (ViT-L) | JFT-300M | >1500 |

Tabel 16.1 ini menyajikan perbandingan konsumsi energi yang diperlukan dalam proses pelatihan tiga model computer vision terkemuka, yaitu ResNet-50, EfficientNet-B7, dan Vision Transformer (ViT-L). Data ini mengungkapkan dampak lingkungan dari pengembangan model AI yang semakin kompleks, sekaligus menyoroti trade-off antara kinerja model dan keberlanjutan lingkungan.

ResNet-50, yang dilatih pada dataset ImageNet dengan 1,2 juta gambar, mengonsumsi energi sekitar 256 kWh. Sebagai arsitektur yang relatif efisien namun powerful, ResNet-50 menjadi standar dalam banyak aplikasi praktis karena mencapai keseimbangan antara akurasi dan konsumsi energi yang moderat.

EfficientNet-B7, juga dilatih pada ImageNet, memerlukan energi lebih besar, yaitu 624 kWh. Meskipun dirancang untuk optimasi efisiensi melalui scaling compound, peningkatan

kedalaman dan resolusi input pada versi B7 menyebabkan kebutuhan komputasi—dan consequently konsumsi energi—yang lebih tinggi dibandingkan pendahulunya.

Vision Transformer (ViT-L), yang dilatih pada dataset massive JFT-300M (300 juta gambar), mengonsumsi energi sangat signifikan, yakni lebih dari 1500 kWh. Arsitektur transformer yang memanfaatkan mekanisme perhatian (attention) global memang mencapai akurasi state-of-the-art, tetapi membutuhkan kapasitas komputasi yang sangat besar, sehingga meninggalkan jejak karbon yang tidak trivial.

Tabel 13.1 menggarisbawahi tantangan kritikal dalam pengembangan AI modern: bagaimana menyeimbangkan pencapaian akurasi tertinggi dengan pertimbangan keberlanjutan. Konsumsi energi yang tinggi tidak hanya mempengaruhi biaya operasional tetapi juga memiliki implikasi lingkungan yang nyata, mendorong perlunya inovasi dalam teknik pelatihan yang lebih efisien, penggunaan hardware hemat energi, dan pertimbangan yang lebih cermat dalam memilih arsitektur model sesuai dengan kebutuhan aplikasi nyata.

**d. Privasi dan Etika Penggunaan**
Dalam ranah pengawasan publik dan deteksi wajah, CV rentan menimbulkan konflik dengan prinsip-prinsip privasi dan HAM. Dapat dilihat pada pelarangan penggunaan facial recognition di beberapa kota di AS karena dikhawatirkan menimbuntu pengawasan massal tanpa persetujuan.
Pendekatan teknologi septi differential privacy, enkripsi homomorfik, dan federated learning menjadi penting untuk mengentasional model vision yang privasi-sentris.
.
**e. Robustness terhadap Lingkungan Dunia Nyata**
Sistem CV harus visual seperti blur, occlusion, atau pencahayaan ekstrem. Ini masih besar, robustness ini. Pengujian model dalam kondisi ideal tidak Benjamin performa baik saat menghadapi kondisi tidak terkendiri.

## 16.2 Arah Masa Depan dan Inovasi Potensial
Perkembangan dalam Computer Vision sangat menjanjikan, terutama dengan munculnya teknologi baru yang menggabungkan efisiensi, fleksibilitas, dan kemampuan adaptif.

**a. Edge AI dan Vision on the Edge**
Alih-alih menghandalkan cloud, tren baru mendorong pemrosesan data visual langsung di perangkat septi drone, kamera IoT, atau sensor pertanian. Hal ini memungkkan sistem bekerja tanpa real-time tanpa latensi dan dengan tingkat privasi lebih baik.
Contoh platform: ESP32-CAM, Google Coral, NVIDIA Jetson.
Gambar: Konsep Pemrosesan Visual di Edge

**b. Self-Supervised Learning (SSL)**
SSL memungkinkan pelatihan model vision tanpa memerlukan label eksplisit, melainkan melalui prediksi bagian gambar dari bagian lain.
Model seperti SimCLR, BYOL, dan DINO sudah menunjukkan hasil luar biasa dengan efisiensi tinggi dalam banyak domain. Dengan SSL, tantangan data berlabel mahal dan bias bisa dikurangi.

**c. Multimodal Learning**
Penggabungan antara gambar, teks, audio, dan data sensorik lainnya (multimodal) memungkinkan CV memiliki pemahaman kontekstual lebih dalam. Sistem seperti CLIP (OpenAI) dan Flamingo (DeepMind) menjadi pionir dalam pendekatan ini.
Gambar: Arsitektur Multimodal Learning

**d. Explainable Vision**
Dalam domain kritikal seperti medis atau kendaraan otonom, kemampuan menjelaskan keputusan model menjadi sangat krusial. Teknik visualisasi seperti Grad-CAM, Saliency Maps, serta integrasi metode SHAP dan LIME akan menjadi fitur wajib model masa depan.

**e. Zero-shot dan Few-shot Learning**
Model seperti CLIP dan BLIP memungkinkan sistem memahami objek baru hanya dengan sedikit atau bahkan tanpa contoh pelatihan
(zero/few-shot). Hal ini memperluas adaptasi model pada lingkungan baru dengan efisiensi tinggi.

**16.3 Kolaborasi Lintas Disiplin dan Keberlanjutan**
Keberhasilan vision masa depan akan sangat ditentukan oleh kolaborasi dengan berbagai bidang ilmu:

- Pertanian: pemantauan tanaman dan deteksi penyakit berbasis citra
- Kesehatan: interpretasi citra radiologi dan patologi
- Ekologi dan Lingkungan: pemantauan satelit untuk deforestasi
- Sosioteknologi: pengembangan AI yang etis dan inklusif

Ke depan, konsep green computing akan menjadi standar moral dan teknis. Peneliti dan pengembang didorong untuk menciptakan model
yang efisien secara energi dan dapat dijalankan secara lokal, demi keberlanjutan lingkungan dan keterjangkauan teknologi.

Tabel 16.2 Aspek Penting Komputer Vision Masa Depan

| Aspek | Tantangan | Arah Solusi |
|---|---|---|
| Privasi | Data wajah tersebar | Federated learning, Enkripsi |
| Efisiensi Energi | GPU mahal | Edge AI, Model Lightweight |
| Generalisasi | Kondisi tak terduga | Data sintetik, Augmentasi |
| Etika dan Bias | Ketidakadilan algoritmik | Dataset inklusif, XAI |

Pada table 16.2 menjelaskan perkembangan computer vision di masa depan tidak hanya ditentukan oleh peningkatan akurasi model, tetapi juga oleh kemampuan mengatasi sejumlah tantangan kritis yang meliputi aspek privasi, efisiensi, energi, generalisasi, serta etika dan bias. Tabel ini merangkum tantangan utama tersebut beserta arah solusi yang sedang dikembangkan oleh para peneliti dan praktisi.

Dalam aspek privasi, tantangan utama terletak pada perlindungan data sensitif seperti citra wajah dan informasi personal lainnya yang sering digunakan dalam pelatihan model. Solusi yang menjanjikan termasuk penerapan federated learning, yang memungkinkan pelatihan

model tanpa sentralisasi data, serta teknik enkripsi lanjutan untuk memastikan bahwa data tetap terlindungi selama proses pelatihan dan inferensi.

Tantangan efisiensi komputasi dihadapi melalui pengembangan edge AI, yang memindahkan pemrosesan data ke perangkat tepi (edge devices) sehingga mengurangi ketergantungan pada cloud dan menurunkan latensi. Selain itu, desain model ringan (lightweight models) yang mempertahankan akurasi tinggi dengan kompleksitas komputasi minimal menjadi fokus utama penelitian.

Isu konsumsi energi yang tinggi, terutama akibat penggunaan GPU yang mahal dan boros daya, diatasi dengan rekayasa model hemat energi yang dioptimalkan untuk perangkat dengan sumber daya terbatas, serta pemanfaatan perangkat keras khusus yang dirancang untuk efisiensi energi.

Tantangan generalisasi model dalam menghadapi kondisi lingkungan yang tak terduga (seperti perubahan cahaya, cuaca, atau sudut pengambilan gambar) disiasati melalui penggunaan data sintetik yang menghasilkan variasi data pelatihan yang lebih luas, serta teknik augmentasi data canggih yang memperkaya keragaman data latih tanpa perlu pengumpulan data baru yang mahal.

Akhirnya, tantangan etika dan bias algoritmik yang dapat menyebabkan ketidakadilan dalam output model (misalnya, bias terhadap kelompok demografi tertentu) diatasi melalui penyusunan dataset yang lebih
inklusif dan representatif, serta pengembangan AI yang dapat dijelaskan (XAI - Explainable AI) untuk meningkatkan transparansi
dan akuntabilitas pengambilan keputusan oleh model. Dengan mengatasi tantangan-tantangan ini, computer vision masa depan tidak
hanya akan menjadi lebih canggih, tetapi juga lebih bertanggung jawab, berkelanjutan, dan bermanfaat bagi masyarakat luas.

## 16.4 Green AI dan Explainable AI dalam Computer Vision

### 16.4.1 Green AI: Efisiensi Energi dalam Era Model Skala Besar

Salah satu isu kritis dalam pengembangan computer vision modern adalah biaya energi yang sangat besar pada saat melatih (training) dan menerapkan (deployment) model. Model seperti Vision Transformer (ViT), Swin Transformer, atau foundation models (CLIP, SAM, DINO) memiliki miliaran parameter, dan proses latihannya dapat menghasilkan jejak karbon setara dengan konsumsi listrik rumah tangga selama bertahun-tahun. Penelitian Strubell et al. (2019)

menunjukkan bahwa melatih satu model NLP skala besar dapat melepaskan lebih dari 300.000 kg $CO_2$, dan angka ini diproyeksikan serupa atau bahkan lebih tinggi dalam computer vision.

Isu ini memunculkan gerakan Green AI, yaitu paradigma riset yang menekankan:

1. Efisiensi Energi dan Komputasi: Menurunkan kebutuhan daya dengan algoritma yang lebih hemat.
2. Efektivitas Sumber Daya: Mengurangi ketergantungan pada data dan infrastruktur raksasa agar riset lebih inklusif.
3. Keberlanjutan Lingkungan: Meminimalkan kontribusi terhadap jejak karbon global.

**Pendekatan Teknis Green AI**

- Model Compression (pruning, quantization, weight sharing): Mengurangi kompleksitas parameter tanpa mengorbankan performa signifikan. Misalnya, Tiny-YOLO dapat berjalan di ESP32-CAM dengan daya terbatas.
- Knowledge Distillation: Model besar (teacher) mentransfer pengetahuan ke model kecil (student) sehingga akurasi tetap terjaga dengan energi jauh lebih rendah.
- Neural Architecture Search (NAS) berbasis efisiensi: Bukan hanya mencari model akurat, tetapi juga yang energy-aware. EfficientNet merupakan contoh arsitektur yang dioptimalkan berdasarkan trade-off antara FLOPs, akurasi, dan konsumsi energi.
- Edge Computing dan Federated Learning: Memindahkan sebagian komputasi ke perangkat tepi agar tidak semua proses dilakukan di cloud. Selain hemat energi, pendekatan ini juga mengurangi latensi dan risiko privasi.

**Dampak Ilmiah dan Etis**

Green AI menantang paradigma lama yang menekankan bigger is better. Alih-alih mengejar model dengan parameter tak terbatas, komunitas riset diarahkan untuk berpikir kritis tentang rasio manfaat terhadap biaya energi. Hal ini membuka diskursus etis: apakah pantas melatih model dengan konsumsi energi setara ribuan ton $CO_2$ hanya untuk meningkatkan akurasi 0,1% pada benchmark?

### 16.4.2 Explainable AI (XAI): Transparansi dan Akuntabilitas Model Vision

Jika Green AI berfokus pada sisi efisiensi, maka Explainable AI (XAI) menyoroti sisi kepercayaan dan interpretabilitas. Model computer vision modern (CNN, ViT, foundation models) bekerja sebagai black box: mereka dapat mencapai akurasi tinggi, tetapi sulit dipahami alasan di balik prediksi yang dihasilkan.

Dalam konteks aplikasi sensitif seperti diagnosis medis, sistem keamanan publik, atau deteksi ancaman militer keterbatasan transparansi ini sangat berisiko. Kesalahan prediksi tidak hanya menurunkan akurasi, tetapi juga berpotensi mengancam nyawa atau melahirkan bias diskriminatif.

**Teknik-teknik XAI dalam Vision**

- Saliency Maps: Memvisualisasikan piksel atau area gambar yang paling berpengaruh terhadap keputusan model.

- Grad-CAM (Gradient-weighted Class Activation Mapping): Menghasilkan heatmap yang menyoroti area kritis yang memicu keputusan klasifikasi/deteksi.
- Occlusion Analysis: Menghapus sebagian citra untuk melihat perubahan hasil prediksi, sehingga diketahui bagian mana yang penting.
- LIME dan SHAP: Menjelaskan kontribusi fitur pada tingkat lokal dan global. Walaupun awalnya populer di NLP/tabular data, adaptasi untuk computer vision berkembang pesat.

**Tantangan XAI**

- Skalabilitas: Model multimodal (gambar + teks) membuat interpretasi lebih kompleks.
- Trade-off antara interpretabilitas dan akurasi: Model yang terlalu sederhana mudah dijelaskan, tetapi kurang kuat untuk data dunia nyata.
- Potensi "false sense of trust": Visualisasi XAI bisa terlihat meyakinkan, padahal tidak selalu konsisten dengan mekanisme internal model.

Signifikansi Etis

XAI berperan dalam menciptakan sistem vision yang adil, transparan, dan dapat diaudit. Hal ini penting agar teknologi tidak hanya berfungsi secara teknis, tetapi juga dapat diterima masyarakat luas. Misalnya, ketika sebuah sistem CCTV berbasis vision menandai seseorang sebagai "tersangka", masyarakat berhak tahu alasan spesifik di balik keputusan itu.

### 16.4.3 Perbandingan Green AI dan XAI dalam Computer Vision

**Tabel 16.3 Perbandingan Green AI dan XAI dalam Computer Vision**

| Aspek | Green AI (Efisiensi Energi) | Explainable AI (Transparansi) |
|---|---|---|
| Fokus | Mengurangi biaya energi, jejak karbon, dan sumber daya komputasi | Membuka mekanisme internal model agar dapat dijelaskan |
| Strategi Teknis | Compression, distillation, efficient NAS, edge computing | Saliency maps, Grad-CAM, LIME, SHAP, occlusion analysis |
| Dampak Ilmiah | Riset inklusif, keberlanjutan, ramah lingkungan | Akuntabilitas, fairness, mengurangi bias |
| Tantangan | Trade-off efisiensi vs akurasi, keterbatasan hardware kecil | Kompleksitas interpretasi, risiko interpretasi semu |
| Relevansi Industri 4.0 | Mendukung IoT, edge devices, smart farming, transportasi cerdas | Penting pada aplikasi kritis: medis, keamanan, hukum |

Tabel 7.4.3 menyajikan perbandingan antara dua isu kontemporer yang sama-sama krusial dalam perkembangan computer vision, yakni Green AI dan Explainable AI (XAI). Meskipun

keduanya memiliki fokus yang berbeda, keduanya berkontribusi pada arah pengembangan teknologi yang lebih berkelanjutan, transparan, dan bertanggung jawab.

**Fokus**
Green AI berfokus pada efisiensi energi dan sumber daya komputasi, yaitu bagaimana merancang dan melatih model agar tidak boros energi, tidak membutuhkan perangkat keras mahal, dan lebih ramah lingkungan. Sebaliknya, XAI berfokus pada transparansi algoritma. Ia menjawab pertanyaan penting: "mengapa model mengambil keputusan tertentu?" Hal ini sangat penting untuk membangun kepercayaan, terutama dalam aplikasi medis, hukum, dan keamanan publik.

**Strategi Teknis**
Pada Green AI, strategi yang diterapkan bersifat teknis untuk meringankan model tanpa mengurangi performa secara signifikan. Contohnya adalah compression (pemangkasan parameter), knowledge distillation (mentransfer pengetahuan dari model besar ke model kecil), serta penggunaan arsitektur efisien seperti MobileNet atau EfficientNet. Pada XAI, strategi teknis diarahkan pada membuka kotak hitam (black box) model. Teknik seperti saliency maps dan Grad-CAM memvisualisasikan area citra yang paling berpengaruh terhadap keputusan model, sementara LIME atau SHAP memberikan penjelasan berbasis kontribusi fitur.

**Dampak Ilmiah**
Green AI membawa dampak dalam bentuk riset yang lebih inklusif dan berkelanjutan. Dengan model hemat energi, riset tidak hanya bisa dilakukan oleh laboratorium besar, tetapi juga oleh universitas kecil atau peneliti independen. XAI, di sisi lain, berkontribusi pada akuntabilitas dan keadilan. Dengan adanya interpretasi, model bisa diaudit, bias bisa diidentifikasi, dan pengguna akhir lebih percaya pada hasil sistem vision.

**Tantangan**
Keduanya menghadapi tantangan yang tidak ringan. Green AI menghadapi trade-off antara efisiensi dan akurasi: semakin kecil model, semakin mungkin terjadi penurunan performa. XAI menghadapi tantangan dalam hal kompleksitas interpretasi, terutama untuk model multimodal yang menggabungkan teks, gambar, dan konteks. Selain itu, ada risiko "ilusi transparansi", di mana visualisasi penjelasan terlihat meyakinkan tetapi tidak benar-benar mencerminkan mekanisme internal model.

**Relevansi dalam Industri 4.0**
Dalam konteks Industri 4.0, Green AI sangat relevan untuk perangkat IoT dan edge computing. Misalnya, deteksi penyakit daun menggunakan kamera di lahan pertanian memerlukan model hemat energi agar bisa berjalan di perangkat terbatas seperti ESP32-CAM. Sebaliknya, XAI sangat relevan untuk aplikasi kritis yang menuntut kepercayaan tinggi, seperti sistem radiologi berbasis AI atau kamera cerdas untuk keamanan kota.

# BAB XVII
# Kesimpulan dan Rekomendasi

## 17.1 Ringkasan Perjalanan Algoritma Vision

Dalam beberapa dekade terakhir, Computer Vision (CV) telah berkembang pesat dari pendekatan berbasis algoritma sederhana hingga teknologi kompleks berbasis deep learning dan artificial intelligence. Buku ajar ini telah menelusuri perjalanan panjang algoritma vision dari masa ke masa:

- Bab 1 memperkenalkan definisi, sejarah, dan latar belakang teknologi Computer Vision.
- Bab 2 membahas berbagai komponen utama dalam sistem CV, dari akuisisi hingga interpretasi citra.
- Bab 3 mengupas secara mendalam tentang perbandingan algoritma vision klasik dan modern.
- Bab 4 mengeksplorasi integrasi CV dengan teknologi terkini seperti IoT dan Cloud Computing.
- Bab 5 hingga Bab 7 menelusuri aplikasi dunia nyata, tantangan terkini, serta arah masa depan pengembangan teknologi vision.

Perjalanan ini menunjukkan bahwa CV bukan hanya masalah teknis, melainkan juga menantang kita untuk berpikir secara lintas disiplin, mempertimbangkan etika, dan mengejar efisiensi serta keadilan dalam penerapannya.

Tabel 17.1 Evolusi Teknologi Vision dari Masa ke Masa

| Era | Ciri Khas | Teknologi Dominan |
|---|---|---|
| Sebelum 2000 | Algoritma klasik | Thresholding, Edge Detection |
| 2000–2010 | Citra Digital dan ML awal | SVM, HOG, Haar Features |
| 2010–2020 | Deep Learning dan CNN | AlexNet, ResNet, YOLO |
| 2020–sekarang | Edge AI, XAI, Multimodal AI | ViT, CLIP, Federated Learning |

Pada tabel 17.1 bercerita tentang perkembangan bidang computer vision dapat dipetakan melalui beberapa era yang menandai perubahan paradigma baik dalam pendekatan algoritmik maupun kemampuan teknologi yang mendasarinya. Setiap era tidak hanya mencerminkan kemajuan teknis, tetapi juga perluasan aplikasi dan tantangan baru yang dihadapi oleh para peneliti dan praktisi.

Pada era sebelum tahun 2000, computer vision didominasi oleh algoritma klasik yang mengandalkan teknik-teknik pemrosesan sinyal dan citra dasar. Pendekatan ini bertumpu pada metode seperti thresholding (pemisahan objek berdasarkan nilai intensitas pixel) dan edge detection (pendeteksian tepi objek menggunakan operator seperti Sobel atau Canny). Meskipun efektif untuk tugas-tugas terbatas dalam kondisi terkendali, metode ini sangat sensitif terhadap variasi lingkungan seperti perubahan pencahayaan dan clutter latar belakang.

Era 2000–2010 ditandai dengan adopsi luas citra digital dan pendekatan machine learning awal. Pada periode ini, fitur engineering menjadi kunci kesuksesan dengan metode seperti Haar features untuk deteksi objek dan HOG (Histogram of Oriented Gradients) untuk deskripsi bentuk objek. Algoritma SVM (Support Vector Machine) menjadi pilihan utama untuk

klasifikasi, memberikan kemampuan generalisasi yang lebih baik dibandingkan metode sebelumnya. Era ini meletakkan fondasi untuk sistem vision yang lebih adaptif.

Revolusi terjadi pada era 2010–2020 dengan kemunculan deep learning dan CNN (Convolutional Neural Networks). AlexNet (2012) menandai titik balik dengan kinerja superior dalam kompetisi ImageNet, diikuti oleh pengembangan arsitektur yang semakin dalam dan kompleks seperti ResNet yang mengatasi masalah vanishing gradient. Pada periode ini juga lahir algoritma deteksi objek real-time seperti YOLO (You Only Look Once) yang mengubah paradigma dari pendekatan berbasis region ke pendekatan single-shot.

Era 2020-sekarang menandai fase matang computer vision dengan fokus pada Edge AI (deploy model pada perangkat edge), XAI (Explainable AI) untuk meningkatkan transparansi, dan pendekatan multimodal yang menggabungkan informasi dari berbagai sensor. Arsitektur Vision Transformer (ViT) menantang dominasi CNN dengan mekanisme attention yang lebih global. Model seperti CLIP menunjukkan kemampuan memahami gambar dalam konteks bahasa alami, sementara Federated Learning muncul sebagai solusi privasi dengan melatih model secara terdistribusi tanpa sentralisasi data.

Perkembangan ini menunjukkan evolusi dari pendekatan yang terbatas dan hand-crafted menuju sistem yang semakin cerdas, terdistribusi, dan dapat beradaptasi dengan kebutuhan real-world yang kompleks, sekaligus memperhatikan aspek etika dan keberlanjutan.

### 17.2 Implikasi dan Signifikansi dalam Dunia Nyata

Penerapan algoritma vision tidak lagi terbatas pada lingkungan akademik atau industri teknologi besar. Kini, algoritma CV telah masuk dalam berbagai sektor masyarakat:

a. Kesehatan
- Deteksi dini kanker menggunakan analisis citra radiologi
- Pelacakan pertumbuhan tumor melalui model segmentasi

b. Pertanian
- Identifikasi penyakit tanaman secara real-time menggunakan drone dan kamera IoT
- Penghitungan hasil panen menggunakan CV pada citra udara

Gambar: CV dalam Deteksi Daun Terserang Penyakit (contoh ilustrasi)

c. Transportasi dan Smart City
- Kendaraan otonom mengandalkan CV untuk deteksi rambu, jalur, dan objek
- CCTV cerdas untuk pengawasan lalu lintas dan keamanan publik

d. Keamanan dan Forensik
- Penggunaan facial recognition dalam investigasi
- Deteksi deepfake dengan algoritma spatio-temporal

Dampak ini menunjukkan bahwa algoritma vision telah menjadi kekuatan pengubah (transformative force) dalam sistem sosial, lingkungan, dan ekonomi global.

### 17.3 Rekomendasi Pengembangan di Masa Depan

Berdasarkan analisis yang telah disampaikan, beberapa rekomendasi strategis dapat dirumuskan untuk arah pengembangan algoritma vision:

**a. Peningkatan Efisiensi dan Ringannya Model**

Model seperti MobileNet, EfficientNet, dan Tiny-YOLO perlu dikembangkan lebih lanjut agar dapat dijalankan di perangkat edge dengan keterbatasan memori dan daya.

Tabel 8.2 Perbandingan Model Lightweight Computer Vision (2024)

| Model | Parameter (Juta) | FPS (Jetson Nano) | Akurasi Top-1 (%) |
|---|---|---|---|
| MobileNetV3-Small | 2.5 | 52 | 67.5 |
| EfficientNet-Lite0 | 4.7 | 38 | 75.1 |
| Tiny-YOLOv7 | 6.2 | 28 | 68.9 |
| NanoDet-Plus | 1.1 | 65 | 63.7 |
| YOLO-NAS-Small | 8.3 | 22 | 78.2 |
| GhostNetV2 | 3.2 | 45 | 73.4 |

Table 8.2 Menjelaskan perkembangan model computer vision yang ringan dan efisien telah mengalami kemajuan signifikan dalam beberapa tahun terakhir, khususnya untuk aplikasi pada perangkat dengan kemampuan komputasi terbatas seperti NVIDIA Jetson Nano. Tabel perbandingan ini menyajikan enam model terkini yang mewakili trade-off antara kompleksitas parameter, kecepatan pemrosesan, dan tingkat akurasi.

MobileNetV3-Small, dengan hanya 2.5 juta parameter, mampu mencapai kecepatan 52 frame per second (FPS) yang menjadikannya salah satu model tercepat, meskipun dengan akurasi yang relatif moderat sekitar 67.5%. Di sisi lain, YOLO-NAS-Small yang memiliki parameter lebih banyak (8.3 juta) menunjukkan akurasi tertinggi hingga 78.2%, namun dengan kecepatan pemrosesan yang lebih rendah yaitu 22 FPS.

Model-model terkini seperti NanoDet-Plus dan GhostNetV2 menawarkan kombinasi yang menarik antara efisiensi dan kinerja. NanoDet-Plus menjadi standout dengan kecepatan 65 FPS berkat arsitektur yang sangat dioptimalkan, sementara GhostNetV2 mempertahankan akurasi 73.4% dengan parameter yang relatif sedikit.

EfficientNet-Lite0 dan Tiny-YOLOv7 berada di tengah spektrum ini, menawarkan keseimbangan antara kecepatan dan akurasi yang sesuai untuk berbagai aplikasi praktis. Perbandingan ini menggarisbawahi bahwa tidak ada model "terbaik" yang universal, melainkan pilihan yang harus disesuaikan dengan kebutuhan spesifik aplikasi, apakah prioritasnya adalah kecepatan, akurasi, atau efisiensi resource

**b. Peningkatan Keadilan dan Reduksi Bias**

Peneliti dan pengembang wajib meninjau ulang dataset dan pipeline pelatihan agar model dapat berperforma secara adil di berbagai kelompok demografis.
Upaya seperti data balancing, augmentasi inklusif, dan transfer learning lintas domain bisa menjadi solusi.

**c. Penguatan Explainability dan Trustworthy AI**
XAI (Explainable AI) harus menjadi fitur bawaan dalam pengembangan algoritma vision, terutama untuk sektor berisiko tinggi seperti kesehatan, hukum, dan transportasi.
Integrasi Grad-CAM, LIME, dan SHAP ke dalam pipeline visualisasi menjadi sangat krusial.

**d. Pendidikan dan Literasi Teknologi**
Pengembangan modul pembelajaran, buku ajar, dan platform edukasi terbuka akan mendorong penyebaran pemahaman teknologi vision ke kalangan pelajar, praktisi, dan masyarakat umum.

**Penutup: Menuju Computer Vision yang Berkelanjutan dan Manusiawi**
Masa depan Computer Vision bukan hanya soal performa atau kecepatan. Lebih dari itu, ia adalah pertaruhan tentang bagaimana teknologi dapat bekerja demi kemanusiaan, bukan menggantikannya.
Pengembangan algoritma vision harus memperhatikan:

- Efisiensi energi untuk keberlanjutan lingkungan
- Transparansi keputusan untuk menjaga kepercayaan pengguna
- Keadilan sosial agar teknologi dapat diakses dan dimanfaatkan oleh semua kalangan
- Kolaborasi lintas ilmu untuk mendorong inovasi bermakna

Dengan membekali generasi baru ilmuwan dan praktisi dengan pemahaman holistik tentang tantangan dan potensi teknologi vision,
kita membuka jalan bagi sistem-sistem cerdas yang tidak hanya akurat, tapi juga adil, inklusif, dan berkelanjutan.

**Visualisasi: Pilar Etika dalam Vision Masa Depan**
Buku ini diakhiri dengan semangat untuk terus mengeksplorasi, mengembangkan, dan membumikan teknologi vision agar tidak hanya
menjadi pencapaian teknis, tetapi juga kontribusi nyata bagi masa depan umat manusia.

# BAB X VIII
# Penutup dan Arah Lanjutan Penelitian

## 18.1 Refleksi Akhir terhadap Tren Algoritma Vision

Sepanjang buku ajar ini, kita telah menyusuri perjalanan algoritma vision dari tahap awal berbasis metode klasik hingga era algoritma modern yang mengintegrasikan deep learning, edge computing, dan kecerdasan buatan multimodal. Refleksi terhadap perkembangan ini menunjukkan bahwa Computer Vision telah berevolusi dari sekadar alat bantu analisis citra menjadi teknologi strategis dalam revolusi industri 4.0 dan bahkan 5.0.

Tantangan yang semula berkisar pada segmentasi dan klasifikasi kini melebar ke isu-isu etika, fairness, transparansi, serta keberlanjutan. Computer Vision tidak lagi berdiri sendiri sebagai bidang teknis, melainkan berada dalam jalinan multidisiplin bersama ilmu sosial, hukum, kebijakan, dan bahkan filosofi.

## 18.2 Tantangan Penelitian yang Perlu Dijawab

Meskipun banyak kemajuan telah dicapai, terdapat sejumlah tantangan terbuka yang dapat dijadikan ladang penelitian masa depan:

### a. Keterbatasan Dataset Lokal

Sebagian besar dataset standar berasal dari negara maju. Hal ini menyebabkan algoritma vision sulit diadaptasi untuk konteks lokal seperti pertanian tropis, wajah Asia Tenggara, atau tulisan tangan dalam aksara non-Latin.

Tabel 9.1 Perbandingan Dataset Global dan Tantangan Dataset Lokal

| Aspek | Dataset Global (COCO, ImageNet) | Tantangan Dataset Lokal |
|---|---|---|
| Representasi Etnis | Dominan wajah kulit putih | Kurang representasi lokal |
| Variasi Cuaca | Mayoritas kondisi terang | Banyak kondisi ekstrem/tropis |
| Bahasa/Teks | Latin & Inggris dominan | Banyak aksara lokal |

Pada table 9.1 Menjelaskan Perkembangan computer vision sangat bergantung pada ketersediaan dataset yang berkualitas dan representatif. Namun, terdapat kesenjangan signifikan antara dataset global yang umum digunakan dan kebutuhan nyata untuk aplikasi di konteks lokal. Tabel ini menguraikan tiga aspek kritikal dimana dataset global seperti COCO dan ImageNet seringkali tidak sesuai dengan karakteristik lingkungan dan demografi lokal.

Dalam hal representasi etnis, dataset global didominasi oleh citra yang menampilkan individu dengan kulit putih dan fitur wajah kaukasoid. Ketidakseimbangan representasi ini menimbulkan bias algoritmik yang sistemik, dimana model computer vision yang dilatih dengan data tersebut cenderung memiliki akurasi lebih rendah ketika dihadapkan pada wajah-wajah dari etnis lain, khususnya dari wilayah Asia, Afrika, atau Amerika Latin.

Aspek variasi cuaca menunjukkan keterbatasan dataset global yang sebagian besar berisi gambar dalam kondisi pencahayaan ideal dan cuaca sedang. Sebaliknya, banyak wilayah lokal mengalami kondisi cuaca ekstrem seperti kabut, hujan deras, atau pencahayaan tropis yang

silau. Model yang hanya dilatih pada dataset global seringkali gagal beradaptasi dengan variasi atmosferik yang khas di daerah tropis atau empat musim.

Tantangan dalam bahasa dan teks muncul ketika model computer vision perlu mengenali dan memproses teks dalam aksara lokal. Dataset global didominasi oleh teks dalam aksara Latin dan bahasa Inggris, sehingga kurang mampu mengenali aksara non-Latin seperti Hanzi, Arab, Devanagari, atau aksara-aksara lokal lainnya yang memiliki karakteristik struktural yang unik.

Kesenjangan ini menggarisbawahi pentingnya pengembangan dataset yang merepresentasikan keragaman global yang sesungguhnya, serta perlunya pendekatan yang lebih inklusif dalam pengumpulan dan pelabelan data untuk memastikan bahwa teknologi computer vision dapat berfungsi secara adil dan efektif across different cultures and environments.

**b. Model yang Ringan dan Andal**
Masih diperlukan model-model vision yang tidak hanya akurat tetapi juga ringan dan hemat energi, untuk mendukung penerapan pada perangkat edge seperti kamera IoT atau smartphone murah.

**c. Explainability dan Keamanan Algoritma**
Diperlukan pendekatan XAI (Explainable AI) yang benar-benar intuitif, mudah dipahami pengguna awam, dan mampu menjelaskan prediksi
dalam konteks nyata. Keamanan model juga perlu diperkuat agar tidak mudah diserang oleh adversarial input.

**d. Integrasi dengan Konteks Sosial-Budaya**
Penelitian ke depan perlu mengeksplorasi bagaimana algoritma vision dapat memperhatikan norma, nilai, dan sensitivitas budaya lokal agar tidak menimbulkan bias atau ketimpangan sosial.

## 18.3 Rekomendasi untuk Peneliti dan Praktisi

Berdasarkan pembelajaran dari buku ini, berikut adalah beberapa rekomendasi strategis bagi peneliti, dosen, maupun praktisi:

- Kolaborasi Multidisiplin: Gali potensi kolaborasi antara bidang Computer Vision, antropologi, linguistik, pertanian, dan pendidikan untuk membangun solusi kontekstual.
- Open Dataset Lokal: Inisiasi pembuatan dataset terbuka berbasis lokal (tanaman lokal, signage lokal, wajah lokal) sebagai kontribusi global dari Indonesia atau Asia Tenggara.
- Etika sebagai Dasar Inovasi: Jadikan prinsip etika dan keberlanjutan sebagai fondasi dalam merancang algoritma, bukan sekadar pelengkap.
- Pendidikan Inklusif: Rancang kurikulum dan buku ajar yang mampu menjembatani pemahaman algoritma vision bagi siswa SMA, mahasiswa non-teknik, dan masyarakat umum.

**Penutup**

Buku ini diakhiri dengan sebuah keyakinan bahwa masa depan Computer Vision berada di tangan para peneliti, pendidik, dan inovator
muda yang tidak hanya cerdas secara teknis, tetapi juga bijak secara sosial. Perkembangan algoritma vision bukan hanya soal model yang
semakin dalam (deep), tetapi juga dampaknya yang semakin luas (wide).
Dengan pendekatan yang inklusif, etis, dan berorientasi pada kebermanfaatan nyata, kita dapat memastikan bahwa teknologi vision
akan menjadi mitra sejati dalam membangun masa depan yang adil, cerdas, dan berkelanjutan.

**Daftar Pustaka**

Forsyth, D. A., & Ponce, J. (2011). Foundations of computer vision. Addison-Wesley. Retrieved from https://visionbook.mit.edu

Madenda, S. (2023). Komputer vision, kecerdasan buatan, dan sistem tertanam (e-book). Penerbit Gunadarma. Retrieved from https://penerbit.gunadarma.ac.id/wp-content/uploads/2024/08/Komputer-Vision_Fullbook-pasca-ISBN-watermarkcompressed-1.pdf

Marpaung, F. (2022). Computer vision dan pengolahan citra digital. Unimed Press. Retrieved from https://digilib.unimed.ac.id/id/eprint/53012/1/Book.pdf

Munawar, Z. (2023). Visi komputer: Konsep, metode, dan aplikasi. Kaizen Publisher. Retrieved from https://repositori.kaizenpublisher.co.id/publications/569133/visi-komputer-konsep-metode-dan-aplikasi

Parker, J. R. (2010). Algorithms for image processing and computer vision (2nd ed.). Wiley. Retrieved from https://books.google.co.id/books/about/Algorithms_for_Image_Processing_and_Comp.html?id=gPVCp56TYGYC&redir_esc=y

Szeliski, R. (2010). Computer vision: Algorithms and applications. Springer. Retrieved from https://szeliski.org/Book

Tunggal Waras, N. G., Saptadi, N. T. S., Wardani, A. K., Pardosi, V. B. A., Hasanah, Q., Kurniasari, A. A., Firmansyah, M. H., Arifianto, A. S., Maulani, G., & Nur Iin, J. (2024). Kuasai machine learning & computer vision dalam sekejap. CV HEI Publishing Indonesia. Retrieved from https://www.researchgate.net/publication/386131229_KUASAI_MACHINE_LEARNING_COMPUTER_VISION_DALAM_SEKEJAP

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342. Retrieved from https://arxiv.org/abs/2106.11342

Kholis. (2013, September 17). Konversi citra RGB ke grayscale, biner, dan HSV serta menyimpan gambar hasil konversi. WordPress. Retrieved from https://kholisilkom45.wordpress.com/2013/09/17/konversi-citra-rgb-ke-grayscale-biner-dan-hsv-serta-menyimpan-gambar-hasil-konversi/

ResearchGate. (2019). Traditional computer vision workflow vs. deep learning workflow [Figure]. Retrieved from https://www.researchgate.net/figure/a-Traditional-Computer-Vision-workflow-vs-b-Deep-Learning-workflow-Figure-from-8_fig1_331586553

Pemrograman Matlab. (2024, March 1). Arsitektur convolutional neural network (CNN) untuk pengolahan citra digital. Retrieved from

https://pemrogramanmatlab.wordpress.com/category/pengolahan-citra-2/?iframe=true&preview=true%2Ffeed%2F

Pareto AI. (2024). YOLO object detection. Retrieved from https://pareto.ai/blog/yolo-object-detection

Nature. (2024). Artificial intelligence for medical imaging. Nature Machine Intelligence. Retrieved from https://www.nature.com/articles/s44260-024-00006-y

DANI SASMOKO, S.T., M. ENG.

# COMPUTER VISION MODERN

## MODEL, ARSITEKTUR, DAN APLIKASI

## Biodata Penulis

Penulis adalah Dosen di Universitas Sains dan Teknologi Komputer yang pernah menempuh pendidikan S1 di Universitas Islam Indonesia, S2 di Universitas Gajah Mada dan Sekarang Sedang Menempuh S3 di Universitas Kristen Satya Wacana di Bidang Ilmu Komputer, Pengalaman Mengajar di Bidang Internet Of Things , AI dan Pemprograman Mobile dengan tertarik di Penelitian Rekayasa Digital dan Ketehnikan di bidang Terapan.

YAYASAN PRIMA AGUS TEKNIK

**PENERBIT :**
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-51-5 (PDF)
9 786347 227515