BAB II

KAJIAN LITERATUR

2.1 Computer Vision

Computer Vision merupakan bidang dalam kecerdasan buatan (*Artificial Intelligence/AI*) yang berfokus pada bagaimana komputer dapat memahami, menafsirkan, dan menganalisis informasi visual dari citra maupun video sebagaimana manusia melakukannya. Tujuan utama dari computer vision adalah mengekstraksi informasi bermakna dari data visual untuk melakukan klasifikasi, deteksi objek, segmentasi, maupun pengenalan pola [45]. Proses ini melibatkan beberapa tahapan penting seperti akuisisi citra, pra-pemrosesan, ekstraksi fitur, serta interpretasi hasil berbasis algoritma pembelajaran mesin atau jaringan saraf tiruan.

Seiring perkembangan teknologi, computer vision semakin banyak diterapkan pada berbagai bidang seperti keamanan (*face recognition* dan *surveillance*), industri (*defect detection*), kesehatan (diagnosis berbasis citra medis), serta biologi (identifikasi spesies). Perkembangan *deep learning* menjadikan *computer vision* jauh lebih efektif dibandingkan pendekatan tradisional berbasis fitur manual seperti SIFT atau HOG [46].

2.2 Citra Digital

Citra digital merupakan representasi visual dari suatu objek atau pemandangan yang disimpan dalam bentuk numerik melalui elemen-elemen terkecil yang disebut piksel . Setiap piksel menyimpan nilai intensitas cahaya atau warna dari bagian tertentu pada gambar. Proses pembentukan citra digital diawali dengan konversi dari citra analog menggunakan dua tahap utama, yaitu *sampling* dan *quantization*, sehingga citra dapat diproses oleh komputer [20].

Menurut penelitian yang diterbitkan oleh MDPI (2024), citra digital dapat didefinisikan di mana nilai fff pada koordinat spasial (x,y)(x, y)(x,y) merepresentasikan tingkat intensitas atau kecerahan dari titik tersebut. Jika nilai amplitudo dari f(x,y)f(x, y)f(x,y) terbatas pada sejumlah nilai diskret dan baik koordinat spasial maupun amplitudonya bersifat terhingga, maka citra tersebut dikategorikan sebagai citra digital [21].

Citra digital dapat diperoleh melalui berbagai perangkat sensor optik seperti kamera digital, *scanner*, *satellite imaging system*, maupun perangkat akuisisi citra lainnya. Keunggulan utama citra digital dibandingkan citra analog terletak pada kemudahan penyimpanan, manipulasi, dan analisis menggunakan teknik *image processing* dan *computer vision*. Teknologi ini memungkinkan

penerapan berbagai metode komputasi untuk mengenali objek, melakukan segmentasi, serta meningkatkan kualitas visual citra secara otomatis [20].Citra digital dapat diklasifikasikan berdasarkan beberapa kriteria, antara lain dimensi, komponen warna, kedalaman bit, dan format/representasi penyimpanan. Berdasarkan dimensi spasialnya, citra terbagi menjadi: citra dua dimensi (2D), citra tiga dimensi (3D) seperti pada citra medis atau volume scan, dan citra satu dimensi (1D) yang jarang digunakan secara langsung dalam aplikasi pengolahan citra [22].

Menurut sebuah studi terkini, "*types of image*" umumnya mencakup *binary image*, *grayscale image*, dan *colour image* (RGB), di mana *binary image* hanya memiliki dua tingkat (0 dan 1), *grayscale image* memiliki rentang nilai intensitas (misalnya 0-255 pada 8 bit), dan *colour image* terdiri dari beberapa saluran warna yang digabungkan (misalnya tiga saluran pada RGB) [23].

Klasifikasi lainnya berdasarkan kedalaman bit (*bit-depth*) yaitu jumlah bit yang digunakan untuk merepresentasikan satu piksel — misalnya 8 bit/piksel untuk citra grayscale, atau 24 bit/piksel (8 bit per saluran) untuk citra warna RGB. Citra dengan *bit-depth* lebih tinggi memungkinkan representasi yang lebih halus dan rentang warna atau intensitas yang lebih luas, tetapi juga membutuhkan ruang penyimpanan dan waktu pemrosesan lebih besar [23].

Berdasarkan sumber akuisisi, citra dapat dibagi menjadi citra alami (*natural image*) yang diambil dari kamera mode nyata, dan citra sintetis (*synthetic image*) yang dihasilkan oleh komputer atau simulasi. Jenis-jenis ini penting dalam penelitian karena karakteristik noise, tekstur, dan distribusi pikselnya berbeda, yang mempengaruhi teknik pengolahan seperti segmentasi dan klasifikasi [22].

2.3 Operasi Pengolahan Citra

Operasi pengolahan citra digital adalah rangkaian proses matematis yang digunakan untuk memanipulasi, menganalisis, dan meningkatkan kualitas citra digital agar dapat memberikan informasi yang lebih berguna. Tujuannya adalah untuk memperbaiki tampilan visual atau menyiapkan citra sebagai masukan (*Input*) bagi sistem analisis lanjutan seperti *object detection*, *pattern recognition*, dan *computer vision* [24].

Secara umum, operasi pengolahan citra dibagi menjadi tiga kategori utama:

1. Operasi titik Operasi ini memodifikasi nilai intensitas tiap piksel secara independen tanpa memperhatikan piksel di sekitarnya. Contohnya meliputi *contrast adjustment*, *brightness*

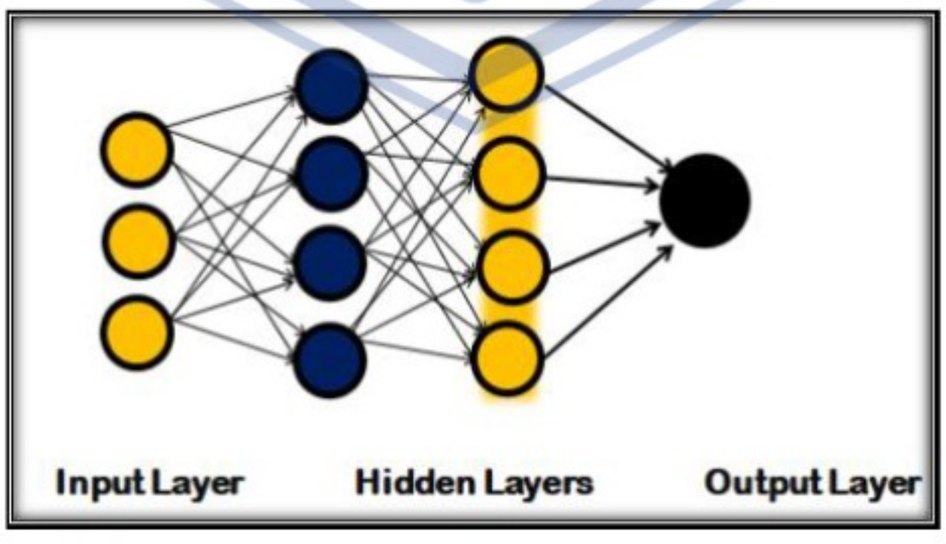
correction, *thresholding*, dan *histogram equalization* yang berfungsi untuk meningkatkan kontras dan kecerahan citra [21].

2. Operasi spasial Berbeda dengan operasi titik, operasi ini mempertimbangkan hubungan antar-piksel di sekitarnya dengan menggunakan *filter kernel* atau *mask*. Operasi seperti *smoothing*, *blurring*, *sharpening*, dan *edge detection* termasuk dalam kategori ini. Teknik ini banyak digunakan dalam penghilangan derau (*noise removal*) dan penajaman tepi citra (*edge enhancement*) [25].
3. Operasi geometrik Operasi ini melibatkan perubahan bentuk atau posisi citra, seperti *rotation*, *translation*, *scaling*, dan *reflection*. Operasi geometrik penting dalam proses *data augmentation* karena dapat menghasilkan variasi citra baru yang meningkatkan performa model pembelajaran mesin [26].

Selain itu, terdapat pula transformasi domain frekuensi seperti *Fourier Transform* dan *Wavelet Transform*, yang digunakan untuk menganalisis pola frekuensi pada citra. Transformasi ini berguna dalam aplikasi seperti *image compression*, *medical imaging*, serta penghilangan derau frekuensi tinggi (*high-frequency noise filtering*) [27].

2.4 Artificial Neural Network (ANN)

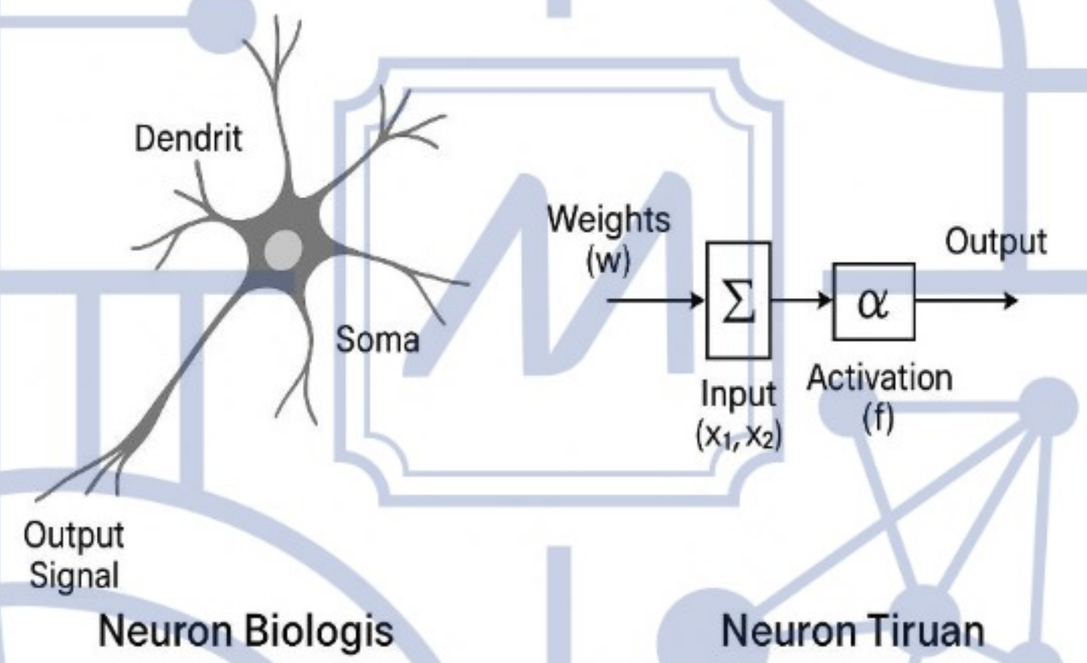
Artificial Neural Network, adalah model komputasi yang terinspirasi oleh struktur dan fungsi otak manusia, di mana sekumpulan unit pemrosesan dasar (*neuron*) saling terhubung untuk mengenali pola, melakukan klasifikasi, atau memprediksi *Output* dari *Input* tertentu. ANN merupakan komponen kunci dalam metode pembelajaran mesin dan *deep learning*, dan telah banyak digunakan dalam pengolahan citra, pengenalan objek, serta deteksi spesies dalam bidang biologi dan medis [39].



Gambar 2. 1 *Artificial Neural Network* (ANN)

2.4.1 Neuron

Neuron dalam *Artificial Neural Network* (ANN) merupakan unit dasar pemrosesan informasi yang terinspirasi dari cara kerja *neuron* biologis pada otak manusia. Pada sistem biologis, *neuron* terdiri atas tiga bagian utama: dendrit, badan sel (*soma*), dan akson. Dendrit berfungsi menerima sinyal dari *neuron* lain, kemudian sinyal tersebut diolah di soma dan diteruskan melalui akson menuju *neuron* lain. Proses ini menyerupai bagaimana informasi disalurkan dalam jaringan saraf buatan, di mana setiap *neuron* menerima *Input*, memprosesnya, dan mengirimkan *Output* ke *neuron* berikutnya. Gambar berikut menunjukkan perbandingan antara *neuron* biologis dan *neuron* tiruan (*Artificial Neuron*) yang menjadi dasar desain jaringan saraf tiruan modern:



Gambar 2. 2 Visualisasi Perbandingan *Neuron* Biologis dan *Neuron* Tiruan

Secara konseptual, setiap *neuron* tiruan (*artificial neuron*) menerima sejumlah *Input* (*x* 1, *x* 2, . . ., *x n*) (x 1 , x 2 ..., x n) yang masing-masing memiliki bobot (*w* 1, *w* 2, . . ., *w n* w 1 , w 2 ..., w n). Semua *Input* tersebut dijumlahkan bersama dengan nilai bias (b b), kemudian hasilnya diproses melalui fungsi aktivasi (f f) untuk menghasilkan *Output* (y y).

$$y = \mathcal{F}\left(\sum_{i=1}^{n} w_i x_i + b\right) \qquad \text{.....(0.1)}$$

Keterangan:

x i = *Input* ke-i

w i = bobot masing-masing *Input*

b b = bias yang menggeser fungsi aktivasi

f f = fungsi aktivasi *non-linear*

y y = *Output* neuron

Hubungan antar-*neuron* ini membentuk jaringan (*network*) yang saling terhubung dan mampu "belajar" dari data dengan menyesuaikan bobot-bobotnya selama proses pelatihan (*training*). Mekanisme ini dikenal sebagai backpropagation, yang memungkinkan jaringan meminimalkan kesalahan prediksi dan meningkatkan akurasi model [40].

2.4.2 Layer

Layers dalam ANN mengelompokkan *neuron-neuron* menjadi tingkatan fungsi yang berbeda. Tiga jenis lapisan utama adalah:

1. *Input layer*: menerima data fitur mentah sebagai *Input* ke jaringan.
2. Hidden *layer*(s): satu atau lebih lapisan tersembunyi yang melakukan pemrosesan fitur dan ekstraksi pola.
3. *Output layer*: menghasilkan *Output* akhir berupa prediksi atau klasifikasi. Semakin banyak *hidden layer*, maka jaringan menjadi lebih "dalam" sehingga disebut *Deep Neural Network (DNN)*. Arsitektur yang dalam memungkinkan model mempelajari representasi kompleks dari data, seperti fitur-morfologi nyamuk dalam citra digital [19].

2.4.3 Activation Function

Fungsi aktivasi merupakan komponen penting dalam *Artificial Neural Network (ANN)* yang menentukan keluaran dari sebuah *neuron* setelah menerima sinyal agregasi. Fungsi ini menambahkan *non-linearitas* ke dalam jaringan, memungkinkan ANN untuk mempelajari hubungan yang kompleks antara *Input* dan *Output* yang tidak dapat diselesaikan oleh model linier [42].

Beberapa fungsi aktivasi yang umum digunakan dalam model modern antara lain:

1. *Sigmoid Function*

$$\mathcal{F}(\mathcal{X}) = \frac{1}{1 + e^{-x}} \qquad \text{.....(0.2)}$$

Fungsi ini menghasilkan nilai antara 0 dan 1, cocok untuk kasus klasifikasi biner. Namun, pada jaringan yang sangat dalam, sigmoid rentan terhadap masalah *vanishing gradient* karena gradiennya mendekati nol untuk nilai ekstrem.

2. ReLU (*Rectified Linear Unit)*

$$\mathcal{F}(\mathrm{x}) = \max(0, \mathrm{x}) \qquad \text{.....(0.3)}$$

ReLU menjadi fungsi aktivasi paling populer dalam *deep learning* karena kesederhanaannya dan kemampuannya mengatasi *vanishing gradient*. Versi turunannya seperti *Leaky* ReLU, *Parametric* ReLU (PReLU), dan ELU (*Exponential Linear Unit*) juga banyak digunakan untuk meningkatkan stabilitas pelatihan [43].

3. *Softmax Function*

Fungsi ini digunakan pada *Output layer* model klasifikasi multikelas untuk menghasilkan distribusi probabilitas dari setiap kelas:

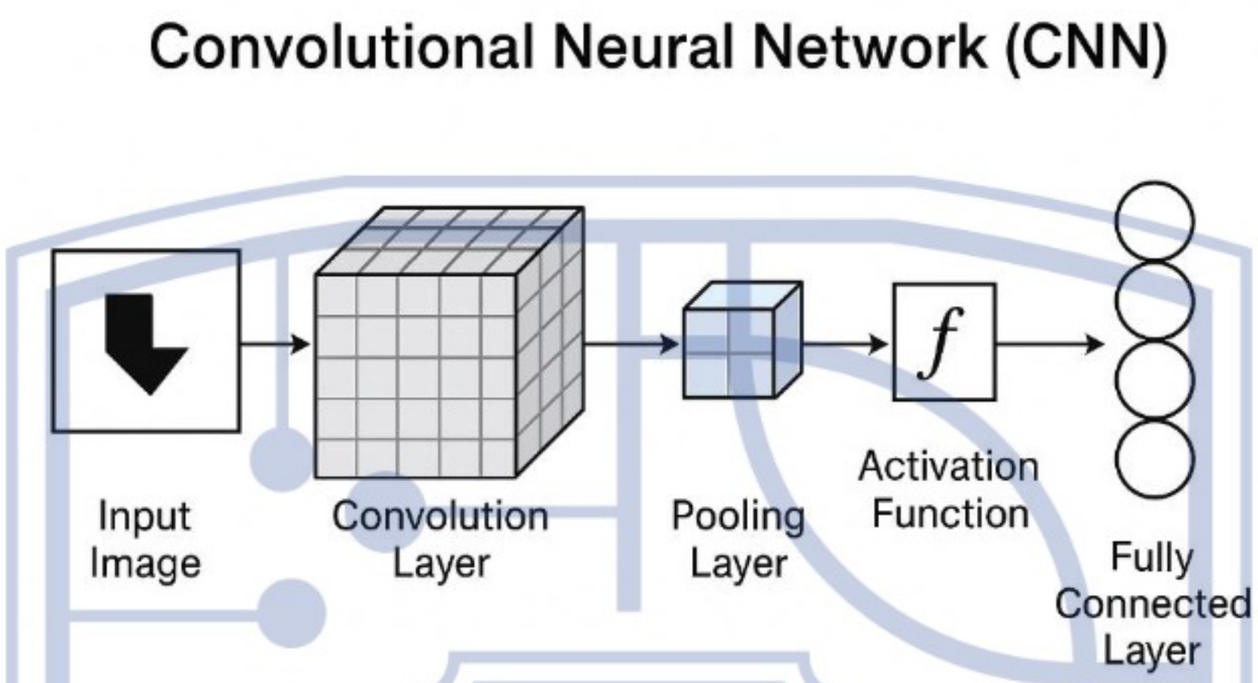
$$\mathcal{F}\left(\mathcal{X}_{\mathrm{i}}\right) = \frac{e^{\mathcal{X}_{\mathrm{i}}}}{\sum_{\mathcal{J}=1}^{n} e^{\mathrm{x_j}}} \qquad \text{.....(0.4)}$$

Softmax memastikan bahwa total seluruh *Output* bernilai 1, sehingga dapat diinterpretasikan sebagai probabilitas dari setiap kelas target [44].

2.5 Convolutional Neural Network

Struktur umum *Convolutional Neural Network* (CNN) terdiri dari beberapa komponen utama seperti *convolution layer*, *pooling layer*, *activation function*, dan *fully connected layer*. Lapisan konvolusi berfungsi untuk mengekstraksi fitur penting dari citra, sedangkan lapisan *pooling* digunakan untuk mereduksi dimensi data tanpa menghilangkan informasi penting. Fungsi aktivasi seperti *ReLU (Rectified Linear Unit)* membantu memperkenalkan sifat non-linear pada jaringan, dan lapisan *fully connected* menggabungkan semua fitur yang telah diekstraksi untuk menghasilkan prediksi akhir. Kombinasi lapisan-lapisan tersebut membuat CNN sangat efektif dalam berbagai tugas pengolahan citra, seperti klasifikasi gambar, deteksi objek, serta segmentasi semantik [47][48].

CNN telah diterapkan secara luas dalam beragam bidang, misalnya untuk pengenalan wajah, diagnosis penyakit berbasis citra medis, sistem kendaraan otonom, dan sistem pengawasan cerdas, karena kemampuannya mengenali pola visual secara otomatis dengan tingkat akurasi tinggi.



Gambar 2. 3 *Convolutional Neural Network* (CNN)

2.6 Object Detection

Object detection merupakan salah satu cabang penting dalam *computer vision* yang bertujuan untuk mengidentifikasi dan menentukan lokasi objek di dalam sebuah citra atau video. Berbeda dengan *image classification* yang hanya mengenali kelas dari seluruh citra, *object detection* menghasilkan koordinat posisi objek dalam bentuk *bounding box* beserta label kelasnya.

Perkembangan metode *object detection* telah mengalami kemajuan pesat seiring dengan evolusi *deep learning*. Awalnya, pendekatan berbasis *handcrafted features* seperti *Haar-like features* dan *Histogram of Oriented Gradients (HOG)* digunakan dalam model seperti *Viola–Jones detector* untuk deteksi wajah. Namun, metode ini memiliki keterbatasan dalam mendeteksi objek kompleks dengan variasi ukuran, orientasi, dan pencahayaan.

Kemunculan *Convolutional Neural Network (CNN)* mengubah paradigma deteksi objek secara signifikan. CNN memungkinkan ekstraksi fitur secara otomatis dari citra tanpa perlu perancangan manual. Model deteksi modern kemudian berkembang menjadi dua kategori utama, yaitu *Two-stage detectors* (misalnya R-CNN, *Fast* R-CNN, dan *Faster* R-CNN) serta *one-stage detectors* (misalnya YOLO dan SSD) yang memiliki keunggulan masing-masing dalam hal akurasi dan kecepatan [49].

1. *Two-stage detector* (R-CNN, *Fast* R-CNN)

Two-stage detector adalah arsitektur deteksi objek yang memproses citra dalam dua tahap. Tahap pertama menghasilkan sejumlah region proposals yang berpotensi mengandung objek, sedangkan tahap kedua melakukan klasifikasi pada setiap proposal tersebut.

Pendekatan ini pertama kali diperkenalkan oleh Ross Girshick melalui *Regions with* CNN *features* (R-CNN). Model ini mengekstraksi sekitar 2000 *region* proposals menggunakan metode *Selective Search*, kemudian setiap *region* diklasifikasikan menggunakan CNN. Meskipun akurat, R-CNN memiliki kelemahan pada efisiensi komputasi.

Untuk mengatasi hal tersebut, *Fast* R-CNN dikembangkan dengan menyatukan ekstraksi fitur dan klasifikasi ke dalam satu jaringan CNN, sehingga seluruh citra hanya diproses sekali [50]. Peningkatan berikutnya dilakukan oleh *Faster* R-CNN [51], yang memperkenalkan *Region Proposal Network* (RPN) guna menghasilkan proposal secara otomatis dan *end-to-end*, meningkatkan kecepatan tanpa menurunkan akurasi.

Model-model *two-stage* ini memiliki performa tinggi pada deteksi objek yang kompleks dan tetap menjadi acuan utama untuk penelitian berbasis presisi tinggi.

2. *One-Stage Detector* (YOLO, SSD, dsb)

Berbeda dengan pendekatan *two-stage*, *one-stage detector* langsung memprediksi kelas dan posisi objek dalam satu langkah jaringan tunggal. Arsitektur ini dirancang untuk mencapai deteksi *real-time* dengan kompromi tertentu terhadap akurasi.

Salah satu model paling terkenal adalah *You Only Look Once (YOLO)* yang diperkenalkan oleh [52]. YOLO membagi citra menjadi grid dan pada setiap sel grid, jaringan CNN memprediksi *bounding box* beserta probabilitas kelas secara bersamaan. Pendekatan ini memungkinkan kecepatan deteksi yang sangat tinggi.

Selanjutnya, varian seperti *YOLOv2*, *YOLOv3*, hingga *YOLOv8* terus dikembangkan untuk meningkatkan keseimbangan antara akurasi dan kecepatan, dengan perbaikan pada *backbone network*, *feature fusion*, dan *anchor box optimization*. Selain YOLO, model lain seperti *Single Shot MultiBox Detector (SSD)* juga mengadopsi konsep satu tahap deteksi dengan menambahkan *multi-scale feature maps* agar lebih sensitif terhadap objek kecil.

Menurut survei terbaru oleh [49], pendekatan *one-stage* telah mendominasi aplikasi dunia nyata seperti pengawasan lalu lintas, sistem keamanan, dan deteksi medis karena kemampuannya melakukan inferensi secara cepat tanpa perangkat keras mahal.

Aspek	YOLOv5	YOLOv7	YOLOv8
Pendekatan	Anchor-based	Anchor-based	Anchor-free
Backbone	CSPDarknet	E-ELAN	C2f
Kecepatan Inferensi	Cepat	Sangat cepat	Lebih efisien
Deteksi Objek Kecil	Baik	Baik	Lebih optimal
Stabilitas Training	Baik	Baik	Lebih stabil
Kompleksitas Model	Sedang	Tinggi	Lebih ringan
Kelebihan Utama	Stabil	Akurasi tinggi	Efisien dan modern

Berdasarkan Tabel 2.X, YOLOv8 dipilih dalam penelitian ini karena memiliki pendekatan anchor-free detection dan struktur backbone C2f yang lebih efisien dalam mendeteksi objek kecil seperti nyamuk dibandingkan versi sebelumnya.

2.7 YOLO

You Only Look Once (YOLO) merupakan algoritma *object detection* yang pertama kali diperkenalkan oleh Joseph Redmon pada tahun 2016. Algoritma ini menggunakan pendekatan *one-stage object detection*, yaitu melakukan proses deteksi lokasi objek dan klasifikasi kelas objek secara bersamaan dalam satu kali proses prediksi sehingga menghasilkan kecepatan deteksi yang tinggi dibandingkan metode *two-stage object detection* [2]

Seiring perkembangannya, algoritma YOLO mengalami berbagai penyempurnaan hingga hadir YOLOv8 yang dikembangkan oleh Ultralytics pada tahun 2023. YOLOv8 dirancang untuk meningkatkan akurasi deteksi, efisiensi komputasi, dan kecepatan inferensi melalui penyempurnaan arsitektur jaringan serta penggunaan pendekatan *anchor-free detection* [5] [28]

YOLOv8 menggunakan tiga komponen utama, yaitu *backbone*, *neck*, dan *detection head*. *Backbone* berfungsi mengekstraksi fitur-fitur penting dari citra masukan, seperti bentuk, tekstur, dan pola objek. Selanjutnya, *neck* menggabungkan informasi fitur dari berbagai tingkat resolusi menggunakan kombinasi *Feature Pyramid Network* (FPN) dan *Path Aggregation Network* (PAN), sehingga objek berukuran kecil maupun besar dapat dikenali dengan lebih baik. Setelah itu, *detection head* menghasilkan prediksi berupa lokasi objek (*bounding box*), kelas objek, serta nilai *confidence score* untuk setiap objek yang berhasil dideteksi [5]

2.7.1 Cara Kerja Yolov8

Cara kerja algoritma YOLOv8 dimulai ketika citra masukan (*input image*) diberikan ke dalam model. Sebelum diproses, citra akan melalui tahap *preprocessing*, seperti penyesuaian ukuran (*resize*) sesuai ukuran masukan model dan normalisasi nilai piksel agar proses komputasi menjadi lebih efisien.

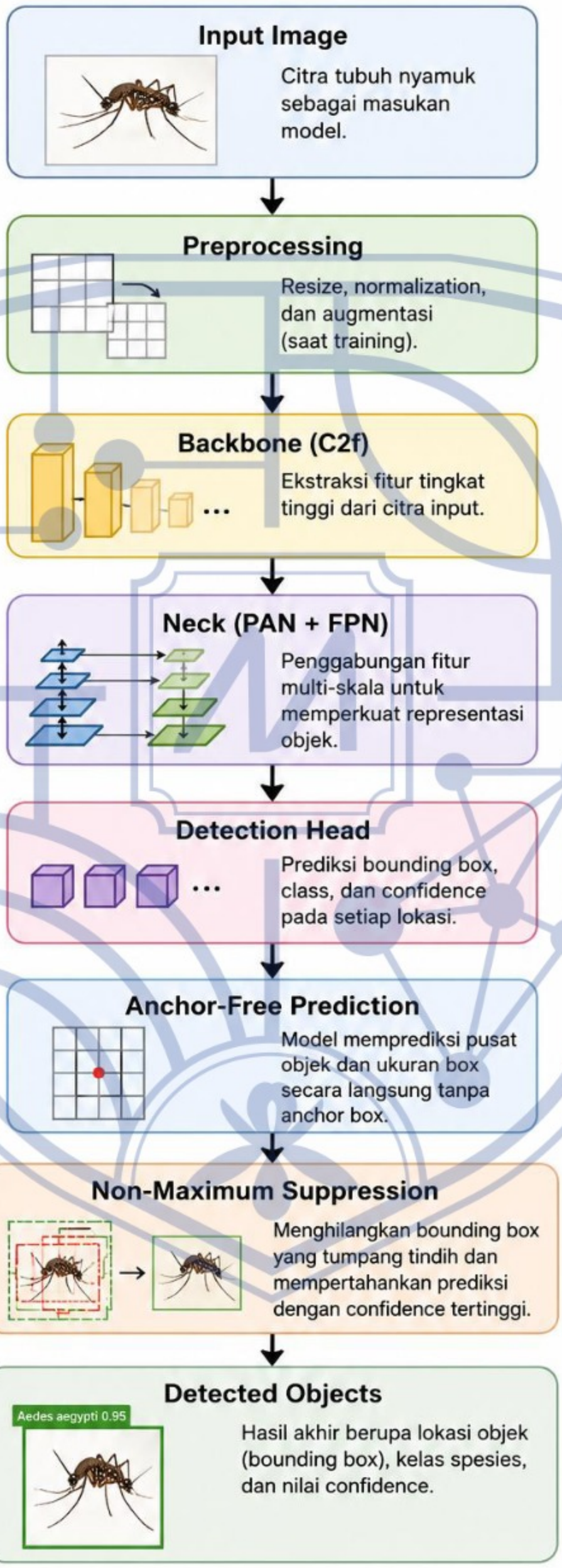
Pada tahap berikutnya, citra diproses oleh *backbone* C2f untuk melakukan ekstraksi fitur. *Backbone* bertugas mengenali karakteristik penting dari objek, seperti bentuk tubuh, pola warna, tekstur, dan tepi objek. Hasil ekstraksi fitur ini kemudian diteruskan ke bagian *neck*.

Bagian *neck* menggunakan kombinasi *Feature Pyramid Network* (FPN) dan *Path Aggregation Network* (PAN) untuk menggabungkan fitur dari berbagai tingkat resolusi. Proses ini memungkinkan model mempertahankan informasi objek berukuran kecil tanpa kehilangan informasi semantik dari objek berukuran besar, sehingga meningkatkan kemampuan deteksi pada berbagai skala objek [5]

Selanjutnya, fitur yang telah digabungkan diteruskan ke *detection head*. Pada tahap ini model memprediksi lokasi objek dalam bentuk *bounding box*, menentukan kelas objek, serta menghitung nilai *confidence score* untuk setiap hasil prediksi. Berbeda dengan versi sebelumnya, YOLOv8 menggunakan pendekatan *anchor-free detection*, sehingga model tidak lagi bergantung pada *anchor box* yang telah ditentukan sebelumnya. Pendekatan ini membuat proses pelatihan menjadi lebih sederhana serta meningkatkan akurasi deteksi, terutama pada objek berukuran kecil [5]

Tahap terakhir adalah *Non-Maximum Suppression* (NMS), yaitu proses untuk menghilangkan beberapa *bounding box* yang saling bertumpang tindih dan mempertahankan prediksi dengan nilai *confidence score* tertinggi. Hasil akhir dari proses ini berupa lokasi objek (*bounding box*), nama kelas objek, dan nilai *confidence score* yang ditampilkan pada citra hasil deteksi.

Secara umum, alur kerja YOLOv8 dapat digambarkan sebagai berikut.



2.8 Metrik Evaluasi

Dalam studi deteksi objek pencapaian model tidak hanya diukur dari akurasi klasifikasi, tetapi juga dari kemampuannya melakukan lokalisasi objek (menentukan *bounding box* yang tepat). Berikut adalah beberapa metrik evaluasi utama yang sering digunakan.

1. Definisi Dasar

Sebelum ke metrik lengkap, penting memahami istilah-istilah berikut:

a. *True Positive* (TP): Prediksi model yang dianggap benar—yaitu objek terdeteksi dan cocok dengan ground-truth (kelas benar dan *bounding box* cukup overlap).

b. *False Positive* (FP): Prediksi model yang salah—misalnya mendeteksi objek yang sebenarnya tidak ada atau kelasnya salah atau *bounding box* tidak cukup overlap.

c. *False Negative* (FN): Objek nyata (ground-truth) yang tidak terdeteksi oleh model.

d. *Intersection over Union* (IoU): Ukuran overlap antara *bounding box* yang diprediksi dan *bounding box* ground-truth.

2. *Precision, Recall,* dan *F1-Score*

Semakin tinggi IoU, semakin baik posisi dan ukuran prediksi mendekati kebenaran. [30].

$$Precision = \frac{TP}{TP + FP} \quad \text{.....(0.5)}$$

a. Menunjukkan proporsi prediksi positif yang benar.

$$Recall = \frac{TP}{TP + FN} \quad \text{.....(0.6)}$$

b. Menunjukkan proporsi objek nyata yang berhasil dideteksi

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad \text{.....(0.7)}$$

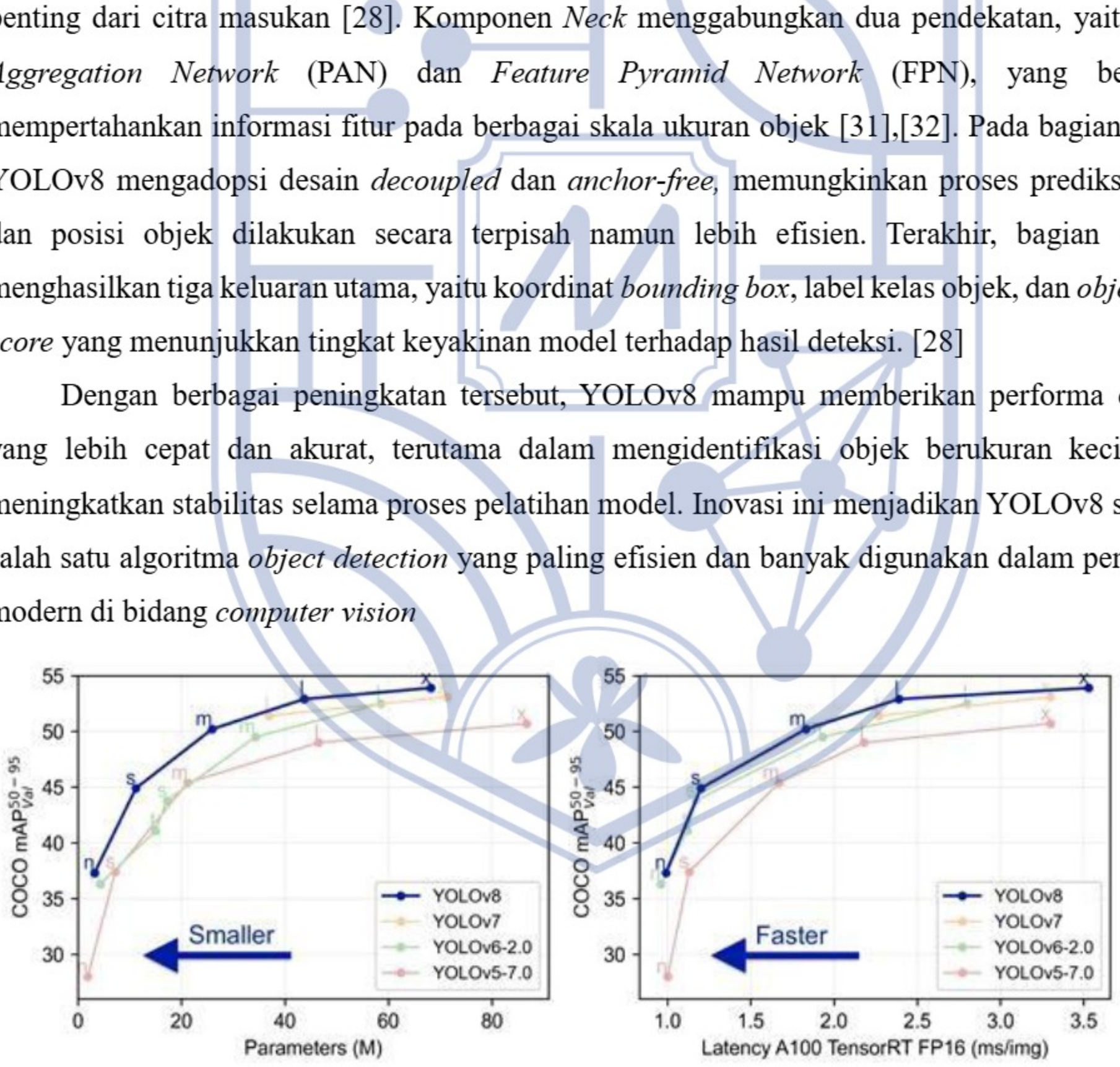
c. Merupakan rata-harmonik antara precision dan recall sehingga cocok bila keduanya penting. [30].

3. *Mean Average Precision* (mAP)

Untuk deteksi objek dengan berbagai kelas dan berbagai ukuran objek, metrik yang sering dipakai adalah *Average Precision (AP)* per kelas, kemudian dirata-rata menjadi *Mean Average Precision (mAP)*. AP mengukur area di bawah kurva *Precision-Recall* untuk satu kelas. Contoh: dalam tantangan seperti COCO (*Common Objects in Context*), sering digunakan mAP pada berbagai *threshold* IoU (misalnya IoU = 0.50 hingga 0.95) sebagai metrik utama [30].

Arsitektural YOLOv8 terdiri atas empat komponen utama, yakni *Backbone, Neck, Head*, dan *Output*. Bagian *Backbone* menggunakan struktur C2f (*Cross Stage Partial Fusion*) yang merupakan hasil pengembangan dari CSPDarknet, berfungsi untuk mengekstraksi fitur-fitur penting dari citra masukan [28]. Komponen *Neck* menggabungkan dua pendekatan, yaitu *Path Aggregation Network* (PAN) dan *Feature Pyramid Network* (FPN), yang berperan mempertahankan informasi fitur pada berbagai skala ukuran objek [31],[32]. Pada bagian *Head*, YOLOv8 mengadopsi desain *decoupled* dan *anchor-free,* memungkinkan proses prediksi kelas dan posisi objek dilakukan secara terpisah namun lebih efisien. Terakhir, bagian *Output* menghasilkan tiga keluaran utama, yaitu koordinat *bounding box*, label kelas objek, dan *objectness score* yang menunjukkan tingkat keyakinan model terhadap hasil deteksi. [28]

Dengan berbagai peningkatan tersebut, YOLOv8 mampu memberikan performa deteksi yang lebih cepat dan akurat, terutama dalam mengidentifikasi objek berukuran kecil serta meningkatkan stabilitas selama proses pelatihan model. Inovasi ini menjadikan YOLOv8 sebagai salah satu algoritma *object detection* yang paling efisien dan banyak digunakan dalam penelitian modern di bidang *computer vision*



Gambar 2. 4 Evaluasi YOLOv8 (Sumber: Dokumentasi YOLOv8 oleh *Ultralytics*) [28]

Deteksi objek berukuran kecil merupakan salah satu tantangan utama dalam *object detection* karena objek kecil memiliki jumlah piksel yang terbatas, sehingga informasi spasialnya mudah hilang selama proses *downsampling* pada jaringan konvolusional. Ketika resolusi citra diperkecil melalui operasi *stride convolution* atau *pooling*, detail fitur halus dari objek kecil cenderung berkurang secara signifikan [5][8].

YOLOv8 mengatasi permasalahan tersebut melalui beberapa mekanisme arsitektural. Pertama, penggunaan *Feature Pyramid Network* (FPN) memungkinkan penggabungan fitur dari berbagai tingkat resolusi. *Layer* dengan resolusi tinggi yang berasal dari tahap awal jaringan tetap dipertahankan dan digabungkan dengan *layer* resolusi rendah yang mengandung informasi semantik lebih kuat. Proses ini membantu menjaga detail spasial yang penting untuk mendeteksi objek kecil [31].

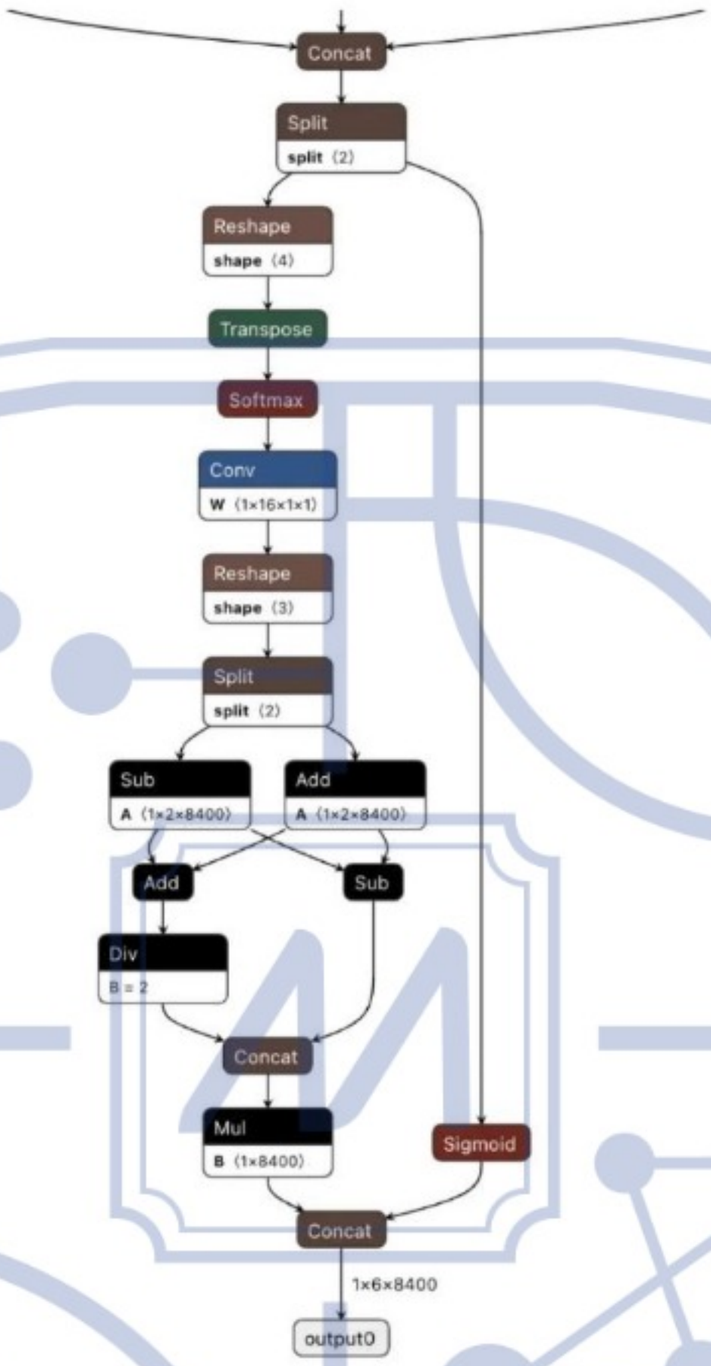
Kedua, YOLOv8 mengintegrasikan *Path Aggregation Network* (PAN) yang memperkuat aliran informasi dari lapisan bawah ke lapisan atas, sehingga fitur resolusi tinggi dapat berkontribusi langsung terhadap proses prediksi akhir. Kombinasi FPN dan PAN menghasilkan representasi fitur multi-skala yang lebih kaya dan stabil terhadap variasi ukuran objek [32]
Ketiga, YOLOv8 mengadopsi pendekatan *anchor-free detection*, di mana model tidak lagi menggunakan *anchor box* tetap dengan ukuran tertentu. Pada metode *anchor-based*, objek kecil seringkali tidak cocok dengan ukuran anchor yang telah ditentukan sebelumnya, sehingga menurunkan akurasi deteksi. Dengan pendekatan anchor-free, YOLOv8 secara langsung memprediksi koordinat *bounding box* berdasarkan titik pusat objek (*center-based prediction*), sehingga lebih fleksibel terhadap variasi skala dan rasio aspek objek [5].

Selain itu, struktur *backbone* yang menggunakan modul C2f (*Cross Stage Partial with feature reuse*) membantu mempertahankan propagasi gradien yang lebih stabil serta menjaga distribusi informasi fitur penting selama proses pelatihan. Hal ini berkontribusi pada peningkatan sensitivitas model terhadap detail morfologi objek berukuran kecil [5]

1. *Anchor-Free Detection*

YOLOv8 adalah model *Anchor-Free Detection* model ini memprediksi langsusng pusat suatu objek, alih-alih offset dari kotak jangkar yang diketahui. Kotak jangkar merupakan bagian yang sangat rumit pada model YOLO sebelumnya, karena kotak tersebut mungkin mewakili distribusi kotak tolok ukur target tetapi bukan distribusi kumpulan data khusus. Deteksi bebas

jangkar mengurangi jumlah prediksi kotak, yang mempercepat Penekanan *Non-Maksimum* (NMS), langkah pasca-pemrosesan rumit yang menyaring deteksi kandidat setelah inferensi.

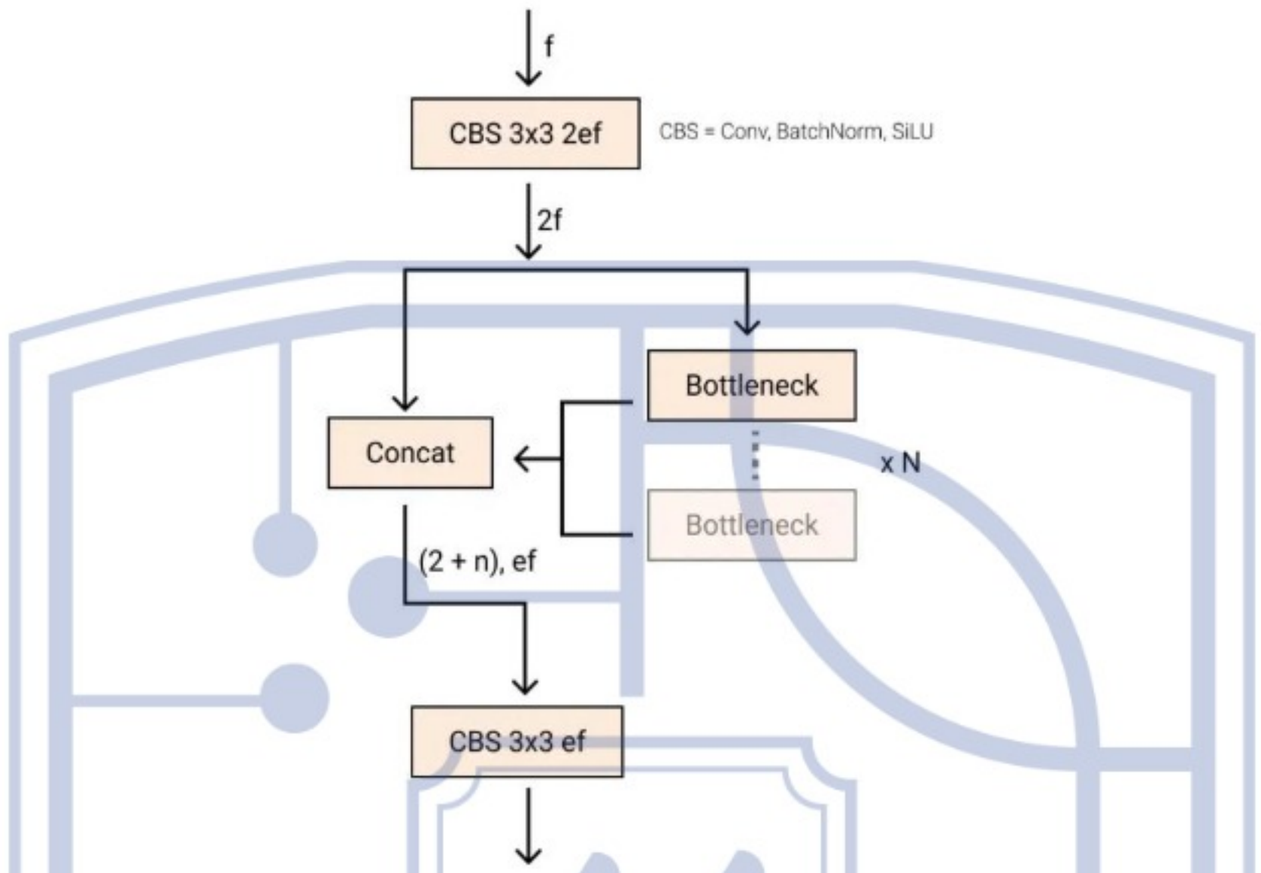


Gambar 2. 5 Visualisasi Kepala deteksi untuk YOLOv8 (Sumber: *Netron* dan dokumentasi YOLOv8) [28][33]

2. *Konvolusi* Baru (C2f *Module*)

Konvolusi pertama stem `6x6` digantikan oleh a `3x3`, blok penyusun utama diubah, dan C2f menggantikan C3 . Modul dirangkum dalam gambar di bawah ini, di mana "f" adalah jumlah fitur, "e" adalah laju ekspansi, dan CBS adalah blok yang terdiri dari a `Conv`, a , `*BatchNorm*` dan a `SiLU`. Dalam C2f, semua keluaran dari `*Bottleneck*` (nama keren untuk dua 3x3 convs dengan koneksi residual) digabungkan. Sementara di , C3 hanya keluaran terakhir `*Bottleneck*` yang digunakan. Sama `*Bottleneck*` seperti di YOLOv5, tetapi ukuran kernel konv pertama diubah dari `1x1`. `3x3` Dari informasi ini, kita dapat melihat bahwa YOLOv8 mulai kembali ke blok ResNet yang didefinisikan pada tahun 2015. Di bagian leher, fitur-fitur digabungkan secara

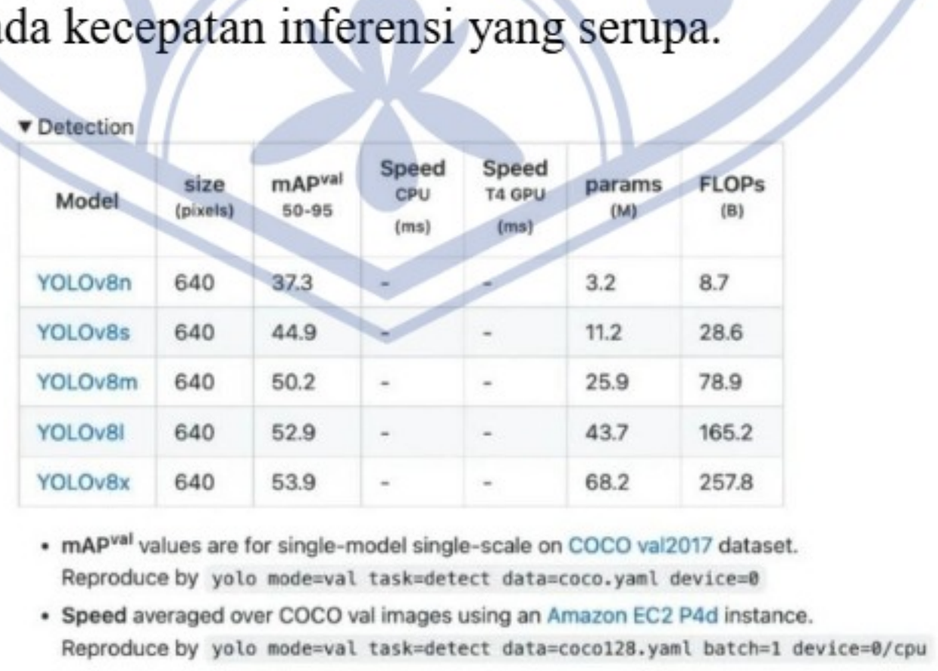
langsung tanpa memaksakan dimensi kanal yang sama. Hal ini mengurangi jumlah parameter dan ukuran keseluruhan tensor.



Gambar 2.6 Arsitektur Modul C2f pada YOLOv8 (Sumber: Dokumentasi YOLOv8 oleh *Ultralytics*) [28]

3. Peningkatan Akurasi YOLOv8

Penelitian YOLOv8 terutama dimotivasi oleh evaluasi empiris pada tolok ukur COCO . Seiring setiap bagian jaringan dan rutinitas pelatihan disempurnakan, eksperimen baru dijalankan untuk memvalidasi efek perubahan pada pemodelan COCO. COCO (*Common Objects in Context*) adalah tolak ukur standar industri untuk mengevaluasi model deteksi objek. Saat membandingkan model pada COCO, kami melihat nilai mAP dan pengukuran FPS untuk kecepatan inferensi. Model harus dibandingkan pada kecepatan inferensi yang serupa.



▼ Detection

Model	size (pixels)	mAP^{val} 50-95	Speed CPU (ms)	Speed T4 GPU (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	-	-	3.2	8.7
YOLOv8s	640	44.9	-	-	11.2	28.6
YOLOv8m	640	50.2	-	-	25.9	78.9
YOLOv8l	640	52.9	-	-	43.7	165.2
YOLOv8x	640	53.9	-	-	68.2	257.8

- mAP^{val} values are for single-model single-scale on COCO val2017 dataset.
  Reproduce by `yolo mode=val task=detect data=coco.yaml device=0`
- **Speed** averaged over COCO val images using an Amazon EC2 P4d instance.
  Reproduce by `yolo mode=val task=detect data=coco128.yaml batch=1 device=0/cpu`

Gambar 2. 6 Evaluasi COCO YOLOv8 (Sumber: Dokumentasi YOLOv8 oleh *Ultralytics*) [28]

Penambahan Mosaik Penelitian pembelajaran mendalam cenderung berfokus pada arsitektur model, tetapi rutinitas pelatihan dalam YOLOv5 dan YOLOv8 merupakan bagian penting dari keberhasilannya. YOLOv8 melakukan augmentasi gambar selama pelatihan daring.

Gambar 2. 7 Penambahan mosaik pada foto papan catur (Sumber: Dokumentasi YOLOv8 oleh *Ultralytics*) [28]

Setiap periode, model melihat variasi yang sedikit berbeda dari gambar yang telah diberikan. Salah satu augmentasi tersebut disebut augmentasi mosaik . Augmentasi ini dilakukan dengan menggabungkan empat gambar, memaksa model untuk mempelajari objek di lokasi baru, dalam oklusi parsial, dan terhadap piksel-piksel di sekitarnya yang berbeda.

Namun, augmentasi ini secara empiris terbukti menurunkan performa jika dilakukan selama keseluruhan rutinitas latihan. Sebaiknya dimatikan selama sepuluh periode latihan terakhir. Perubahan semacam ini merupakan contoh perhatian cermat yang diberikan pada pemodelan YOLO selama ini dalam repo YOLOv5 dan penelitian YOLOv8.

2.9 Nyamuk

Nyamuk termasuk dalam ordo *Diptera* dan famili *Culicidae*, yang secara global telah diidentifikasi sebanyak lebih dari 3.500 spesies dan subspecies dalam sekitar 44 genera [34]. Morfologi nyamuk betina umumnya mencakup probosis yang panjang untuk mengisap darah, sedangkan pada jantan antena-berbulu dan probosis yang lebih pendek digunakan untuk mengisap nektar bunga [11].

Siklus hidupnya terdiri dari empat tahap utama: telur → larva → pupa → imago (dewasa), di mana tahap larva dan pupa tumbuh dalam media perairan stagnan seperti genangan air, wadah terbuka, dan lingkungan lembap yang mendukung [12]. faktor lingkungan seperti suhu, kelembapan, serta kualitas air menjadi komponen penting yang menentukan tingkat keberhasilan pertumbuhan dan reproduksi nyamuk [11].

Lebih lanjut, nyamuk dikenal sebagai vektor utama berbagai penyakit menular yang menyerang manusia dan hewan. Beberapa genus penting, antara lain *Aedes*, *Anopheles*, dan *Culex*, berperan dalam penularan penyakit seperti demam berdarah dengue, *Zika*, *chikungunya*, malaria, *West Nile virus*, dan *filariasis* [11].

1. *Aedes*

Aedes merupakan kelompok nyamuk yang memiliki peranan penting secara medis karena kemampuannya dalam menularkan berbagai penyakit virus dan parasit pada manusia. Secara taksonomi, *Aedes* termasuk dalam famili *Culicidae* dan subfamili *Culicinae*, dengan distribusi luas di wilayah tropis dan subtropis di seluruh dunia [13].Spesies dari genus ini dikenal dengan perilaku menggigit di siang hari dan umumnya berkembang biak di habitat buatan manusia seperti wadah air bersih, pot bunga, kaleng bekas, dan tempat penampungan air lainnya [14].Selain itu, beberapa spesies penting dari genus *Aedes* seperti *Aedes aegypti* dan *Aedes albopictus* merupakan vektor utama penyakit Demam Berdarah Dengue(DBD), Chikungunya, *Zika Virus*, dan *Yellow Fever*. Karakteristik ekologis dan perilaku spesifik kedua spesies ini menjadi faktor penting dalam st*rate*gi pengendalian vektor secara efektif [15]. Secara morfologi, *Aedes aegypti* memiliki tubuh berwarna hitam kecokelatan dengan pola sisik putih keperakan pada bagian toraks yang menyerupai bentuk lyre (kecapi). Kaki memiliki belang putih yang khas sehingga mudah dibedakan dari genus nyamuk lainnya. Probosis berukuran panjang dan tubuh cenderung berukuran kecil hingga sedang.

2. *Anopheles*

Anopheles merupakan salah satu genus nyamuk dari famili *Culicidae* yang dikenal sebagai vektor utama penyakit malaria pada manusia. Genus ini memiliki distribusi yang luas di daerah tropis dan subtropis, termasuk wilayah Asia Tenggara dan Indonesia. Hingga saat ini, lebih dari 460 spesies *Anopheles* telah diidentifikasi di seluruh dunia, namun hanya sebagian yang berperan sebagai vektor malaria [16].

Secara morfologi, nyamuk Anopheles memiliki ciri khas posisi tubuh miring (membentuk sudut ±45°) saat hinggap, yang membedakannya dari genus *Aedes* dan *Culex*. Selain itu, palpus pada *Anopheles* betina memiliki panjang yang hampir sama dengan probosis, sedangkan pada genus lain palpus umumnya lebih pendek. Ciri morfologi ini sering digunakan sebagai dasar identifikasi manual oleh ahli entomologi [17].

3. *Culex*

Culex merupakan kelompok nyamuk yang juga memiliki peranan penting sebagai vektor penyakit. Spesies-spesies dalam genus ini seperti *Culex* quinquefasciatus dan *Culex* pipiens tersebar di wilayah tropis hingga sedang dan dikenal sebagai vektor utama untuk penyakit seperti *West Nile virus,* Japanese *encephalitis*, dan *filariasis* [18]. Tinjauan lain menunjukkan bahwa faktor lingkungan seperti suhu dan kelembapan sangat mempengaruhi rentang sebaran dan kelimpahan *Culex*, yang pada gilirannya berdampak pada risiko penularan penyakit [19]. Spesies *Culex* umumnya berkembang biak dalam air kumuh atau tercemar, got, dan kolam stagnan – berbeda dengan *Aedes* yang lebih banyak di wadah bersih – yang menjadikannya sasaran berbeda dalam st*rate*gi pengendalian vektor. Secara morfologi, *Culex quinquefasciatus* memiliki tubuh berwarna cokelat keabu-abuan tanpa pola belang mencolok pada kaki maupun toraks. Saat hinggap, posisi tubuh cenderung sejajar dengan permukaan. Antena dan probosis berkembang baik untuk mendukung aktivitas mencari sumber makanan.