

Deteksi Bahan Pangan Tinggi Protein Menggunakan Model *You Only Look Once (YOLO)*

Restu Agil Yuli Arjun*, Esi Putri Silmina

Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Aisyiyah Yogyakarta, Yogyakarta, Indonesia

Email: ^{1,*}restuagil.ya@gmail.com, ²esiputrisilmina@unisayogya.ac.id

Email Penulis Korespondensi: restuagil.ya@gmail.com

Submitted: 31/01/2025; Accepted: 27/02/2025; Published: 07/03/2025

Abstrak—Stunting memiliki prevalensi yang tinggi yakni 21,6% dari target pemerintah sebesar 14% dan menjadi salah satu masalah kesehatan di Indonesia. Kekurangan nutrisi, terutama protein menjadi penyebab utama yang berperan dalam pertumbuhan anak. Salah satu solusi pencegahan ialah dengan pemberian Makanan Pendamping ASI (MP-ASI) yang kaya protein. Teknologi yang mampu mendeteksi bahan pangan tinggi protein secara cepat dan akurat sangat diperlukan. Penelitian ini bertujuan untuk membuat model deteksi bahan pangan tinggi protein menggunakan Model *YOLOv11*. Model ini dipilih karena kemampuannya dalam mendeteksi objek secara efisien meskipun dalam lingkungan yang kompleks dan objek yang tumpang tindih. Metode penelitian melalui beberapa tahapan, yakni pengumpulan dan anotasi *dataset*, pra-pemrosesan data, pelatihan model, evaluasi model, dan uji coba model. *Dataset* yang digunakan terbagi dalam tiga bagian, *train set* 70%, *valid set* 20%, dan *test set* 10%. Proses pelatihan menggunakan Model *Yolo11s*. Evaluasi model dilakukan dengan mengukur metrik *precision*, *recall*, dan *mean Average Precision (mAP)* untuk memastikan akurasi deteksi. Hasil evaluasi model mendapatkan nilai *precision* 96%, *recall* 92,3 %, *mAP50* 96,4%, dan *mAP50-95* 81,5%. Hasil uji coba keberhasilan model mendapatkan nilai 98,2%.

Kata Kunci: Stunting; Gizi; Protein; *YOLOv11*; Deteksi Objek

Abstract—Stunting has a high prevalence of 21.6% from the government target of 14% and is one of the health problems in Indonesia. Lack of nutrition, especially protein, is the main cause that plays a role in child growth. One of the preventive solutions is to provide protein-rich complementary foods (MP-ASI). To enhance this solution, technology that can swiftly and precisely identify high-protein food components is imperative. This research seeks to create a high-protein food detection model utilizing the *YOLOv11* framework, chosen for its efficacy in object detection, particularly in intricate environments and with overlapping items. The research methodology includes several stages: dataset collection and annotation, data pre-processing, model training, model evaluation, and model testing. The dataset is divided into three parts: 70% for the training set, 20% for the validation set, and 10% for the test set. The *YOLOv11s* model is used for training. Evaluation is based on precision, recall, and mean Average Precision (mAP) metrics to ensure the model's detection accuracy. The evaluation results indicate a precision of 96%, recall of 92.3%, mAP50 of 96.4%, and mAP50-95 of 81.5%. During testing, the model achieved a success rate of 98.2%. These results demonstrate the model's potential in detecting protein-rich foods, which could significantly contribute to addressing malnutrition and stunting.

Keywords: Stunted; Nutrition; Proteins; *YOLOv11*; Object Detection

1. PENDAHULUAN

Stunting adalah kondisi hambatan proses tumbuh dan kembang anak akibat kekurangan gizi, rekurensi infeksi, hingga kekurangan stimulasi psikososial. Pembelahan sel dan sintesis DNA selama pertumbuhan, terutama energi dan protein, akan lebih cepat jika kebutuhan gizi terpenuhi. Protein membantu memperbaiki jaringan yang rusak dan membangun jaringan baru selama proses tumbuh kembang anak. Anak yang kekurangan asupan protein kronis akan mengalami stunting [1]. Jika stunting dibiarkan, maka akan berdampak pada penurunan kognitif, kesulitan konsentrasi, kesulitan belajar, mudah sakit, dan berisiko terkena gangguan metabolik, berupa obesitas dan penyakit kardiovaskular [2].

Hasil Survei yang dilakukan pada tahun 2022 oleh Survei Status Gizi Indonesia (SSGI) menunjukkan masih tingginya angka stunting yakni 21,6%, jauh dari 14% yang menjadi target pemerintah Indonesia pada tahun 2024 [3]. Penelitian lain menunjukkan bahwa risiko stunting meningkat 1,6 kali lipat dari 13,7% menjadi 22,4% pada kelompok usia 6-11 bulan ke kelompok usia 12-23 bulan. Peningkatan ini membuktikan terdapat kendala dalam ketepatan pemberian MP-ASI baik itu dari segi umur, kuantitas, tekstur MPASI yang diberikan, kandungan protein, hingga ragam makanan [4].

Salah satu cabang ilmu komputer yang dikenal sebagai kecerdasan buatan (*Artificial Intelligence*) memungkinkan sistem komputer meniru perilaku cerdas manusia [5]. *Metode Naive Bayes Classifier* yang merupakan cabang dari kecerdasan buatan pernah diterapkan pada kasus stunting [6]. Selain itu, kecerdasan buatan juga diterapkan pada deteksi objek. Deteksi objek merupakan bagian dari visi komputer, yang memungkinkan komputer untuk mengenali dan mengidentifikasi berbagai objek dalam sebuah citra. Teknologi ini juga terkait dengan *deep learning*, yang digunakan untuk tugas klasifikasi dan deteksi objek secara otomatis [7].

You Only Look Once (YOLO) dibandingkan dengan model deteksi lain seperti *Faster R-CNN* dan *SSD*, *YOLO* memiliki keunggulan dalam kecepatan deteksi (*FPS*) tanpa mengorbankan akurasi. *YOLO* menunjukkan performa terbaik pada kasus identifikasi obat dengan *FPS* tertinggi sebesar 51, meskipun akurasinya lebih rendah dari *Faster R-CNN* [8]. Permasalahan yang mirip juga terjadi pada deteksi kendaraan, *YOLO* memiliki performa yang terbaik diantara Metode *Faster R-CNN* dan *SSD*. *Faster R-CNN* menjadi yang tercepat dibandingkan model *RCNN*, tetapi

tidak memiliki *FPS* yang memuaskan. *SSD* memiliki kecepatan deteksi yang cepat, namun akurasinya lebih rendah karena menggunakan model yang ringan [9].

Penerapan *YOLO* khususnya pada kasus deteksi makanan telah banyak diterapkan. Rahma et al. menerapkan *YOLO* untuk deteksi 31 jenis makanan khas Palembang yang mendapatkan performa rata-rata 96% dengan kecepatan deteksi 40.486.129 *millisecond* [10]. Selain itu, Handayani juga mengimplementasikan *YOLO* dengan melatih model agar mampu mendeteksi telur kategori segar dan tidak segar dengan nilai *precision* 82,2%, *recall* 92,45%, *mAP50* 95,05%, dan *mAP50-95* sebesar 93,16%, membuktikan performa yang baik dalam mendeteksi citra cangkang telur [11]. *YOLO* juga telah terbukti efektif dalam mendeteksi kesegaran ikan layur dengan nilai *precision* 97,6%, *recall* 99,6%, dan *mAP50* sebesar 99,1% [12], serta dalam deteksi kesegaran ikan beku dengan nilai *mAP* 94,7% dan mampu mendeteksi sebanyak 94% citra ikan dengan tepat [12].

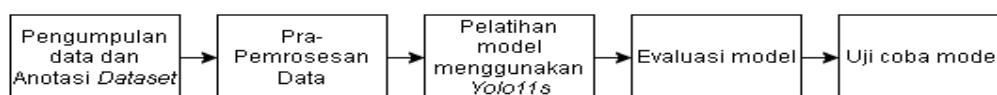
Model *YOLOv11*, yang merupakan perkembangan terbaru dari *YOLO*, menunjukkan peningkatan signifikan dalam hal akurasi, kecepatan, dan efisiensi komputasi dibandingkan dengan versi sebelumnya [13]. *YOLOv11* dinilai lebih unggul dibandingkan dengan *YOLOv10* pada kasus deteksi buah apel, baik secara *precision*, *recall*, dan *mean Average Precision (mAP)* [14]. Jegham et al., juga menganalisis kinerja berbagai Algoritma *YOLO* dimulai dari *YOLOv3*, *YOLOv9*, *YOLOv10*, dan *YOLOv11* berupa objek rambu lalu lintas, satwa liar afrika, dan kapal. *YOLOv9* menunjukkan akurasi yang substansial namun kesulitan dalam mendeteksi objek kecil. *YOLOv10* menunjukkan akurasi yang relatif lebih rendah dalam objek yang tumpang tindih, namun *YOLOv10* unggul dalam kecepatan. Sementara itu, *YOLOv11* secara konsisten menunjukkan kinerja unggul dalam hal akurasi, kecepatan, efisiensi komputasi, dan ukuran model [15]. Penelitian lain pada deteksi kendaraan menemukan bahwa *YOLOv11* melampaui versi sebelumnya (*YOLOv8* dan *YOLOv10*) dalam mendeteksi kendaraan yang lebih kecil dan tumpang tindih. *YOLOv11* juga mempertahankan waktu inferensi yang tinggi sehingga mempercepat proses deteksi [16]. Perkembangan ini menjadikan *YOLOv11* pilihan ideal untuk digunakan dalam deteksi bahan pangan tinggi protein, khususnya dalam konteks pencegahan stunting.

Mempertimbangkan tingginya angka stunting di Indonesia [3] dan pentingnya pemberian MP-ASI yang tepat sebagai pencegahan [4], maka penggunaan teknologi menjadi salah satu upaya yang dibutuhkan masyarakat [17]. Pemanfaatan teknologi memungkinkan masyarakat untuk mengakses informasi dan edukasi mengenai stunting melalui platform daring yang dapat diakses melalui ponsel mereka [17]. Studi menunjukkan bahwa penggunaan teknologi dalam memberikan edukasi terkait stunting memberikan peningkatan bermakna terhadap pemahaman masyarakat [18]. Pemanfaatan teknologi menjadi inovasi baru sebagai salah satu strategi pencegahan stunting dengan lebih efisien dan efektif dengan disajikannya informasi gizi yang tepat serta saran penyajian makanan yang sesuai [19]. Pentingnya intervensi pada periode *golden age* anak dan pemberian MP-ASI yang sesuai dengan rekomendasi menjadi kunci untuk menurunkan angka stunting. Pemberian MPASI yang tidak dilakukan sesuai rekomendasi dapat meningkatkan risiko penyakit hingga kematian pada anak [20]. Oleh karena itu, teknologi yang mampu mendeteksi bahan pangan tinggi protein secara cepat dan akurat sangat dibutuhkan. *YOLOv11* dipilih sebagai model deteksi dalam penelitian ini karena kemampuannya yang unggul dalam mendeteksi objek secara efisien, bahkan pada lingkungan objek yang kompleks dan tumpang tindih [16][8][9][13]. Keunggulan *YOLOv11* dalam hal akurasi, kecepatan dan efisiensi komputasi menjadikannya sebagai teknologi yang sangat cocok untuk diterapkan dalam mendeteksi bahan pangan yang dapat mendukung pencegahan stunting [15].

Penelitian ini bertujuan untuk membuat model deteksi bahan pangan tinggi protein menggunakan Model *YOLOv11*. Performa model akan diukur menggunakan metrik *precision*, *recall*, *mAP* (*mean Average Precision*), dan tingkat keberhasilan dalam pengujian. Penelitian ini diharapkan menjadi salah satu upaya dari pencegahan stunting berbasis teknologi serta menjadi dasar yang berguna bagi peneliti lain dalam mengembangkan sistem deteksi, misalnya sistem deteksi bahan pangan tinggi protein untuk pemilihan menu MP-ASI. Sistem tersebut nantinya dapat mendeteksi bahan pangan kemudian memberikan rekomendasi menu yang sesuai dengan hasil deteksi bahan pangan, usia anak, serta berdasarkan pedoman ahli gizi. Penelitian yang dilakukan dalam pemanfaatan teknologi ini diharapkan akan bermakna dalam penurunan angka stunting khususnya di Indonesia.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan beberapa tahapan yang dilakukan untuk memastikan penelitian terlaksana secara sistematis. Tahapan tersebut memiliki peran dan kepentingannya tersendiri, sehingga tercapai tujuan penelitian. Gambar 1, menampilkan tahapan-tahapan yang dilakukan dalam penelitian.



Gambar 1. Tahapan Penelitian

Gambar 1, terdapat lima tahapan yang dilakukan dalam penelitian ini. Tahapan pertama adalah pengumpulan dan anotasi *dataset*, yang mengumpulkan data gambar bahan pangan tinggi protein yang kemudian di anotasi. Tahap kedua yaitu pra-pemrosesan data, yang berfungsi untuk mempersiapkan *dataset* sehingga siap digunakan pada tahap

pelatihan model. *Dataset* yang telah dipersiapkan, kemudian masuk kepada tahap ketiga yaitu pelatihan, dimana model dibangun agar mampu mendeteksi bahan pangan tinggi protein. Model yang telah dibangun kemudian di evaluasi pada tahap keempat menggunakan metrik *precision*, *recall*, dan *mAP*. Tahap kelima, dilakukan uji coba model guna mengukur performa model pada kondisi yang sebenarnya.

2.1 Pengumpulan dan Anotasi Dataset

Dataset yang akan digunakan dalam penelitian ini diunduh melalui platform *roboflow universe* [21] dengan format *COCO* berjumlah 2360 gambar. *Dataset* kemudian diunggah ke platform *roboflow* dengan format *COCO* agar *dataset* teranotasi secara otomatis. *Dataset* ini terdiri dari 10 kelas diantaranya, Daging Ayam, Daging Sapi, Ikan Kembung, Ikan Lele, Ikan Nila, Ikan Patin, Tahu, Telur Ayam Broiler, Tempe, dan Udang. *Dataset* dilakukan penambahan data pada masing-masing kelas sebanyak 10 gambar yang diperoleh dari internet. Data tersebut kemudian di anotasi pada platform *roboflow* yang berfungsi untuk melabeli objek sesuai dengan kelasnya [22]. Tabel 1, menunjukkan *dataset* Bahan Pangan Tinggi Protein.

Tabel 1. *Dataset* Bahan Pangan Tinggi Protein

Kelas	Total/kelas
Daging Ayam	131
Daging Sapi	170
Ikan Kembung	291
Ikan Lele	252
Ikan Nila	251
Ikan Patin	278
Tahu	290
Telur Ayam Broiler	287
Tempe	292
Udang	128
Total Data	2370

Tabel 1 menunjukkan distribusi jumlah gambar pada setiap kelas dalam *dataset* Bahan Pangan Tinggi Protein yang digunakan dalam penelitian ini. *Dataset* terdiri dari 10 kelas yang mencakup berbagai jenis bahan pangan tinggi protein, antara lain Daging Ayam, Daging Sapi, Ikan Kembung, Ikan Lele, Ikan Nila, Ikan Patin, Tahu, Telur Ayam Broiler, Tempe, dan Udang. Kelas dengan jumlah gambar terbanyak adalah Tempe dengan 292 gambar, diikuti oleh Ikan Kembung dengan 291 gambar, dan Tahu dengan 290 gambar. Sedangkan kelas yang memiliki jumlah gambar paling sedikit adalah Udang dengan 128 gambar dan Daging Ayam dengan 131 gambar.

2.2 Pra-Pemrosesan Data

Dataset yang digunakan dalam penelitian ini melalui proses *preprocessing* pada platform *roboflow* untuk meningkatkan kualitas data yang agar optimal [23]. Tahap awal dalam pra-pemrosesan data yaitu pembagian *dataset*. *Dataset* akan dibagi menjadi 3 bagian yaitu *train set* (70%), *valid set* (20%), dan *test set* (10%). *Train set* untuk melatih model, *valid set* akan melakukan evaluasi pada model, sedangkan *test set* adalah data yang tidak pernah diketahui oleh model sehingga dapat digunakan untuk pengujian [24]. Langkah selanjutnya adalah penggunaan fitur *Auto-Orient*, yang berfungsi untuk menyesuaikan gambar secara otomatis. Proses ini bertujuan untuk menyeragamkan orientasi gambar dalam *dataset* sehingga model dapat mempelajari pola secara konsisten. Selain itu, gambar-gambar dalam *dataset* juga diubah ukurannya menjadi 640x640 piksel. Penyesuaian ukuran ini dilakukan untuk menyeragamkan *input* model sesuai dengan persyaratan *YOLO*, yang membutuhkan dimensi tetap untuk memastikan efisiensi dalam proses pelatihan dan inferensi [25]. Penelitian ini juga menerapkan teknik augmentasi untuk meningkatkan variasi *dataset* [22]. Augmentasi data bertujuan untuk memperkaya *dataset* tanpa memerlukan pengumpulan data tambahan secara langsung, sehingga model dapat belajar dari lebih banyak variasi pola, meningkatkan generalisasi, dan mengurangi risiko *overfitting* [26][27].

2.3 Pelatihan Model

Proses pelatihan model dilakukan menggunakan Model *Yolo11s*. *Dataset* akan dilatih pada platform *Google Colaboratory*, yang mendukung penggunaan *Python* dan *GPU Tesla 4* untuk mempercepat komputasi selama pelatihan [28]. Proses pelatihan bertujuan untuk melatih model agar dapat mengenali pola dari data pada *train set* [29], sesuai dengan tujuan penelitian. Model akan dilatih secara iteratif dengan parameter yang disesuaikan untuk memastikan bahwa proses pembelajaran berjalan optimal [25]. Setelah pelatihan selesai, model siap untuk diuji dan dievaluasi pada tahap selanjutnya.

2.4 Evaluasi Model

Evaluasi yang dilakukan pada model bertujuan dalam mengukur performa model dalam mendeteksi bahan pangan tinggi protein secara akurat. Kinerja model akan dievaluasi menggunakan metrik *Precision*, *Recall*, dan *mean Average Precision (mAP)* [30][31][22] yang didasarkan pada *True Positive (TP)*, *False Positive (FP)*, dan *False Negatif (FN)*.

True Positive (TP) adalah pendeteksian yang akurat terhadap objek yang benar-benar ada didalam gambar. *False Positive (FP)* adalah kondisi dimana model mendeteksi objek yang tidak ada didalam gambar. *False Negative (FN)* terjadi ketika model tidak dapat mendeteksi objek yang sebenarnya ada didalam gambar. Persamaan (1) dan (2) menjelaskan rumus perhitungan metrik *precision* dan *recall*. Sedangkan rumus *mAP* (*mean Average Precision*) diuraikan dalam persamaan (3) [22].

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Precision pada persamaan 1 mengukur sejauh mana prediksi model yang dihasilkan benar, yaitu proporsi antara prediksi yang sesuai dengan data anotasi dibandingkan dengan keseluruhan prediksi model. *Recall* pada persamaan 2 mengevaluasi kemampuan model untuk mendeteksi semua objek yang ada pada gambar [14], sehingga mencerminkan sensitivitas model [23] terhadap keberadaan bahan pangan yang sebenarnya. Evaluasi model juga dilakukan menggunakan metrik *mAP* (*mean Average Precision*) pada persamaan 3, yaitu nilai rata-rata akurasi model yang dihitung berdasarkan nilai *threshold* ambang batas tumpang tindih (*Intersection over Union* atau *IoU*) antara prediksi dan anotasi *ground truth* [24]. Variasi *mAP50* dan *mAP@50-95* ditetapkan pada penelitian ini sebagai parameter evaluasi pada model. *mAP50* merupakan nilai *mAP* yang didapatkan dengan *threshold IoU* sebesar 50%. *mAP@50-95* mencakup rata-rata akurasi model pada berbagai ambang batas *IoU*, mulai dari 0,50 hingga 0,95 dengan langkah sebanyak 0,05. Metrik *mAP@50-95* memberikan penilaian yang lebih ketat terhadap performa model karena mengevaluasi konsistensinya pada berbagai tingkat tumpang tindih [32].

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (3)$$

Rumus *mAP* diatas mengandung beberapa notasi, *n* digunakan untuk merepresentasikan jumlah kelas dalam *dataset*, sedangkan *i* menunjukkan kelas ke *i* sampai dengan total kelas. Notasi *AP* mengacu pada nilai *Average Precision* untuk kelas ke *i* [33]. Proses evaluasi dilakukan menggunakan *valid set* yang berisi gambar-gambar bahan pangan tinggi protein yang telah dilakukan anotasi dengan label *ground truth*. Model mendeteksi gambar bahan pangan dalam *valid set*, dan hasil deteksinya dibandingkan dengan anotasi *ground truth* untuk menghitung metrik *Precision*, *Recall*, dan *mAP*.

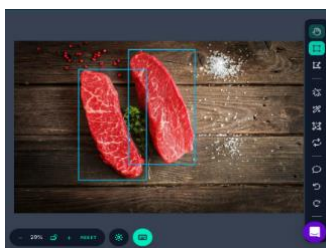
2.5 Uji Coba Model

Kemampuan model yang telah dilatih sebelumnya diuji dalam mendeteksi bahan pangan tinggi protein dengan benar. *Dataset* yang digunakan untuk uji coba adalah *test set*, yang terdiri dari data yang tidak pernah digunakan selama pelatihan maupun validasi [24]. Proses uji coba dimulai dengan membuat fungsi untuk melakukan inferensi menggunakan model yang telah dilatih sebelumnya. Proses inferensi dilakukan dengan memasukkan gambar dari *test set* ke dalam model dengan pengaturan nilai *confidence* yang disesuaikan dengan nilai Kurva F1. Model kemudian memprediksi objek dalam gambar, menghasilkan *output* berupa *bounding box*, label kelas, dan nilai *confidence*. *Bounding box* menunjukkan lokasi objek yang terdeteksi, label kelas menunjukkan kelas yang terdeteksi, sedangkan nilai *confidence* mengindikasikan tingkat keyakinan model terhadap hasil prediksi. Hasil inferensi juga dihitung keberhasilannya dalam mendeteksi bahan pangan tinggi protein pada *test set*. Proses ini membantu dalam mengukur akurasi model dalam mendeteksi bahan pangan tinggi protein pada kasus yang sebenarnya.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Pengumpulan dan Anotasi Data

Pengumpulan data dilakukan untuk membangun *dataset* bahan pangan tinggi protein, yang terdiri dari 10 kelas bahan pangan, yaitu Daging Ayam, Daging Sapi, Ikan Kembung, Ikan Lele, Ikan Nila, Ikan Patin, Tahu, Telur Ayam Broiler, Tempe, dan Udang. *Dataset* yang diunduh melalui platform *roboflow universe* kemudian dilakukan penambahan data sebanyak 10 gambar per kelas, sehingga total *dataset* menjadi 2370 gambar. Data baru kemudian dilakukan anotasi pada Gambar 2 menggunakan platform *roboflow*, dimana setiap citra akan dilengkapi dengan label yang mencakup nama kelas, koordinat titik, dan ukuran objek [33].



Gambar 2. Proses Anotasi Data

Gambar 2 menunjukkan proses anotasi data menggunakan platform *roboflow*. Gambar 2 tampak sepotong daging sapi yang terletak di atas permukaan kayu, dengan dua area yang diberi tanda kotak (*bounding box*). Tanda kotak ini menunjukkan objek yang diidentifikasi dalam gambar, dalam hal ini daging sapi, yang merupakan salah satu kelas bahan pangan tinggi protein dalam *dataset*. Kotak-kotak ini digunakan untuk membantu model dalam mengidentifikasi jenis objek dalam gambar dengan lebih akurat [22]. Platform *roboflow* mempermudah proses ini dengan memungkinkan pengguna untuk menambahkan koordinat titik dan ukuran objek, yang akan berguna dalam pelatihan model deteksi objek.

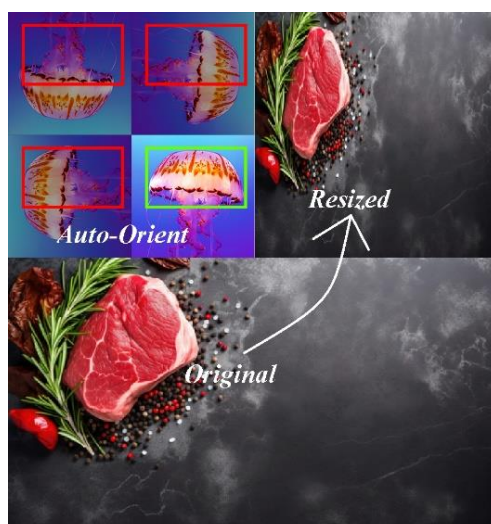
3.2 Hasil Pra-Pemrosesan Data

Tahap pertama dilakukan pembagian data menjadi tiga bagian yaitu *train set*, *valid set*, dan *test set* [22]. Tabel 2 menunjukkan detail hasil pembagian data.

Tabel 2. Hasil Pembagian *Dataset*

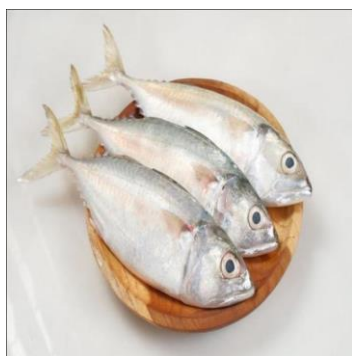
Kelas	<i>Train set</i>	<i>Valid set</i>	<i>Test set</i>	Total/kelas
Daging Ayam	94	23	14	131
Daging Sapi	124	31	15	170
Ikan Kembung	203	56	32	291
Ikan Lele	194	41	17	252
Ikan Nila	171	53	27	251
Ikan Patin	202	54	22	278
Tahu	191	60	39	290
Telur Ayam Broiler	196	66	25	287
Tempe	215	50	27	292
Udang	79	37	12	128
Total Data	1669	471	230	2370

Pembagian *dataset* pada Tabel 2 dibagi ke dalam tiga bagian utama, yaitu *train set*, *valid set*, dan *test set*. *Train set* digunakan untuk melatih model [24] dan merupakan bagian terbesar, yaitu sekitar 70% dari total data. *Valid set* digunakan untuk mengevaluasi performa model selama pelatihan [24], dengan alokasi sekitar 20% dari total data. Sementara itu, *test set* berfungsi untuk menguji model setelah pelatihan selesai [24], dengan porsi sekitar 10% dari keseluruhan *dataset*. Pembagian data memastikan bahwa setiap kategori memiliki representasi yang cukup dalam setiap *subset*, sehingga model dapat belajar dengan baik dan memiliki kemampuan generalisasi yang lebih baik [22]. Tahap selanjutnya dapat dilihat pada Gambar 3, yang menunjukkan sampel gambar yang berfokus pada manipulasi gambar melalui proses *Auto-Orient* dan *Resize*. Proses *Auto-Orient* berguna untuk memperbaiki orientasi gambar secara otomatis ketika gambar dalam kondisi yang tidak konsisten. Proses ini penting karena akan memudahkan model dalam mempelajari pola secara konsisten. Proses *Resize* juga dilakukan dengan menyeragamkan ukuran menjadi 640x640 piksel, sesuai persyaratan *YOLO* yang membutuhkan dimensi tetap untuk memastikan efisiensi dalam proses pelatihan dan inferensi [25].



Gambar 3. Proses *Resized* dan *Auto-Orient*

Penerapan augmentasi pada penelitian ini meliputi *flip* horizontal dan vertikal, *90° Rotate Clockwise*, *Counter-Clockwise*, dan *Upside Down*, *Crop Zoom in* sebesar 20%, Rotasi antara -15° dan $+15^\circ$, serta penambahan *Noise* sebesar 0,5% per piksel. Augmentasi hanya dilakukan pada *train set* [33], sehingga jumlah pada *train set* meningkat dua kali lipat, dari 1669 gambar menjadi 3338 gambar. Gambar 4a menunjukkan citra asli, sedangkan Gambar 4b menunjukkan citra hasil augmentasi.



(a) Citra Awal



(b) Rotasi +15°, Noise 0,5% per piksel

Gambar 4. Hasil Perbandingan Citra Sebelum dan Sesudah Augmentasi

Gambar 4 menunjukkan proses augmentasi citra dengan menerapkan rotasi dan penambahan *noise* pada gambar ikan. Gambar 4a, citra awal menampilkan empat ekor ikan yang tersusun di atas piring kayu tanpa adanya modifikasi. Sementara itu, Gambar 4b, citra telah mengalami transformasi dengan rotasi sebesar +15° serta penambahan *noise* sebesar 0,5% per piksel. Teknik ini digunakan untuk meningkatkan variasi data gambar tanpa mengubah informasi utama yang terkandung dalam citra.

3.3 Hasil Pelatihan Model

Model *YOLOv11* dilatih menggunakan *dataset* deteksi bahan pangan tinggi protein yang terdiri dari 10 kelas. *Dataset* ini mencakup total 3338 gambar untuk pelatihan dan 471 gambar untuk validasi. Proses pelatihan menghabiskan waktu 2 jam 43 menit 11 detik. Model dilatih sebanyak 100 *epoch* menggunakan parameter *default*. Parameter pelatihan ditampilkan dalam Tabel 3.

Tabel 3. Paramater Pelatihan

Model	<i>YOLO11s</i>
<i>Optimizer</i>	<i>AdamW</i>
<i>Leraning Rate Awal</i>	0,01
<i>Loss Function</i>	<i>Box Loss, Cls Loss, Dfl Loss</i>
Ukuran Gambar	640x640 piksel
<i>Batch Size</i>	16

Tabel 3 menunjukkan berbagai parameter yang digunakan dalam proses pelatihan model *YOLOv11*. Pelatihan menggunakan Model *YOLO11s*, versi ringan dari Model *YOLOv11*. *Optimizer* yang digunakan adalah *AdamW*, yang memiliki kemampuan dalam mengurangi *overfitting* [34]. Selain itu, *learning rate* yang digunakan adalah 0,01, yang mempengaruhi seberapa besar langkah yang diambil model dalam menyesuaikan bobotnya pada setiap iterasi [25]. Pengaturan ini penting untuk memastikan model dapat belajar dengan cepat tetapi tetap stabil. Parameter fungsi *loss*, terdapat beberapa komponen penting yang digunakan dalam perhitungan kesalahan model, yaitu *Box Loss*, *Class Loss*, dan *Dfl Loss*. *Box Loss* berfungsi untuk mengukur ketepatan prediksi posisi *bounding box* [25], *Class Loss* untuk mengukur kesalahan klasifikasi objek [25], dan *Dfl Loss* untuk mengukur nilai *loss* dalam distribusi objek [12]. Ukuran gambar yang digunakan dalam pelatihan adalah 640x640 piksel. *Batch size* yang digunakan dalam pelatihan adalah 16, yang menentukan jumlah gambar yang diproses sekaligus dalam satu iterasi [35]. Parameter-parameter ini disesuaikan untuk mencapai keseimbangan antara akurasi model dan efisiensi pelatihan. Tabel 4 menunjukkan perkembangan model secara sederhana.

Tabel 4. Perkembangan Model

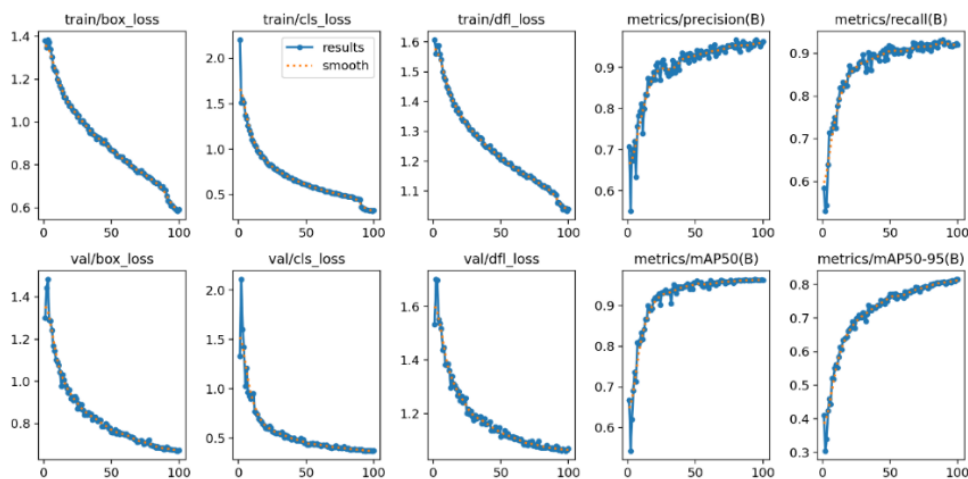
<i>Epoch</i>	<i>Precision</i>	<i>Recall</i>	<i>mAP50</i>	<i>mAP50-95</i>	<i>Box loss</i>	<i>Cls Loss</i>	<i>Dfl Loss</i>
1	70,7%	58,5%	66,7%	41,1%	1.38	2.205	1.608
50	92,7%	90,5%	95,7%	77,1%	0.8669	0.611	1.204
100	96,3%	91,9%	96,3%	81,5%	0.5918	0.3265	1.038

Tabel 4 menunjukkan perkembangan performa model *YOLOv11* yang dilatih selama 100 *epoch*, dengan berbagai metrik yang mengukur kualitas prediksi model. Pada *epoch* pertama, model menunjukkan hasil yang cukup rendah, dengan *precision* 70,7%, *recall* 58,5%, *mAP50* 66,7%, dan *mAP50-95* sebesar 41,1%. Selain itu, nilai *box loss*, *cls loss*, dan *dfl loss* berada pada angka yang lebih tinggi, yaitu 1.38, 2.205, dan 1.608, yang mengindikasikan bahwa model masih dalam tahap pembelajaran awal sehingga angka *loss* masih tinggi. Meskipun demikian, nilai-nilai ini mulai menunjukkan peningkatan yang signifikan di *epoch* ke-50, dengan *precision* dan *recall* meningkat masing-masing menjadi 92,7% dan 90,5%, serta *mAP50* dan *mAP50-95* mencapai 95,7% dan 77,1%. Nilai *loss* juga

menunjukkan penurunan yang signifikan di *epoch* ke-50, dengan *box loss*, *cls loss*, dan *dfl loss* mendapatkan nilai masing-masing 0.8669, 0.611, dan 1.204.

Iterasi pada *epoch* ke-100, model menunjukkan performa yang sangat baik, dengan *precision* mencapai 96,3% dan *recall* 91,9%. Nilai *mAP50* semakin mendekati 100% di angka 96,3%, sementara *mAP50-95* juga mengalami peningkatan signifikan hingga 81,5%. Selain itu, nilai *box loss*, *cls loss*, dan *dfl loss* semakin menurun, masing-masing menjadi 0.5918, 0.3265, dan 1.038, yang menandakan bahwa model berhasil meminimalkan kesalahan dalam prediksi posisi, klasifikasi, dan distribusi objek [12]. Secara keseluruhan, Nilai-nilai pada Tabel 4 menunjukkan bahwa model berhasil mempelajari fitur dari *dataset* dengan baik.

Performa model selama pelatihan dapat dilihat pada Gambar 5, yang menampilkan grafik dari berbagai metrik, meliputi *box loss*, *cls loss*, *dfl loss*, *precision*, *recall*, *mAP50*, dan *mAP50-95*.



Gambar 5. Grafik Hasil Pelatihan

Grafik *train loss* dan *val loss* memperlihatkan konsistensi dengan penurunan nilai *loss* pada kedua *dataset*. Grafik *train loss* dan *val loss* menunjukkan bahwa model berhasil belajar tanpa mengalami *overfitting*, seperti yang ditunjukkan oleh nilai *val loss* yang tetap stabil dan tidak jauh berbeda dari *train loss*. Nilai-nilai tersebut memperlihatkan bahwa Model *YOLOv11* dengan baik melakukan generalisasi data yang belum pernah dilihat sebelumnya. Hasil pelatihan secara keseluruhan yang dilakukan pada model menunjukkan performa yang memuaskan, dengan semua metrik *error* mengalami penurunan yang konsisten dan *mAP* meningkat signifikan selama 100 *epoch*. Hal ini menunjukkan bahwa Model *YOLOv11* berhasil mempelajari fitur *dataset* untuk mendeteksi bahan pangan tinggi protein.

3.4 Hasil Evaluasi Model

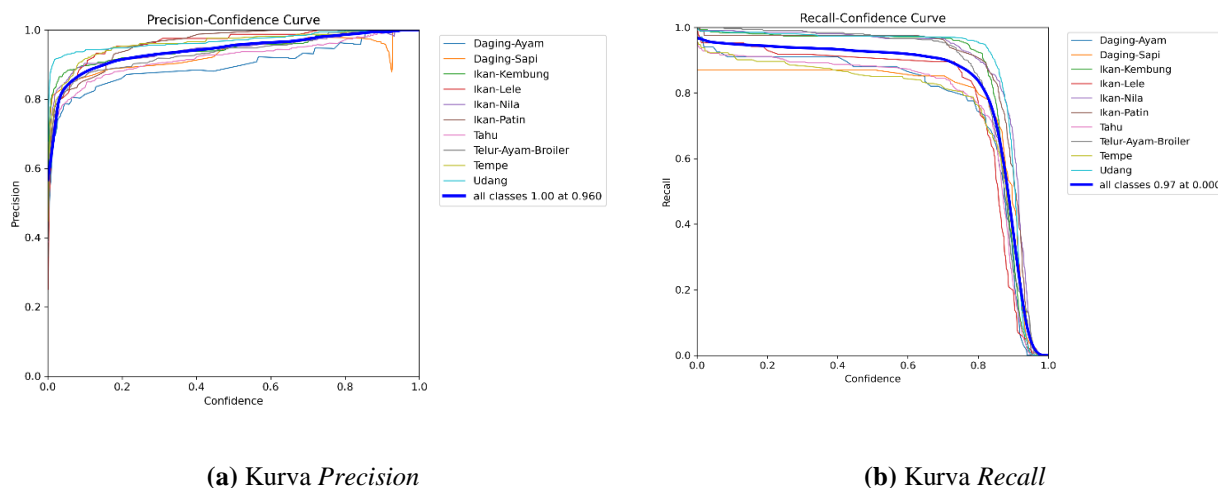
Evaluasi model dilakukan untuk menilai performa deteksi bahan pangan tinggi protein berdasarkan metrik *precision*, *recall*, *mean Average Precision* pada *IoU threshold* 50% (*mAP50*), dan rata-rata *mAP* pada berbagai *threshold IoU* (*mAP50-95*). Hasil evaluasi menggunakan data validasi yang terdiri dari 469 gambar dengan total 2.225 *instance* objek dari 10 kelas bahan pangan tinggi protein. Tabel 5 menunjukkan hasil evaluasi model keseluruhan.

Tabel 5. Hasil Evaluasi Model

<i>Precision</i>	<i>Recall</i>	<i>mAP 50</i>	<i>mAP 50-95</i>
96%	92,3%	96,4%	81,5%

Hasil evaluasi pada Tabel 5 menunjukkan performa model yang sangat baik dalam hal *presisi* dan *recall*. *Precision* model mencapai 96%, yang menunjukkan bahwa sebagian besar deteksi objek oleh model adalah akurat, dengan sedikit kesalahan prediksi positif. *Recall* sebesar 92,3% mengindikasikan bahwa model mampu mendeteksi sebagian besar objek yang ada, meskipun ada beberapa objek yang terlewatkan dalam deteksi. Metrik evaluasi pada *mAP50* juga mencapai 96,4% menunjukkan bahwa model dapat mencapai deteksi yang sangat baik pada *threshold Intersection over Union (IoU)* 50%, yang digunakan untuk menilai kecocokan prediksi *bounding box* dengan objek sebenarnya. Metrik *mAP50-95* yang mencapai 81,5% menunjukkan performa model yang cukup baik pada berbagai *threshold IoU*, meskipun ada penurunan sedikit dibandingkan dengan *mAP50*. Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa model *YOLOv11* memiliki kemampuan yang sangat baik dalam mendeteksi objek bahan pangan tinggi protein secara akurat, dengan tingkat kesalahan yang sangat rendah. Hasil pada Tabel 5 memperlihatkan performa dan akurasi yang sangat baik pada model. Berikut adalah analisis mendalam dari evaluasi model berdasarkan kurva evaluasi.

3.4.1 Kurva Precision dan Recall

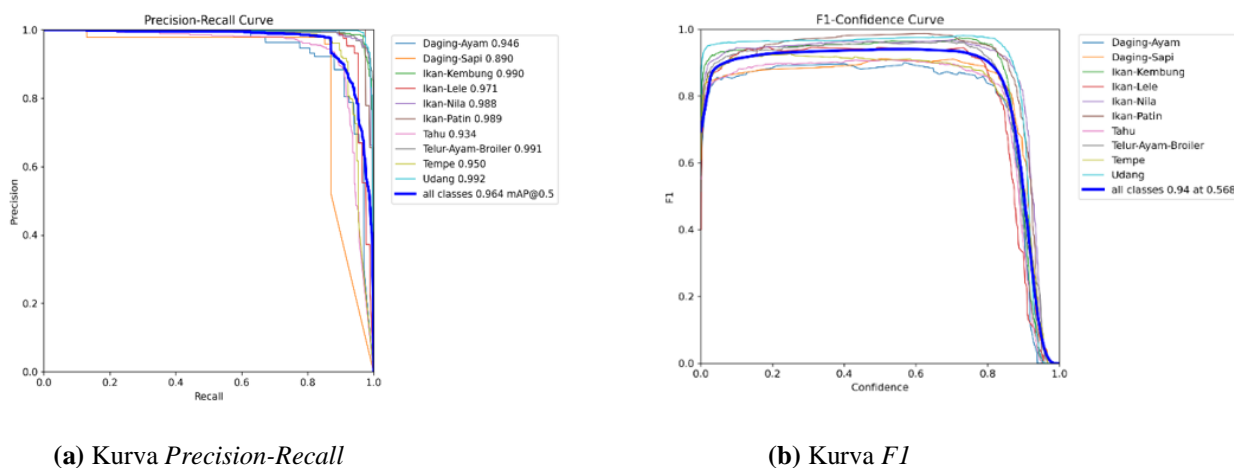


Gambar 6. Kurva Precision dan Recall

Gambar 6a menunjukkan kurva *precision* yang menggambarkan hubungan antara nilai *confidence threshold* dengan nilai *precision* dari model. Berdasarkan kurva, *Precision* cenderung stabil diatas 90% pada berbagai level *confidence* hingga mendekati nilai *confidence* maksimum. Stabilitas *precision* ini mengindikasikan bahwa model memiliki kemampuan deteksi yang cukup konsisten, meskipun *confidence threshold* dinaikkan. Hal ini sangat penting dalam aplikasi nyata di mana nilai kepercayaan tinggi diperlukan untuk mengurangi kesalahan prediksi.

Kurva *recall* pada Gambar 6b menggambarkan kemampuan model untuk mendeteksi semua *instance* yang relevan. Berdasarkan kurva, *Recall* mencapai nilai maksimum mendekati 1 pada *confidence threshold* yang rendah dan tetap berada di kisaran tinggi hingga *confidence threshold* meningkat. Penurunan *recall* yang sedikit pada nilai *confidence threshold* yang tinggi menunjukkan adanya kompromi dengan nilai *precision*, yang sesuai dengan konsep *trade-off* dalam metrik evaluasi.

3.4.2 Kurva Precision-Recall dan F1



Gambar 7. Kurva Precision-Recall dan F1

Kurva *PR* pada Gambar 7a memberikan gambaran menyeluruh mengenai hubungan antara *precision* dan *recall* untuk semua kelas. Kurva menunjukkan bahwa model mampu mempertahankan *precision* tinggi meskipun *recall* meningkat. Area bawah kurva *PR* yang besar (*AUC-PR*) mengindikasikan performa model yang kuat dalam mendeteksi bahan pangan secara efektif, terutama pada kelas-kelas dengan jumlah data yang mencukupi.

Kombinasi yang baik antara *precision* dan *recall* ditunjukkan oleh hubungan antara *confidence threshold* dengan nilai *F1-Score* Kurva *F1* pada Gambar 7b [27]. Kurva ini menunjukkan bahwa nilai *F1-Score* maksimum dicapai pada *confidence threshold* 0,568 dengan *F1* sebesar 0,94. Hal ini menegaskan performa optimal model pada titik *threshold* tertentu yang dapat dijadikan referensi untuk implementasi.

3.4.3 Evaluasi Setiap kelas

Evaluasi dilakukan untuk menentukan kelas dengan performa unggul dan kelas yang masih membutuhkan peningkatan. Tabel 6 menunjukkan hasil evaluasi model pada setiap kelas.

Tabel 6. Evaluasi Model Setiap Kelas

Kelas	<i>Precision</i> (P)	<i>Recall</i> (R)	<i>mAP 50</i>	<i>mAP 50-95</i>
Daging Ayam	90,5%	88,1%	94,6%	72,2%
Daging Sapi	95,9%	86,3%	89,1%	77,8%
Ikan Kembung	95,8%	97,2%	99%	84,8%
Ikan Lele	98,7%	90,3%	97,1%	79,3%
Ikan Nila	95,4%	96,6%	98,8%	89,7%
Ikan Patin	99,6%	97,6%	98,9%	92,2%
Tahu	93,4%	87,6%	93,3%	71,6%
Telur Ayam Broiler	94,8%	96,6%	99,1%	81,8%
Tempe	98%	85%	95,1%	80%
Udang	97,7%	97,6%	99,2%	85,4%
Keseluruhan	96%	92,3%	96,4%	81,5%

a. Performa Kelas Unggul

1. Kelas Ikan Patin menunjukkan performa terbaik berdasarkan *mAP50* (98,9%) dan *mAP50-95* (92,2%). Tingkat *precision* (99,6%) dan *recall* (97,6%) yang sangat tinggi menunjukkan model mampu mendeteksi objek dengan sangat akurat.
2. Kelas Udang juga menunjukkan hasil unggul dengan *precision* sebesar 97,7%, *recall* 97,6%, *mAP50* sebesar 99,2%, dan *mAP50-95* sebesar 85,4%. Hal ini mengindikasikan kemampuan model yang konsisten dalam mengklasifikasikan objek di kelas Udang.

b. Performa Kelas Perlu Peningkatan

1. Kelas Daging Ayam memiliki performa terendah pada *mAP50-95*, yaitu 72,2%, meskipun *precision* cukup baik di angka 90,5%. Hal ini menunjukkan bahwa model memiliki kesulitan dalam mendeteksi objek pada berbagai skenario atau kondisi tertentu yang lebih kompleks. Perlu dilakukan evaluasi lebih lanjut pada *dataset* kelas ini, misalnya memperbaiki representasi data atau mengatasi masalah bias pada gambar.
2. Kelas Tahu juga menunjukkan *mAP50-95* yang relatif rendah (71,6%) dibandingkan kelas lain. Namun, nilai *precision* (93,4%) masih cukup baik, mengindikasikan perlunya perbaikan pada *recall* atau kondisi deteksi lainnya.

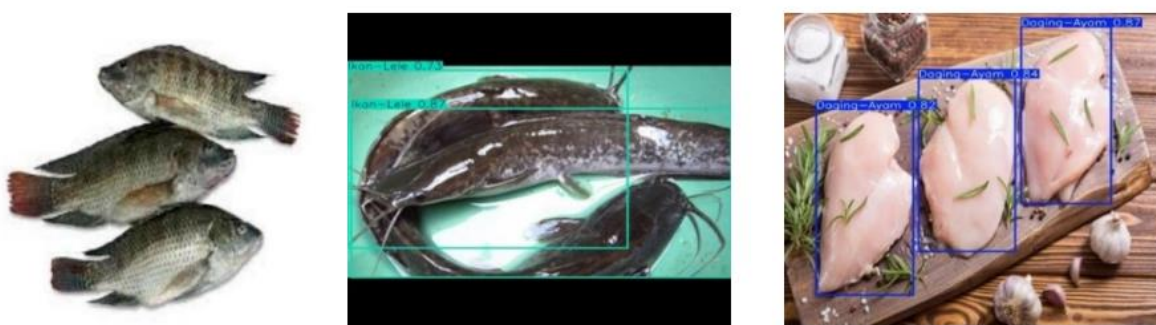
Model *YOLOv11* menunjukkan performa yang sangat baik untuk tugas deteksi bahan pangan tinggi protein, dengan *mAP50* mencapai 96,4% dan *mAP50-95* sebesar 81,5%. Namun, diperlukan pengoptimalan lebih lanjut, terutama pada kelas dengan performa terendah seperti Daging Ayam dan Tahu, untuk meningkatkan kemampuan deteksi secara menyeluruh.

3.5 Hasil Uji Coba Model

Proses uji coba menerapkan model *best.pt* yang merupakan hasil dari pelatihan pada tahap sebelumnya. Model diuji menggunakan 230 gambar yang ada pada *test set*. Proses inferensi dilakukan dengan waktu 18,5 ms. Hasil uji coba menunjukkan bahwa model berhasil mendeteksi objek benar pada 226 gambar, sedangkan model tidak berhasil mendeteksi 4 gambar objek sesuai dengan *groundtruth*. Rumus pada Persamaan (4) bertujuan untuk menghitung tingkat keberhasilan deteksi.

$$\text{Tingkat Keberhasilan (\%)} = \frac{\text{Jumlah Gambar Benar}}{\text{Total Gambar}} \times 100 = \frac{226}{230} \times 100 = 98,26\% \quad (4)$$

Hasil perhitungan pada Persamaan 4 menunjukkan bahwa model mendeteksi objek pada gambar tes dengan sangat baik. Namun, terdapat ketidakberhasilan deteksi pada beberapa gambar, terutama pada kondisi tertentu seperti resolusi gambar yang rendah. Contoh deteksi gagal dapat dilihat pada Gambar 8.



(a) Ikan Nila dan Ikan Lele

(b) Daging Ayam

Gambar 8. Hasil Uji Coba

Gambar 8a, terdapat gambar ikan nila dan ikan lele, di mana model berhasil mendeteksi dua objek ikan lele dari tiga yang ada, namun gagal mendeteksi ikan nila yang seharusnya terdeteksi. Hal ini menunjukkan bahwa model memiliki keterbatasan dalam menangani gambar dengan resolusi rendah atau kondisi tertentu yang mempengaruhi kejelasan objek dalam gambar. Ketidakberhasilan ini berpengaruh pada akurasi deteksi yang mengarah pada penurunan performa pada gambar tertentu.

Gambar 8b, hasil deteksi model menunjukkan performa yang lebih baik dengan mendeteksi semua objek pada gambar daging ayam. Setiap potongan daging ayam terdeteksi dengan benar dan dibatasi dengan kotak pembatas yang tepat. Pengujian pada Gambar 8b menunjukkan bahwa model memiliki tingkat keberhasilan yang sangat baik pada gambar dengan kualitas dan kontras yang lebih jelas, seperti yang terlihat pada gambar daging ayam ini. Meskipun ada beberapa kegagalan deteksi pada gambar lain, hasil ini menggambarkan potensi besar dari model untuk mendeteksi objek dengan akurat pada kondisi yang mendukung.

4. KESIMPULAN

Penelitian ini berhasil membuat model deteksi bahan pangan tinggi protein menggunakan Model *YOLOv11*, yang menunjukkan performa unggul dalam mendeteksi objek. Berdasarkan hasil evaluasi, model mencapai *precision* 96%, *recall* 92,3%, *mAP50* sebesar 96,4% dan *mAP50-95* sebesar 81,5%. Hasil ini menunjukkan bahwa Model *YOLOv11* mampu mendeteksi bahan pangan seperti daging ayam, daging sapi, ikan kembung, ikan lele, ikan nila, ikan patin, tahu, telur ayam broiler, tempe, dan udang dengan akurasi tinggi, menjadikannya teknologi yang potensial untuk membantu pencegahan stunting melalui pemilihan bahan pangan kaya protein yang tepat. Keterbatasan penelitian ini perlu dipertimbangkan untuk pengembangan berikutnya. Pertama, performa deteksi pada beberapa kelas, seperti Daging Ayam dan Tahu, masih perlu ditingkatkan, terutama pada kondisi gambar yang kompleks atau dengan variasi objek yang kurang representatif dalam *dataset*. Kedua, meskipun augmentasi data telah diterapkan, *dataset* yang digunakan relatif terbatas dalam hal jumlah dan variasi, yang dapat memengaruhi generalisasi model terhadap data dunia nyata. Selain itu, model belum diintegrasikan dengan sistem rekomendasi menu MP-ASI berbasis hasil deteksi, usia anak, dan pedoman ahli gizi, yang dapat memperluas aplikasi teknologi ini dalam pencegahan stunting. Penelitian di masa depan dapat berfokus pada peningkatan kualitas *dataset* dengan memperbanyak jumlah data dan memperluas variasi kondisi gambar, seperti pencahayaan, sudut pengambilan gambar, dan resolusi. Selain itu, penerapan model yang lebih ringan dan efisien juga dapat menjadi pilihan untuk memastikan implementasi yang lebih praktis pada perangkat dengan sumber daya terbatas. Integrasi dengan sistem rekomendasi berbasis kecerdasan buatan juga dapat memberikan solusi yang lebih komprehensif dalam mendukung program pencegahan stunting di masyarakat.

REFERENCES

- [1] T. Sudiarti, "Nutrition Intake and Stunting of Under-Five Children in Bogor West Java, Indonesia," *Food Sci Nutr*, vol. 7, no. 3, hlm. 1–7, Jul 2021, doi: 10.24966/FSN-1076/100104.
- [2] R. Yulina Widiastuti dan R. Dania Faiza, "Upaya Kader Posyandu dalam Mengurangi Tingkat Stunting di Desa Pakel Kabupaten Jombang," *Jurnal Pendidikan Luar Sekolah*, Sep 2022.
- [3] Kemenkes, *Panduan Kegiatan Hari Gizi Nasional: MP-ASI Kaya Protein Hewani Cegah Stunting*. 2024.
- [4] Kemenkes, *Panduan Kegiatan Hari Gizi Nasional: Protein Hewani Cegah Stunting*. 2023.
- [5] E. Putri Silmin dan T. Hardiani, "Perancangan Sistem Pakar Penyakit Pneumonia Pada Balita Menggunakan Algoritme K-NN (K-Nearest Neighbor)," *Jurnal Pseudocode*, Sep 2018.
- [6] T. Hardiani dan R. N. Putri, "Implementasi Metode Naïve Bayes Classifier Untuk Klasifikasi Stunting Pada Balita," *Digital Transformation Technology*, vol. 4, no. 1, hlm. 621–627, Agu 2024, doi: 10.47709/digitech.v4i1.4481.
- [7] N. K. Negoro, E. Utami, dan A. Yaqin, "Klasifikasi Deteksi Penggunaan Masker Menggunakan Metode Convolutional Neural Network," *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 2, hlm. 664–674, Mei 2023, doi: 10.29100/jipi.v8i2.3748.
- [8] L. Tan, T. Huangfu, L. Wu, dan W. Chen, "Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification," 30 Juli 2021. doi: 10.21203/rs.3.rs-668895/v1.
- [9] J. Kim, J.-Y. Sung, dan S. Park, "Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition," dalam *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, IEEE, Nov 2020, hlm. 1–4. doi: 10.1109/ICCE-Asia49877.2020.9277040.
- [10] L. Rahma, H. Syaputra, A. H. Mirza, dan S. D. Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once)," *Jurnal Nasional Ilmu Komputer*, vol. 2, no. 3, hlm. 213–232, Nov 2021, doi: 10.47747/jurnalnik.v2i3.534.
- [11] T. Handayani, "Implementasi Metode You Only Look Once (Yolo) Untuk Deteksi Kesegaran Telur Ayam Berdasarkan Citra Cangkar," *Kohesi: Jurnal Multidisiplin Saintek*, vol. 3, no. 9, 2024.
- [12] Aldra Kasyfil Aziz, Ledya Novamizanti, dan Suryo Adhi Wibowo, "Perancangan dan Implementasi Deep Learning untuk Deteksi Kesegaran Ikan Beku pada Aplikasi FishQ menggunakan YOLOv8," *Jurnal Nasional SAINS dan TEKNIK*, vol. 2, no. 2, hlm. 1–6, 2024.
- [13] Ultralytics, "Ultralytics YOLO11." Diakses: 18 Desember 2024. [Daring]. Tersedia pada: <https://docs.ultralytics.com/models/yolo11/>
- [14] R. Sapkota, Z. Meng, dan M. Karkee, "Synthetic meets authentic: Leveraging LLM generated datasets for YOLO11 and YOLOv10-based apple detection through machine vision sensors," *Smart Agricultural Technology*, vol. 9, hlm. 100614, Des 2024, doi: 10.1016/j.atech.2024.100614.



- [15] N. Jegham, C. Y. Koh, M. Abdelatti, dan A. Hendawi, “Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors,” Okt 2024.
- [16] M. A. R. Alif, “YOLOv11 for Vehicle Detection: Advancements, Performance, and Applications in Intelligent Transportation Systems,” Okt 2024.
- [17] M. N. Al Jihad dkk., “Cegah Stunting Berbasis Teknologi, Keluarga, Dan Masyarakat,” *SALUTA: Jurnal Pengabdian Kepada Masyarakat*, vol. 1, no. 2, hlm. 31, Feb 2022, doi: 10.26714/sjpkm.v1i2.8683.
- [18] K. K. Arum Nurcahyanti dan I. Rahmansyah, “Efektivitas Pemanfaatan Stunting Mobile untuk Pencegahan Stunting dengan Meningkatkan Pemahaman Kebutuhan Asupan Makan di Kabupaten Banyumas,” *Jurnal Multidisiplin Indonesia*, vol. 2, no. 3, hlm. 594–599, Mar 2023, doi: 10.58344/jmi.v2i3.199.
- [19] Yanuardi Yanuardi, Rohmat Taufiq, Sumardi Sadi, Sri Mulyati, dan Faridi Faridi, “Pemanfaatan Teknologi Dalam Pencegahan Stunting di Lingkungan Masyarakat Desa Suka Asih Kecamatan Pasar Kemis Kabupaten Tangerang,” *Journal of Social Science and Technology for Community Services (JSSTCS)*, vol. 5, no. 1, Mar 2024.
- [20] E. F. Grathima, E. Yunitasari, dan R. Indarwati, “Dampak Intervensi Complementary Food pada Anak dalam Pencegahan Stunting,” *Journal of Telenursing (JOTING)*, vol. 6, no. 1, hlm. 367–375, Feb 2024, doi: 10.31539/joting.v6i1.9024.
- [21] N. J. Hayati, D. Singasatia, dan M. R. Muttaqin, “Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan,” *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, hlm. 91–99, Nov 2023, doi: 10.34010/komputa.v12i2.10654.
- [22] D. Jayakumar dan S. Peddakrishna, “Performance Evaluation of YOLOv5-based Custom Object Detection Model for Campus-Specific Scenario,” *International Journal of Experimental Research and Review*, vol. 38, hlm. 46–60, Apr 2024, doi: 10.52756/ijerr.2024.v38.005.
- [23] I. Nasution, A. P. Windarto, dan M. Fauzan, “Penerapan Algoritma K-Means Dalam Pengelompokan Data Penduduk Miskin Menurut Provinsi,” *Building of Informatics, Technology and Science (BITS)*, vol. 2, no. 2, hlm. 76–83, Des 2020, doi: 10.47065/bits.v2i2.492.
- [24] N. T. Adam, Z. A. Tyas, dan T. Hardiani, “Deteksi Gestur Sistem Isyarat Bahasa Indonesia Menggunakan Metode Deep learning SSD MobileNet V2 FPN Lite,” *Sainteks*, vol. 21, no. 2, hlm. 129, Okt 2024, doi: 10.30595/sainteks.v21i2.24006.
- [25] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, dan E. R. Swedia, “SISTEM PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN ARSITEKTUR YOLOV8,” *Sebatik*, vol. 27, no. 2, hlm. 753–761, Des 2023, doi: 10.46984/sebatik.v27i2.2374.
- [26] Jesica Trivena Sinaga, Haniifa Aliila Faudyta, dan Egia Rosi Subhiyacto, “Klasifikasi Kanker Kulit menggunakan Convolutional Neural Network dengan Optimasi Arsitektur VGG16,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 3, Des 2024.
- [27] Syadina A. Prasetya, Mihuandayani Mihuandayani, Yansen Abast, Michael Mangole, dan Jonathan Rahman, “Implementasi Convolutional Neural Network untuk Klasifikasi Tingkat Kematangan Buah Nanas Menggunakan YOLOv8,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 1, Jun 2024.
- [28] M. A. Rohman, P. Mudjirahardjo, dan M. A. Muslim, “Implementasi Filter Gray Level Co-Occurance Matriks Terhadap Sistem Klasifikasi Kanker Payudara Dengan Metode Convolutional Neural Network,” *Transmisi*, vol. 23, no. 4, hlm. 160–168, Agu 2021, doi: 10.14710/transmisi.23.4.160-168.
- [29] A. F. Oklilas, S. Sukemi, dan R. Apriliyanto, “Model Yolo Versi 4 Pada Pengenalan Kendaraan Di Jalan Raya Kota Palembang,” *Transmisi: Jurnal Ilmiah Teknik Elektro*, vol. 25, no. 3, hlm. 136–139, Agu 2023, doi: 10.14710/transmisi.25.3.136-139.
- [30] B. Prihasto, N. R. Fadhliana, A. Hariyani, F. M. Alwafi, dan T. B. Askarin, “Development of Object Detection System on Non-Helmeted Riders Using YOLOv8,” *Jurnal Pendidikan Multimedia (Edsence)*, vol. 5, no. 2, hlm. 79–90, Des 2023, doi: 10.17509/edsence.v5i2.65910.
- [31] U. Ali, M. A. Ismail, R. A. Ariyaluran Habeeb, dan S. R. Ali Shah, “Performance Evaluation of YOLO Models in Plant Disease Detection,” *Journal of Informatics and Web Engineering*, vol. 3, no. 2, hlm. 199–211, Jun 2024, doi: 10.33093/jiwe.2024.3.2.15.
- [32] A. K. Wang dan Thoyyibah, “Pengembangan Sistem Deteksi Digit pada Meteran Air PDAM Menggunakan Model Deep Learning YOLOv5,” *Jurnal Ilmu Komputer*, vol. 1, no. 2, Des 2023.
- [33] Raihan Restu Putra, Maimunah Maimunah, dan Dimas Sasongko, “Implementasi Algoritma YOLO V8 (You Only Look Once) Dalam Deteksi Penyakit Daun Durian,” *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 3, Des 2024.
- [34] Kaustubh Yadav, “A Comprehensive Study on Optimization Strategies for Gradient Descent In Deep Learning,” *ArXiv*, 2021.
- [35] A. Akram, K. Fayakun, dan H. Ramza, “Klasifikasi Hama Serangga pada Pertanian Menggunakan Metode Convolutional Neural Network,” *Building of Informatics, Technology and Science (BITS)*, vol. 5, no. 2, Sep 2023, doi: 10.47065/bits.v5i2.4063.