

Analisis Keandalan YOLOv8m untuk Deteksi Varian Produk Kemasan Kotak pada Sistem Manajemen Ketersediaan *Stock*

Richo

Politeknik Perkapalan Negeri Surabaya, Jalan Teknik Kimia Kampus ITS, Surabaya 60111, Indonesia

e-mail: richo@student.ppns.ac.id

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi : 19 Oktober 2024

Revisi Akhir : 05 November 2024

Diterbitkan *Online* : 30 November 2024

Kata Kunci:

Deteksi, Manajemen, *Stock*, *You Only Look Once* (YOLO), YOLOv8m

Korespondensi :

Telepon / Hp : +62 8233 9739 702

E-mail : richo@student.ppns.ac.id

A B S T R A K

Sistem manajemen ketersediaan *stock* secara otomatis dapat meningkatkan efisiensi waktu ketika proses pendataan produk, mempercepat laju penjualan, efisiensi *budget*, hingga meminimalisir terjadinya tindakan kecurangan pendataan oleh petugas toko. Demi terciptanya tujuan tersebut sudah seharusnya sistem pendataan ketersediaan *stock* dioptimalisasi dengan sistem otomatisasi. Sayangnya, sistem pendataan saat ini masih dilakukan secara manual dengan basis pengecekan produk pada *display shelving* sehingga memicu terjadinya kesalahan pendataan oleh petugas, tentunya akar permasalahan ini harus segera diputuskan sehingga efisiensi manajemen persediaan *stock* dapat terorganisir. Sistem deteksi *real-time* menggunakan metode YOLO telah beberapa kali dilakukan dan terbukti telah berkontribusi positif pada keakuratan hasil deteksi, salah satunya yakni metode YOLOv8. Pada penelitian ini menggunakan komparasi 4 metode YOLOv8 diantaranya yakni YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l, sebagai langkah untuk menciptakan akurasi model terbaik. Produk deteksi yang digunakan pada penelitian ini menggunakan produk "Pocky 70gram Biscuit Sticks" dengan total varian produk sebanyak 8 *class*. Penelitian ini menggunakan 400 data *training* dan 400 data *validation*. Hasil temuan dari penelitian ini yakni metode YOLOv8m memberikan kontribusi akurasi *training* terbaik diantara model lainnya dengan persentase *precision* mencapai 95.45% dan mAP50 sebesar 96.32%. Adanya penyempurnaan pada model YOLOv8m yang telah dilakukan berhasil meningkatkan *precision of accuracy model* sebesar 97.85% dari sebelumnya yang hanya 95.45%. Hasil *persentase average confidence score* pada penelitian ini sebesar 86.08% atas kontribusi pengujian keseluruhan varian produk yang diujikan. Keandalan sistem deteksi ketersediaan *stock* ini menunjukkan bahwa sistem *application* mampu mengenali setiap jenis produk secara tepat dan efektif sehingga *output* pada penelitian ini dapat bermanfaat untuk efisiensi manajemen ketersediaan *stock*.

1. PENDAHULUAN

Pengenalan varian produk kemasan seringkali menjadi tantangan signifikan bagi industri ritel dan distribusi khususnya pada sistem manajemen ketersediaan *stock*, mengingat variasi dan detail kemasan suatu jenis produk yang beragam, termasuk paduan warna dan kompleksitas ornamen desain suatu produk [1]. Saat ini, proses manajemen pendataan ketersediaan *stock* masih dilakukan secara manual, dimana staff *sales* atau petugas minimarket melakukan pengecekan berbasis *check list* terhadap produk yang masih tersedia dengan mencatat secara manual jenis produk yang ada pada *display shelving*. Ketidaklengkapan produk pada *display shelving* menandakan bahwa jenis produk tersebut telah habis terjual, yang mana perlu dilakukan *restock* produk secara cepat dan tepat. Sayangnya sistem manual yang saat ini diterapkan belum dapat memutuskan akar permasalahan tersebut.

Proses pendataan secara manual berpotensi menimbulkan kesalahan pengamatan, yang dapat berdampak negatif terhadap efisiensi laju penjualan produk karena keterlambatan tindakan *re-stock* produk [2]. Padahal kegiatan pendataan ketersediaan *stock* yang akurat dapat mempengaruhi *omzet* penjualan,

meminimalisir terjadinya keterlambatan pergantian produk, efisiensi *budget* toko, mempercepat laju penjualan, hingga meminimalisir terjadinya tindakan kecurangan pendataan oleh petugas toko.

Menyoroti solusi yang pernah ada, sistem deteksi produk yang dilakukan masih bersifat *uni-class*, dimana sistem hanya dapat mengenali tidak lebih dari *dua class* yang ditetapkan dalam tingkatan item produk, sehingga sistem aplikasi kurang efektif jika diterapkan pada jenis produk yang tergolong kompleks dan beragam [3]. Adanya permasalahan yang telah dibahas sebelumnya, peneliti menetapkan parameter deteksi jenis produk berdasarkan keberagaman detail desain pada setiap jenis produk berbasis aplikasi deteksi *real-time* menggunakan metode YOLO (*You Only Look Once*) sebagai *logic* utama dalam menentukan hasil keputusan deteksi [4].

Penelitian sejenis terkait implementasi metode YOLO untuk sistem deteksi objek telah beberapa kali dilakukan, salah satunya penelitian yang telah dilakukan oleh (Maulana, dkk), penelitian tersebut mengimplementasikan metode YOLOv5 untuk deteksi dua jenis produk hasil proses manufaktur yang bersifat *uni-class* menggunakan sistem Tesla T4 16 GB dengan kontribusi akurasi mencapai 97.7% [5]. penelitian serupa lainnya yang dilakukan oleh (Agnes & Lina, 2023)

dengan melakukan pengembangan sistem pengenalan produk yang terfokus pada *non-item* produk artinya skala pengenalan dilakukan pada produk yang bersifat non-terjal dengan total data *training* mencapai 400 data dengan akurasi deteksi mencapai 75% [6]. Penelitian produk dengan metode YOLO juga dilakukan oleh (Yuning, dkk) dengan penelitian yang terfokus pada monitoring produk berbasis *multi scale* dengan arsitektur metode yang beragam diantaranya yakni YOLOv3, YOLOv4, Faster R-CNN Inception V2 dan lain sebagainya, pada penelitian ini didapatkan temuan bahwa metode YOLOv4 menghasilkan *positive decision rate* paling tinggi yang mencapai 97.68% [7]. Adanya pembuktian tersebut, membuktikan bahwa metode YOLO memberikan kontribusi positif untuk sistem deteksi yang akurat.

Namun, meskipun pencapaian sebelumnya telah memberikan kontribusi positif, masih terdapat beberapa tantangan dalam menciptakan inovasi baru sebagai bentuk pengembangan dan rekayasa teknologi *Artificial Intelligence*, serta menciptakan solusi berlandaskan pada permasalahan yang saat ini sedang membutuhkan solusi perubahan khususnya pada sistem manajemen kesediaan *stock*.

Metode YOLO telah banyak dilakukan pengembangan model, salah satu versi yang saat ini sedang naik daun adalah metode YOLOv8 atau (*You Only Look Once Version 8*) [8]. Metode YOLOv8 memiliki keakuratan deteksi dalam hal klasifikasi deteksi objek karena sifat konfigurasi model yang *independent*, kompleks, multiparameter *build*, dan rekayasa arsitektur yang fleksibel membuat model ini terus berkembang pada berbagai objek penelitian deteksi dengan sifat objek yang relatif terjal dan beragam [8].

Sebagai langkah untuk mendukung pengembangan berkelanjutan, peneliti melakukan studi penelitian yang berfokus pada pengembangan sistem untuk mengenali berbagai jenis varian produk kemasan kotak. Objek deteksi yang dimaksud merupakan produk “Pocky 70gram *Biscuit Sticks*” dengan keberagaman jenis varian yang ada. Penelitian ini bertujuan untuk mengeksplorasi berbagai arsitektur model YOLOv8 guna memperoleh keputusan deteksi yang optimal.

Penelitian ini dirancang dengan total keseluruhan data yang mencapai 400 data *training* dan 400 data *validation* dengan total *pixel* gambar yang beragam dan dinormalisasikan dalam bentuk *scale* yang seragam yakni 300x300 *pixel*. Pendekatan ini bertujuan untuk memperkaya pola *dataset* dan memastikan representasi yang konsisten dalam pelatihan model deteksi.

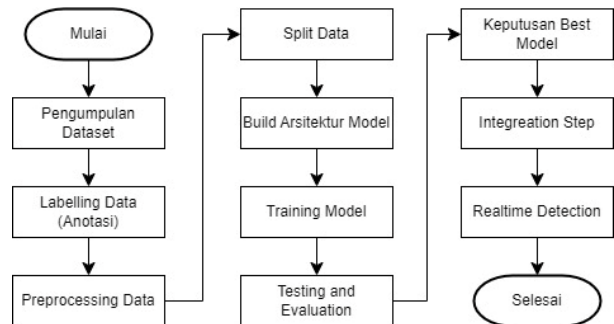
Pada implementasinya, arsitektur model YOLOv8 diujikan dengan membandingkan berbagai arsitektur, yaitu YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l. Kompilasi arsitektur model dalam penelitian ini bertujuan untuk mengevaluasi dan mengidentifikasi arsitektur mana yang memberikan kontribusi terbaik dalam hal performa deteksi, serta mencapai target akurasi yang paling tinggi dalam rekayasa model YOLOv8. Implementasi sistem otomatisasi ini diharapkan dapat mengurangi frekuensi kesalahan

pengecekan (*human error*), mempercepat proses verifikasi kesediaan *stock*, serta meningkatkan efektivitas manajemen persediaan secara keseluruhan. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan terhadap inovasi terbaru dalam praktik nyata yang terfokus pada sistem manajemen kesediaan *stock* produk. Sehingga diharapkan dapat meningkatkan efisiensi proses *restock* produk serta memastikan sistem manajemen penjualan yang unggul dan lebih terstruktur.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini secara sistematis menjabarkan tahap prosedural yang dilakukan dalam proses prediksi kesediaan *stock* dengan mengimplementasikan metode YOLOv8. Tahapan penelitian ini diuraikan dalam Gambar 1 berikut ini.



Gambar 1. Alur Sistematis Penelitian

Tahapan awal dimulai dari pengumpulan *dataset*, objek penelitian ini berfokus pada produk kemasan kotak “Pocky 70 gram *Biscuit Sticks*” dengan beragam jenis varian yang mencapai 8 varian produk sebagai total *class* klasifikasi. Adapun visualisasi *dataset* yang digunakan pada penelitian ini sebagaimana yang tertera pada Gambar 2 berikut ini:



Gambar 2. Teknik Pengambilan *Dataset*

```

p_chocolate
p_strawberry
p_cookies
p_double_choco
p_matcha
p_almond
p_dark_choco
p_milk_choco
    
```

Gambar 3. *Class Labelling* yang digunakan

Berdasarkan Gambar 2 diatas, teknik pengambilan *dataset* dilakukan dengan memperhatikan variansi dalam skala *pixel* gambar, jarak, tingkat pencahayaan, dan posisi pengambilan yang beragam. Pendekatan ini bertujuan untuk memperkaya pola *dataset* sehingga memberikan indikasi peluang deteksi yang lebih komprehensif dan kompleks. Data yang dikumpulkan merupakan citra RGB dengan total data *training* sebanyak 200 data dan *validation* sejumlah 200 data. Selanjutnya *dataset* yang telah dikumpulkan akan dinormalisasi dengan ukuran *pixel* yang seragam yakni dengan *scale* 300x300 *pixel*.

Labelling data dilakukan pada setiap data gambar yang telah dinormalisasi sebelumnya. *Labelling* data bertujuan untuk tahapan pengenalan parsial yang mengindikasikan deklarasi pengenalan produk, *output* pada langkah ini dilakukan *ekspor* data *labelling* dalam format YOLO dengan kompleksitas data berformat “txt”. Teknik *labelling* pada penelitian ini disajikan sebagaimana pada Gambar 4 berikut.



Gambar 4. Teknik *Labelling* Data

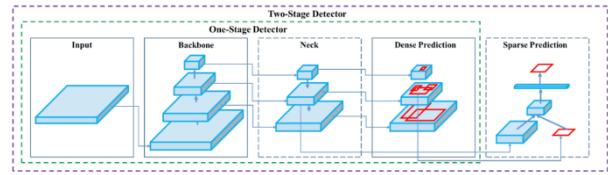
Selanjutnya dilakukan *preprocessing* data seperti *flip*, *rotation*, *brightness adjustment* dan berbagai parameter lainnya [9]. *Split* data dilakukan untuk membagi data berbasis *random state* dengan pembagian antara data *training* dan data *testing*.

Tahapan yang tak kalah penting yakni *build* arsitektur YOLOv8, komparasi arsitektur pada penelitian ini terdiri dari YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l. Setelah dilakukannya *build* parameter tahapan selanjutnya yakni *training* model yang bertujuan untuk melatih model dalam pengenalan data *training*. Model yang telah didapatkan pada proses *training* kemudian dilakukan *testing* model untuk mengevaluasi performa model dengan kompleksitas pengenalan yang akurat. Model dengan tingkat akurasi terbaik dan hasil pengenalan yang paling akurat ditetapkan sebagai model arsitektur yang paling baik. Selanjutnya model arsitektur yang terpilih akan menjadi penentu hasil keputusan deteksi dalam implementasi *real-time*.

2.2 YOLOv8

Metode *You Only Look Once version 8* atau (YOLOv8) merupakan rumpun ilmu YOLO yang memanfaatkan 4 komponen dasar yakni *Backbone*, *Neck*, *Dense Prediction*, dan *Sparse Prediction* [10]. Struktur

kompleks metode YOLOv8 disusun dengan *comperative* seperti pada Gambar 5 berikut.



Gambar 5. Arsitektur Utama YOLOv8 [11]

Pada mulanya *input* citra akan diproses ke tahap *backbone* sebagai lapisan yang berperan dalam mengekstraksi fitur *input* melalui beberapa lapisan *convolutional* dengan menerapkan pendekatan CSPDarknet53 sebagai CSPDarknet backbone [12]. Proses *convolutional* ini menerapkan persamaan berikut:

$$\text{Conv}(Xq + Wq) = \partial (Xq * Wq + c) \quad (1)$$

Dimana Xq sebagai *input* parameter, Wq adalah filter, c adalah bias, sedangkan ∂ adalah *activation function non-linier*. Setelah melalui proses *backbone*, *input* data akan diteruskan ke tahap *Neck* atau *Pyramid Feature Network* yang bertujuan untuk menskalasi fitur pada berbagai sudut pandang resolusi [13]. Selanjutnya yakni tahapan *Dense Prediction* dimana *bounding box* dapat diprediksi secara otomatis melalui pendekatan *anchor free boxes* dan *keypoint-based regression* yakni prediktif titik koordinat pusat terhadap dimensional objek, dengan formulasi sebagai berikut:

$$(x_i, y_i, w_i, h_i) = \text{MLP}_{reg}(\text{fitur}) \quad (2)$$

$$\text{Loss} = -\frac{1}{K} \sum_{i=0}^K [y_i \log p_i + (1-y_i) \log (1-p_i)] \quad (3)$$

Berdasarkan persamaan 2 tersebut, x dan y mencerminkan *center point bounding box*, sedangkan w dan h merupakan lebar dan tinggi *bounding box*. Sedangkan persamaan 3 merupakan *objective loss* dengan *basis binary cross-entropy*. Pada label y memberikan indikasi label objek (bernilai 1 pada aktual objek dan 0 mengindikasikan non objek), parameter p_i adalah *range probability of detect prediction* [14].

Metode YOLOv8 merupakan rumpun keluarga YOLO yang sedang ramai diterapkan pada implementasi *detection realtime*. Penelitian ini menggunakan kontribusi metode YOLOv8 dengan komparasi arsitektur yang dikembangkan pada penelitian ini yakni diantara lain YOLOv8n (*version 8 nano*), YOLOv8s (*version 8 small*), YOLOv8m (*version 8 medium*), dan YOLOv8l (*version 8 large*).

Adanya komparasi tersebut ditujukan untuk memperoleh model deteksi terbaik dan paling konsisten dalam pengenalan produk, serta memastikan efisiensi implementasi model secara komprehensif yang ditinjau berdasarkan akurasi, hasil deteksi, serta FPS selama proses deteksi berlangsung. Setiap model YOLO version 8 memiliki keunikan arsitektur yang beragam sehingga

pola *layer* arsitektur dan hasil deteksi yang dihasilkan bergantung pada model yang dihasilkan pada proses *training* data. Perbedaan konfigurasi arsitektur yang digunakan dapat dilihat pada Tabel 1 berikut.

Tabel 1. Perbedaan Struktural Arsitektur YOLOv8

Parameter	YOLOv8n (Nano)	YOLOv8s (Small)	YOLOv8m (Medium)	YOLOv8l (Large)
Ukuran Model	± 3.25M	± 11.3 M	± 26.1 M	± 47.2 M
Backbone Architecture	CSPDarknet dengan porsi convolutional tidak lebih dari 2	CSPDarknet dengan jumlah convolutional sebanyak 3 hingga 4 layer	CSPDarknet dengan jumlah convolutional yang kompleks serta penambahan CSP blok multilateral	CSPDarknet dengan jumlah convolutional expend serta penambahan CSP excellent
Jumlah Lapisan	150 hingga 200 layer	± 300 layer	± 400 layer	± 500 layer
Dimensional Fitur	Ekstraksi fitur sederhana	Kompromi antara ringan dan presisi	Fitur kompleks dan detail	Resolusi tinggi untuk fitur
Lapisan Unik	layer unsample yang tidak bersentuhan dengan layer shortcut kernel sehingga ekstraksi lebih cepat	konfigurasi upsample yang berkontribusi langsung pada setiap layer sehingga proses inferensi relatif lebih lama	adanya layer atrous convolutional dan efisiensi lapisan upsample yang hanya dilakukan 1 kali yakni setelah preprocess model	(Menggunakan lebih dalam CSP excellent dan PAN)
FPN dan Path Aggregation Network (PAN)	Feature Pyramid Network (FPN) + PAN dengan sedikit layer	FPN + PAN sedikit lebih kompleks	FPN + PAN dengan beberapa layer tambahan	Feature Pyramid Network (FPN) + PAN paling kompleks
Ukuran File	Paling Kecil	Kecil	Menengah	Paling besar

Sumber: [15]

Adanya perbedaan keunikan *layer* pada setiap arsitektur, penelitian ini melakukan komparasi dari metode yang telah dibandingkan untuk mendapatkan model terbaik yang akan diimplementasikan sebagai penentu hasil keputusan deteksi pada sistem deteksi *realtime*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Training Model

Tahapan *training* dilakukan dengan konfigurasi nilai *parametric* yang sama untuk setiap arsitektur yang diimplementasikan, *ratio training* yang diujikan pada pembobotan *training* dan *validation* dengan perbandingan sebesar 60% dan 40%, data *testing* tidak digunakan pada proses *training* model sehingga tidak mendapatkan alokasi *rasio* dalam proses ini. *Input* data *training* menggunakan *scale shape* sebesar 300x300 *pixel* dan total *epoch* sejumlah 300 *epoch* untuk setiap arsitektur model, sedangkan *leaning rate* yang digunakan 0.01 dengan *patience* sebesar 150. Berdasarkan hasil *training* yang dilakukan pada keseluruhan arsitektur, didapatkan hasil yang disajikan pada Tabel 2 berikut.

Tabel 2. Akumulasi Training Parameter Metode YOLOv8

	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l
Inferensi (second/ iter)	14.87 s/it	17.44 s/it	18.23 s/it	27.48 s/it
Last Instance	517	623	654	659
Precision	84.56 %	89.23 %	95.45 %	92.33 %
Recall	83.71 %	87.15 %	94.88 %	90.22 %
mAP50	84.78 %	89.75 %	96.32 %	92.24 %
mAP50-95	73.25 %	82.31 %	91.17 %	90.89 %

Berdasarkan hasil *training* pada Tabel 2 diatas, diperoleh bahwa setiap model arsitektur memiliki

inferensi yang beragam. Model YOLOv8l cenderung memiliki inferensi *training* yang lebih lama, hal ini disebabkan karena kompleksitas arsitektur yang kompleks. Jika ditinjau berdasarkan parameter *futurestic accuracy (precision, recall, mAP50, mAP50-95)* diperoleh bahwa arsitektur model YOLOv8m memberikan kontribusi nilai yang paling tinggi dibanding dengan model arsitektur yang lainnya dengan persentase *precision* mencapai 95.45% dan mAP50 sebesar 96.32%. Namun, tingginya nilai akurasi model YOLOv8m tidak selalu mencerminkan bahwa model telah sempurna, sebagai tahapan lanjutan penulis akan membandingkan nilai *training* dengan akurasi *testing* yang akan dilakukan pengujian lanjutan.

3.2 Hasil Testing Model

Tahapan *testing* bertujuan untuk pengujian dan mengevaluasi akurasi model dengan membandingkan hasil prediksi dan aktualnya. Proses ini bertujuan sebagai referensi untuk mengevaluasi performa model yang diperoleh selama tahap pelatihan. Tahap *testing* pada penelitian ini dilakukan dengan meng-*input sample* data *testing* yang ada untuk dilakukan trial prediksi sebelum arsitektur model diterapkan secara nyata dalam *fase deployment*. Adapun hasil *testing* model yang disajikan pada Tabel 3 berikut.

Tabel 3. Akumulasi Testing Parameter Metode YOLOv8

	YOLO v8n	YOLO v8s	YOLO v8m	YOLO v8l
Total sample	15	15	15	15
Jumlah Varian Benar	101	107	111	110
Jumlah Varian Salah	19	13	9	10
Persentase Akurasi	81.18 %	87.85 %	91.89 %	90.91 %

Berdasarkan hasil akumulasi *testing* berbagai model arsitektur YOLOv8 pada Tabel 3, diperoleh bahwa model dengan arsitektur YOLOv8m memberikan kontribusi akurasi yang paling optimal dibandingkan dengan arsitektur lainnya dengan total data terdeteksi benar sebanyak 111 dari 120 varian produk dengan capaian akurasi sebesar 91.89%. Hal ini selaras dengan *persentase* akurasi pada tahap *training* yang menyatakan bahwa arsitektur dengan model YOLOv8m memiliki akurasi terbaik. Keputusan ini memberikan pandangan bagi peneliti untuk melanjutkan pengembangan rekayasa arsitektur model YOLOv8m. Dengan adanya *insight* ini, membukakan peluang bagi peneliti untuk menciptakan sistem deteksi optimal dibuktikan dengan kontribusi akurasi model YOLOv8m yang sangat baik. Dengan ini memberikan keputusan final bahwa metode YOLOv8m diidentifikasi sebagai model terbaik untuk mengatasi problematika deteksi pada penelitian ini, sehingga dapat menjadi pilihan utama dalam pengembangan arsitektur

yang lebih kompleks dengan akurasi dan efisiensi yang efektif.

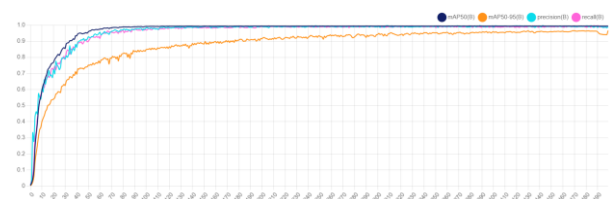
3.3 Tahap Penyempurnaan Best Model YOLOv8m dan Optimalisasi Model

Setelah dilakukan perbandingan dengan berbagai arsitektur YOLOv8 didapatkan temuan baru yakni model terbaik dalam hal deteksi jenis varian produk kemasan yakni dengan model YOLOv8m dengan akurasi *training* dan *testing* yang sangat baik yakni sebesar 93.05% dan 91.89%. Hal ini memicu penulis untuk melakukan pengembangan, penyempurnaan, dan optimalisasi arsitektur model YOLOv8m guna meningkatkan akurasi dan mempertajam keputusan hasil deteksi dengan optimal. Tahapan ini dilakukan penambahan *epoch training* sebanyak 500 *epochs* yang sebelumnya hanya 300 *epochs*. Tabel 4 berikut ini merupakan arsitektur YOLOv8m yang telah dikembangkan oleh peneliti.

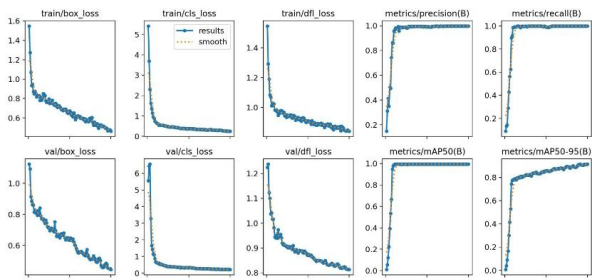
Tabel 4. Arsitektur Penyempurnaan dan Pengembangan Model YOLOv8m

Layer Construction	Output Size	Count Layer	Detail Construction
Input Data	300x300	1	-
Convolutional	150x150x128	4	Detail filter of number → 64, 64, 128, Kernel layer size → 4,4,1 dan stride = 3,2,2
Max Pooling	75x75x128	3	Kernel size = 3x3 dengan stride = 2
Residual Block 1	Convolutional	5	Filter → 128, Kernel layer → 1 and stride layer = 1
	Convolutional		Filter → 128, Kernel layer → 1 and stride = 1 or 2
	Convolutional		Filter → 128, Kernel layer → 1 dan stride = 1 or 2
	Average Pooling str		Kernel layer → 2x2, stride → 2
Residual Block 2	Convolutional str	7	Filter → 128, Kernel layer → 1 dan stride final = 1
	Convolutional		Filter → 128, Kernel layer → 1 dan stride reduce = 1
	Convolutional		Filter → 128, Kernel layer → 1 dan stride reduce = 1 or 2
	Average Pooling str		Filter → 128, Kernel layer → 1 dan stride final = 1 or 2
Residual Block 3	Convolutional	3	Kernel layer → 2x2, stride final = 2
	Convolutional		Filter → 256, Kernel layer = 1 dan stride = 1
	Convolutional		Filter → 256, Kernel layer = 1 dan stride = 1 or 2
	Average Pooling str		Filter → 256, Kernel layer = 1 dan stride = 1 or 2
FPN Branch1	Convolutional	5	Kernel layer → 2x2, stride = 2
	Convolutional		Filter → 256 Kernel layer → 1 dan stride = 1
	Convolutional		Filter → 256, Kernel layer = 1 dan stride = 1
	Convolutional str		Filter → 128, 256, 128, 256, Kernel layer → 1,3,1,3 dan stride final = 1,1,2,1
YOLO head 1	Convolutional	3	Filter → 256, 512, 256, 512, Kernel layer → 1,3,1,3 dan stride final = 1,1,2,1
FPN Branch2	Convolutional	5	Filter → 256, 512, 256, 512 Kernel layer → 1,3,1,3 dan stride final = 1,1,2,1
YOLO head 2	Convolutional	3	Filter → 128, 256, 128, 256, Kernel layer → 1,3,1,3 dan stride final = 1,1,2,1
Unique Layer (Atrous Convolutional)	Convolutional	5	Filter → 64, Kernel layer → 1 dan stride final = 1
	Convolutional		Filter → 64, Kernel layer → 1 dan stride final = 1,2
	Convolutional		Filter → 64, Kernel layer → 1 dan stride final = 1,2
	Max Pooling str		Kernel layer = 2x2, stride result final= 2
Up sampling layer	Halve Channels Layer	3	Kernel layer = 1, stride result final= 1
	Up Sampling		Range of scale = 1

Menggunakan rekayasa penyempurnaan arsitektur model YOLOv8m yang telah dikembangkan oleh peneliti menghasilkan konfigurasi *training* yang jauh lebih baik dengan pola grafik yang cenderung konstan, sebagaimana yang telah disajikan pada Gambar 6 berikut.

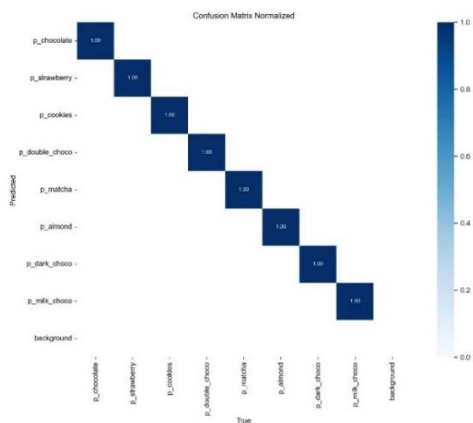


Gambar 6. Arsitektur Utama YOLOv8

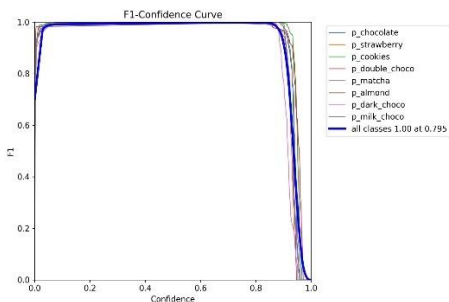


Gambar 7. Arsitektur Utama YOLOv8

Berdasarkan hasil visual pada Gambar 7, hasil *loss* dan *accuracy* dengan pembaruan arsitektur model YOLOv8m menghasilkan konfigurasi yang positif. Konstruksi grafik *loss*, *recall*, *precision*, *mAP50*, *mAP50-95* pada setiap *epoch* pelatihan yang sangat baik, sehingga dapat ditarik kesimpulan bahwa grafik tersebut tidak terjadi indikasi *underfitting* (bebas dari *point of loss prediction*) karena kecenderungan menurun pada grafik *loss* dan mengalami peningkatan *smooth* pada grafik akurasi. Adapun grafik *headmap correlation matrix* dan *confidence score* yang disajikan pada Gambar 8 dan Gambar 9 berikut.



Gambar 8. Nilai Heatmap Confusion Matrix Normalized



Gambar 9. Confidence Score setiap Class Predict

Konfigurasi Gambar 8 didapatkan grafik *confusion matrix* dengan *confidence score* setiap *class* memiliki nilai yang sempurna yakni sebesar 1.0 atau 100%. Bergeser pada Gambar 9 menunjukkan performa *training* pada setiap *epochs*, berdasarkan pola grafik tersebut memiliki kecenderungan yang sangat baik dengan performa meningkat dan stabil artinya model mampu mempertahankan akurasi terbaik pada setiap

epochs sehingga akurasi dapat meningkat dibanding ketika sebelum dilakukan pemodelan arsitektur model YOLOv8m. Berdasarkan penyempurnaan dan pengembangan arsitektur model YOLOv8m diperoleh perubahan nilai akurasi *training* sebelum dan sesudah rekayasa arsitektur YOLOv8m yang disajikan pada Tabel 5 berikut.

Tabel 5. Akumulasi Training Parameter Metode YOLOv8

	Sebelum Penyempurnaan Arsitektur	Setelah Penyempurnaan Arsitektur
Precision	95.45 %	97.82 %
Recall	94.88 %	95.86 %
mAP50	96.32 %	97.98 %
mAP50-95	93.05 %	94.72 %

Konfigurasi yang ditunjukkan pada Tabel 5 menunjukkan bahwa penyempurnaan rekayasa arsitektur model YOLOv8m memberikan kontribusi positif terhadap performa *training*, didapatkan bahwa penyempurnaan model arsitektur YOLOv8m yang dikembangkan peneliti memberikan peningkatan pada 4 aspek *training* dengan nilai yang lebih besar dari arsitektur sebelumnya. Dengan kenaikan tertinggi pada persentase *precision* dari 95.45% menjadi 97.82% atau meningkat sebesar 2,37%. Tahapan selanjutnya dilakukan proses *testing* model YOLOv8m yang telah dilakukan penyempurnaan dan pengembangan arsitektur model.

3.4 Testing Best Model YOLOv8m

Tahapan *testing* dilakukan sebelum model diterapkan pada sistem *deployment*, langkah ini memastikan bahwa model yang dipersiapkan memiliki akurasi dan performa yang baik. *Testing* model dilakukan dengan melakukan *upload* gambar pada sistem untuk menguji keandalan model terhadap ketepatan keputusan deteksi. Pada tahapan ini hanya dilakukan pengujian pada model YOLOv8m yang telah dikembangkan dan disempurnakan oleh peneliti. Pengujian ini menggunakan 16 *sample* data *testing* yang diambil secara acak untuk melatih prediksi pada setiap jenis varian produk. Gambar 10 berikut merupakan *testing image* yang telah dilakukan oleh peneliti.



Gambar 10. Testing Best Model YOLOv8m

Berdasarkan hasil pengujian yang ditampilkan pada Gambar 10, data *testing* yang terdiri dari 16 data *image* menunjukkan tingkat akurasi deteksi yang sangat baik untuk berbagai varian produk yang terdapat pada setiap komponen 16 gambar yang diujikan. Keberhasilan ini menunjukkan bahwa model YOLOv8m memenuhi kriteria yang ditetapkan dan akan dilanjutkan pada tahap implementasi deteksi *real-time* berbasis aplikasi.

3.5 Implementasi Application Detection Real-Time

Tahapan puncak dari penelitian ini yakni melibatkan *integrasi best model YOLOv8m* pada perangkat GPU untuk mengevaluasi hasil prediksi yang berbeda-beda bergantung pada kondisi lingkungan yang diujikan. Hasil evaluasi menunjukkan variasi dalam akurasi deteksi yang dipengaruhi oleh kondisi lingkungan sekitar, seperti tingkat pencahayaan dan jarak pengambilan objek. Sistem deteksi *real-time* yang dihasilkan dapat dilihat pada Gambar 11 berikut.



Gambar 11. Integrasi Deteksi Real-Time

Berdasarkan hasil pengujian *real-time*, sistem pengujian menggunakan perangkat OPPO A5 2020 V11.1 memiliki rata-rata inferensi waktu deteksi mencapai 230 ms dan *frame per second* sebesar 3.3 FPS, selain itu didapatkan rata-rata akurasi setiap jenis varian produk seperti yang disajikan pada Tabel 6 berikut. Nilai akurasi ini mencerminkan seberapa jauh sistem mampu mengenali setiap jenis produk dengan baik.

Tabel 6. Average Confidence Score Persentase Setiap Jenis Produk

Varian Produk	Label	Average Confidence Score (%)
Pocky Chocolate	p_chocolate	82.1
Pocky Strawberry	p_strawberry	72.3
Pocky Cookies and Cream	p_cookies	87.6
Pocky Double Choco	p_double_choco	86.9
Pocky Matcha	p_matcha	82.6
Pocky Almond	p_almond	89.8
Pocky Dark Choco	p_dark_choco	94.5
Pocky Milk Choco	p_milk_choco	92.9
Akumulasi Average Confidence Score		86,08 %

Meninjau data yang didapatkan pada Tabel 6, ditemukan nilai akurasi paling tinggi adalah produk Pocky Dark Choco dengan nilai *confidence score of accuracy* mencapai angka sebesar 94.5%. Sehingga didapatkan akurasi pengujian *real-time* dengan kontribusi yang cukup baik yakni sebesar 86,08% dari rata-rata akurasi ketepatan deteksi pada keseluruhan jenis produk.

4. KESIMPULAN

Berdasarkan analyzing data dan pembahasan yang telah dilakukan, sistem deteksi ini secara umum memberikan wawasan tentang klasifikasi multiclass yang dirancang untuk meningkatkan efisiensi manajemen ketersediaan *stock* produk kemasan kotak, dengan total 8 *class*. Keragaman varian metode YOLOv8 yang dikembangkan peneliti mampu mempelajari setiap pola dan karakteristik jenis produk dengan sangat baik. Komparasi metode telah dilakukan dengan membandingkan performa *training* dan *testing* pada arsitektur YOLOv8n, YOLOv8s, YOLOv8m, dan YOLOv8l. Model dengan akurasi paling tinggi yakni menggunakan arsitektur YOLOv8m dengan akurasi *training* dan *testing* masing-masing sebesar 93.05% dan 91.89%.

Sebagai langkah keberlanjutan, peneliti melakukan sistem penyempurnaan dan pengembangan rekonstruksi parameter model YOLOv8m pada *atrous convolutional layer*, *unique contest layer*, *Feature Pyramid Network (FPN)*, dan *Path Aggregation Network (PAN)* disertai peningkatan nilai *epochs* yang bertujuan untuk meningkatkan akurasi model. Penelitian ini menemukan temuan baru bahwa adanya langkah penyempurnaan dan pengembangan rekonstruksi parameter model YOLOv8m memberikan peningkatan performa akurasi dari 95.45% menjadi 97.82% dengan peningkatan sebesar 2.37%, sehingga ditetapkan YOLOv8m sebagai *best model* pada penelitian ini. Selain itu, penelitian ini dilakukan sistem integrasi model berbasis application pada perangkat OPPO A5 2020 V11.1 untuk sistem deteksi *real-time* dengan rata-rata inferensi waktu deteksi mencapai 230 ms dan *frame per second* sebesar 3.3 FPS dengan performa *average confidence score* deteksi *real-time* pada keseluruhan jenis produk mencapai 86,08%.

Keandalan sistem deteksi kesediaan *stock* ini menunjukkan bahwa sistem *application* mampu mengenali setiap jenis produk secara tepat dan efektif serta mampu memberikan performa akurasi *confidence score* yang sangat baik. Melalui studi kelayakan yang kompatibel ini, diharapkan adanya potensi penggunaan dan pengembangan aplikasi dengan efektivitas yang tinggi, sehingga *output* penelitian ini dapat berkontribusi positif dalam mewujudkan efisiensi sistem aplikasi manajemen kesediaan *stock* pada proses deteksi persediaan produk dalam skala industri yang meluas.

DAFTAR PUSTAKA

- [1] R. Putra and Yoannita, "Perangkat Lunak Pengenalan Jenis Roti Pada Toko Roti Dengan Menggunakan Model Deteksi Objek YOLOv8," MDP Student Conf. 2024, vol. 3, no. 1, pp. 973–980, 2024.
- [2] M. Basirun, M. Prodi, S. Informasi, P. S. Komputer, and P. S. Informasi, "PENERAPAN SCM DALAM UPAYA PENGELOLAAN STOCK BAHAN BAKU PRODUKSI TAHU PADA PABRIK TAHU HELEN PENDAHULUAN Industri rumah tangga , yaitu industri yang menggunakan tenaga kerja kurang dari empat orang . Ciri industri ini memiliki modal yang sangat terbatas , ," vol. 2, no. 3, pp. 211–218, 2022.
- [3] R. Richo, R. Y. Adhitya, M. K. Hasin, M. Syai'in, and E. Setiawan, "Eksplorasi Keandalan Sistem Sortir dan Klasifikasi Kecacatan Perkat Kemasan Menggunakan Arsitektur UNet-Inception Convolutional Neural Network," J. Elektron. dan Otomasi Ind., vol. 10, no. 3, pp. 321–333, 2023, doi: 10.33795/elkolind.v10i3.3835.
- [4] R. Rizal, "Perancangan Early Warning System Untuk Mendukung Sistem Persediaan Barang Dagang," Pros. Semin. Nas. SISFOTEK, vol. 9, no. September, pp. 228–236, 2018, [Online]. Available: <http://seminar.iaii.or.id/index.php/SISFOTEK/article/view/83>
- [5] A. Maulana, M. Suherman, A. F. N. Masruriyah, and H. Y. Novita, "PENERAPAN ALGORITMA CNN MENGGUNAKAN FRAMEWORK YOLO UNTUK DETEKSI OBJEK PRODUK DI PERUSAHAAN MANUFAKTUR," INTI NUSA MANDIRI, vol. 18, no. 2, pp. 107–114, 2024.
- [6] A. Carolina and L. Lina, "Sistem Penjualan Dengan Pengenalan Produk Secara Otomatis Menggunakan Metode Yolo," Simtek J. Sist. Inf. dan Tek. Komput., vol. 8, no. 2, pp. 271–275, 2023, doi: 10.51876/simtek.v8i2.224.
- [7] Y. Li, H. Huang, Q. Chen, Q. Fan, and H. Quan, "Research on a Product Quality Monitoring Method Based on Multi Scale PP-YOLO," IEEE Access, vol. 9, pp. 80373–80387, 2021, doi: 10.1109/ACCESS.2021.3085338.
- [8] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios," Sensors, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.
- [9] R. Richo, R. Yudha Adhitya, M. Khoirul Hasin, M. Syai'in, and E. Setiawan, "Analisis Pengaruh Optimizer pada Model CNN untuk Identifikasi Cacat pada Perkat Kemasan Optimizer Analysis on the CNN Model for Identification Packaging Defects," J. Sisfotenika, vol. 13, no. 2, pp. 217–229, 2023, [Online]. Available: <http://sisfotenika.stmikpontianak.ac.id/index.php/S>
- [10] I. Maulana, N. Rahaningsih, and T. Suprapti, "Analisis Penggunaan Model YOLOv8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya," JATI (Jurnal Mhs. Tek. Inform., vol. 7, no. 6, pp. 3621–3627, 2024, doi: 10.36040/jati.v7i6.8271.
- [11] A. Ma'aruf and M. Hardjianto, "Application of the You Only Look Once Version 8 Algorithm for Indonesian Sign Language Alphabet," Semin. Nas. Mhs. Fak. Teknol. Inf., vol. 2, no. September, pp. 567–576, 2023.
- [12] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," Komputa J. Ilm. Komput. dan Inform., vol. 12, no. 2, pp. 91–99, 2023, doi: 10.34010/komputa.v12i2.10654.
- [13] M. IKBAL and R. A. Saputra, "Pengenalan Rambu Lalu Lintas Menggunakan Metode YOLOv8," JIKA (Jurnal Inform., vol. 8, no. 2, p. 204, 2024, doi: 10.31000/jika.v8i2.10609.
- [14] L. Wang et al., "A defect detection method for industrial aluminum sheet surface based on improved YOLOv8 algorithm," Front. Phys., vol. 12, no. June, pp. 1–14, 2024, doi: 10.3389/fphy.2024.1419998.
- [15] B. Luo, Z. Kou, C. Han, and J. Wu, "A 'Hardware-Friendly' Foreign Object Identification Method for Belt Conveyors Based on Improved YOLOv8," Appl. Sci., vol. 13, no. 20, 2023, doi: 10.3390/app132011464.