BAB I
PENDAHULUAN

1.1 Latar Belakang

Industri akuakultur, khususnya budidaya ikan nila (*Oreochromis niloticus*), telah mengalami pertumbuhan yang signifikan dalam beberapa dekade terakhir (Dailami M, 2021). Ikan nila merupakan jenis ikan air tawar yang banyak digemari oleh masyarakat dan bernilai ekonomis tinggi, serta memiliki keunggulan lainnya yaitu mudah dibudidayakan dan dapat diolah menjadi makanan yang bernilai tinggi (Ilham I, 2024).

Budidaya ikan nila di Indonesia telah mengalami perkembangan yang sangat pesat dalam beberapa tahun terakhir. Berdasarkan data statistik produksi perikanan yang dirilis oleh Dinas Perikanan dan Kelautan, tercatat adanya lonjakan produksi yang signifikan, dengan peningkatan lebih dari 800.000 ton antara tahun 2010 hingga 2022. Angka ini mencerminkan kontribusi penting sektor budidaya ikan nila terhadap ketahanan pangan dan ekonomi perikanan di Indonesia, sekaligus memperkuat posisinya sebagai salah satu komoditas unggulan dalam industri akuakultur nasional. Namun, salah satu tantangan utama dalam budidaya ikan nila adalah penghitungan bibit ikan secara akurat dan efisien, terutama pada tahap awal produksi. Penghitungan yang tidak tepat dapat menyebabkan ketidakseimbangan dalam manajemen pakan, pemeliharaan, dan ruang, yang pada akhirnya dapat mempengaruhi produktivitas dan efisiensi operasional.

Metode penghitungan bibit ikan secara manual yaitu dengan memindahkan ikan dari wadah satu ke wadah yang lain sambil menghitungnya masih umum digunakan hingga saat ini dan seringkali memerlukan waktu yang lama dan rentan terhadap kesalahan manusia. Selain itu, variasi ukuran dan gerakan bibit ikan yang cepat dalam lingkungan air yang dinamis membuat tugas ini semakin menantang. Dalam upaya `untuk mengatasi tantangan ini, teknologi pengolahan citra digital dan kecerdasan buatan (AI) telah diusulkan sebagai solusi potensial.

Salah satu pendekatan yang mendapatkan perhatian dalam deteksi objek di berbagai bidang, termasuk akuakultur, adalah model deteksi berbasis *deep learning*

seperti *You Only Look Once* (YOLO). YOLO adalah salah satu metode deteksi objek *real-time* yang paling populer karena kecepatan dan akurasinya dalam mengenali dan melokalisasi objek dalam gambar atau video. Studi oleh (Redmon J, 2016) pertama kali memperkenalkan YOLO, yang sejak saat itu telah berkembang melalui beberapa versi, dengan setiap iterasi menawarkan peningkatan kinerja yang signifikan dalam hal presisi dan kecepatan deteksi.

Deteksi objek secara real-time selalu menjadi titik fokus penelitian di bidang visi komputer, yang bertujuan untuk memprediksi secara akurat kategori dan posisi objek dalam gambar dengan latensi rendah (Wang C.Y, 2024). Awalludin E.A (2020) menyatakan bahwa pemanfaatan teknologi komputer berbasis teknik pengolahan citra untuk menghitung jumlah larva ikan membutuhkan waktu pengolahan yang lebih singkat. Teknologi komputer yang digunakan sebagai solusi alternatif metode pendekatan penghitungan manual.

Penelitian lebih lanjut telah menunjukkan bahwa YOLO dapat diterapkan secara efektif dalam konteks akuakultur. Misalnya, (Kuswantori A, 2023) mengembangkan Deteksi dan klasifikasi ikan untuk sistem penyortiran otomatis dengan algoritma yolo yang dioptimalkan. Demikian pula, Wang A (2024) menggunakan YOLO untuk mendeteksi penyakit ikan dalam akuakultur, dengan hasil yang menunjukkan bahwa model ini mampu mengidentifikasi gejala penyakit dengan cepat dan akurat, membantu peternak ikan dalam mengambil tindakan preventif secara tepat waktu.

Meskipun banyak penelitian telah dilakukan untuk mengaplikasikan YOLO dalam berbagai aspek akuakultur, penerapannya dalam perhitungan otomatis bibit ikan nila masih relatif terbatas. Padahal, keberhasilan dalam penghitungan yang akurat pada tahap awal budidaya sangat krusial untuk memastikan distribusi pakan yang tepat, pemeliharaan yang efektif, dan optimalisasi ruang budidaya. Oleh karena itu, untuk mengisi celah tersebut, penulis mengajukan sebuah judul "**Sistem Perhitungan Otomatis Bibit Ikan Nila Menggunakan Metode YOLO (*You Only Look Once*)**".

Pengembangan sistem otomatisasi ini diharapkan tidak hanya meningkatkan efisiensi dan akurasi penghitungan bibit ikan, tetapi juga memberikan kontribusi yang signifikan terhadap peningkatan produktivitas dan manajemen budidaya ikan

nila secara keseluruhan. Dengan adanya penelitian ini, diharapkan bahwa teknologi pengolahan citra berbasis YOLO dapat diadopsi lebih luas dalam industri akuakultur, membawa kemajuan yang signifikan dalam praktik-praktik budidaya ikan di masa depan.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, masalah yang ditemukan, yaitu:

1. Bagaimana merancang sistem yang dapat menghitung otomatis bibit ikan nila menggunakan metode YOLO?
2. Berapa besar keakuratan metode YOLO untuk mendeteksi serta menghitung bibit ikan nila?

1.3 Tujuan Penelitian

Terkait dengan rumusan masalah, maka tujuan penelitiannya antara lain:

1. Untuk mengembangkan sistem yang berfungsi menghitung otomatis bibit ikan nila menggunakan metode YOLO.
2. Untuk mengetahui akurasi yang dihasilkan menggunakan metode YOLO dalam mendeteksi dan menghitung bibit ikan nila.

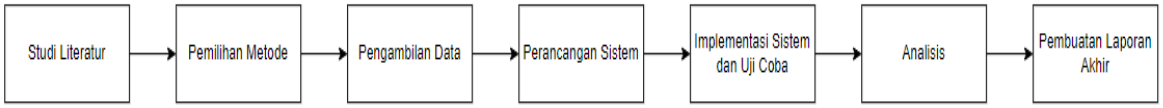
1.4 Manfaat Penelitian

1. Bagi Masyarakat
   a. Penelitian ini dapat membantu pembudidaya ikan nila untuk meningkatkan produktifitas budidayanya.
   b. Penelitian ini memudahkan pembudidaya ikan nila dala menghitung jumlah bibit secara cepat dan tepat.
2. Bagi peneliti, peneltiian ini berguna untuk menambah pengetahuan dan kemampuan mengenai pemrosesan citra dan perhitungan objek dalam sebuah gambar dengan *image processing*.
3. Bagi institusi Pendidikan, diharapkan penelitian ini dapat menambah kajian ilmu dan informasi bacaan mengenai perhitungan objek dengan *image processing*. Disamping itu hasil penelitian ini dapat dijadikan sebagai bahan referensi bagi mahasiswa yang akan melakukan penelitian yang sama dikemudian hari.

1.5 Ruang Lingkup Penelitian

1. Data yang diperoleh berupa data gambar dengan format JPEG (*Joint Photographic Experts Group*).
2. Data training menggunakan data primer.
3. Metode yang digunakan adalah YOLO (*You Only Look Once*) v10.
4. Objek penelitian adalah bibit ikan nila dengan ukuran 2-4 cm.
5. Dalam pengambilan gambar, bibit ikan diletakkan ke dalam sebuah wadah dengan latar warna yang jelas.
6. Pengambilan citra menggunakan kamera handphone dengan resolusi 1080p dengan kecepatan 30 fps.

1.6 Metode Penelitian



Gambar 1 Tahapan Penelitian

1. Studi Literatur

   Pada tahap ini, peneliti melakukan kajian terhadap literatur yang ada untuk memahami topik yang diteliti, metode yang telah digunakan sebelumnya, dan menemukan celah atau masalah yang perlu dipecahkan yaitu mengenai sistem perhitungan otomatis objek yang bergerak secara dinamis terutama bibit ikan nila.

2. Pemilihan Metode

   Setelah melakukan studi literatur, peneliti memilih metode yang tepat untuk digunakan dalam penelitian yang berdasarkan pada temuan dari studi literatur dan tujuan penelitian serta dengan mempertimbangkan kebaruan metode dan teknologi yang akan digunakan.

3. Pengambilan Data

   Pengambilan data dilakukan di ruang tertutup dengan pencahayaan yang sepenuhnya bergantung pada blitz kamera untuk menjaga konsistensi cahaya. Ini memungkinkan pengaturan pencahayaan yang bebas dari cahaya eksternal. Dengan menggunakan blitz, setiap gambar memiliki tingkat pencahayaan yang sama, yang membuat bibit ikan nila lebih jelas, terutama

karena berada dalam wadah gelap yang memberikan kontras tinggi dengan objek. Kontras ini sangat penting untuk membantu model YOLOv10 yang akan dilatih menemukan objek.

Untuk mengambil gambar bibit ikan dari berbagai sudut pandang, pengambilan gambar dilakukan dalam posisi tegak lurus diatas wadah dan menyamping wadah. Pengambilan gambar bibit ikan dari berbagai sudut pandang, yaitu tegak lurus di atas wadah dan **menyamping** wadah, bertujuan untuk menciptakan dataset yang beragam sehingga model YOLOv10 dapat mengenali objek dalam berbagai kondisi. Pada posisi tegak lurus, kamera ditempatkan secara vertikal di atas wadah untuk menghasilkan tampilan bird's-eye view.

Pendekatan ini memungkinkan model melihat keseluruhan bentuk ikan tanpa distorsi perspektif, meminimalkan noise dari latar belakang, dan membantu dalam pengukuran dimensi ikan secara lebih akurat. Posisi ini juga melatih model untuk mengenali ikan yang terlihat sepenuhnya tanpa hambatan. Sebaliknya, pada posisi menyamping, kamera ditempatkan secara horizontal atau miring terhadap wadah, memberikan sudut pandang lateral yang mensimulasikan bagaimana ikan terlihat dalam skenario nyata, seperti tumpang tindih atau sebagian terhalang. Sudut pandang menyamping ini memperkaya data dengan variasi bentuk dan orientasi ikan, melatih model untuk mengenali objek yang tidak sepenuhnya terlihat, dan mencerminkan kondisi kompleks di kolam budidaya.

Dengan mengombinasikan kedua posisi ini, dataset yang dihasilkan akan lebih realistis dan beragam, memastikan model YOLOv10 mampu mendeteksi ikan dalam berbagai orientasi, posisi, dan kondisi lingkungan, meningkatkan akurasi dan fleksibilitas dalam aplikasi nyata.

Gambar 2 (a) Proses Pengambilan Data; (b) Ilustrasi Pengambilan Data

Gambar diambil berulang kali untuk meningkatkan volume data yang representatif, dengan variasi kondisi bibit. Teknik ini memastikan bahwa dataset yang dikumpulkan memiliki variasi yang cukup untuk melatih model YOLOv10 secara efektif dalam mendeteksi dan menghitung bibit ikan nila dalam berbagai skenario.

Berikut merupakan contoh data citra yang akan digunakan sebagai sampel pada penelitian ini:

(a) (b)

Gambar 3 Contoh Data Citra (a) Tampak Depan (b)Tampak Samping

4. Perancangan Sistem

   Setelah data terkumpul, peneliti merancang sistem yang akan digunakan. Ini termasuk mendefinisikan spesifikasi sistem, alur kerja, dan komponen yang dibutuhkan.

5. Implementasi dan Uji Coba

   Pada tahap ini, sistem yang telah dirancang diimplementasikan, yaitu algoritma YOLOv10 untuk memdeteksi dan menghitung objek bibit ikan nila dengan Bahasa pemrograman python. Peneliti kemudian melakukan uji coba untuk memastikan sistem berfungsi sesuai yang diharapkan.

6. Analisis

   Setelah uji coba, peneliti melakukan analisis terhadap hasil yang diperoleh. Analisis ini penting untuk mengevaluasi performa sistem dan apakah tujuan penelitian tercapai.

7. Pembentukan Laporan Akhir

   Pada tahap akhir, peneliti menyusun laporan akhir yang merangkum seluruh proses penelitian, temuan, dan kesimpulan dalam bentuk skripsi.

BAB II
TINJAUAN PUSTAKA

2.1 Bibit Ikan Nila

Bibit ikan nila (*Oreochromis niloticus*) adalah tahap awal dalam siklus hidup ikan nila yang biasanya berada pada fase larva hingga benih. Ikan nila merupakan salah satu komoditas utama di sektor perikanan, baik di tingkat nasional maupun internasional, karena tingkat pertumbuhannya yang cepat, kemampuannya beradaptasi dengan berbagai kondisi lingkungan, dan tingginya nilai ekonomis.

Bibit ikan nila adalah hasil dari proses pemijahan yang dapat dilakukan secara alami maupun menggunakan teknik pembenihan buatan. Dalam metode alami, pemijahan biasanya dilakukan di kolam pemijahan dengan pengaturan lingkungan yang mendukung, sedangkan pembenihan buatan sering menggunakan metode yang lebih terkontrol seperti *hatchery* untuk meningkatkan efisiensi dan kualitas bibit yang dihasilkan.

Gambar 4 Bibit Ikan Nila

Secara morfologi, bibit ikan nila memiliki tubuh yang kecil, dengan panjang yang bervariasi antara beberapa milimeter hingga beberapa sentimeter, tergantung pada tahap perkembangan. Pada fase awal, bibit ikan nila memiliki transparansi tubuh sebagian, yang mempermudah pengamatan pertumbuhan organ dalam. Meskipun kecil, bibit ikan nila telah memiliki kemampuan untuk mencari makan

secara mandiri, terutama pada pakan alami seperti plankton. Seiring waktu, bibit ini dapat diberi pakan buatan untuk mendorong pertumbuhan lebih lanjut.

Pemeliharaan bibit ikan nila memerlukan perhatian khusus terhadap beberapa faktor penting, seperti kualitas air, pakan, kepadatan populasi, dan pengelolaan lingkungan. Kualitas air yang buruk dapat mempengaruhi tingkat kelangsungan hidup bibit, sementara pemberian pakan yang tidak sesuai akan menghambat pertumbuhan. Kepadatan populasi juga harus dikontrol agar bibit dapat tumbuh optimal tanpa menghadapi kompetisi sumber daya yang berlebihan. Bibit ikan nila biasanya dipelihara hingga mencapai ukuran tertentu, umumnya 5–10 cm, sebelum dipindahkan ke kolam pembesaran untuk budidaya lebih lanjut.

Dalam konteks budidaya, bibit ikan nila memiliki peran yang sangat krusial karena keberhasilan tahap pembibitan akan sangat menentukan produktivitas pada tahap pembesaran. Oleh karena itu, keberlanjutan produksi bibit ikan nila yang berkualitas tinggi menjadi salah satu prioritas utama dalam industri perikanan. Dengan permintaan yang terus meningkat di pasar domestik maupun internasional, peningkatan kualitas bibit melalui penerapan teknologi modern dan manajemen budidaya yang baik menjadi kunci keberhasilan budidaya ikan nila secara keseluruhan.

Seiring dengan perkembangan teknologi, bibit ikan nila telah menjadi salah satu objek penelitian yang menarik dalam penerapan *deep learning*, terutama dalam deteksi dan penghitungan objek berbasis citra. Karakteristik visual bibit ikan nila, seperti bentuk tubuh kecil, transparansi sebagian, serta kondisi tumpang tindih atau berdekatan di dalam wadah pemeliharaan, menimbulkan tantangan unik bagi sistem berbasis kecerdasan buatan. Tantangan ini membuka peluang bagi algoritma deteksi objek, seperti YOLO (You Only Look Once), untuk membuktikan efektivitasnya dalam mendeteksi objek dinamis dalam lingkungan yang kompleks.

Penggunaan *deep learning* dalam mendeteksi bibit ikan nila memberikan berbagai manfaat, terutama di sektor akuakultur. Algoritma seperti YOLO memungkinkan penghitungan jumlah bibit ikan secara otomatis dan real-time, yang sebelumnya dilakukan secara manual dengan tingkat akurasi yang terbatas. Sistem ini dapat digunakan untuk memantau kondisi bibit, mengidentifikasi pola distribusi dalam wadah, atau bahkan mendeteksi anomali yang menunjukkan masalah

kesehatan ikan. Keuntungan lain dari pendekatan ini adalah kemampuan untuk mengolah data secara efisien dalam skala besar, sehingga mendukung manajemen budidaya yang lebih baik.

Selain itu, *deep learning* memungkinkan pengembangan sistem yang lebih adaptif terhadap variasi kondisi pencahayaan, posisi kamera, atau orientasi objek. Hal ini sangat relevan dalam pengambilan data di lingkungan nyata, di mana bibit ikan sering berada dalam posisi tumpang tindih atau sebagian terhalang oleh bibit lainnya. Dengan teknik augmentasi data, seperti flip gambar, transformasi kecerahan, atau pengaturan saturasi, model *deep learning* dapat dilatih untuk mengenali bibit ikan dalam berbagai kondisi tersebut.

Penggunaan bibit ikan nila sebagai objek penelitian berbasis *deep learning* juga membuka peluang untuk mengatasi keterbatasan metode manual yang membutuhkan waktu lama dan tenaga kerja intensif. Teknologi ini memungkinkan pengembangan sistem otomatisasi yang lebih efisien, akurat, dan dapat diterapkan di berbagai skala budidaya, mulai dari unit kecil hingga skala industri. Dengan demikian, integrasi teknologi berbasis *deep learning* dalam deteksi bibit ikan nila tidak hanya meningkatkan efisiensi, tetapi juga memberikan kontribusi signifikan terhadap keberlanjutan industri perikanan modern.

2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah teknik manipulasi dan analisis gambar yang dilakukan pada citra digital untuk meningkatkan kualitas visual, mengekstrak informasi penting, atau mengotomatisasi pengenalan pola. Teknik ini mencakup berbagai proses seperti peningkatan kualitas gambar, segmentasi, klasifikasi, dan pengenalan objek. Proses pengolahan citra digital sangat penting di berbagai bidang, termasuk industri, kesehatan, dan pertanian, karena dapat membantu pengambilan keputusan berbasis data visual secara efisien (Irfan M, 2020).

Citra digital, yang merupakan hasil representasi visual dalam format numerik, terdiri dari piksel-piksel yang membentuk matriks. Setiap piksel mewakili intensitas warna atau kecerahan pada titik tertentu dalam gambar, yang menentukan resolusi dan kualitas visual citra. Semakin banyak piksel yang dimiliki, semakin tinggi resolusi dan detail gambar tersebut (Irfan M, 2020). Keunggulan citra digital adalah kemudahannya untuk dimanipulasi dan diolah secara otomatis

menggunakan algoritma komputer. Hal ini memungkinkan analisis citra untuk berbagai tujuan, seperti meningkatkan kualitas gambar, mengurangi noise, hingga mendeteksi objek spesifik dalam gambar (Rahmawati D, 2022).

Dalam konteks akuakultur, pengolahan citra digital memungkinkan pengelolaan yang lebih efisien melalui identifikasi otomatis, penghitungan jumlah, dan pengukuran ukuran ikan secara real-time tanpa intervensi manusia. Hal ini mengurangi risiko kesalahan manusia dan meningkatkan produktivitas. Contohnya, citra digital digunakan untuk menganalisis visual tanda-tanda penyakit pada ikan, yang membantu meningkatkan kualitas produksi perikanan (Santoso T., 2021). Di sektor lain, seperti medis, citra digital dari MRI atau CT Scan sering diproses lebih lanjut untuk membantu diagnosis yang lebih akurat (Susanto B., 2021).

Proses pengolahan citra digital biasanya dilakukan melalui tahapan berikut:

1. Pengambilan Gambar: Menggunakan perangkat sensor seperti kamera digital atau sensor gambar lainnya.
2. Pra-pemrosesan Gambar: Menggunakan teknik seperti filterisasi untuk mengurangi noise dan meningkatkan kualitas gambar.
3. Segmentasi: Memisahkan objek utama dari latar belakang dalam gambar.
4. Ekstraksi Fitur dan Klasifikasi: Mengidentifikasi atau menghitung objek berdasarkan parameter tertentu (Rahmawati D, 2022).

Dengan kemajuan teknologi, pengolahan citra digital kini dapat diterapkan dalam berbagai tugas kompleks, seperti analisis citra 3D, pengenalan wajah, dan perhitungan otomatis objek di bidang agrikultur atau akuakultur. Ini membuka peluang untuk efisiensi dan produktivitas yang lebih tinggi di banyak sektor industri.

2.3 Deteksi Objek

Deteksi objek adalah teknik dalam pengolahan citra dan visi komputer yang bertujuan untuk mengenali dan mengidentifikasi objek tertentu dalam gambar atau video. Deteksi objek tidak hanya menentukan apakah suatu objek ada dalam gambar, tetapi juga menyediakan informasi tentang posisi dan batas objek tersebut melalui *bounding box*. Teknologi ini sangat penting dalam berbagai aplikasi seperti pengawasan video, pengenalan wajah, hingga deteksi penyakit pada tanaman atau hewan (Syarifuddin A., 2020).

Pada prinsipnya, deteksi objek bekerja dengan cara mengidentifikasi pola-pola visual yang spesifik dalam gambar. Metode tradisional deteksi objek biasanya bergantung pada fitur visual tertentu, seperti bentuk, warna, atau tekstur. Namun, perkembangan metode berbasis pembelajaran mendalam (*deep learning*) telah membawa peningkatan signifikan dalam kinerja deteksi objek. Algoritma seperti YOLO (You Only Look Once) mampu melakukan deteksi objek secara real-time dengan akurasi tinggi, bahkan dalam kondisi lingkungan yang kompleks (Gunawan A., 2021).

Aplikasi deteksi objek di Indonesia semakin berkembang, terutama dalam bidang pertanian dan perikanan. Sistem otomatis berbasis deteksi objek digunakan untuk mengukur dan menghitung jumlah ikan dalam kolam budidaya, memantau kondisi kesehatan ikan, serta mendeteksi hama pada tanaman secara lebih efisien. Dalam sektor keamanan, deteksi objek digunakan untuk mengidentifikasi ancaman potensial, seperti deteksi kendaraan yang tidak dikenal dalam sistem pengawasan lalu lintas (Wibowo A., 2021).

2.4 Visi Komputer

Visi komputer (*computer vision*) adalah cabang ilmu komputer yang fokus pada bagaimana komputer dapat mengekstraksi, menginterpretasi, dan memahami informasi visual dari gambar atau video. Visi komputer memungkinkan sistem komputer untuk secara otomatis mengenali dan menganalisis pola-pola dalam citra, mirip dengan bagaimana mata manusia berfungsi dalam memproses penglihatan. Ini melibatkan berbagai teknik pengolahan citra, seperti segmentasi, klasifikasi, dan deteksi objek untuk mengotomatisasi tugas berbasis visual (Wibowo A., 2021).

Visi komputer memiliki berbagai aplikasi praktis, termasuk dalam sistem pengenalan wajah, diagnosis medis berbasis citra, serta kendaraan otonom. Dalam beberapa tahun terakhir, teknologi ini telah berkembang pesat karena kemajuan dalam algoritma pembelajaran mendalam, seperti jaringan saraf konvolusi (CNN). CNN digunakan secara luas dalam visi komputer untuk tugas-tugas seperti klasifikasi gambar dan deteksi objek. Contoh aplikasi lainnya termasuk sistem keamanan cerdas yang mampu mengenali individu berdasarkan ciri wajah atau identifikasi kendaraan dalam lalu lintas (Kusumawati R, 2020).

Di sektor akuakultur dan pertanian, visi komputer digunakan untuk meningkatkan efisiensi produksi melalui deteksi otomatis dan pengukuran objek. Sistem ini dapat digunakan untuk memonitor perkembangan tanaman atau hewan, menganalisis kondisi lingkungan, dan mendeteksi gangguan yang mungkin terjadi. Sebagai contoh, visi komputer dapat diterapkan untuk menghitung jumlah bibit ikan dalam kolam budidaya secara otomatis, yang sangat penting dalam pengelolaan sumber daya yang lebih efisien (Rahmawati D, 2022).

2.5 Python

Python adalah bahasa pemrograman tingkat tinggi yang sangat populer dan banyak digunakan dalam pengembangan aplikasi, terutama di bidang data sains, pembelajaran mesin, dan pengolahan citra. Keunggulan utama Python adalah sintaksisnya yang sederhana dan mudah dipahami, sehingga memudahkan programmer untuk mengembangkan kode dengan cepat dan efisien. Selain itu, Python didukung oleh ekosistem pustaka yang luas, seperti OpenCV untuk pengolahan citra, TensorFlow dan PyTorch untuk pembelajaran mendalam, serta NumPy dan Pandas untuk analisis data (Fauzi, 2020).

Dalam konteks visi komputer dan deteksi objek, Python sering digunakan bersama pustaka seperti OpenCV untuk pengolahan citra dasar dan TensorFlow atau PyTorch untuk membangun dan melatih model pembelajaran mendalam. Python memungkinkan pengembang untuk dengan mudah mengimplementasikan algoritma canggih seperti YOLO dan Mask R-CNN untuk deteksi objek, segmentasi citra, dan klasifikasi gambar. Misalnya, Python telah digunakan dalam pengembangan sistem otomatis untuk menghitung jumlah ikan dalam kolam menggunakan YOLO yang diintegrasikan dengan OpenCV (Rahmatullah A, 2021).

Selain itu, Python juga banyak digunakan di berbagai sektor industri untuk mengembangkan aplikasi berbasis kecerdasan buatan (AI) dan analitik data. Sifatnya yang serba guna menjadikannya pilihan utama bagi para peneliti dan pengembang dalam membuat aplikasi yang cepat dan skalabel. Python juga terus diperbarui dengan alat-alat dan pustaka baru yang mendukung berbagai jenis pemrosesan data, dari pengenalan suara hingga analisis gambar dan video (Susanto B., 2021).

2.6 YOLO (*You Only Look Once*)

You only look once adalah algoritma deteksi objek berbasis deep learning yang dirancang untuk mendeteksi objek secara real-time. YOLO memiliki pendekatan unik dengan memproses gambar secara keseluruhan hanya dalam satu kali "pandangan," berbeda dari metode deteksi objek tradisional seperti R-CNN yang memecah proses menjadi beberapa tahap, seperti ekstraksi fitur dan klasifikasi objek secara terpisah. Dengan mendeteksi semua objek dalam gambar dalam satu langkah, YOLO dapat mencapai kecepatan yang sangat tinggi tanpa mengorbankan akurasi (Redmon J., 2018).

YOLO bekerja dengan cara membagi gambar input menjadi grid-grid kecil. Setiap grid bertanggung jawab mendeteksi objek yang berada di area tersebut. Setiap grid menghasilkan beberapa bounding box yang mencakup prediksi posisi, ukuran, dan probabilitas keberadaan objek di dalam bounding box tersebut. Probabilitas ini menunjukkan tingkat keyakinan model terhadap keberadaan objek dalam gambar (Bochkovskiy A., 2020).

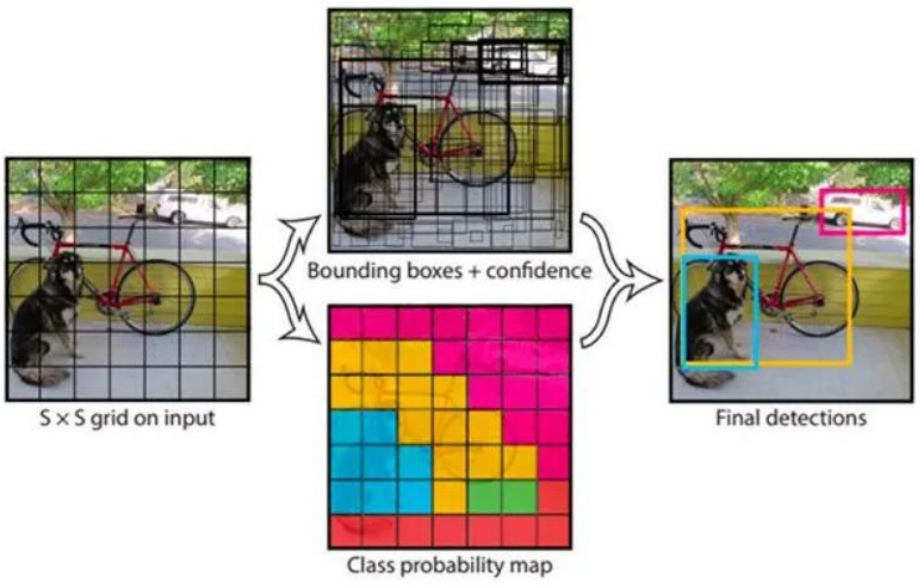
Salah satu fitur utama YOLO adalah efisiensi waktu pemrosesan yang tinggi. Hal ini menjadikannya sangat ideal untuk aplikasi yang membutuhkan deteksi objek secara real-time, seperti sistem pengawasan berbasis video atau kendaraan otonom. Selain itu, YOLO juga memiliki kemampuan untuk mendeteksi banyak objek berbeda dalam satu gambar secara bersamaan. Misalnya, dalam gambar kompleks seperti jalanan yang ramai, YOLO dapat mendeteksi kendaraan, pejalan kaki, dan rambu lalu lintas dengan cepat dan akurat (Gleen Jocher, 2023).

Pada versi YOLOv10, algoritma ini membawa sejumlah perbaikan signifikan dalam kecepatan, akurasi, dan efisiensi. YOLOv10 memanfaatkan pendekatan baru dalam optimasi model dengan meningkatkan kemampuan generalisasi dan mengurangi overfitting, terutama pada dataset besar. Selain itu, YOLOv10 dirancang agar lebih adaptif terhadap kondisi lingkungan yang kompleks, seperti pencahayaan rendah atau gambar dengan noise tinggi. Dengan peningkatan ini, YOLOv10 menjadi solusi yang sangat efisien untuk berbagai aplikasi industri, termasuk sektor akuakultur (Rahmawati D., 2022).

Dalam konteks akuakultur, YOLOv10 digunakan untuk menghitung jumlah bibit ikan secara otomatis. Sistem ini memberikan solusi yang lebih cepat dan akurat dibandingkan dengan metode manual tradisional, sehingga membantu

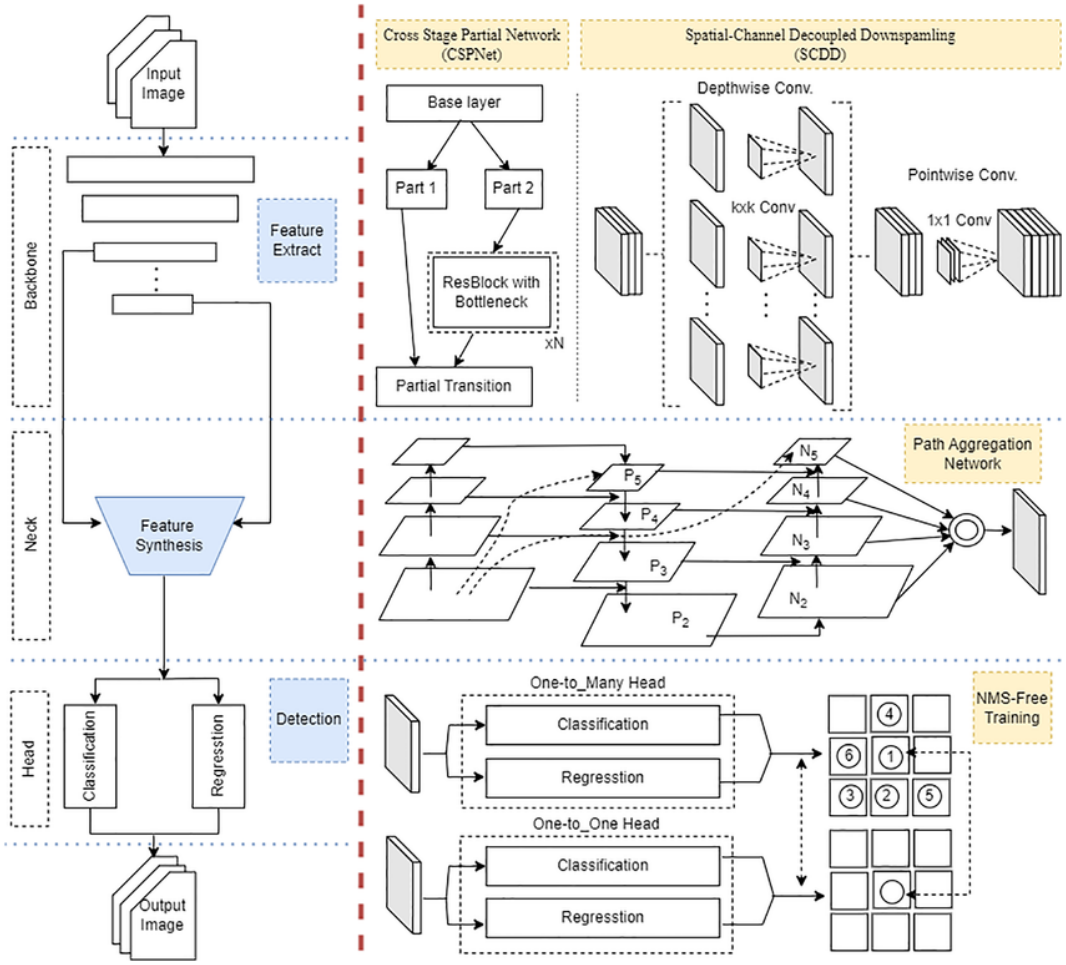
menjaga keseimbangan populasi ikan di kolam budidaya. Dengan kecepatan dan akurasinya, YOLOv10 memungkinkan pengelolaan akuakultur yang lebih efisien, mengurangi risiko kesalahan manusia, dan meningkatkan produktivitas operasional.

Dengan fleksibilitasnya, YOLOv10 dapat dilatih menggunakan berbagai dataset untuk mendeteksi objek spesifik sesuai kebutuhan aplikasi. Penerapannya mencakup pengawasan lalu lintas, deteksi ancaman keamanan, serta aplikasi di bidang agrikultur dan perikanan. Keunggulan YOLOv10 dalam efisiensi dan performa menjadikannya alat yang andal untuk tugas-tugas deteksi objek di berbagai sektor industri.



Gambar 5 Cara Kerja YOLO

Arsitektur YOLO V10



Gambar 6 Arsitektur YOLO

1. Backbone

Backbone adalah bagian dari arsitektur YOLOv10 yang bertugas mengekstraksi fitur dasar dari gambar input. Dalam gambar ini, backbone menggunakan:

a. Cross Stage Partial Network (CSPNet): CSPNet membagi aliran fitur menjadi dua jalur, yaitu *Part 1* dan *Part 2*. Setelah itu, fitur digabungkan kembali melalui transisi parsial. CSPNet dirancang untuk meningkatkan efisiensi dan mengurangi redundansi fitur sambil mempertahankan akurasi deteksi.

b. ResBlock with Bottleneck: Blok residual dengan bottleneck digunakan untuk meningkatkan efisiensi pemrosesan data fitur. Beberapa blok residual ini digunakan untuk mengekstraksi fitur pada berbagai skala.

c. Spatial-Channel Decoupled Downsampling (SCDD): Digunakan untuk melakukan downsampling (mengurangi resolusi) secara lebih efisien dengan memisahkan informasi spasial dan kanal. Teknik ini memanfaatkan konvolusi depthwise dan pointwise untuk mempercepat proses tanpa kehilangan informasi penting.

2. Neck

Neck bertugas menggabungkan fitur dari berbagai skala untuk memastikan deteksi objek dapat dilakukan pada ukuran objek yang berbeda. Komponen utama dalam neck meliputi:

a. Feature Synthesis: Bagian ini mensintesis fitur dari backbone agar dapat digunakan untuk deteksi objek.
b. Path Aggregation Network (PAN): PAN digunakan untuk meningkatkan aliran informasi fitur dari berbagai resolusi (multiscale features). Pada PAN, fitur dari tingkat rendah hingga tingkat tinggi diolah dan digabungkan untuk mendukung deteksi objek yang kecil maupun besar.

3. Head

Head adalah bagian dari YOLOv10 yang bertanggung jawab menghasilkan bounding box (kotak pembatas) dan klasifikasi objek. Dalam gambar ini, terdapat dua pendekatan:

a. One-to-Many Head: Menghasilkan beberapa bounding box dan klasifikasi sekaligus untuk setiap grid. Ini memungkinkan deteksi pada skenario kompleks dengan banyak objek.
b. One-to-One Head: Memastikan hanya satu prediksi untuk setiap grid, mengurangi redundansi prediksi dan meningkatkan akurasi.
c. Classification dan Regression: Bagian ini bertugas untuk mengklasifikasikan jenis objek (misalnya, ikan, kendaraan) dan memprediksi koordinat bounding box.

4. NMS-Free Training

NMS (Non-Maximum Suppression) biasanya digunakan untuk menghilangkan prediksi bounding box yang saling tumpang tindih, tetapi YOLOv10 memperkenalkan metode *NMS-Free Training*. Ini memungkinkan model untuk belajar menangani redundansi prediksi selama pelatihan, sehingga

NMS tidak diperlukan saat inferensi, yang mempercepat proses deteksi secara keseluruhan.

a. Alur Data

1) Input Image: Gambar dimasukkan ke backbone untuk ekstraksi fitur.
2) Backbone: CSPNet dan SCDD mengekstraksi fitur penting.
3) Neck: PAN menggabungkan fitur dari berbagai resolusi.
4) Head: Bounding box dan klasifikasi dihasilkan, baik melalui pendekatan One-to-Many maupun One-to-One.
5) Output Image: Hasil akhir berupa bounding box dan label objek.

b. Keunggulan Arsitektur YOLOv10

1) Efisiensi Tinggi: CSPNet dan SCDD meningkatkan efisiensi komputasi tanpa kehilangan akurasi.
2) Deteksi Multiskala: PAN memungkinkan deteksi objek pada berbagai ukuran.
3) NMS-Free Training: Mengurangi latensi selama inferensi, cocok untuk aplikasi real-time.
4) Fleksibilitas: Cocok untuk berbagai aplikasi, termasuk deteksi objek kecil dan kompleks.

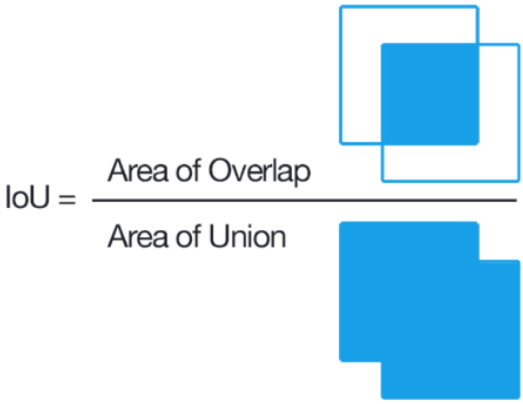
Arsitektur ini dirancang untuk memenuhi kebutuhan deteksi objek secara real-time dengan tingkat akurasi yang tinggi pada berbagai lingkungan dan perangkat keras.

2.7 *Mean Average Precision* (mAP)

mAP adalah metrik yang digunakan untuk mengukur akurasi deteksi objek berdasarkan beberapa komponen penting, yaitu:

Intersection over Union (IoU)**:**

IoU adalah rasio antara area overlap (persilangan) antara *bounding box* prediksi dengan *bounding box* sebenarnya terhadap area total (union) dari keduanya. IoU dihitung dengan rumus berikut :



Gambar 7 Perhitungan IoU

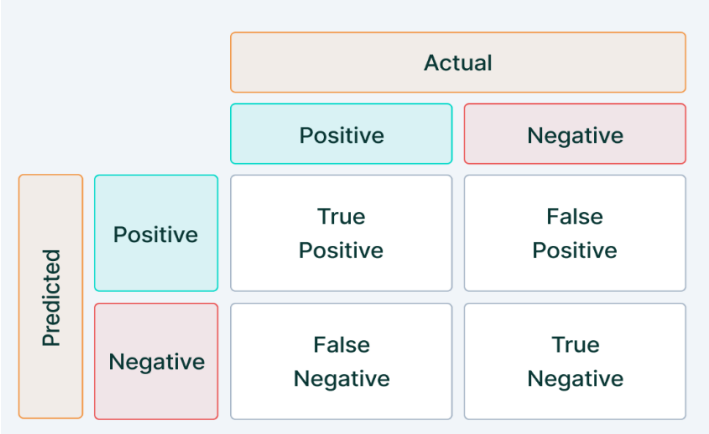
Nilai IoU digunakan untuk menentukan seberapa baik *bounding box* prediksi mencocokkan objek yang sebenarnya. Jika IoU tinggi, berarti prediksi *bounding box* model sangat dekat dengan ground truth.

a. *Confusion Matrix*

Confusion matrix adalah tabel yang menunjukkan hubungan antara prediksi model dengan label sebenarnya. Tabel ini terdiri dari**:**

1) *True Positive*: Jumlah kasus dimana model memprediksi *bounding box* berada di suatu posisi dan benar
2) *False Positive*: Jumlah kasus dimana model memprediksi *bounding box* berada di suatu posisi tetapi salah
3) *False Negative*: Jumlah kasus dimana model tidak memprediksi *bounding box* berada di suatu posisi dan salah, yang artinya ground truth dari *bounding box* berada di lokasi tersebut.
4) *True Negative*: Jumlah kasus dimana model tidak memprediksi *bounding box* dan memang tidak ada *bounding box* di lokasi tersebut.

Nilai ini digunakan untuk menghitung precision dan recall, yang menjadi dasar untuk menghitung mAP.



Gambar 8 Confusion Matrix

b. *Precision*

Precision (Presisi) untuk mengukur seberapa tepat prediksi model. Ini adalah rasio antara jumlah prediksi benar (*True Positive*) dengan total prediksi yang dibuat oleh model (*True Positive* + *False Positive*). Presisi mengukur seberapa banyak dari bibit ikan yang terdeteksi adalah benar-benar bibit ikan.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

c. *Recall*

Recall adalah mengukur seberapa baik model dalam menemukan semua bibit ikan yang ada. Ini adalah rasio antara jumlah prediksi benar (*True Positive*) dengan total jumlah objek yang ada (*True Positive* + *False Negative*). Recall mengukur seberapa banyak objek yang sebenarnya berhasil dideteksi oleh model.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

d. mAP50

mAP50 adalah rata-rata precision pada nilai IoU ≥ 50%. Ini memberikan gambaran dasar apakah model mendeteksi objek dengan tingkat overlap yang moderat.

e. mAP50-95

mAP50-95 adalah rata-rata precision pada berbagai nilai IoU, mulai dari 50% hingga 95% dengan interval 0.05. Nilai ini memberikan gambaran yang lebih rinci tentang kemampuan model dalam mendeteksi objek dengan tingkat ketelitian yang lebih tinggi.

Rumus mAP dihitung dengan menjumlahkan area di bawah kurva precision-recall (AUC) pada setiap nilai IoU dan kemudian mengambil rata-ratanya.

2.8 *Root Mean Square Error* (RMSE)

RMSE digunakan untuk mengukur perbedaan antara jumlah bibit ikan nila yang terdeteksi oleh model dengan jumlah sebenarnya. RMSE dihitung dengan rumus:

1. Selisih **:** Hitung selisih antara nilai yang diprediksi oleh model dan nilai sebenarnya untuk setiap data. Ini disebut sebagai error atau residual.
2. Kuadratkan Error **:** Untuk menghilangkan pengaruh tanda positif atau negatif dan fokus pada besarnya error, kuadratkan setiap error.
3. Rata-Rata **:** Hitung rata-rata dari error yang telah dikuadratkan.
4. Akar Kuadrat **:** Ambil akar kuadrat dari rata-rata tersebut untuk mendapatkan RMSE.

Secara matematis, RMSE dapat dituliskan sebagai :

$$RMSE = \frac{1}{n}\sum_{i=0}^{n}(yi - \hat{y}i)^2$$

Dimana n adalah jumlah total data, yi adalah nilai sebenarnya dan $\hat{y}i$ adalah nilai prediksi model. RMSE mengukur rata-rata kesalahan kuadrat pada prediksi model. Semakin kecil nilai RMSE, semakin baik model dalam menghitung jumlah bibit ikan.

2.9 *Flask*

Flask adalah sebuah framework micro yang digunakan untuk pengembangan aplikasi web berbasis Python. Flask dirancang untuk memberikan

fleksibilitas tinggi kepada pengembang dengan tetap menjaga kemudahan penggunaan. Framework ini pertama kali diperkenalkan oleh Armin Ronacher pada tahun 2010 dan menjadi populer di kalangan pengembang karena sifatnya yang ringan namun powerful.

Flask menawarkan kerangka kerja inti yang ringan, memberikan kebebasan kepada pengembang untuk memilih komponen tambahan yang mereka butuhkan (Ronacher, 2010). Selain itu, Flask mendukung perluasan aplikasi melalui modul dan library tambahan, seperti Flask-RESTful untuk membangun API dan Flask-SQLAlchemy untuk ORM (Object Relational Mapping) (Grinberg, 2018). Framework ini juga dilengkapi dengan Jinja2, sebuah template engine yang memungkinkan pembuatan tampilan HTML secara dinamis dengan sintaks yang sederhana. Filosofi "Batteries Not Included" dari Flask memberikan fleksibilitas bagi pengembang untuk memilih solusi yang paling sesuai dengan kebutuhan aplikasi mereka.

Flask sering digunakan dalam pengembangan aplikasi web kecil hingga menengah, termasuk pembuatan REST API untuk aplikasi modern, prototipe cepat untuk sistem berbasis web, dan sistem pengolahan data berbasis web yang memerlukan integrasi dengan backend Python. Penelitian oleh Doe et al. (2018) menunjukkan bahwa Flask memiliki performa yang baik untuk aplikasi web dengan arsitektur microservices, berkat sifatnya yang modular dan efisien dalam penggunaan sumber daya.

2.10 *Nginx*

Nginx (dibaca "engine-x") adalah sebuah web server open-source yang dirancang untuk menangani beban lalu lintas tinggi. Pertama kali dikembangkan oleh Igor Sysoev pada tahun 2004, Nginx dirancang untuk mengatasi masalah skalabilitas yang sering ditemui pada server tradisional seperti Apache (Sysoev, 2004). Nginx menggunakan pendekatan event-driven yang memungkinkan penanganan ribuan koneksi secara bersamaan dengan penggunaan sumber daya yang minimal. Selain sebagai web server, Nginx sering digunakan sebagai reverse proxy dan load balancer untuk mendistribusikan beban ke beberapa server backend. Dukungan terhadap protokol HTTP/2 memungkinkan Nginx memberikan peningkatan kecepatan dan efisiensi komunikasi antara klien dan

server (Rahman, 2017). Kemampuan caching Nginx juga membantu meningkatkan performa website dengan mengurangi beban pada server backend.

Nginx sering digunakan untuk menangani lalu lintas tinggi pada website atau aplikasi web, berfungsi sebagai proxy untuk aplikasi berbasis Flask atau framework lainnya, serta mengamankan koneksi melalui penerapan SSL/TLS. Penelitian oleh Smith et al. (2020) menunjukkan bahwa penggunaan Nginx sebagai reverse proxy untuk aplikasi Flask mampu meningkatkan performa aplikasi hingga 40%, terutama dalam skenario dengan banyak koneksi simultan.