BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Beberapa penelitian terkait yang menjadi referensi kepada penulis pada penelitian ini yaitu

Pada tahun 2021, D. R Fauzi dkk meneliti tentang "Implementasi Noise Removal Dan Image Restoration Pada Citra Kabur (Blur) Dengan Menggunakan Metode *Lucy – Richardson Algorihm Techniques*"[[1]]. Dalam penelitian ini, algoritma *Lucy Richardson* cukup efektif dalam proses deblurring gambar. Pada iterasi awal, khususnya iterasi ke-1, gambar masih tampak buram karena adanya penambahan blur sebagai bagian dari proses awal. Namun, seiring bertambahnya jumlah iterasi, kualitas gambar semakin membaik. Hasil terbaik diperoleh pada iterasi sekitar 40 kali, meskipun pada iterasi yang terlalu tinggi dapat muncul artifact pada citra. Selain itu, berdasarkan pengujian usability yang melibatkan 23 responden, sistem mendapatkan tingkat kepuasan pengguna sebesar 85,749%. Nilai ini menunjukkan bahwa sistem yang dikembangkan memiliki kualitas yang sangat baik dan diterima dengan baik oleh pengguna.

Dedi Triyanto dkk, melakukan peneltian pada tahun 2024 tentang "Pengembangan Sistem Deteksi Objek Botol Real-Time dengan YOLOv8 untuk

Aplikasi Vision”[2]. Dalam penelitian ini, Penelitian ini menunjukkan bahwa YOLOv8 sangat bagus untuk mendeteksi botol plastik dan kaleng—modelnya akurat hampir sempurna. Dari 2900 gambar dan 25 kali pelatihan, hasilnya sangat tinggi: akurasi, presisi, dan recall semuanya di atas 99%. Model ini nggak bikin kesalahan dalam mengenali objek.

“Yolo-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah ” di lakukan oleh Yanto dkk pada tahun 2023[3]. memperlihatkan bahwa YOLO-v8 Improved Algorithm memiliki hasil akurasi yang tinggi untuk mendeteksi penggunaan masker muka, dengan akurasi kelas badmask sebesar 94%, mask sebesar 97% dan nomask sebesar 95%. Nilai F1-Confidence, Precision, dan Recall semua kelasnya juga tinggi, yaitu masing-masing sebesar 0.94, 0.96, dan 0.978. Waktu komputasi rata-rata yang diperlukan oleh algoritma ini hanya 17ms. Sehingga diharapkan mempermudah deteksi pemakaian masker wajah dan memastikan bahwa pemakaian masker dilakukan secara benar dan tepat, memastikan keselamatan dan kesehatan semua orang yang berada di lingkungan tersebut.

Pada tahun 2023, Lejar Satya dkk meneliti tentang “Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur Yolov8”[4]. Dalam penelitian ini,

YOLOv8 berhasil menghasilkan model yang mampu mendeteksi berbagai jenis kendaraan dan plat nomor dengan tingkat akurasi yang memadai. Skor precision sebesar 0,871 dan nilai recall sebesar 0,85. Penggunaan PaddleOCR sebagai metode pengenalan karakter pada plat nomor juga memberikan hasil yang baik dalam mengenali karakter-karakter pada plat nomor polisi, termasuk pada karakter yang terdistorsi atau memiliki kualitas gambar yang rendah. Pada uji coba menggunakan berbagai gambar, sistem mampu mengenali objek kendaraan dan plat nomor polisi dengan baik pada sebagian besar kasus. Meskipun demikian, masih terdapat beberapa kasus di mana model mengalami kesulitan dalam mengenali objek, terutama pada kelas Truk dan karakter tanggal pada plat nomor polisi. Ini menunjukkan bahwa masih ada ruang untuk peningkatan dalam pengenalan karakter yang lebih akurat.

E.U. Armin dkk, melalukan penelitian tentang "Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada pengendara mobil" pada tahun 2023[5]. Dalam penelitian ini, model YOLOv8 berhasil mencapai nilai mAP sebesar 0.96092 selama proses pelatihan dengan memanfaatkan 2996 citra dari dataset sekunder. Peneliti selanjutnya melakukan evaluasi pada kinerja model YOLOv8 dan membandingkannya dengan model deteksi objek lainnya, termasuk SSD-Resnet, YOLOv3, dan YOLOv5, dengan menggunakan dataset yang sama yang telah dikumpulkan oleh peneliti. Hasil evaluasi menunjukkan bahwa meskipun YOLOv8 memiliki FPS yang lebih rendah dibandingkan dengan SSD-Resnet dengan selisih sebanyak 50

FPS, namun model ini menunjukkan tingkat ketepatan yang lebih baik dalam mendeteksi objek.

Pada tahun 2024, Aditia Putra Pranjaya dkk meneliti tentang “Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya”[6]. Dalam penelitian ini, ta pistol pada penelitian ini menunjukkan nilai mAP sebesar 84% dan F1-Score 80%. Tahap pengujian deteksi terhadap gambar yang telah dimanipulasi seperti grayscale, pengaturan tingkat cahaya dan gambar berotasi tidak berpengaruh secara signifikan pada hasil deteksi. Akan tetapi, pada kondisi gambar blur sangat mempengaruhi model YOLOv8 pada saat uji pendeteksian senjata. Pada pengujian berdasarkan jarak antara objek dan kamera, pendeteksian objek terjauh yang dapat dicapai sejauh 3 meter dengan nilai confidence rendah dibawah 60% namun pada jarak 2 meter nilai confidence mampu mencapai diatas 80%. Hal ini menunjukkan bahwa volume dan variasi dataset berpengaruh terhadap hasil pelatihan seperti pada penelitian-penelitian sebelumnya, augmentasi bisa menjadi salah satu cara yang efektif untuk memperkaya data tanpa perlu melewati proses panjang pada pengumpulan dataset.

2.2 Landasan Teori

2.2.1 *Computer Vision*

Computer Vision adalah cabang dari *Deep Learning* dan *Artificial Intelligence* yang memungkinkan mesin untuk melihat dan memahami lingkungan seperti manusia. Penerapa ini terbagi menjadi tiga kategori utama:

1. *Low-Level Vision* : Berfokus pada pemrosesan dasar gambar seperti penghilangan derau, pemfilteran, penajaman, dan klasifikasi tekstur. Mesin memahami gambar sebagai objek digital.
2. *Intermediate-Level Vision* : Meningkatkan kemampuan mesin untuk mendeteksi garis, tepi, dan pola. Pengenalan pola mencakup analisis bentuk seperti batas, tepi, lingkaran, dan elips, serta pencocokan pola antar objek.
3. *3D Vision* dan Gerakan : Membantu mesin memahami dunia dalam tiga dimensi dengan mengidentifikasi bentuk dan bayangan, serta mengambil informasi dari objek yang bergerak. Teknologi ini telah diterapkan dalam kehidupan sehari-hari, misalnya pada kamera pemantau untuk deteksi plat kendaraan.

2.2.1.1 Cara Kerja *Computer Vision*

Computer Vision adalah ekstraksi otomatis informasi dari gambar. Informasi dapat berarti apa saja mulai dari model 3D, posisi kamera, deteksi dan pengenalan objek hingga pengelompokan dan pencarian konten gambar. Intinya, tugas utama visi komputer adalah tentang bagaimana membuat komputer memahami gambar digital serta data visual dari dunia nyata.

Berikut adalah tabel yang merangkum cara kerja *Computer Vision* berdasarkan berbagai proses utama yang terlibat:

Tabel 2. 1 Cara Kerja *Computer Vision*

Proses	Deskripsi
Pengolahan Citra Digital	Manipulasi dan interpretasi gambar menggunakan komputer untuk meningkatkan kualitas dan mengekstrak informasi.
Image Enhancement	Meningkatkan kualitas gambar, misalnya melalui perbaikan kontras dan penajaman tepian.
Image Restoration	Mengurangi cacat pada gambar seperti noise atau kesamaran akibat fokus kamera yang kurang tepat.
Image Segmentation	Memisahkan elemen dalam gambar untuk memudahkan analisis dan pengenalan pola.
Image Analysis	Mengekstrak fitur penting untuk identifikasi objek dalam gambar.
Image Reconstruction	Menggabungkan beberapa citra untuk membentuk gambar baru, seperti pemetaan organ dari foto rontgen.
Image Compression	Mengurangi ukuran gambar untuk efisiensi penyimpanan dan transmisi data.
Steganografi & Watermarking	Menyembunyikan informasi rahasia dalam gambar agar tidak terlihat oleh orang lain.
Klasifikasi Citra	Identifikasi dan kategorisasi objek dalam gambar menggunakan pola dan ciri visual tertentu.
Convolutional Neural Networks (CNN)	Model deep learning yang digunakan untuk klasifikasi gambar, sering kali dengan dataset besar seperti ImageNet.
Transfer Learning	Teknik menggunakan model yang telah dilatih sebelumnya untuk tugas baru guna meningkatkan

	efisiensi proses.
Pengenalan Bentuk dan Bayangan	Memahami bentuk objek dan efek bayangan yang dapat membantu analisis visual.
Self Shadow	Bayangan yang muncul pada objek itu sendiri, sering kali dengan tingkat kecerahan lebih tinggi.
Cast Shadow (Umbra & Penumbra)	Bayangan yang terbentuk akibat objek menghalangi cahaya langsung, dapat berpengaruh pada interpretasi visual.
Filtering	Menggunakan kernel untuk meningkatkan fitur tertentu dalam gambar dan menghilangkan noise.
Contrast Adjustment	Mengoptimalkan pencahayaan gambar agar objek lebih mudah dikenali.
Shading	Memberikan efek warna untuk meningkatkan persepsi kedalaman dan kontur objek.

2.2.1.2 Penerepan *Computer Vision*

Penerapan visi komputer pada kehidupan manusia beragam pada berbagai aspek. Hal ini dikarenakan efektivitas bantuan mesin dalam menganalisis citra baik 2D dan 3D terbukti dan terjamin hasilnya. Berbagai aspek krusial yang membutuhkan pembuatan keputusan penting bahkan sudah menerapkan bantuan alat pendeteksi citra, seperti contohnya dalam bidang kesehatan dengan adanya klasifikasi sel tumor dari hasil rontgen. Berikut adalah beberapa contoh penerapan computer vision dalam berbagai aspek :

1. **Pemrosesan gambar *biometrik dan biomedis***

a. Digunakan dalam identifikasi biometrik untuk keamanan, identifikasi kriminal, dan verifikasi kartu identitas.

b. Pengenalan wajah menggunakan metode *Eigenfaces/Fisherfaces*, verifikasi tanda tangan, serta analisis citra medis seperti *X-Ray*, *CT*, dan *MRI* untuk diagnosis penyakit.

2. Analisis pemandangan multispektral penginderaan jauh

a. Memanfaatkan citra satelit dan udara untuk mengumpulkan data sumber daya alam, seperti pertanian, hidrologi, mineral, dan geologi.

b. Berguna untuk perencanaan wilayah, pemetaan vegetasi, dan pengukuran kepadatan penduduk menggunakan klasifikasi berbasis pengetahuan.

3. Deteksi dan pelacakan objek bergerak

a. Digunakan dalam pengawasan keamanan, pemantauan lalu lintas, dan industri untuk mengidentifikasi objek yang bergerak dalam video.

b. Prinsip pentingnya adalah pemanfaatan cahaya dan teknologi night vision untuk deteksi optimal di berbagai kondisi pencahayaan.

Computer Vision terus berkembang sebagai teknologi yang berkontribusi besar dalam berbagai industri, termasuk kesehatan, keamanan, dan pengelolaan sumber daya alam.[7]

2.2.2. *Machine Learning*

Machine Learning adalah teknik pemrograman komputer yang memungkinkan sistem belajar dari data dan membangun model atau inferensi

tanpa instruksi eksplisit. Algoritma ini dapat mengenali pola dan membuat prediksi berdasarkan data historis, menjadikannya bagian dari kecerdasan buatan (AI).[8]

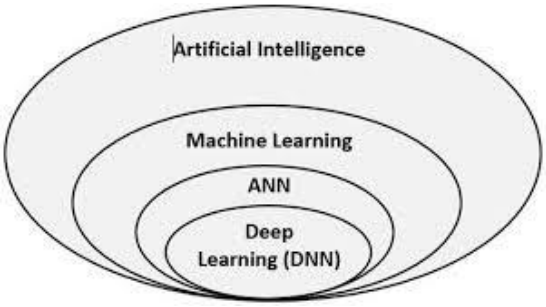
Berikut adalah penerapan *Machine Learning* dalam bentuk tabel:

Tabel 2. 2 Penerapan *Machine Learning*

Aspek	Penjelasan
Penerapan dalam Bisnis	Digunakan untuk analisis data pelanggan, pengambilan keputusan bisnis, optimalisasi strategi pemasaran, dan peningkatan efisiensi operasional.
Penerapan dalam Manufaktur	Mendukung pemeliharaan prediktif, kontrol kualitas produk, optimalisasi rantai pasokan, serta peningkatan efisiensi produksi.
Penerapan dalam Kesehatan	Menganalisis data medis untuk diagnosis penyakit, deteksi tumor dan kanker, serta mendukung keputusan dokter dalam merawat pasien secara real-time.
Penerapan dalam Keuangan	Memungkinkan analisis pasar saham, deteksi penipuan, identifikasi klien berisiko tinggi, serta optimalisasi portofolio investasi.
Penerapan dalam Hiburan	Digunakan untuk merekomendasi konten yang dipersonalisasi, desain trailer dan iklan, serta meningkatkan pengalaman pengguna dalam layanan streaming.

Machine Learning semakin berkembang dan menjadi bagian integral dalam berbagai industri.

2.2.3 *Deep Learning*



Gambar 2. 1 *Deep Learning*

Deep learning adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mengekstrak fitur dari data secara otomatis. Keunggulan utama deep learning dibandingkan metode tradisional adalah kemampuannya dalam menangani data dalam jumlah besar dan kompleks, serta kemampuannya dalam belajar dari data tanpa perlu pemrograman eksplisit.[9]

2.2.3.1 Tantangan dalam *Deep Learning*

Beberapa tantangan utama dalam *deep learning* :

1. Kebutuhan data yang besar : Model deep learning membutuhkan dataset yang luas agar dapat belajar dengan baik.
2. *Interpretabilitas model* : Model deep learning sering dianggap sebagai black box, sehingga sulit untuk memahami bagaimana keputusan dibuat.

3. Masalah *overfitting* : Model dapat terlalu menyesuaikan diri dengan data pelatihan, sehingga kurang efektif dalam menangani data baru.
4. Keterbatasan dalam menangani data yang tidak seimbang : Data yang tidak seimbang dapat menyebabkan model lebih cenderung memprediksi kelas mayoritas.

2.2.3.2 Aplikasi *Deep Learning*

Deep learning telah diterapkan dalam berbagai bidang, termasuk:

1. Pengenalan gambar : digunakan dalam deteksi objek, pengelompokan gambar, dan pengolahan citra medis.
2. Pemrosesan bahasa alami (NLP) : digunakan dalam terjemahan otomatis, chatbot, dan analisis sentimen.
3. Deteksi penyakit dalam bidang medis : model *deep learning* telah digunakan untuk mendiagnosis penyakit berdasarkan gambar medis.
4. Pengenalan pola dalam data industri : digunakan dalam prediksi tren pasar, analisis data keuangan, dan optimasi produksi.

2.2.4 Deteksi Objek

Deteksi objek adalah proses mengenali dan menemukan berbagai objek dalam gambar atau video. Dalam dunia pertanian, deteksi objek sangat berguna, terutama untuk memantau tanaman dan buah-buahan. Misalnya, dalam budidaya buah markisa, teknologi ini membantu petani mengetahui kapan waktu yang tepat untuk memanen buah. Dengan deteksi otomatis, petani bisa melihat apakah buah

markisa sudah matang dan siap dijual, sehingga kualitas dan rasa buah dapat terjaga dengan baik. Ini menjadi solusi yang efisien dan mengurangi risiko memanen buah yang belum matang atau terlalu matang.komputer sebagai perangkat lunak untuk memproses setiap piksel dari sebuah gambar.

2.2.5 Pengolahan Citra Digital

Deteksi objek adalah proses mengenali berbagai jenis objek dan mengklasifikasikannya ke dalam kategori tertentu; ini merupakan salah satu tugas utama dalam *computer vision*. *Computer vision* adalah bidang yang berkembang pesat dan kemajuannya sangat penting bagi revolusi di bidang ilmu komputer. Deteksi objek melibatkan beberapa elemen utama seperti dataset, algoritma, dan teknik. Dalam deteksi objek, biasanya terdapat dataset yang berisi berbagai objek, dan tujuan dari penelitian ini adalah agar komputer dapat mendeteksi objek tersebut tanpa diberitahu secara eksplisit tentang objeknya.

Seluruh proses ini terbagi menjadi beberapa bagian, dan pembelajaran yang dilakukan dapat terdiri dari berbagai jenis. Jenis pembelajaran ini mencakup pembelajaran terawasi (supervised learning) dan pembelajaran tak terawasi (unsupervised learning). Dalam *supervised learning*, model dilatih dengan data berlabel untuk belajar dan kemudian mendeteksi data saat tahap pengujian, sedangkan dalam *unsupervised learning*, model tidak diberikan data berlabel dan belajar secara mandiri melalui perhitungan *loss*. Karena deteksi objek adalah bagian fundamental dari *computer vision*, ini juga menjadi dasar bagi berbagai tugas lainnya dalam bidang tersebut. Beberapa tugas berbasis deteksi objek yang paling penting termasuk pengenalan aktivitas, pembuatan teks dari gambar (*image*

captioning), deteksi dan pengenalan wajah, pelacakan objek, dan segmentasi gambar.

Berdasarkan aplikasinya, deteksi objek dapat dibagi menjadi dua jenis utama. Yang pertama adalah deteksi umum, yaitu deteksi objek di dunia nyata. Banyak penelitian dilakukan di bidang ini, terutama menggunakan algoritma YOLO (*You Only Look Once*), yang dikenal sebagai metode deteksi objek real-time. Jenis lainnya disebut deteksi objek berbasis aplikasi, di mana teknik deteksi objek digunakan sesuai dengan kebutuhan tertentu.

Perkembangan utama dalam deteksi objek mulai mendapatkan perhatian besar setelah tahun 2012 dengan hadirnya model seperti *Regional Convolutional Neural Network* (RCNN), *Fast RCNN*, dan *Faster RCNN*. Model-model ini secara drastis mengubah pendekatan tradisional dalam deteksi objek dan membawa revolusi dalam metode deteksi secara umum. Setelah itu, YOLO dan tiga versinya yang muncul berturut-turut memberikan kontribusi besar bagi kemajuan deteksi objek di berbagai bidang. Meskipun telah banyak kemajuan, penelitian tentang deteksi objek masih menghadapi berbagai tantangan untuk membuatnya lebih cepat dan lebih akurat.[10]

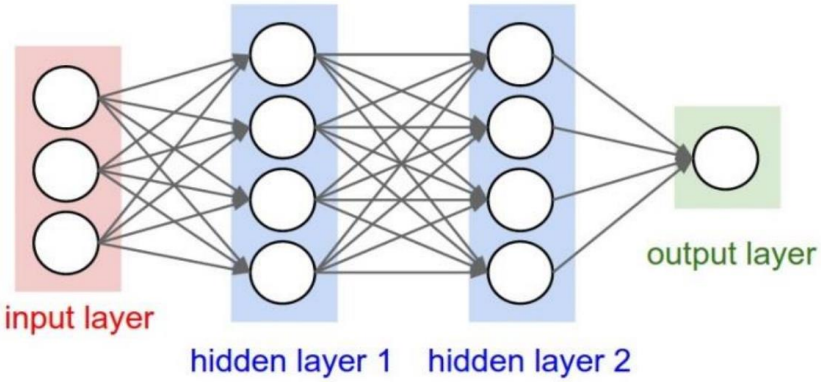
2.2.6 *Convolutional Neural Network (*CNN*)*

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Networ*k karena kedalaman jaringan

yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama *LeNet* berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi *ImageNet Large Scale Visual Recognition Challenge* 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra.[11]

2.2.6.1 Konsep CNN

Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



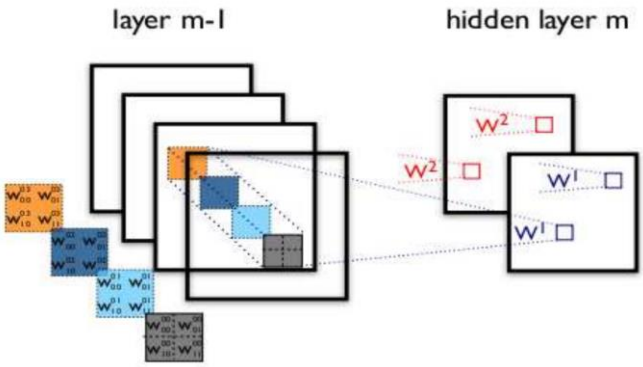
Gambar 2. 2 Arsitektur MLP Sederhana

Sebuah MLP seperti pada Gambar. 1. memiliki i layer (kotak merah dan biru) dengan masing-masing layer berisi ji neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi.

Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar.2. Dimensi bobot pada CNN adalah:

neuron input x neuron output x tinggi x lebar

Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara.



Gambar 2. 3 Proses konvolusi pada CNN

2.2.6 Metode *Lucy Richardson*

Algoritma *Lucy-Richardson* (L-R), yang dikenal juga dengan dekonvolusi *Lucy-Richardson,* dikembangkan secara independen oleh Richardson (1972) dan Lucy (1974). Algoritma ini efektif jika kita mengetahui PSF tetapi hanya mengetahui sedikit mengenai derau aditif pada citra. Algoritma ini pada mulanya digunakan untuk merestorasi citra astonomi, sebelum akhirnya digunakan juga secara luas untuk merestorasi sembarang citra yang mengalami kekaburan. Algoritma ini memaksimumkan kemungkinan (*maximum likelihood*) bahwa sebuah citra bila dikonvolusi dengan PSF hasilnya adalah instansiasi dari citra kabur, dengan mengasumsikan derau tersebat dengan distribusi Poison.

$$p(x) = e{-}a\ ax$$

$$x!$$

Dengan x adalah peubah acak dan a adalah konstanta. Esensi dari iterasi adalah sebagai berikut: estimasi ke-(n +1) dari citra restorasi adalah estimasi ke-n citra restorasi dikali dengan citra koreksi. Persamaan iterasinya adalah:

$$f\hat{}\ \mathrm{n}+1 = f\hat{}\ \mathrm{n}\ (\ g$$

$$\hat{f}\, n{*}PSF\,) * reflect\,(PSF)$$

yang dalam hal ini, operator $*$ menyatakan konvolusi, $\hat{f} = \hat{f}\,(x,y)$ menyatakan estimasi citra restorasi, g = g(x, y) menyatakan citra masukan (yang mengalami degradasi), reflect(PSF) menyatakan pencerminan PSF, yaitu Reflect ((PSF (x, y)) = PSF (–x, –y), dan (g $\hat{f}$ n∗PSF) ∗ reflect (PSF) menyatakan citra koreksi.[12]

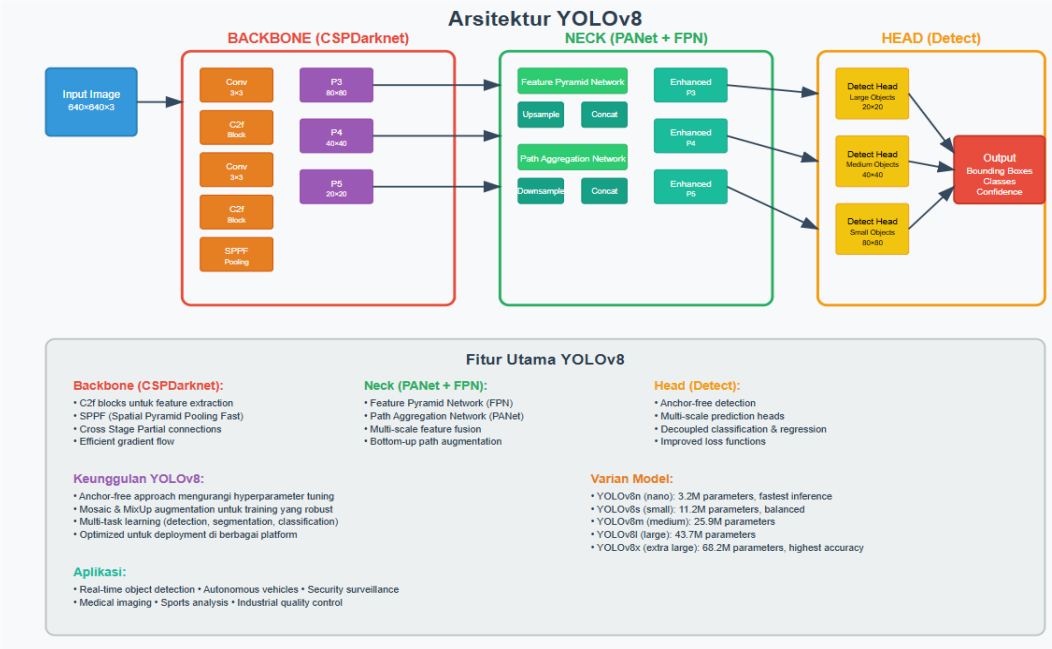
2.2.6 *You Only Look Once* (YOLO)

YOLO adalah metode canggih untuk mendeteksi objek dalam gambar dengan cepat dan efisien. Ketika sebuah gambar dimasukkan, YOLO membaginya menjadi beberapa kotak kecil dan setiap kotak ini memeriksa apakah ada objek di dalamnya. Jika ada, sistem akan menggambar kotak pembatas di sekitar objek tersebut dan memberikan skor kepercayaan yang menunjukkan seberapa yakin dia bahwa objek itu benar-benar ada. Untuk menghindari kebingungan ketika kotak-kotak tersebut tumpang tindih, YOLO memilih kotak yang paling akurat dan mengabaikan yang lainnya. Selain itu, untuk melatih sistem agar lebih baik, YOLO menggunakan teknik yang disebut data augmentation, di mana gambar diubah-ubah agar sistem bisa belajar dari berbagai situasi. Metode ini sangat berguna dalam berbagai aplikasi, seperti pengawasan keamanan, mobil otonom, dan analisis pertanian, karena mampu mendeteksi banyak objek secara bersamaan dengan akurasi tinggi.[13]

2.2.7 *You Only Look Once* (YOLOv8)

YOLOv8, memiliki beberapa varian (*small, medium, large, dan extra large*) yang disesuaikan dengan kebutuhan komputasi dan akurasi. YOLOv8 (You Only Look Once versi 8) adalah model deteksi objek yang sangat populer dan diperbarui dari versi sebelumnya. Model ini dirancang untuk mendeteksi objek dalam gambar atau video dengan cepat dan akurat, sehingga sangat cocok untuk aplikasi seperti pengawasan video dan mobil otonom. Salah satu fitur utamanya adalah kemampuan untuk melakukan deteksi secara real-time. YOLOv8 juga memiliki arsitektur yang ditingkatkan, yang meningkatkan kecepatan dan akurasi deteksi. Selain itu, model ini mampu mendeteksi objek dengan berbagai ukuran melalui prediksi multi-skala dan mendukung transfer learning, memungkinkan pengguna untuk menyempurnakan model dengan dataset mereka sendiri tanpa memerlukan banyak sumber daya.

Dalam praktiknya, untuk melatih model YOLOv8, pengguna perlu mempersiapkan dataset dengan mengumpulkan dan memberi anotasi pada gambar, kemudian mengatur file konfigurasi model. Proses pelatihan dapat dilakukan menggunakan framework seperti PyTorch, dan setelah itu, model harus dievaluasi pada set validasi untuk menyesuaikan parameter jika diperlukan. YOLOv8 memiliki berbagai aplikasi, termasuk dalam mobil otonom untuk mendeteksi pejalan kaki dan rintangan, sistem visi robotik, serta dalam pencitraan medis untuk mendeteksi anomali. Sumber daya seperti dokumentasi resmi dan komunitas online juga tersedia untuk membantu pengguna dalam mengimplementasikan model ini.



Gambar 2. 4 Gambar Arsitektur YOLOv8

Berdasarkan gambar arsitektur YOLOv8, diagram ini menunjukkan struktur lengkap model deep learning untuk deteksi objek yang terdiri dari tiga komponen utama: Backbone (CSPDarknet) berwarna merah yang mengekstrak fitur dari input image 640×640×3 menggunakan Conv layer, blok C2f, dan SPPF pooling untuk menghasilkan peta fitur multi-skala (P3: 80×80, P4: 40×40, P5: 20×20), Neck (PANet + FPN) berwarna hijau yang menggabungkan fitur menggunakan Feature Pyramid Network dan Path Aggregation Network melalui operasi upsample/downsample dan concatenation untuk menghasilkan fitur yang disempurnakan, dan Head (Detect) berwarna kuning yang memiliki tiga kepala deteksi secara terpisah untuk mendeteksi objek berukuran kecil, sedang, dan besar secara bebas jangkar, kemudian menghasilkan keluaran berupa kotak pembatas, kelas, dan skor kepercayaan. Keunggulan YOLOv8 terletak pada pendekatan bebas jangkar yang mengurangi penyetelan hyperparameter, penggunaan blok C2f

untuk ekstraksi fitur efisien, kemampuan pembelajaran multi-tugas, dan tersedia dalam 5 varian (nano hingga ekstra besar) dengan parameter 3.2M-68.2M yang dioptimalkan untuk berbagai aplikasi real-time seperti kendaraan otonom, pengawasan keamanan, pencitraan medis, dan kontrol kualitas industri..[14]

2.2.8 *Python*

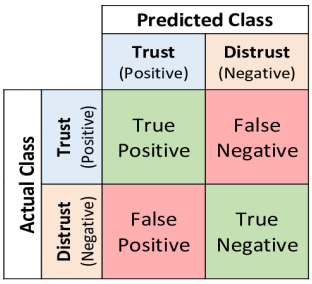
Python adalah bahasa pemrograman yang bersifat *open source*. Bahasa ini dioptimalkan untuk meningkatkan produktivitas pengembang, kualitas perangkat lunak, serta portabilitas program. *Python* telah banyak digunakan dalam pengembangan berbagai jenis perangkat lunak, termasuk *user interface*, *numeric programming*, *product custimization*, *system programming*, dan *scripting internet*. Dalam pemrograman *Python*, library terdiri dari beberapa modul dan paket yang digabungkan untuk menyediakan sekumpulan fungsionalitas. Dengan adanya library dalam *Python*, pengembang dapat menganalisis kode yang telah digunakan dan diuji sebelumnya, sehingga memudahkan dalam pengembangan aplikasi.

2.2.9 Confusion Matrix

Confusion matrix adalah alat yang sering digunakan untuk menghitung akurasi dalam konteks data mining. Matriks ini menghitung empat keluaran utama, yaitu: *recall*, *precision*, akurasi, dan *error rate*. *Confusion matrix* adalah tabel yang menunjukkan jumlah data uji yang diklasifikasikan dengan benar dan

salah. Dalam *confusion matrix*, terdapat empat istilah yang digunakan untuk merepresentasikan hasil proses klasifikasi, antara lain:

1. *True Positive* (TP): data positif yang diprediksi sebagai positif.
2. *True Negative* (TN): data negatif yang diprediksi sebagai negatif.
3. *False Positive* (FP): data negatif yang salah diklasifikasikan sebagai positif.
4. *False Negative* (FN): data positif yang salah diklasifikasikan sebagai negatif.



Gambar 2. 5 C*onfusion Matrix*

Untuk menghitung akurasi dilakukan dengan persamaan

$$\text{Presisi prediksi} = \frac{TP}{TP+FP} \tag{2.1}$$

Presisi merupakan penggambaran seberapa tepat model dalam memprediksi kejadian positif saat kegiatan prediksi. Kemudian untuk menghitung menggunakan persamaan.

$$\text{Akurasi} = \frac{TP+TN}{P+N} \tag{2.2}$$

Akurasi berfungsi sebagai parameter yang akurat pada suatu model dalam melakukan klasifikasi. Recall dapat dihitung menggunakan persamaan.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.3}$$

F1 Score merupakan rata-rata dari precision dan recall yang dapat dihitung dengan persamaan $F1 = 2x \frac{precision\ x\ Recall}{precision+Recall}$ (2.4)

2.2.10 *Mean Average Precision* (mAP)

Mean Average Precision (mAP) adalah metrik evaluasi yang secara luas digunakan dalam bidang visi komputer, terutama untuk tugas pengenalan objek dan pengambilan informasi. Metrik ini mengukur efektivitas model dalam mendeteksi dan mengklasifikasikan objek dalam gambar. Berikut adalah penjelasan lengkap mengenai mAP. *Average Precision* (AP) adalah rata-rata precision pada berbagai level recall. AP memberikan gambaran tentang seberapa baik model dapat mengidentifikasi objek dengan benar pada berbagai ambang batas. *Mean Average Precision* (mAP) adalah rata-rata dari nilai AP untuk semua kelas dalam dataset. mAP memberikan gambaran keseluruhan tentang performa model di semua kategori.[15]

1. Konsep dasar mAP

a. ***Precision****:* Proporsi prediksi positif yang benar. Dihitung sebagai:

Precision=TPTP+FPPrecision=TP+FPTP

di mana TP adalah true positives dan FP adalah *false positives.*

b. ***Recall**:* Proporsi objek yang terdeteksi dengan benar dari total objek yang relevan. Dihitung sebagai:

$$Recall=TPTP+FNRecall=TP+FNTP$$

di mana FN adalah *false negatives*.

2. Proses Perhitungan mAP

Berikut adalah penjelasan proses perhitungan mAP dalam bentuk tabel:

Tabel 2. 3 Tabel Proses Perhitungan mAP

Langkah	Deskripsi
Pengumpulan Data	- Ground Truth: Kumpulkan data asli dengan label objek yang benar dalam dataset. - Prediksi Model: Dapatkan prediksi dari model (bounding boxes dan kelas).
Menghitung Precision dan Recall	- Hitung nilai precision dan recall untuk setiap prediksi yang dihasilkan model pada berbagai threshold.
Menyusun Kurva Precision-Recall	- Plot nilai precision terhadap recall untuk setiap threshold. - Lakukan interpolasi untuk menghaluskan kurva, sehingga setiap nilai recall memiliki precision maksimum.
Menghitung *Average Precision* (AP)	- Hitung AP untuk setiap kelas dengan mengambil rata-rata dari precision pada berbagai level recall.
Menghitung *Mean Average Precision* (mAP)	- Hitung mAP dengan mengambil rata-rata semua nilai AP untuk setiap kelas: $mAP = (1/N) \sum(AP_i)$ di mana N adalah jumlah kelas.

Tabel di atas menyajikan langkah-langkah secara sistematis untuk memudahkan pemahaman proses perhitungan mAP.

2.2.11 *Roboflow*

Roboflow adalah platform berbasis web yang menyediakan solusi lengkap untuk pengembangan model *computer vision*, mulai dari pengumpulan, pengelolaan, hingga optimasi dataset untuk pelatihan model *machine learning*. Platform ini memudahkan pengembang dalam membuat, mengajar, dan menggunakan model deteksi objek dengan berbagai fitur seperti anotasi (*labeling*) gambar menggunakan *bounding box, preprocessing* data, dan augmentasi data untuk meningkatkan kualitas dataset

1. Fungsi dan Peran *Roboflow* dalam *Computer Vision*

a. Pengelolaan Dataset: *Roboflow* membantu mengorganisir dataset agar mudah dianotasi dan diproses, termasuk upload gambar, video, dan file anotasi seperti XML.

b. Anotasi dan *Labeling*: Platform ini menyediakan alat untuk memberikan label dan *bounding box* pada objek dalam gambar, yang merupakan tahap penting dalam pelatihan model deteksi objek.

c. *Preprocessing* Data: Meliputi *resize* gambar agar ukuran seragam, pembersihan data dari noise atau outlier, dan penyesuaian lain yang penting untuk mempersiapkan data sebelum pelatihan model.

d. Augmentasi Data: *Roboflow* mendukung augmentasi otomatis seperti flipping, rotasi, perubahan warna, dan noise untuk memperkaya variasi data sehingga model menjadi lebih robust dan akurat.

e. Pembagian Dataset: Dataset yang telah dianotasi dan diproses dibagi menjadi set pelatihan, validasi, dan pengujian dengan proporsi tertentu untuk memastikan evaluasi model yang baik

2. Integrasi dengan Model dan *Framework Machine Learning*

Roboflow kompatibel dengan berbagai *framework* dan model populer, termasuk YOLOv5, TensorFlow, dan *PyTorch*. Platform ini menyediakan lebih dari 20 pilihan model yang dapat digunakan langsung untuk pengembangan proyek computer vision, memudahkan proses deployment dan inferensi model secara real-time. Roboflow juga mendukung pipeline pengembangan model secara *end-to-end* dari dataset hingga *deployment*, sehingga mempercepat proses pembuatan aplikasi *computer vision.*

3. Manfaat Penggunaan *Roboflow*

a. Meningkatkan efisiensi dalam pengelolaan dan pemrosesan dataset.

Memperbaiki kualitas data latih melalui anotasi yang akurat dan augmentasi yang efektif.

b. Memudahkan kolaborasi antar tim dalam pengembangan model. Mempercepat proses pelatihan dan deployment model computer vision dengan dukungan berbagai format dataset dan model yang siap pakai.

merupakan kerangka kerja yang sangat membantu dalam pengembangan sistem computer vision, khususnya dalam tahap pengumpulan dan persiapan data,

yang merupakan fondasi penting untuk menghasilkan model deteksi objek yang *Roboflow* akurat dan efisien.

Roboflow dan YOLO memiliki hubungan yang erat dalam pengembangan model deteksi objek berbasis computer vision. Roboflow berperan sebagai platform yang menyediakan solusi lengkap untuk pengelolaan dataset, mulai dari anotasi, preprocessing, augmentasi, hingga ekspor data dalam format yang kompatibel dengan YOLO, termasuk YOLOv5, versi terbaru dari arsitektur.

2.2.12 *Google Colab*

Gambar 2. 6 Logo *Google Colab*

Google Colab atau *Google Colaboratory*, adalah sebuah *executable document* yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui *Google Drive*. *Software* ini pada dasarnya serupa dengan Jupyter Notebook gratis berbentuk *cloud* yang dijalankan menggunakan *browser*, seperti *Mozilla Firefox* dan *Google Chrome*. Penggunanya untuk menjalankan kode *Python* tanpa perlu melakukan proses instalasi dan *setup* lain nya. Justru, semua keperluan setting dan keperluan setting dan adjustment akan diserahkan ke *cloud*. Selain itu, *Google Colaboratory* juga terkenal karena dapat mendorong kebutuhan

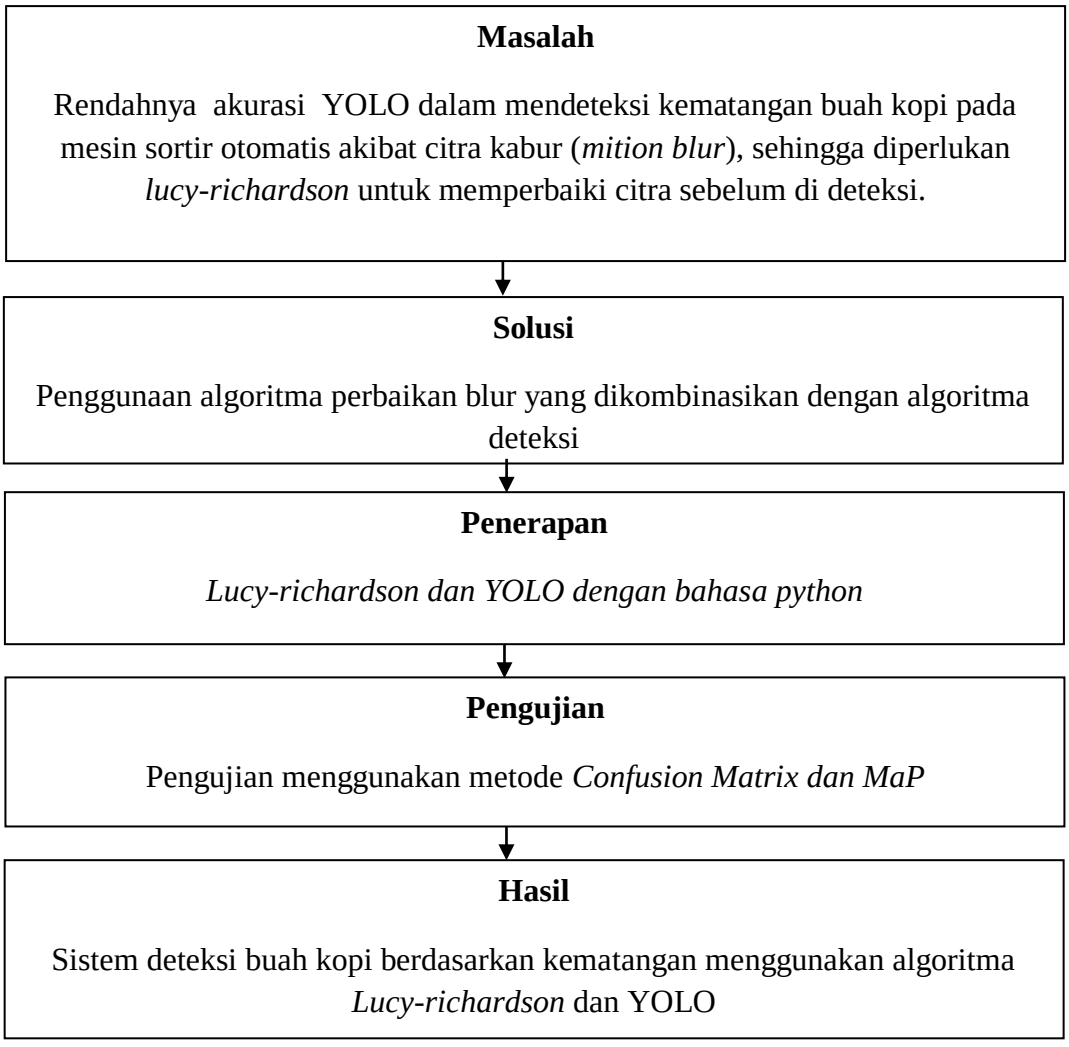
kolaborasi tim . Dapat Menggunakan *Python* dan alat *Google Colab* untuk deteksi sentimen ulasan pengguna *e-learning* menggunakan algoritma *naive bayes*, data yang relevan dikumpulkan dari berbagai sumber untuk mendapatkan informasi yang memadai. Setelah itu, proses pelabelan dilakukan untuk mengkategorikan data sesuai dengan tujuan penelitian. Selanjutnya, data yang telah dilabeli melewati tahap *pre-processing* untuk membersihkan dan mempersiapkan data agar siap digunakan dalam analisis. Setelah klasifikasi selesai, kinerja model dievaluasi menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Proses ini melibatkan pencarian data relevan berdasarkan kata kunci, serta pemahaman lebih mendalam tentang data melalui teknik data mining menggunakan *Python* dan *Google Colab*. Selanjutnya, dilakukan import library Pandas, yang fungsinya untuk memanipulasi data. Kemudian, *library Google Play scraper* digunakan untuk melakukan *crawling* data dari *Play Store*, yang kemudian dikonversi ke dalam format ".csv". Ada beberapa keuntungan *Google Colab* yaitu:

a. Akses GPU gratis untuk menjalankan kode yang memerlukan komputasi intensif.
b. Kemampuan berkolaborasi dengan tim secara real-time.
c. Integrasi dengan *Google Drive* untuk menyimpan dan mengakses file.
d. Fleksibilitas dalam menjalankan kode *Python* dengan berbagai *library* dan *framework.*
e. Tidak memerlukan server atau infrastruktur tambahan.
f. Kemampuan untuk membuat notebook yang interaktif dan dinamis.

g. Dukungan untuk berbagai *library* seperti *TensorFlow*, *PyTorch*, dan *Keras*.

2.3 Kerangka Pikir

Kerangka pikir dalam penelitian ini mengenai deteksi kematangan buah kopi dengan menggunakan algoritma *Lucy-richardson* dan YOLO dapat dilihat pada gambar 2.6



Gambar 2. 7 Kerangka Berfikir