

Model *Mobile* untuk Deteksi Objek pada *On-Shelf Availability* Produk Retail

Raihan Digo Saputra
Program Studi Informatika
Universitas Islam Indonesia
Daerah Istimewa Yogyakarta, Indonesia
raihan.saputra01@students.uui.ac.id

Dhomas Hatta Fudholi
Program Studi Informatika
Universitas Islam Indonesia
Daerah Istimewa Yogyakarta, Indonesia
hatta.fudholi@uui.ac.id

Abstrak—*Supermarket* sebagai toko retail yang menyediakan berbagai kebutuhan produk sehari-hari, dalam menjalankan proses bisnisnya, sering mengalami masalah kehabisan stok barang yang memberikan dampak kerugian paling banyak. Diperkirakan, kehabisan stok barang pada produk retail berkisar antara 7% hingga 10% di seluruh dunia, yang menyebabkan kerugian hingga miliaran tiap tahunnya. Masalah ini sebagian besar disebabkan oleh manajemen barang yang buruk. Adanya dukungan dari sistem deteksi objek diharapkan dapat meningkatkan kepedulian pihak manajemen terhadap pentingnya ketersediaan suatu barang sehingga proses jual beli pada toko retail menjadi lebih baik. Penelitian ini menerapkan arsitektur model deteksi objek YOLO (*You Only Look Once*), yakni YOLOv4-tiny yang akan dijalankan pada perangkat *mobile*. YOLOv4-tiny digunakan karena memiliki berbagai kelebihan diantaranya terbukti memiliki kinerja yang baik, ukuran modelnya yang kecil, kompleksitas komputasi yang rendah, dan dapat memproses gambar dengan cepat. Selain itu, pengolahan data pada penelitian ini meliputi *data collection*, *data preprocessing*, *data annotation*, *training model*, *evaluation*, dan *testing*. Adapun *Dataset* yang digunakan pada penelitian ini sebagai studi kasus adalah produk susu bubuk sebanyak 4637 data gambar yang berisikan 106 kelas. Performa yang dihasilkan dari model yang dibangun cukup tinggi dengan nilai *mean average precision* sebesar 92.14%. Terlebih lagi, pengujian pada perangkat android mendapatkan waktu inferensi rata-rata berkisar antara 600 hingga 700ms.

Kata Kunci— *Kehabisan Stok Barang; Ketersediaan Barang Pada Rak; Deteksi Objek; YOLOv4-tiny; Produk Retail*

I. PENDAHULUAN

Supermarket yang juga dikenal dengan swalayan adalah toko retail yang menyediakan berbagai kebutuhan produk kepada pelanggannya [1]. Di Indonesia terdapat banyak jenis *supermarket* yang mendominasi pasar ritel produk antara lain Super Indo, Hypermart, Carrefour, Lotte Mart, Giant dan lain-lain [2], [3]. Ketika menjalankan proses bisnisnya, masalah umum yang muncul di *supermarket* antara lain pencurian barang [4], barang di rak yang terjual habis [5], dan barang kadaluarsa [6]. Di antara masalah yang ada, OOS (*Out-of-Stock*) atau kehabisan barang memberikan kerugian yang paling banyak [7].

OSA (*On-Shelf Availability*) merupakan suatu istilah untuk menggambarkan ketersediaan suatu produk barang di rak toko [8]. OSA dapat berubah menjadi OOS jika tidak diperhitungkan dengan baik. Banyaknya OOS (*Out-of-Stock*) yang terjadi pada produk retail berkisar antara 7% hingga 10% di seluruh dunia dan sebagian besar dikarenakan manajemen barang yang buruk [9]. Jika tidak diatasi, hal tersebut akan berdampak pada kerugian yang akan diterima. Seperti yang

terjadi di Eropa tingkat produk tidak tersedia atau OOS tercatat 8,3%, hal tersebut berdampak pada kerugian hingga miliaran setiap tahunnya pada sektor industri [10]. Data mengenai pentingnya ketersediaan barang secara menyeluruh di Indonesia sendiri belum ada. Akan tetapi, Berdasarkan penelitian yang telah dilakukan di Bantul, Yogyakarta didapatkan data bahwa nilai koefisien regresi kelengkapan suatu produk sebesar 0,387 dengan nilai signifikansi sebesar 0,000 yang berarti $0,000 < 0,05$. Oleh karena itu, dapat disimpulkan bahwa kelengkapan produk berpengaruh positif dan juga signifikan terhadap kepuasan konsumen [11].

Adanya OSA (*On-Shelf Availability*) yang baik akan meningkatkan penjualan [12]. Akan tetapi, jika terjadi OOS (*Out-of-Stock*) pada produk retail akan menimbulkan reaksi negatif dari pelanggan yang berdampak pada kerugian [13] karena persediaan merupakan faktor utama bagi perusahaan retail, manufaktur, ataupun lainnya [14]. Hal ini terjadi diantaranya disebabkan oleh kesalahan pendataan [15], manajemen *stock* yang tidak akurat [16], dan terlambat untuk melakukan *restock* produk [17]. Terlambat dalam melakukan *restock* merupakan hal yang umum terjadi. Ini dikarenakan pada umumnya pengecekan ketersediaan produk dilakukan secara manual di sebagian besar toko oleh para karyawan. Pendekatan ini dinilai tidak efektif dan berkelanjutan karena terus menerus membutuhkan usaha manusia [12]. Maka dari itu, pengecekan ketersediaan produk di rak perlu dilakukan setiap saat agar tidak terjadi keterlambatan melakukan *restock*.

Terdapat berbagai teknologi terkemuka dalam hal ini untuk mengantisipasi terjadinya OOS, antara lain: sistem informasi ketersediaan produk [18]–[22], penggunaan robot seluler [23], dan *object detection-product recognition* [12], [24]–[26]. *Object detection* merupakan salah satu metode yang dapat digunakan untuk menganalisis OSA pada produk retail. Adanya *object detection* dapat mengurangi beban kerja yang berhubungan dengan manusia karena dapat dikerjakan secara otomatis dan dilakukan secara *real-time* [23].

Terdapat beberapa penelitian mengenai implementasi *object detection* pada produk retail dalam kurun waktu lima tahun terakhir. Pengambilan studi literatur ini dimaksudkan untuk memperoleh pengetahuan terbaru mengenai penerapan sistem deteksi objek dalam menganalisis ketersediaan produk pada produk retail. Pada penelitian yang dilakukan oleh Luka Š., dkk [24], masalah utama yang ingin diatasi dalam kasus tersebut, yaitu mengenai kehabisan stok produk. Penelitian tersebut menerapkan model *Faster R-CNN* dan *Single Shot Multibox detection* (SSD) dengan metode *transfer learning*

pada *custom dataset* yang terdiri dari 684 objek. Hasil dari penelitian tersebut didapatkan *object detection accuracy* sebesar 97.06% pada model *Faster R-CNN* dan 75.42% pada model *SSD*.

Penelitian yang dilakukan oleh Ramiz Y., dkk [12] menyebutkan bahwa ketersediaan produk pada rak merupakan kunci untuk meningkatkan keuntungan dari toko grosir. *Dataset* yang digunakan merupakan *WebMarket dataset* yang terdiri dari 3153 gambar. Gambar tersebut diambil dengan menggunakan tiga kamera digital dengan jarak satu meter. Penelitian tersebut menerapkan beberapa model arsitektur *object detection*. Hasil dari penelitian tersebut didapatkan pengetahuan bahwa *RetinaNet* dan *YOLOv3* merupakan model dengan tingkat akurasi tertinggi. Pada model *RetinaNet* didapatkan *mAP* sebesar 0.71 dengan *F1-score* sebesar 0.74, sedangkan pada *YOLOv3* didapatkan *mAP* sebesar 0.74 dengan *F1-score* 0.66.

Pada penelitian yang dilakukan Dipendra J., dkk [25], masalah yang ingin diatasi berupa pemeriksaan ketersediaan barang yang dilakukan secara manual. Penelitian tersebut menggunakan *custom dataset* yang terdiri dari 1000 gambar dan menerapkan *data cleaning* pada prosesnya. Arsitektur *object detection* yang digunakan pada penelitian ini, yaitu beberapa *family model* dari *EfficientDet* dan *YOLOv5*. Performa terbaik didapatkan pada model *EfficientDet-D1* dengan *precision* sebesar 57.0%, *recall* sebesar 63.8%, dan *F1-score* sebesar 60.2% untuk *family model* *EfficientDet* dan pada model *YOLOv5n6* didapatkan *precision* sebesar 76.3%, *recall* sebesar 63.8%, dan *F1-score* sebesar 71.3% untuk *family model* *YOLOv5*.

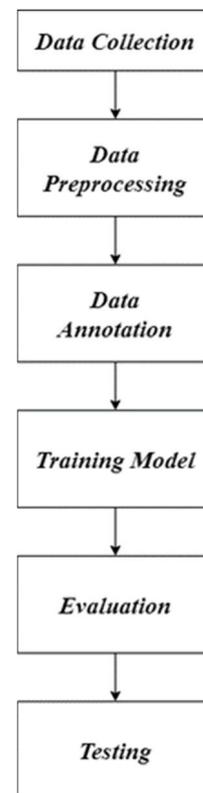
Berikutnya penelitian yang dilakukan Ankit S., dkk [26] bertujuan untuk menyempurnakan *product recognition* pada toko retail. Penelitian tersebut menggunakan dua *dataset* yang berbeda sebagai perbandingan, yaitu *Grozi-32k* dan *GP-180*. Model yang digunakan pada penelitian tersebut, yaitu *Faster R-CNN*. Model yang dilatih dengan *dataset* *GP-180* menghasilkan performa *mAP@0.5* sebesar 82.70 dan *PR@0.5* sebesar 89.70, sedangkan untuk *dataset* *Grozi* didapatkan *mAP* sebesar 47.77% dan *PR* sebesar 55.11%.

Ada berbagai macam model algoritma yang dapat digunakan untuk melakukan *object detection*, salah satunya dengan *YOLOv4-tiny* (*You Only Look Once version 4*) yang diterapkan pada penelitian ini. *YOLOv4-tiny* memiliki berbagai kelebihan diantaranya memiliki kinerja yang baik dalam akurasi, ukuran modelnya yang kecil, kompleksitas komputasi yang rendah [27]. Selain itu, terdapat penelitian yang dilakukan oleh Zuanita S. J., dkk [28] yang menggunakan algoritma *YOLOv4-tiny* untuk melakukan deteksi rias adat nusantara, terbukti mampu mendapatkan hasil rata-rata yang baik dengan akurasi 95.20% dan waktu deteksi selama 327ms. Algoritma ini menerapkan proses *training* pada *dataset* gambar dengan label atau *class* yang telah dilakukan proses pelabelan. Selain itu, algoritma ini juga dapat berjalan di perangkat *mobile*. *Model mobile* dipilih karena memiliki berbagai keunggulan, diantaranya untuk menjaga privasi, tidak memerlukan koneksi internet karena tidak melakukan komunikasi dengan *server* dan *response time* yang cepat karena langsung dijalankan pada perangkat [29].

Penelitian ini akan mengembangkan *object detection* pada produk retail dengan batasan berupa penggunaan produk retail susu bubuk sebagai objek penelitian. Penggunaan susu bubuk sebagai objek penelitian dikarenakan produk tersebut merupakan bagian dari produk retail yang umum dijumpai pada *supermarket*. Salah satu upaya yang dilakukan untuk membedakan dengan penelitian terdahulu, yaitu pada penelitian ini akan menggunakan data gambar dan *class* yang lebih banyak, yakni sebanyak 4637 data gambar dan 106 *class data*. Hal tersebut dikarenakan pada penelitian sebelumnya, telah dilakukan pengembangan deteksi objek dengan menggunakan *class data* yang sedikit. Adanya penelitian ini diharapkan dapat meningkatkan kesadaran pihak manajemen toko retail akan pentingnya ketersediaan suatu produk di rak, dalam upaya memaksimalkan keuntungan.

II. METODOLOGI PENELITIAN

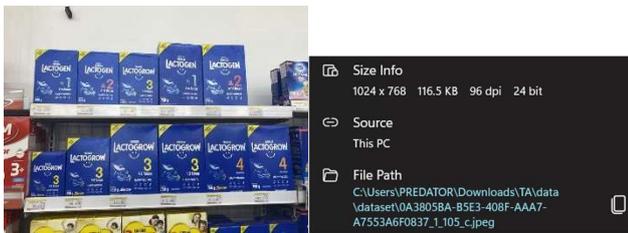
Penelitian ini memiliki beberapa tahapan yang dilakukan oleh peneliti. Tahapan tersebut menerapkan enam alur pengolahan data yang terdiri dari *data collection*, *data preprocessing*, *data annotation*, *training model*, *evaluation*, dan *testing*. Alur tahapan ini ditunjukkan pada Gambar 1.



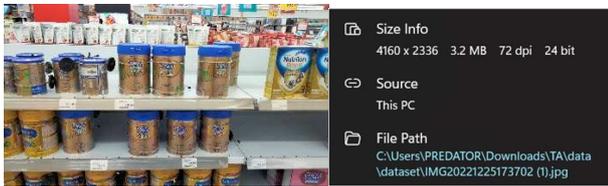
Gambar 1. Alur penerapan arsitektur *YOLOv4-tiny*

A. Data Collection

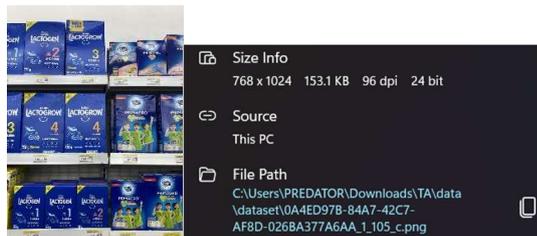
Pada tahap ini dilakukan pengambilan *dataset* menggunakan kamera *smartphone*. Sebanyak 4637 gambar berhasil diperoleh yang akan dibagi ke dalam dua kategori, yaitu *data training* dan *data testing* dengan perbandingan 80:20. *Dataset* yang digunakan merupakan data gambar produk retail yang terdiri dari berbagai macam produk susu bubuk, baik dalam kemasan kaleng dan juga kemasan dus. Contoh data gambar yang akan digunakan pada penelitian ini ditunjukkan pada Gambar 2.



(a) Gambar Beresolusi 1024x768



(b) Gambar Beresolusi 4160x2336



(c) Gambar Beresolusi 768x1024

Gambar 2. Data gambar produk susu bubuk

Gambar 2 merupakan contoh *dataset* yang digunakan. Terdapat beragam resolusi yang ada pada *dataset* gambar tersebut misalnya 1024x768 ditunjukkan pada Gambar 2(a), 4160x2336 ditunjukkan pada Gambar 2(b), dan 768x1024 ditunjukkan pada Gambar 2(c). Selain itu, terdapat tiga format gambar yang ada pada *dataset*, yakni “.jpeg”, “.jpg”, dan “.png”. Hal tersebut terjadi karena pengambilan gambar menggunakan beberapa perangkat. Luas pengambilan data gambar beragam mulai dari satu hingga lima rak seperti yang ditunjukkan pada Gambar 3.



(a) Luas satu rak



(b) Luas dua rak



(c) Luas tiga rak



(d) Luas empat rak



(e) Luas lima rak

Gambar 3. Luas Pengambilan Data Gambar

Pada Gambar 3 dapat dilihat bahwa terdapat berbagai macam luas pengambilan data gambar mulai dari satu rak hingga lima rak, yang secara berurutan ditunjukkan pada Gambar 3(a) hingga 3(e). Selain itu, dapat diketahui pula bahwa semakin luas pengambilan objek maka objek gambar akan semakin kecil. Jika suatu objek semakin kecil maka detail objek tersebut juga semakin tidak terlihat.

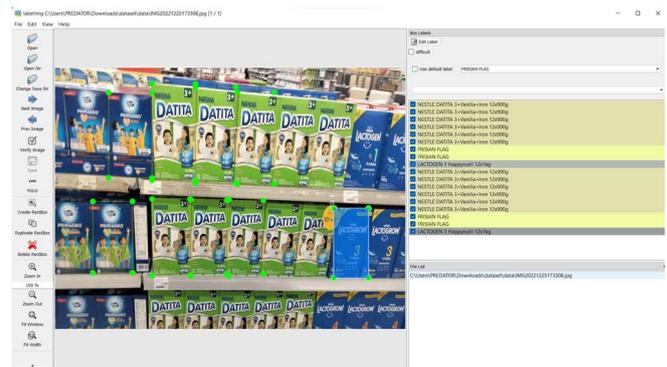
B. Data Preprocessing

Proses *preprocessing* yang dilakukan pada gambar menerapkan pengaturan resolusi, *sharpening*, dan penyalarsan format gambar. Pengaturan resolusi dilakukan untuk menyamakan resolusi dari semua data gambar, yang semula terdapat berbagai macam resolusi akan dilakukan perubahan menjadi 416x416. Perubahan tersebut dilakukan karena model akan dilatih menggunakan jaringan 416x416. Selain itu, perubahan resolusi juga dapat mengurangi *size* pada *dataset*, besar atau kecilnya ukuran ini berpengaruh pada waktu yang dibutuhkan untuk *train model*.

Pada tahap *sharpening* akan dilakukan untuk memperjelas atau mempertajam objek yang terdapat pada gambar. Proses penajaman gambar menggunakan bantuan dari *library* PIL (*Python Imaging Library*). *Library* ini merupakan *library* yang dapat digunakan untuk memanipulasi gambar [30]. Pada YOLOv4-*tiny* format gambar yang digunakan untuk melakukan pelatihan jaringan, yaitu berformat “.jpg” maka semua gambar perlu diubah ke dalam format “.jpg”.

C. Data Annotation

Pada tahap ini data gambar yang telah dilakukan *preprocessing* akan dilakukan pelabelan. Label atau kelas yang digunakan pada penelitian ini berjumlah 106 kelas. Proses pelabelan dilakukan seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Proses pelabelan *dataset*

Gambar 4 merupakan proses pelabelan pada gambar dengan *tools* yang umum digunakan untuk melakukan pelabelan, yaitu *labelimg*. Proses pelabelan dilakukan dengan memberikan *bounding boxes* pada objek sebagai pemisahannya, kemudian diberikan *class* dari objek tersebut yang berupa label berbagai macam produk susu bubuk. Pemberian label harus dilakukan dengan baik karena akan berpengaruh dengan akurasi dari model deteksi objek.

D. Training Model

Dataset yang telah melalui proses pelabelan akan digunakan pada proses *training model*. Pada tahap ini pertama-tama dilakukan konfigurasi parameter YOLOv4-*tiny* dengan *dataset* penelitian. Model ini dipilih karena memiliki berbagai kelebihan, seperti yang dijabarkan oleh Liu Q., dkk. [27], yakni diantaranya ukuran model yang kecil tapi memiliki kinerja yang baik. Hal tersebut merupakan faktor kunci dalam pengembangan *object detection* pada perangkat *mobile*.

Konfigurasi parameter yang dilakukan pada arsitektur YOLOv4-*tiny*, disesuaikan dengan jumlah kelas yang digunakan serta dengan mengikuti aturan yang dijelaskan oleh Badharudheen [31]. Pada konfigurasi yang diterapkan pada penelitian tersebut terbukti menghasilkan akurasi yang cukup baik. Penelitian ini menerapkan konfigurasi model YOLOv4-*tiny* ditunjukkan pada Tabel 1.

TABEL 1. KONFIGURASI MODEL YOLOV4-TINY

Parameter	Nilai
<i>Batch</i>	64
<i>Subdivisions</i>	32
<i>Width</i>	416
<i>Height</i>	416
<i>Channels</i>	3
<i>Momentum</i>	0.9
<i>Decay</i>	0.0005
<i>Angle</i>	0
<i>Saturation</i>	1.5
<i>Exposure</i>	1.5
<i>Hue</i>	0.1
<i>Learning rate</i>	0.00261
<i>Burn in</i>	1000
<i>Max batches</i>	212000
<i>Policy</i>	<i>steps</i>
<i>Steps</i>	169600, 190800
<i>Scales</i>	0.1, 0.1

Pada Tabel 1 *batch* yang digunakan sebesar 64 dengan *subdivisions* sebanyak 32. Hal tersebut diartikan satu iterasi terdiri dari 64 gambar dengan dua gambar per *mini-batch* yang akan diproses. *Width* dan *height* merupakan representasi dari tinggi dan lebar *input* atau *pixel* suatu gambar. *Max_batches* merupakan hasil perkalian antara banyaknya kelas dikali dengan 2000. Nilai *steps* merupakan nilai *max_batches* dikali 80% dan 90% [32]. Selain itu, pada setiap “*yolo layer*” pada baris *classes* akan diubah sesuai dengan jumlah kelas yang dipakai, yakni 106 *class*. *Filter* pada setiap “*convolutional*” *layer* juga diganti mengikuti rumus $filter=(class + 5)*3$ [32].

Pelatihan dilakukan dengan menerapkan konsep *transfer learning*. Pada algoritma YOLOv4-*tiny* hasil *training* akan disimpan dengan tiga kategori, yaitu *best*, *last*, dan setiap 10000 *steps* dengan format “.*weight*”. Kategori *best* merupakan hasil terbaik yang didapatkan selama proses

pelatihan, kategori *last* merupakan hasil terakhir kali model yang dilatih selama proses pelatihan, dan kategori 10000 *steps* merupakan hasil pelatihan yang disimpan setiap 10000 *steps*.

E. Evaluation

Dari beberapa model YOLOv4-*tiny* yang sudah didapatkan, akan dilakukan perhitungan performa untuk menentukan model mana yang paling baik digunakan untuk penelitian ini. Perhitungan performa dilakukan dengan menggunakan *confusion matrix*. *Confusion matrix* sendiri merupakan matriks yang dapat mengklasifikasi benar atau salahnya suatu data uji atau *test* [33]. Dalam *confusion matrix* terdapat empat klasifikasi, yaitu *true positive (TP)*, *false negative (FN)*, *true negative (TN)*, dan *false positive (FP)*. Dari variabel-variabel tersebut, suatu model dapat dihitung tingkat *precision (P)*, *recall (R)*, dan *mean average precision (mAP)* dengan persamaan sebagai berikut:

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

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

Nilai *precision* pada persamaan 1 merupakan nilai kemampuan suatu model dalam melakukan identifikasi hanya pada objek yang relevan. Pada persamaan 2, nilai *recall* merupakan nilai kemampuan suatu model untuk menemukan semua objek yang relevan. Nilai *precision* dan *recall* ini digunakan untuk menghitung *average precision (AP)*.

$$AP = \frac{1}{11} \sum_{R \in \{0,0.1,\dots,0.9,1\}} P_{interp}(R) \quad (3)$$

$$P_{interp}(R) = \max_{\tilde{R}: \tilde{R} \geq R} P(\tilde{R}) \quad (4)$$

Nilai *average precision (AP)* pada persamaan 3 merupakan ringkasan dari nilai *precision* dan *recall*. Nilai tersebut diringkas dengan rata-rata nilai *precision* maksimum pada 11 titik *recall*. Titik ini dimulai dari 0 hingga 1 dan berkelipatan setiap 0.1.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5)$$

Nilai *mean average precision (mAP)* merupakan hasil nilai rata-rata keakuratan dari suatu model dalam melakukan deteksi objek di semua *class*. Nilai N dalam persamaan 5 merupakan jumlah kelas yang digunakan pada model. Pada penelitian ini evaluasi mAP akan digunakan sebagai tolak ukur pemilihan suatu model. Seluruh persamaan tersebut dikutip dari penelitian Padilla R., dkk. [34].

F. Testing

Pada tahap ini dilakukan pengujian untuk mengetahui apakah sistem dapat berjalan dengan baik dalam mendeteksi ketersediaan suatu produk retail, dalam kasus ini susu bubuk yang terdapat pada toko retail. Pengujian dilakukan dengan instalasi sistem pada perangkat bergerak, khususnya dalam hal ini perangkat *smartphone*. Model yang telah dipilih perlu dilakukan perubahan format terlebih dahulu dari “.*weight*” menjadi “.*tflite*” agar dapat digunakan. Pengujian difokuskan pada kebenaran prediksi serta waktu inferensi yang

dibutuhkan. Terdapat dua perangkat *smartphone* yang digunakan pada proses ini dengan spesifikasi pada Tabel 2.

TABEL II. SPESIFIKASI *SMARTPHONE* YANG DIGUNAKAN PADA PROSES *TESTING*

Komponen	Xiaomi Redmi Note 4x	Oppo A53
Processor	Snapdragon 625	Snapdragon 460
RAM	3 Gb	4 Gb
Resolusi Kamera	13 MP	13 MP

Pada Tabel 2 terdapat dua *smartphone* yang digunakan pada proses *testing model* di perangkat *mobile*. Perangkat pertama merupakan Xiaomi Redmi Note 4x merupakan produk *smartphone* yang keluar pada tahun 2017, dilengkapi dengan *processor* snapdragon 625, RAM 3 Gb, dan kamera tunggal dengan resolusi 13 MP. Perangkat kedua merupakan Oppo A53 merupakan produk *smartphone* yang keluar pada tahun 2020, dilengkapi dengan *processor* snapdragon 460, RAM 4 Gb, dan memiliki tiga kamera dengan resolusi kamera utama 13 MP. Berdasarkan pemaparan sebelumnya, Oppo A53 memiliki keunggulan pada sektor kamera, hal ini cukup penting untuk diketahui karena pengujian pada perangkat *mobile* berkaitan langsung dengan kamera yang digunakan.

G. Perangkat Penelitian

Perangkat keras atau *hardware* merupakan faktor utama yang diperlukan dalam melakukan penelitian. Hal tersebut dikarenakan *hardware* sebagai otak untuk menjalankan pengolahan. *Hardware* yang mumpuni akan berpengaruh pada cepat atau lambatnya proses pengolahan, baik pengolahan *dataset* maupun *training model*. Peneliti menyelesaikan penelitian dengan menggunakan spesifikasi *hardware* seperti yang ada pada Tabel 3.

TABEL III. SPESIFIKASI *HARDWARE* YANG DIGUNAKAN DALAM PENELITIAN

Komponen	Spesifikasi
Processor	Intel(R) Core(TM) i5-9300H @ 2.40Ghz
RAM	8 Gb
GPU	NVIDIA GeForce GTX 1650 4Gb

Pada Tabel 3 perangkat penelitian menggunakan spesifikasi *hardware* dengan *processor* 4 core dan 8 threads, didukung NVIDIA GeForce GTX 1650 yang memiliki *compute capability CUDA* sebesar 7.5. Selain itu, perangkat ini juga dilengkapi dengan RAM 8 Gb untuk menunjang performanya. Komponen ini merupakan komponen yang cukup untuk melakukan *training model YOLOv4-tiny*.

III. HASIL DAN PEMBAHASAN

Pada tahap ini akan dijelaskan mengenai hasil dari berbagai proses penerapan objek deteksi mulai dari hasil pengolahan *dataset*, hasil pelatihan dari model objek deteksi, serta hasil dari penerapan *YOLOv4-tiny* pada perangkat *mobile*.

A. Hasil Pengolahan *Dataset*

Hasil dari *dataset* yang telah dikumpulkan akan dilakukan *preprocessing* dan juga pelabelan. Proses ini merupakan proses penentu bagus atau tidaknya suatu *dataset*. Hasil *Preprocessing* pada penelitian ini ditunjukkan pada Gambar 5.



Gambar 5. Hasil *preprocessing* gambar

Gambar 5 merupakan contoh hasil dari proses *preprocessing* yang menerapkan *sharpening*, pengaturan resolusi, dan perubahan format. Data gambar yang digunakan pada penelitian ini memiliki resolusi 416x416 dengan format gambar “.jpg”. Proses perubahan resolusi ini akan membuat struktur gambar menjadi lebih kecil dari pada gambar sebelum dilakukan pengaturan resolusi.

Setelah dilakukan *preprocessing*, proses dilanjutkan dengan pemberian label pada data. Pemberian label harus dilakukan dengan baik karena akan berpengaruh dengan akurasi model. Pemberian label dilakukan dengan *tools labeling* dengan format data YOLO dan disimpan dalam file “.txt”. Setiap satu file “.txt” berisikan semua *object* pada gambar yang telah diberi label. Hasil dari proses pelabelan ditunjukkan pada Gambar 6.



Gambar 6. Hasil *Image Annotation*

Gambar 6 merupakan contoh hasil dari proses *image annotation*. Kolom pertama pada file merupakan *class id* dari suatu *object*. Kolom kedua dan ketiga mewakili titik *center* dari sumbu x dan titik *center* dari sumbu y. Selain itu, Kolom keempat dan kelima melambangkan lebar dan tinggi dari kotak pembatas label.

B. Hasil *Training Model*

Hasil dari *training model* dengan arsitektur *YOLOv4-tiny* menghasilkan model *YOLOv4-tiny* dengan file ekstensi “.weight”. Saat melakukan *training*, model akan menyimpan tiga pembobotan antara lain pembobotan setiap 10.000 *steps*, “*best.weight*” yang merupakan pembobotan terbaik selama melakukan proses *training*, dan “*last.weight*” merupakan *steps* terakhir yang dilakukan pada saat *training* dan disimpan setiap 100 *steps*.

Proses pelatihan ini memakan waktu sekitar 111 jam. Hal tersebut berbanding lurus dengan banyaknya kelas yang digunakan, yakni 106 kelas dengan *max batches* 212.000. Pada proses pelatihan ini mendapatkan hasil mAP terbaik sebesar 92.14%. Tabel 4 merupakan rincian hasil terbaik dari proses *training*.

TABEL IV. RINCIAN *EVALUATION TRAINING* TERBAIK

<i>Evaluation</i>	Nilai
<i>TP</i>	5833
<i>FP</i>	1633
<i>FN</i>	1287
<i>Precision</i>	0.78
<i>Recall</i>	0.82
<i>F1-score</i>	0.80
<i>Average IoU</i>	62.56%
<i>mAP</i>	92.14%

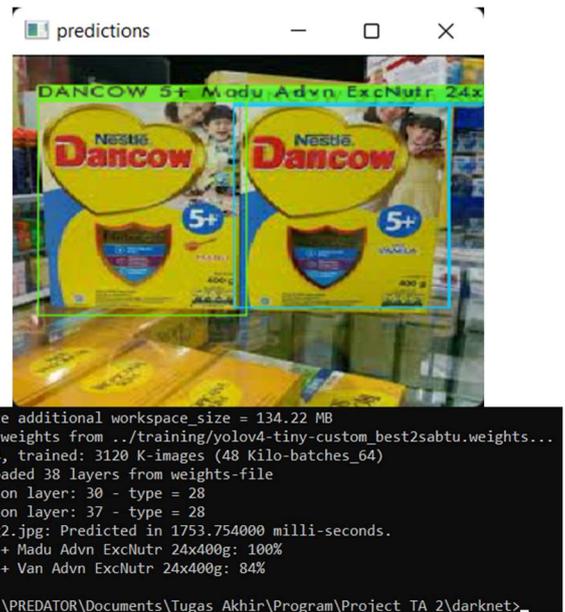
Berdasarkan hasil *evaluation* pada Tabel 4, nilai yang dihasilkan dari bobot tersebut sudah cukup baik untuk digunakan. Maka dari itu bobot tersebut akan digunakan sebagai model YOLOv4-*tiny* pada penelitian ini. Beberapa hasil *AP* (*Average Precision*) per label dari proses *training* ditunjukkan pada Tabel 5.

TABEL V. *RANDOM* SAMPEL SETIAP LABEL

<i>Class_id</i>	Nama	<i>TP</i>	<i>FP</i>	<i>AP</i>
9	LACTOGEN 2 Happynutri 24x350g	88	23	93.60%
27	DANCOW 3+ Van Advn ExcNutr 12x800g	119	59	93.62%
34	DANCOW 1+ Madu Advn ExcNutr 12x800g	128	76	94.37%
71	HILO	7	13	28.88%
81	DANCOW 1+ Madu Advn ExcNutr 24x400g REDESIGN	11	0	100%

Dari Tabel 5 dapat diketahui bahwa beberapa *class* memiliki nilai *AP* yang sudah cukup baik, yakni mencapai lebih dari 90%. Selain itu, terdapat beberapa *class* memiliki nilai *AP* yang sempurna mencapai 100%, hal ini dapat terjadi karena model dapat dengan benar melakukan deteksi untuk *class* tersebut. Terdapat pula *class* yang memiliki nilai *AP* yang rendah dengan nilai 28.88%, hal tersebut terjadi dikarenakan model banyak melakukan kesalahan dalam mendeteksi objek pada *class* tersebut.

Setelah mendapatkan bobot yang terbaik, tahap selanjutnya uji coba pada model YOLOv4-*tiny*. Proses uji coba ini dilakukan untuk melihat kemampuan dari model. Gambar yang digunakan pada proses ini merupakan gambar yang terdiri dari dua objek susu bubuk. Proses uji coba ditunjukkan pada Gambar 7.

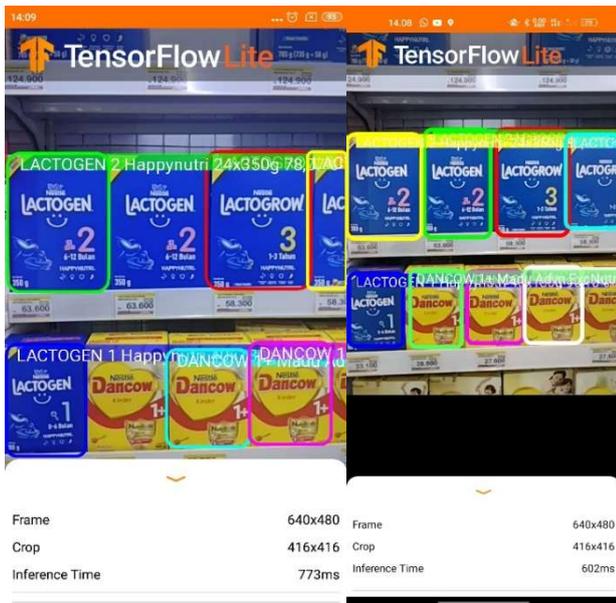


Gambar 7. Uji Coba Model Terhadap Gambar Produk Susu Bubuk

Pada Gambar 7 model YOLOv4-*tiny* mampu mendeteksi dua objek dengan benar. Produk yang berhasil dideteksi, yaitu “Dancow 5+ Madu *Advanced* Excelnutri 400g” dan “Dancow 5+ Vanila *Advanced* Excelnutri 400g”. Hal tersebut membuktikan bahwa YOLOv4-*tiny* cukup akurat dalam mengenali objek produk retail susu bubuk yang ditampilkan dengan bantuan *bounding box* walaupun resolusi gambar berkurang yang dikarenakan pengaturan resolusi.

C. Hasil Penerapan pada Perangkat *Mobile*

Sebelum digunakan pada perangkat *mobile*, model yang telah dipilih dilakukan konversi terlebih dahulu dari format “.weight” menjadi “.tflite”. Hal tersebut dilakukan karena aplikasi yang dikembangkan pada perangkat *mobile* hanya dapat menjalankan model dengan format “.tflite”. Setelah itu, penerapan pada perangkat *mobile* menggunakan *project tensorflow-yolov4-tflite* yang dikembangkan oleh Haroon Shakeel [35]. Pada tahap ini akan dilihat bagaimana inferensi waktu yang dihasilkan. Hal tersebut penting dilakukan untuk menunjukkan bahwa model dapat berjalan secara *real-time*. Uji coba menggunakan perangkat *mobile* ditunjukkan pada Gambar 8.



Gambar 8. Hasil penerapan model pada perangkat *mobile*

Pada Gambar 8 diketahui bahwa model dapat berjalan pada kedua perangkat *android* walaupun kemampuan mengenali objek dengan *bounding box* tidak sempurna, namun cukup akurat. Jika diperhatikan dengan seksama, kemampuan mengenali objek juga bergantung pada kualitas kamera yang digunakan. Gambar sebelah kanan merupakan hasil pengujian dari perangkat Oppo, sedangkan gambar sebelah kiri merupakan hasil pengujian dari perangkat Xiaomi. Berdasarkan gambar tersebut dapat diketahui bahwa perangkat sebelah kanan lebih banyak menghasilkan *bounding box* daripada perangkat yang sebelah kiri. Selain itu, waktu inferensi yang dihasilkan dari kedua perangkat ditunjukkan pada Tabel 6.

TABEL VI. KECEPATAN WAKTU INFERENSI PADA PERANGKAT ANDROID

Nama Perangkat	Waktu Inferensi
Xiaomi Redmi Note 4x	773ms
Oppo A53	602ms

Dari Tabel 6 disimpulkan bahwa model YOLOv4-*tiny* mampu melakukan *object detection* secara *real-time* pada produk susu bubuk, dengan rata-rata *inference time* yang diperlukan 600 hingga 700ms dari kedua perangkat. Model *mobile* ini tidaklah menjadi sistem OSA yang utuh. Akan tetapi, menjadi alat yang dapat membantu proses deteksi sehingga pencatatan yang dapat dilakukan lebih cepat dan otomatis.

IV. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan *object detection* pada produk retail khususnya susu bubuk, baik dalam kemasan dus maupun kaleng. Data produk susu yang digunakan sebanyak 4637 data gambar dan 106 *class data*. Selain itu, Arsitektur *object detection* yang diterapkan pada penelitian ini yaitu YOLOv4-*tiny*. Dalam pengembangan *object detection* pada penelitian ini melalui enam alur pengolahan data yang terdiri dari *data collection*, *data preprocessing*, *data annotation*, *training model*, *evaluation*, dan *testing*.

Berdasarkan hasil penerapan dan pengujian, diketahui bahwa *object detection* yang menerapkan model arsitektur YOLOv4-*tiny* berhasil diimplementasikan untuk mendeteksi produk retail susu bubuk secara *real-time*. Performa *object detection* yang dihasilkan dengan arsitektur YOLOv4-*tiny* mencapai angka yang cukup tinggi, yaitu dengan nilai *mean average precision* sebesar 92.14%. Selain itu, berdasarkan pengujian yang dilakukan pada perangkat *mobile* khususnya perangkat android, menghasilkan waktu inferensi rata-rata berkisar 600 hingga 700ms. Waktu tersebut merupakan waktu yang cukup untuk menjalankan *object detection* secara *real-time*.

Terdapat beberapa hal yang dapat dijadikan saran untuk penelitian selanjutnya, yaitu dengan menggunakan *dataset* yang memiliki resolusi lebih tinggi. Hal tersebut bertujuan untuk melihat pengaruh resolusi terhadap performa dari model. Selain itu, dapat juga mencoba model arsitektur yang lebih ringan mengingat lamanya waktu yang diperlukan untuk melakukan *training* pada model.

REFERENSI

- [1] R. Puspa *et al.*, "FAKTOR YANG MEMPENGARUHI KEPUASAN PELANGGAN BERDASARKAN BAURAN PEMASARAN PADA SUPERMARKET K-STORE KRAKATAU JUNCTION 1)," vol. 13, no. 02, 2020, doi: 10.46306/jbbe.v13i2.
- [2] D. ANGELIA, "Supermarket dengan Gerai Terbanyak di Indonesia 2021," 2022. <https://goodstats.id/article/supermarket-dengan-gerai-terbanyak-di-indonesia-2021-QVGHG> (accessed Oct. 09, 2022).
- [3] D. Suryadarma, A. Poesoro, S. Budiyati, Akhmadi, and M. Rosfadhila, *Dampak Supermarket terhadap Pasar dan Pedagang Ritel Tradisional di Daerah Perkotaan di Indonesia*. Smeru, 2007. [Online]. Available: www.smeru.or.id
- [4] E. N. ALFIAH, "PENGENDALIAN INTERNAL ATAS PERSEDIAAN BARANG DAGANG DI SWALAYAN SENDY'S KOTA PALANGKA RAYA."
- [5] N. Hafiz Ar, M. Kurniawan, and A. T. Surabaya, "Analisis Fast Moving Consumer Goods untuk Memprakirakan Penjualan Barang Menggunakan Metode Triple Exponential Smoothing 115."
- [6] H. S. Disemandi and P. Ariesta Nadia, "PRODUK BAHAN PANGAN KADALUARSA YANG DIIPERJUALBELIKAN DI SUPERMARKET: SUATU KAJIAN HUKUM PERLINDUNGAN KONSUMEN," *Maleo Law Journal*, 2021.
- [7] K. Campo, E. Gijsbrechts, and P. Nisol, "Towards understanding consumer response to stock-outs," *Journal of Retailing*, vol. 76, no. 2, pp. 219–242, Jun. 2000, doi: 10.1016/S0022-4359(00)00026-9.
- [8] C. V. B. Saragih, M. S. Shihab, and W. Purwanto, "PENGARUH KUALITAS PRODUK, KETERSEDIAAN PRODUK DAN GAYA HIDUP TERHADAP KEPUTUSAN PEMBELIAN PRODUK LULUR MANDI SUMBER AYU DI JAKARTA," *Jurnal MIX*, vol. 6, no. 2, 2013, Accessed: Oct. 01, 2022. [Online]. Available: [01/10/2022http://digilib.mercubuana.ac.id/manager/t!@file_artikel_a_bstrak/Isi_Artikel_949723712677.pdf](http://digilib.mercubuana.ac.id/manager/t!@file_artikel_a_bstrak/Isi_Artikel_949723712677.pdf)
- [9] J. Ranjan and S. Puri, "Out of Stock conditions affecting Customer satisfaction and customer loyalty," *ABRM*, 2012.
- [10] A. Mitchell, "Improving On-Shelf Availability It Matters More," 2012. [Online]. Available: www.SymphonyIRI.eu
- [11] S. Anjarwan, "PENGARUH KELENGKAPAN PRODUK, PERSEPSI HARGA DAN LOKASI TERHADAP KEPUASAN KONSUMEN PADA TOSERBA DM BARU 1 PLERET BANTUL YOGYAKARTA," 2018.
- [12] R. Yilmazer and D. Birant, "Shelf Auditing Based on Image Classification Using Semi-Supervised Deep Learning to Increase On-Shelf Availability in Grocery Stores," *Sensors*, vol. 21, no. 2, p. 327, Jan. 2021, doi: 10.3390/s21020327.
- [13] M. Ezhilkumar, "ENHANCING BEHAVIORAL INTENTION IN OUT-OF-STOCK SITUATIONS-THE MEDIATING ROLE OF PERCEIVED PRODUCT UNIQUENESS AND PERCEIVED CONSUMPTION RISK," 2020.
- [14] L. Saresa, S. Afriani, and Y. Fitriano, "Analisis Sistem Pengendalian Internal Persediaan Barang Dagang pada Alfamart Merapi Kebun Tebeng Bengkulu," *Jurnal Ekonomi, Manajemen, Akuntansi dan Keuangan*, vol. 2, no. 3.

- [15] A. Hijriani, J. A. Safitri, R. I. Adi Pribadi, and R. Andrian, "Pengembangan Sistem Informasi Manajemen Supplier dan Barang dengan Extreme Programming," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 1, Apr. 2020, doi: 10.28932/jutisi.v6i1.2132.
- [16] E. Muningsih and D. S. Kiswati, "Penerapan Metode K-Means Untuk Clustering Produk Online Shop Dalam Penentuan Stok Barang," *Jurnal Bianglala Informatika*, vol. 3, 2015.
- [17] C. D. Rumiarti, B. R. Setiawan, I. Dewa, and M. P. Wiana, "KAJIAN PERENCANAAN STRATEGIS SISTEM INFORMASI PADA BISNIS RITEL BERBASIS METODOLOGI WARD & PEPPARD: STUDI KASUS PT. GRAMEDIA ASRI MEDIA," vol. 6, no. 3, pp. 245–256, 2019, doi: 10.25126/jtiik.20196926.
- [18] R. Akbar and J. Juliastriozza, "Penerapan Enterprise Resource Planning (ERP) untuk Sistem Informasi Pembelian, Persediaan dan Penjualan Barang pada Toko EMI GROSIR dan ECERAN," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 1, no. 1, pp. 7–17, Oct. 2015, doi: 10.25077/TEKNOSI.v1i1.2015.7.
- [19] A. S. Arandhea and R. Puspitasari, "Penerapan Sistem Informasi Akuntansi Untuk Persediaan Barang Dagang," *Jurnal Aplikasi Bisnis Kesatuan*, vol. 1, no. 2, Dec. 2021, doi: 10.37641/jabkes.v1i2.1180.
- [20] T. Rahmasari, "Perancangan Sistem Informasi Akuntansi Persediaan Barang Dagang Pada Toserba Selamat Menggunakan Php Dan Mysql," *is The Best Accounting Information Systems and Information Technology Business Enterprise this is link for OJS us*, vol. 4, no. 1, pp. 411–425, Jun. 2019, doi: 10.34010/aisthebest.v4i1.1830.
- [21] M. A. Swasono and A. T. Prastowo, "ANALISIS DAN PERANCANGAN SISTEM INFOMASI PENGENDALIAN PERSEDIAAN BARANG," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, 2021.
- [22] S. Limanto, "Pengembangan Aplikasi Sistem Informasi untuk Membantu Mengontrol Stok dan Meningkatkan Layanan kepada Pelanggan," 2018.
- [23] M. Paolanti, L. Romeo, M. Martini, A. Mancini, E. Frontoni, and P. Zingaretti, "Robotic retail surveying by deep learning visual and textual data," *Rob Auton Syst*, vol. 118, pp. 179–188, Aug. 2019, doi: 10.1016/j.robot.2019.01.021.
- [24] L. Šećerović and V. Papić, "Detecting missing products in commercial refrigerators using convolutional neural networks," *Symposium on Neural Networks and Application (NEUREL)*, Nov. 2018.
- [25] D. Jha, A. Mahjoubfar, and A. Joshi, "Designing an Efficient End-to-end Machine Learning Pipeline for Real-time Empty-shelf Detection," May 2022, [Online]. Available: <http://arxiv.org/abs/2205.13060>
- [26] A. Sinha, S. Banerjee, and P. Chattopadhyay, "An Improved Deep Learning Approach For Product Recognition on Racks in Retail Stores," Feb. 2022, [Online]. Available: <http://arxiv.org/abs/2202.13081>
- [27] Q. Liu, X. Fan, Z. Xi, Z. Yin, and Z. Yang, "Object detection based on Yolov4-Tiny and Improved Bidirectional feature pyramid network," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Feb. 2022. doi: 10.1088/1742-6596/2209/1/012023.
- [28] Z. S. Jannah and F. A. Sutanto, "Implementasi Algoritma YOLO (You Only Look Once) Untuk Deteksi Rias Adat Nusantara," *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 22, no. 3, p. 1490, Oct. 2022, doi: 10.33087/jjubj.v22i3.2421.
- [29] X. Dai, I. Spasic, S. Chapman, and B. Meyer, "The State of the Art in Implementing Machine Learning for Mobile Apps: A Survey," in *Conference Proceedings - IEEE SOUTHEASTCON*, Institute of Electrical and Electronics Engineers Inc., Mar. 2020. doi: 10.1109/SoutheastCon44009.2020.9249652.
- [30] H. Gujar *et al.*, "Python Based Image Processing", Accessed: May 26, 2023. [Online]. Available: https://www.researchgate.net/publication/309208697_Python_Based_Image_Processing
- [31] J. K. Badharudheen, "Vision System for Apple Harvesting Robot," 2021, Accessed: Jun. 16, 2023. [Online]. Available: <https://digikogu.taltech.ee/et/Download/4d27e33c-0c66-493e-a56a-6b02e1b9e4af>
- [32] A. Bochkovskiy, "How to train (to detect your custom objects)." <https://github.com/AlexeyAB/darknet> (accessed Mar. 23, 2023).
- [33] D. Normawati and S. A. Prayogi, "Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," 2021.
- [34] R. Padilla, S. L. Netto, and E. A. B. da Silva, *A Survey on Performance Metrics for Object-Detection Algorithms*. 2020.
- [35] Haroon Shakeel, "Convert Darknet YOLOv4 or YOLOv3 to TensorFlow Model," 2020. <https://github.com/haroonshakeel/tensorflow-yolov4-tflite> (accessed Mar. 25, 2023).