

Penerapan Algoritma *You Only Look Once* (YOLO) untuk Deteksi Kekosongan Rak pada Sistem Pemantauan Barang

Mirawati^{1*}, Binanda Wicaksana²

^{1,2}Program Studi Teknologi Informasi, Fakultas Informatika dan Komputer, Universitas Binaniaga Indonesia

¹e-mail: mirawati1833@gmail.com

²e-mail: binandawicaksana@gmail.com

*Corresponding Author

ABSTRACT

The availability of goods on retail shelves is an important factor for customer satisfaction and smooth sales. The monitoring process, which is still carried out manually, often causes delays in detecting empty shelves. This study aims to develop an Internet of Things (IoT)-based goods monitoring system with the application of the *You Only Look Once* (YOLO) algorithm to detect empty shelves quickly and accurately. The system uses an ESP32-CAM device as an image capture device that is processed through the YOLO algorithm to identify product availability. The type of research used is Research and Development (R&D) with stages of needs analysis, design, prototyping, testing, and evaluation. The results show that the system is capable of detecting empty shelves with an accuracy of 81.8% and received a feasibility rating of 100% from experts and 85% from users. This proves that the application of IoT-based YOLO is effective in improving stock monitoring efficiency and reducing delays in restocking. This system is suitable for implementation in retail environments and has the potential to be further developed into a smart inventory management solution based on computer vision.

Keywords: Vacancy detection, ESP32-CAM, Internet of Things, monitoring system, YOLO.

ABSTRAK

Ketersediaan barang di rak ritel menjadi faktor penting bagi kepuasan pelanggan dan kelancaran penjualan. Proses pemantauan yang masih dilakukan secara manual sering menyebabkan keterlambatan dalam mendeteksi kekosongan rak. Penelitian ini bertujuan mengembangkan sistem pemantauan barang berbasis Internet of Things (IoT) dengan penerapan algoritma *You Only Look Once* (YOLO) untuk mendeteksi kekosongan rak secara cepat dan akurat. Sistem menggunakan perangkat ESP32-CAM sebagai pengambil gambar yang diolah melalui algoritma YOLO untuk mengidentifikasi ketersediaan barang. Jenis penelitian yang digunakan adalah *Research and Development* (R&D) dengan tahapan analisis kebutuhan, perancangan, pembuatan prototipe, pengujian, dan evaluasi. Hasil penelitian menunjukkan sistem mampu mendeteksi kekosongan rak dengan akurasi 81,8% dan memperoleh penilaian kelayakan 100% dari ahli serta 85% dari pengguna. Hal ini membuktikan bahwa penerapan YOLO berbasis IoT efektif dalam meningkatkan efisiensi pemantauan stok dan mengurangi keterlambatan pengisian barang. Sistem ini layak diterapkan di lingkungan ritel dan berpotensi dikembangkan lebih lanjut menjadi solusi manajemen inventaris cerdas berbasis visi komputer.

Kata Kunci: Deteksi kekosongan, ESP32-CAM, Internet of Things, sistem pemantauan, YOLO.

A. PENDAHULUAN

Ketersediaan barang di rak ritel merupakan faktor penting yang memengaruhi kepuasan pelanggan dan kelancaran penjualan. Namun, banyak swalayan masih mengandalkan sistem pemantauan manual yang bergantung pada karyawan toko, sehingga sering terjadi keterlambatan dalam mendeteksi kekosongan rak. Kondisi tersebut dapat menyebabkan kehilangan pelanggan karena sebagian besar konsumen cenderung berpindah ke toko lain ketika produk yang diinginkan tidak tersedia. Oleh karena itu, diperlukan sistem pemantauan berbasis teknologi yang mampu menyajikan informasi ketersediaan barang secara lebih cepat, akurat, dan efisien.

Beberapa penelitian sebelumnya menunjukkan penerapan teknologi Internet of Things (IoT) dan *computer vision* dalam meningkatkan efektivitas pemantauan stok barang. Aplikasi berbasis web untuk pemantauan stok secara akurat dan efisien telah dikembangkan guna mendukung pengelolaan stok di CV Bina Anak Papua (Samanui & Prayitno, 2024). Sistem manajemen gudang berbasis *Radio Frequency Identification* (RFID) juga dirancang untuk memantau stok secara *real-time* dan mempercepat proses keluar-masuk barang (Moh Muthohir, Sindhu Rakasiwi, 2023). Selain itu, sistem IoT berbasis RFID terbukti dapat meningkatkan akurasi data stok di gudang bahan baku serta memperbaiki proses pemantauan inventaris (Butsianto et al., 2021). Penelitian-penelitian tersebut masih berfokus pada pelacakan stok dan belum mengintegrasikan teknologi visi komputer untuk mendeteksi kekosongan rak secara langsung.

Berdasarkan penelitian sebelumnya, penelitian ini mengembangkan sistem pemantauan barang berbasis IoT dengan algoritma *You Only Look Once* (YOLO) yang dikombinasikan dengan perangkat ESP32-CAM. Sistem ini berfungsi menangkap citra kondisi rak secara berkala, kemudian melakukan deteksi objek guna mengidentifikasi keberadaan atau kekosongan produk. Data hasil deteksi ditampilkan melalui antarmuka web agar pengelola toko dapat mengetahui kondisi rak dan melakukan pengisian ulang barang dengan lebih efisien.

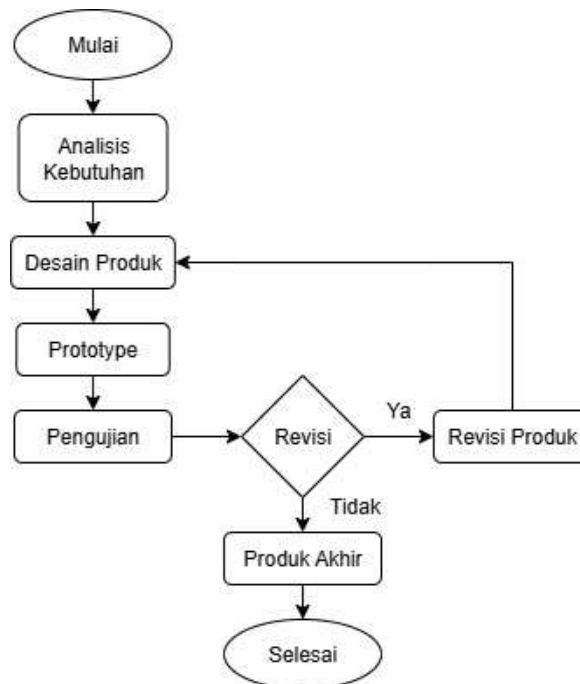
Tujuan penelitian ini adalah merancang dan menguji sistem pemantauan barang berbasis IoT yang mampu mendeteksi kekosongan rak dengan menggunakan algoritma YOLO dan modul ESP32-CAM. Melalui pengujian

sistem, penelitian ini juga bertujuan mengevaluasi tingkat akurasi serta efektivitasnya dalam membantu proses pengelolaan stok. Berdasarkan landasan teori dan hasil penelitian terdahulu, hipotesis yang diajukan adalah bahwa penerapan teknologi IoT dengan algoritma YOLO dapat menghasilkan sistem pemantauan yang efektif dan akurat dalam mendeteksi kekosongan rak, serta berpotensi diterapkan di lingkungan ritel untuk meningkatkan efisiensi operasional.

B. METODE

1. Tahap Penelitian

Proses pengembangan sistem pemantauan barang dalam penelitian ini diawali dengan analisis kebutuhan untuk mengidentifikasi aspek-aspek yang harus dipenuhi oleh sistem, termasuk spesifikasi fungsional dan non- fungsional, jenis barang yang akan dipantau, serta perangkat yang digunakan, seperti ESP32-CAM dan server untuk pemrosesan data. Berdasarkan hasil analisis tersebut, dilakukan tahap desain produk yang berfokus pada perancangan sistem secara menyeluruh, mulai dari arsitektur, alur kerja modul, antarmuka pengguna, hingga integrasi antara modul pengambilan gambar, algoritma deteksi YOLO, dan server. Setelah desain selesai, sistem dikembangkan menjadi prototype sebagai bentuk awal implementasi, yang mencakup pengaturan kamera, implementasi awal algoritma YOLO, serta antarmuka dasar untuk menampilkan hasil deteksi. Prototype kemudian diuji untuk memastikan sistem berfungsi sesuai rancangan, dengan fokus pada akurasi deteksi, kecepatan respons, dan kestabilan koneksi antara ESP32-CAM dan server. Jika pengujian menemukan kekurangan, dilakukan revisi terhadap desain, model YOLO, atau antarmuka pengguna, kemudian diuji kembali hingga sistem memenuhi kriteria efektivitas yang ditentukan. Setelah semua perbaikan selesai dan pengujian menunjukkan hasil yang memuaskan, sistem dikembangkan menjadi produk akhir yang siap digunakan untuk memantau kondisi rak secara berkala. Gambar 1 menunjukkan alur prosedur pengembangan sistem yang diterapkan dalam penelitian ini.



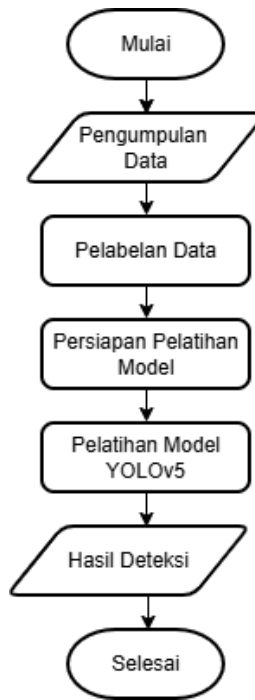
Gambar 1 Prosedur Pengembangan

2. Algoritma YOLO (*You Only Look Once*)

YOLO (*You Only Look Once*) merupakan algoritma deteksi objek berbasis *deep learning* yang bekerja secara menyeluruh dalam satu tahap pemrosesan untuk menentukan letak dan jenis objek pada suatu citra (Eko Supriyanto et al., 2025, p. 36). Algoritma ini menggunakan jaringan *deep convolutional neural network* yang membagi citra menjadi sejumlah grid, di mana setiap sel memprediksi keberadaan objek dengan menetapkan *bounding box* dan kelas objek secara langsung (Eko Supriyanto et al., 2025, p. 37). Pengembangan model YOLO dari versi pertama hingga YOLOv5 menunjukkan peningkatan kinerja dalam hal kecepatan, efisiensi pemrosesan, dan kemampuan penerapan pada perangkat dengan spesifikasi rendah seperti *edge device* (Ir. Eko Supriyanto et al., 2025, pp. 37–40).

Dalam penelitian ini, YOLOv5 digunakan karena memiliki keunggulan dalam kecepatan melakukan deteksi objek serta kemudahan penerapan pada sistem berbasis Internet of Things (IoT). Model YOLOv5 dilatih

menggunakan dataset gambar berbagai jenis barang yang diambil melalui kamera ponsel. Alur proses pelatihan model tersebut digambarkan pada Gambar 2, yang mencakup tahapan pengumpulan data, pelabelan data, persiapan pelatihan model, pelatihan YOLOv5, hingga menghasilkan hasil deteksi yang digunakan dalam sistem pemantauan barang.



Gambar 1 Alur Proses Model YOLOv5

3. Teknik Analisis Data

Untuk mengevaluasi akurasi sistem dalam mendeteksi kekosongan rak, digunakan metode *confusion matrix*. Metode ini digunakan untuk mengukur kinerja model deteksi dalam mengenali lima kelas produk yang diuji, sehingga berbentuk *multi-class confusion matrix*. Melalui *confusion matrix*, diperoleh nilai *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN) yang menjadi dasar dalam perhitungan metrik evaluasi. Berikut ini tabel confusion matrix.

Tabel 1 Confusion Matrix

		Prediksi		
		A	B	C
Ground Truth	A	TP_A	E_{AB}	E_{AC}
	B	E_{BA}	TP_B	E_{BC}
	C	E_{CA}	E_{CB}	TP_C

Sumber: (Muflikhah et al., 2023, p. 114)

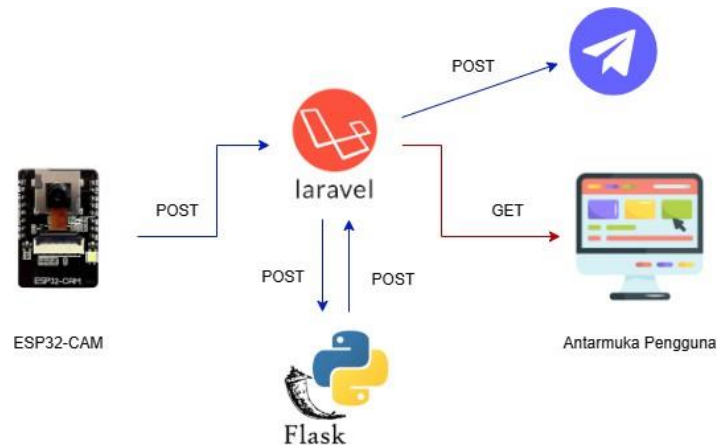
C. HASIL DAN PEMBAHASAN

1. HASIL

a. Analisis Metode Internet of Things

Penerapan metode Internet of Things (IoT) dalam penelitian ini diawali dengan perancangan sistem untuk memantau kondisi barang di rak secara berkala. Berdasarkan hasil analisis kebutuhan, sistem dikembangkan agar mampu mengambil citra rak pada waktu tertentu dan menampilkannya melalui antarmuka web. Pendekatan IoT digunakan untuk menghubungkan ESP32-CAM, server Laravel, server Flask Python, serta sistem deteksi objek YOLOv5 agar seluruh komponen dapat saling berkomunikasi dalam satu jaringan. ESP32-CAM dikonfigurasi untuk mengambil gambar kondisi rak setiap satu menit dan mengirimkannya ke server Laravel menggunakan HTTP POST, kemudian diteruskan ke server Flask

Python untuk diproses oleh model YOLOv5. Hasil deteksi berupa jumlah dan jenis produk dikembalikan dalam format JSON dan ditampilkan pada dashboard web. Jika stok barang kurang dari tiga, sistem mengirimkan notifikasi melalui Telegram. Arsitektur komunikasi sistem ditunjukkan pada Gambar 3, yang memperlihatkan alur pertukaran data antar komponen melalui jaringan Wi-Fi menggunakan protokol HTTP.



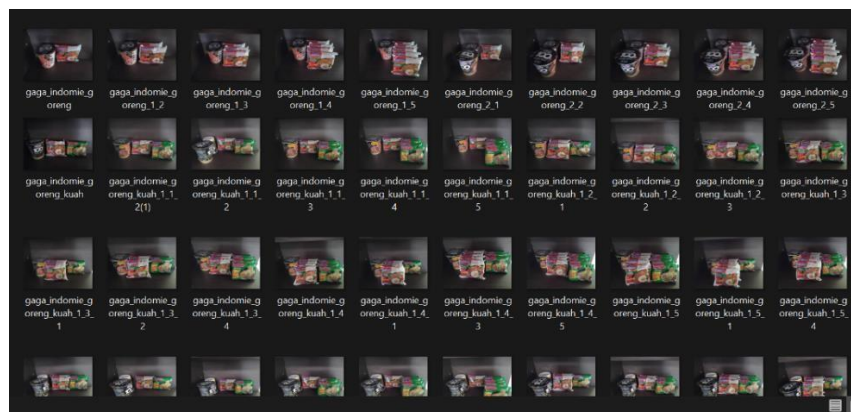
Gambar 3 Alur Komunikasi IoT

b. Hasil Analisis Metode Deteksi Objek Berbasis YOLOv5

Penerapan YOLOv5 dalam penelitian ini dilakukan melalui beberapa tahapan, meliputi pengumpulan data, pelabelan data, persiapan pelatihan model, pelatihan model yolov5, dan hasil deteksi. Setiap tahapan dijelaskan secara berurutan untuk menggambarkan proses penerapan metode deteksi objek pada sistem pemantauan barang.

1) Pengumpulan Data

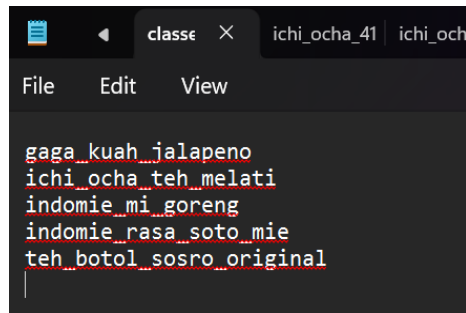
Dataset dikumpulkan dengan mengambil gambar produk menggunakan kamera ponsel, mencakup lima jenis produk, yaitu Indomie Mi Goreng, Indomie Rasa Soto Mie, Ichi Ocha Teh Melati, Teh Botol Sosro, dan Mie Gaga Extra Pedas Kuah Jalapeno. Pengambilan gambar memperhatikan variasi latar belakang serta jumlah produk dalam satu frame, mulai dari satu hingga lima barang, agar model dapat mengenali objek dalam berbagai kondisi. Seluruh gambar kemudian dibagi menjadi data pelatihan dan data validasi dengan proporsi 70% untuk pelatihan dan 30% untuk validasi. Data pelatihan digunakan untuk melatih model YOLOv5 mengenali objek sesuai kategori, sedangkan data validasi berfungsi mengevaluasi kinerja model dan mencegah overfitting. Contoh dataset ditampilkan pada Gambar 4.



Gambar 4 Dataset

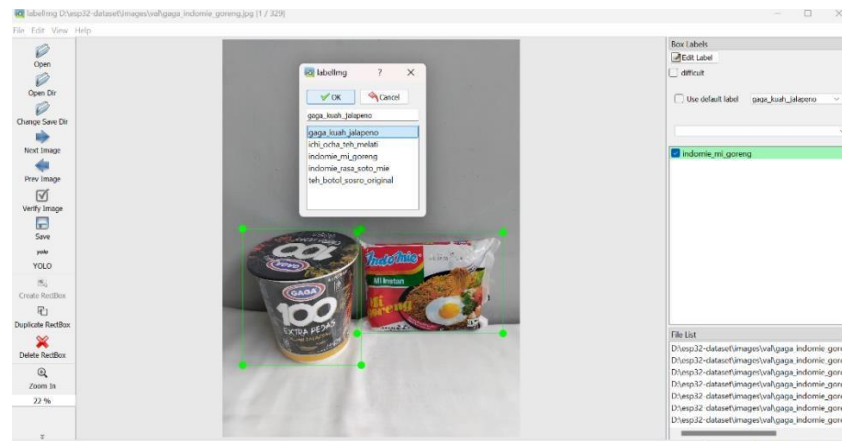
2) Pelabelan Data

Setelah proses pengumpulan dan pembagian dataset, tahap berikutnya adalah pelabelan setiap gambar. Pelabelan dilakukan menggunakan aplikasi LabelImg, yang memungkinkan penentuan posisi objek melalui *bounding box*. Sebelum pelabelan dilakukan, disiapkan file *classes.txt* yang memuat daftar kategori produk sebagai acuan penetapan label, sehingga setiap objek dapat diberi label kelas secara konsisten.



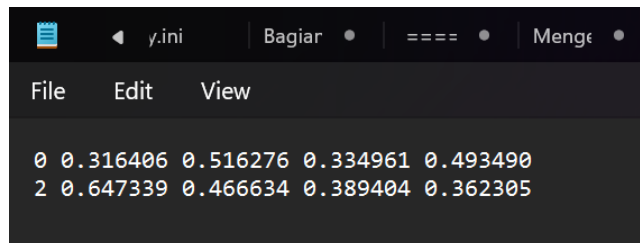
Gambar 5 Isi dari file classes.txt

Selanjutnya, proses pelabelan dilakukan dengan cara memberikan *bounding box* pada objek di dalam gambar, kemudian menetapkan kelas sesuai kategori yang tercantum pada classes.txt.



Gambar 6 Proses Pelabelan Gambar

Hasil dari proses pelabelan menghasilkan file dengan ekstensi .txt yang memuat informasi berupa nomor indeks kelas produk yang merujuk pada file classes.txt, serta koordinat dan ukuran *bounding box*.



Gambar 7 Isi dari file “.txt” dari hasil pelabelan

3) Persiapan Pelatihan Model

Sebelum proses pelatihan dilakukan, diperlukan sejumlah file pendukung yang harus disiapkan terlebih dahulu. File-file tersebut meliputi dataset, konfigurasi, dan daftar kelas yang akan digunakan pada YOLOv5. Daftar lengkap file yang dipersiapkan dapat dilihat pada Tabel 2.

Tabel 2 Daftar folder/file persiapan dataset

No	Nama File / Folder	Keterangan
1	train/	Folder yang berisi kumpulan gambar dataset yang digunakan pada tahap pelatihan (<i>training set</i>).
2	val/	Folder yang berisi kumpulan gambar dataset untuk tahap validasi (<i>validation set</i>).
3	classes.txt	File teks yang berisi daftar nama kategori produk yang digunakan sebagai label acuan.
4	train.txt	File teks yang menyimpan hasil pelabelan gambar pada kelompok pelatihan.

No	Nama File / Folder	Keterangan
5	val.txt	File teks yang menyimpan hasil pelabelan gambar pada kelompok validasi dengan format yang sama seperti <i>train.txt</i> .
6	data.yaml	File konfigurasi utama YOLOv5 yang berisi informasi penting seperti lokasi dataset, jumlah kelas (<i>nc</i>), serta daftar nama kelas (<i>names</i>).
7	yolov5s.pt	File bobot awal (<i>pre-trained weights</i>) yang digunakan sebagai dasar sebelum proses pelatihan dilakukan.

4) Pelatihan Model YOLOv5

Proses pelatihan model YOLOv5 dilakukan setelah seluruh dataset, file label, serta file konfigurasi dipersiapkan. Model dasar YOLOv5s dengan bobot awal (*pre-trained weights*) dari dataset COCO digunakan untuk mempercepat pelatihan dan meningkatkan akurasi. Selanjutnya, dilakukan *fine-tuning* menggunakan dataset penelitian agar model mampu mengenali kategori produk secara spesifik. Proses pelatihan dijalankan dengan perintah sebagai berikut:

```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Version 10.0.22631.5909]
(c) Microsoft Corporation. All rights reserved.

D:\yolov5-master>python train.py --img 640 --batch 4 --epochs
50 --data D:/esp32-dataset/data.yaml --weights yolov5s.pt --
name deteksi_produk_final
```

Gambar 8 Syntax untuk training data

Pelatihan awal dilakukan menggunakan 200 gambar latih untuk memastikan konfigurasi dan parameter berjalan dengan baik. Setiap gambar berukuran 640 piksel, dengan batch sebanyak 4 dan 50 epoch, memanfaatkan bobot awal *yolov5s.pt*. File *data.yaml* berisi pengaturan lokasi dataset dan kategori objek, sedangkan parameter *name* menandai hasil pelatihan. Dokumentasi pelatihan awal ditunjukkan pada Gambar 9.

```
C:\Windows\System32\cmd.e x + v - □ x
10      -1 1 131584 models.common.Conv [512, 256, 1, 1]
11      -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
12      [-1, 6] 1 0 models.common.Concat [1]
13      -1 1 361984 models.common.Conv [512, 256, 1, False]
14      -1 1 33824 models.common.Conv [256, 128, 1, 1]
15      -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
16      [-1, 6] 1 0 models.common.Concat [1]
17      -1 1 90880 models.common.Conv [256, 128, 1, False]
18      -1 1 147712 models.common.Conv [128, 128, 3, 2]
19      [-1, 14] 1 0 models.common.Concat [1]
20      -1 1 296448 models.common.Conv [256, 256, 1, False]
21      -1 1 590336 models.common.Conv [256, 256, 3, 2]
22      [-1, 10] 1 0 models.common.Concat [1]
23      -1 1 1182720 models.common.Conv [512, 512, 1, False]
24      [17, 20, 23] 1 26970 models.yolo.Detect [5, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 48, 39, 19],
[116, 90, 156, 198, 373, 326]], [128, 256, 512]]
Model summary: 214 layers, 7033114 parameters, 7033114 gradients, 16.0 GFLOPs

Transferred 343/349 items from yolov5s.pt
optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias
Scanning D:/esp32-dataset/labels/train...: 0% | 0/200 [00:00<, 71it/s]D:\yolov5-master\utils\general.py:32: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to setuptools<81.
import pkg_resources as pkg
D:\yolov5-master\utils\general.py:32: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to setuptools<81.
import pkg_resources as pkg
D:\yolov5-master\utils\general.py:32: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to setuptools<81.
import pkg_resources as pkg
Scanning D:/esp32-dataset/labels/train...: 200 images, 0 backgrounds, 0 corrupt: 100% | 200/200 [01:00<]
29C
Q search
```

Gambar 9 Proses Pelatihan

Setelah uji coba awal, pelatihan utama dilakukan secara bertahap menggunakan total 1.319 gambar. Tahap pertama melibatkan 550 gambar latih selama 50 epoch, kemudian dilanjutkan *fine-tuning* dengan 769 gambar latih selama 20 epoch. Metode bertahap ini memungkinkan model menyesuaikan bobot awal dan belajar dari variasi data yang lebih beragam, sehingga meningkatkan kemampuan deteksi produk secara optimal.

5) Hasil Deteksi

Setelah pelatihan selesai, model YOLOv5 diuji pada citra uji untuk menilai kemampuan deteksi objek

di rak. Model menggunakan bobot hasil pelatihan dan menghasilkan deteksi berupa *bounding box*, kelas objek, serta tingkat kepercayaan (*confidence score*). Ambang batas kepercayaan ditetapkan sebesar 0,25 untuk menyaring prediksi dengan keyakinan rendah, sedangkan ambang batas IoU sebesar 0,45 digunakan dalam *Non-Maximum Suppression* (NMS) untuk menggabungkan *bounding box* yang saling bertumpuk.

Pengujian dilakukan menggunakan citra hasil tangkapan kamera ESP32-CAM. Model menghasilkan keluaran berupa *bounding box* berwarna yang menunjukkan posisi dan jenis objek

yang dikenali, disertai *confidence score*. Gambar 10 menunjukkan contoh hasil deteksi pada citra *mi_goreng_35.jpg*, di mana model berhasil mengenali produk Indomie Mi Goreng dengan *confidence score* 0,95. *Bounding box* berwarna putih menandai area objek dengan label kelas “*indomie_mi_goreng*”, menunjukkan bahwa model mampu mendeteksi objek dengan akurasi tinggi pada kondisi pencahayaan normal.























Gambar 10 Hasil Model YOLO















2. PEMBAHASAN

Untuk mengevaluasi kinerja sistem secara teknis, langkah pertama dilakukan perbandingan antara data training dengan hasil prediksi sistem. Perbandingan ini bertujuan untuk melihat konsistensi sistem dalam mengenali produk sesuai dengan label yang telah dilatih. Hasil perbandingan ditunjukkan pada Tabel 3.

Tabel 3 Perbandingan data training dan data hasil prediksi

No	Produk	Gambar Training	Label	Gambar Prediksi	Label Prediksi
1	Indomie Mi Goreng		indomie_mi_goreng (1)		indomie_mi_goreng (1)
2	Indomie Mi Goreng		indomie_mi_goreng (2)		indomie_mi_goreng (2)

No	Produk	Gambar Training	Label	Gambar Prediksi	Label Prediksi
3	Indomie Mi Goreng		indomie_mi_goreng (3)		indomie_mi_goreng (3)
4	Indomie Mi Goreng		indomie_mi_goreng (4)		indomie_mi_goreng (4)
5	Indomie Mi Goreng		indomie_mi_goreng (5)		indomie_mi_goreng (5)
6	Indomie Rasa Soto Mie		indomie_rasa_soto_mie (1)		indomie_rasa_soto_mie (1)
7	Indomie Rasa Soto Mie		indomie_rasa_soto_mie (2)		indomie_rasa_soto_mie (2)
8	Indomie Rasa Soto Mie		indomie_rasa_soto_mie (3)		indomie_rasa_soto_mie (3)
9	Indomie Rasa Soto Mie		indomie_rasa_soto_mie (4)		indomie_rasa_soto_mie (2)
10	Indomie Rasa Soto Mie		indomie_rasa_soto_mie (5)		indomie_rasa_soto_mie (5)
11	Mi Gaga Extra Pedas Kuah Jalapeno		gaga_kuah_jalapeno (1)		gaga_kuah_jalapeno (1)

No	Produk	Gambar Training	Label	Gambar Prediksi	Label Prediksi
12	Mi Gaga Extra Pedas Kuah Jalapeno		gaga_kuah_jalapeno (2)		gaga_kuah_jalapeno (2)
13	Teh Botol Sosro		teh_botol_sosro_original (1)		teh_botol_sosro_original (1)
14	Teh Botol Sosro		teh_botol_sosro_original (2)		teh_botol_sosro_original (2)
15	Ichi Ocha Teh Melati		ichi_ocha_teh_melati (1)		ichi_ocha_teh_melati (1)
16	Indomie Mi Goreng, Indomie Rasa Soto Mie				indomie_mi_goreng (4), indomie_rasa_soto_mie (4)
17	Indomie Mi Goreng, Indomie Rasa Soto Mie				indomie_mi_goreng (5), indomie_rasa_soto_mie (5)
18	Indomie Mi Goreng, Indomie Rasa Soto Mie				indomie_rasa_soto_mie (3), indomie_mi_goreng (5)
19	Indomie Mi Goreng, Indomie Rasa Soto Mie				indomie_rasa_soto_mie (2), indomie_mi_goreng (4)
20	Indomie Mi Goreng, Indomie Rasa Soto Mie				indomie_mi_goreng (3), indomie_rasa_soto_mie (2)
21	Mi Gaga Extra Pedas Kuah Jalapeno				gaga_kuah_jalapeno (1)

No	Produk	Gambar Training	Label	Gambar Prediksi	Label Prediksi
22	Mi Gaga Extra Pedas Kuah Jalapeno				gaga_kuah_jalapeno (2)
23	Ichi Ocha Teh Melati				ichi_ocha_teh_melati (1)
24	Ichi Ocha Teh Melati				indomie_rasa_soto_mie (1)
25	Teh Botol Sosro				indomie_rasa_soto_mie (1)
26	Teh Botol Sosro				teh_botol_sosro_original (1)
27	Teh Botol Sosro				teh_botol_sosro_original (1)
28	Teh Botol Sosro, Indomie Rasa Soto Mie				ichi_ocha_teh_melati (1), indomie_rasa_soto_mie (2)
29	Mi Gaga Extra Pedas Kuah Jalapeno				gaga_kuah_jalapeno (1)
30	Indomie Mi Goreng				indomie_mi_goreng (1)

Untuk memperoleh gambaran kinerja sistem secara menyeluruh, dilakukan analisis perbandingan antara label aktual dan hasil prediksi pada seluruh kelas produk. Hasil pengujian ini disajikan dalam bentuk confusion matrix, seperti ditunjukkan pada Tabel 4.

Tabel 4 Hasil Pengujian Confusion Matrix

kelas awal		Prediksi YOLO															
		indomie mi goreng					indomie rasa soto mie					gaga kuah jalapeno		teh sosro		ichi ocha	
		1	2	3	4	5	1	2	3	4	5	1	2	1	2		
indomie mi goreng	1	2															
	2		2														
	3			3													
	4				12												
	5			2		18											
indomie rasa soto mie	1						1										
	2							6									
	3								3								
	4							2		6							
	5							3	2		15						
gaga kuah jalapeno	1											4	1				
	2											1	2				
teh sosro	1						2								3		
	2															2	
ichi ocha							2								1	2	2

Berdasarkan hasil pengujian confusion matrix pada Tabel 4, analisis dilanjutkan untuk mengetahui jumlah objek yang terdeteksi dengan benar (*True Positive*), jumlah prediksi yang salah (*False Positive*), serta jumlah objek yang tidak terdeteksi (*False Negative*) per kelas. Hasil perhitungan disajikan pada Tabel 5.

Tabel 5 Tabel Hasil Perhitungan TP, FP, dan FN

Kelas	TP	FP	FN
Indomie Mi Goreng	37	0	2
Indomie Rasa Soto Mie	31	0	9
Mi Gaga Extra Pedas Kuah Jalapeno	6	1	1
Teh Botol Sosro	5	0	3
Ichi Ocha Teh Melati	2	1	1
Total	81	2	16

Berdasarkan Tabel 5, metrik *Precision*, *Recall*, dan *F1 Score* dihitung untuk masing-masing kelas produk menggunakan nilai *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN). Perhitungan per kelas adalah sebagai berikut:

- 1) Indomie Mi Goreng

Untuk kelas Indomie Mi Goreng, diperoleh TP = 37, FP = 0, dan FN = 2.

Dengan menggunakan rumus:

$$Precision = \frac{37}{37 + 0} = \frac{37}{37} = 1,0$$

$$Recall = \frac{37}{37 + 2} = \frac{37}{39} = 0,949$$

$$F1Score = 2x \frac{1,0x0,949}{1,0 + 0,949} = 0,974$$

- 2) Indomie Rasa Soto Mie

Untuk kelas Indomie Rasa Soto Mie, diperoleh TP = 31, FP = 0, dan FN = 9.

Dengan menggunakan rumus:

$$Precision = \frac{31}{31+0} = \frac{31}{31} = 1,0$$

$$Recall = \frac{31}{31+9} = \frac{31}{40} = 0,775$$

$$F1Score = 2x \frac{1,0x0,775}{1,0+0,775} = 0,873$$

- 3) Mi Gaga Extra Pedas Kuah Jalapeno

Untuk kelas Mi gaga Extra Pedas Kuah Jalapeno, diperoleh TP = 6, FP = 1, FN = 1.

Dengan menggunakan rumus:

$$Precision = \frac{6}{6+0} = \frac{6}{6} = 1,0$$

$$Recall = \frac{6}{6+1} = \frac{6}{7} = 0,857$$

$$F1Score = 2x \frac{1,0x0,857}{1,0+0,857} = 0,857$$

- 4) The Botol Sosro

Untuk kelas Teh Botol Sosro, diperoleh TP = 5, FP = 0, FN = 3.

Dengan menggunakan rumus:

$$Precision = \frac{5}{5+0} = \frac{5}{5} = 1,0$$

$$Recall = \frac{5}{5+3} = \frac{5}{8} = 0,625$$

$$F1Score = 2x \frac{1,0x0,625}{1,0+0,625} = 0,769$$

- 5) Ichi Ocha The Melati

Untuk kelas Ichi Ocha Teh Melati, diperoleh TP = 2, FP = 1, FN = 1.

Dengan menggunakan rumus:

$$Precision = \frac{2}{2+1} = \frac{2}{3} = 0,667$$

$$Recall = \frac{2}{2+1} = \frac{2}{3} = 0,667$$

$$F1Score = 2x \frac{0,667x0,667}{0,667+0,667} = 0,667$$

Berdasarkan perhitungan diatas, akurasi keseluruhan sistem dapat dihitung dengan rumus:

$$Akurasi = \frac{TP_{Total}}{TP_{Total} + FP_{Total} + FN_{Total}} = \frac{81}{81 + 2 + 16} = \frac{81}{99} = 0,818\%$$

Berdasarkan perhitungan *Precision*, *Recall*, dan *F1 Score* untuk masing-masing kelas produk serta akurasi keseluruhan sistem, dapat disimpulkan bahwa sistem pemantauan barang yang dikembangkan memiliki kinerja yang baik. Nilai akurasi sebesar 81.8% menunjukkan bahwa sistem mampu mendeteksi dan mengklasifikasikan sebagian besar produk dengan benar, meskipun masih terdapat beberapa kesalahan prediksi dan objek yang tidak terdeteksi. Hasil ini memberikan gambaran bahwa sistem telah bekerja secara efektif untuk mendukung pemantauan ketersediaan barang.

D. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Sistem pemantauan barang berbasis Internet of Things (IoT) yang dikembangkan menggunakan algoritma You Only Look Once (YOLO) dan perangkat ESP32-CAM mampu mendeteksi kekosongan rak serta menampilkan hasilnya melalui antarmuka web.
2. Hasil pengujian menunjukkan sistem memiliki tingkat akurasi sebesar 81,8%, sehingga sistem dinilai efektif dan layak diterapkan untuk pemantauan stok barang di lingkungan ritel.
3. Penerapan algoritma YOLO terbukti meningkatkan efektivitas proses deteksi kekosongan rak, sehingga membantu pengelola toko dalam memantau ketersediaan barang dengan lebih cepat dan akurat.

E. DAFTAR PUSTAKA

- [1] Butsianto, Herdiyan, Bangsa, Danas, & Pusat, C. (2021). *Sistem inventori berbasis internet of things menggunakan radio frequency identification (studi kasus pt. grafitecindo ciptaprima)*. 13(2), 2298–2314.
- [2] Eko Supriyanto, Rizal Isnanto, Sutrisno Hadi Purnomo & Hanny Nurrani. (2025). *Deteksi Dan Klasifikasi Kesehatan Ayam Menggunakan Yolo Dengan Google Colab*. <https://books.google.co.id/books?id=50RIEQAAQBAJ>
- [3] Moh Muthohir, Sindhu Rakasiwi, L. U. (2023). *Warehouse management system berbasis radio frequency identification*. 3(1).
- [4] Muflikhah, L., Mahmudy, W. F., & Kurnianingtyas, D. (2023). *Machine Learning*. Universitas Brawijaya Press. https://books.google.co.id/books?id=tu_uEAAAQBAJ
- [5] Samanui, I. B., & Prayitno, G. (2024). *Peningkatan Efisiensi Pengelolaan Stok Barang Melalui Aplikasi Berbasis Web di CV Bina Anak Papua*. 2(1).