

ANALYSIS AND IMPLEMENTATION OF YOLOV7 IN DETECTING PIN DEL IN REAL-TIME

Iustisia Natalia Simbolon^{*1}, Daniel Fernandez Lumbanraja², Kristina Tampubolon³

^{1,2,3}Informatics, Faculty of Informatics and Electrical Engineering, Institut Teknologi Del, Indonesia
Email: ¹iustisia.simbolon@del.ac.id, ²fdaniel029@gmail.com, ³kristinatampubolon45@gmail.com

(Article received: August 04, 2023; Revision: September 27, 2023; published: April 15, 2024)

Abstract

Real-time object detection is the process of identifying and tracking objects instantly and directly without any delay between image input and output. Carrying out real-time detection is a challenge in detection systems because it requires speed and accuracy of detection. This research proposes the application of the YOLOv7 algorithm which allows object localization and classification in one stage. This detection is carried out in real time on two objects, namely PinDel and Students. This research focuses on applying the YOLOv7 algorithm to detect real-time use of Pin Del by students. In this research, several hyperparameters were adjusted until the optimal value was found, including epoch with a value of 300, as well as confidence threshold, and IoU threshold with a value of 0.5. The model evaluation results from hyperparameter experiments show good results, with precision of 0.946, recall of 0.959, and mAP@0.5 of 0.977. This research has succeeded in detecting Pin Del objects in real time by obtaining a detection speed of between 7 and 40 FPS, which shows a fast response in detecting objects in real time. This research has contributed to the development of real-time object detection technology and its application in Pin Del use cases by students.

Keywords: Confidence Threshold, IoU Threshold, Real-time, Recall, YOLOv7.

ANALISIS DAN IMPLEMENTASI YOLOV7 DALAM MENDETEKSI PIN DEL SECARA REAL-TIME

Abstrak

Deteksi objek secara *real-time* adalah proses pengidentifikasian dan pelacakan objek secara instan dan langsung tanpa adanya penundaan antara input gambar dan output. Melakukan deteksi secara *real-time* menjadi tantangan dalam sistem deteksi karena membutuhkan kecepatan dan ketepatan deteksi. Penelitian ini mengusulkan penerapan algoritma YOLOv7 yang memungkinkan lokalisasi dan klasifikasi objek dalam satu tahapan. Pendeteksi ini dilakukan secara *real-time* terhadap dua objek, yaitu PinDel dan Mahasiswa. Penelitian ini berfokus pada penerapan algoritma YOLOv7 untuk mendeteksi penggunaan Pin Del oleh mahasiswa secara *real-time*. Dalam penelitian ini, beberapa *hyperparameter* diatur hingga ditemukan nilai optimal, termasuk *epoch* dengan nilai 300, serta *confidence threshold*, dan *IoU threshold* dengan nilai 0.5. Hasil evaluasi model dari eksperimen *hyperparameter* menunjukkan hasil yang baik, dengan *precision* sebesar 0.946, *recall* sebesar 0.959, dan *mAP@0.5* sebesar 0.977. Penelitian ini telah berhasil mendeteksi objek Pin Del secara *real-time* dengan diperolehnya kecepatan deteksi antara 7 hingga 40 FPS dimana hal ini menunjukkan *respons* yang cepat dalam mendeteksi objek secara *real-time*. Penelitian ini telah memberikan kontribusi pada pengembangan teknologi deteksi objek secara *real-time* dan aplikasinya pada kasus penggunaan Pin Del oleh mahasiswa.

Kata kunci: Confidence Threshold, IoU Threshold, Real-time, Recall, YOLOv7.

1. PENDAHULUAN

Deteksi objek adalah sebuah teknik dalam computer vision yang digunakan untuk mengidentifikasi dan melokalisasi objek dalam gambar ataupun video. Deteksi objek bertujuan untuk menemukan dan mengklasifikasikan objek yang ada di dalam sebuah gambar, dan melabelinya dengan *bounding boxes* untuk menunjukkan tingkat

confidences objek tersebut [1]. Ada banyak algoritma yang bisa digunakan untuk melakukan pendeteksian pada object secara *real-time* seperti CNN, R-CNN, Fast R-CNN, faster R-CNN dan juga YOLO. Algoritma *sliding windows* atau *two stage detection* seperti CNN, R-CNN, faster R-CNN melakukan konvolusi pada keseluruhan image sehingga waktu deteksi akan menjadi lama dan algoritma akan

kebingungan jika terdapat lebih dari satu objek dalam suatu gambar [2]. Ada dua tahapan terpisah yang dilakukan pada algoritma *sliding windows* yaitu identifikasi objek dan klasifikasi objek. Di sisi lain, YOLO yang merupakan algoritma *one stage detection* melakukan prediksi *bounding box* dan klasifikasi *class* dalam satu tahapan [2]. Dari ketiga algoritma *two stage detection* tersebut, algoritma *faster R-CNN* memiliki kecepatan 7 fps [3] yang merupakan kecepatan tertinggi dari ketiga algoritma tersebut.

Algoritma YOLO mampu mendeteksi objek secara *real-time* dengan baik. Pengembangan YOLOv4 meningkatkan nilai AP dan FPS YOLOv3 sebesar 10% dan 12% [3]. Hal ini berbeda dengan YOLO dan YOLOv2 tidak efektif dalam mendeteksi target yang kecil sehingga deteksi multi-skala ditambahkan pada YOLOv3 [4]. Pada Algoritma YOLOv4 yang diperkenalkan oleh Alexey Bochkovskiy memiliki kecepatan mencapai 45 fps [5]. Namun, pada Algoritma YOLOv7 merupakan salah satu algoritma pendeteksian objek yang dapat dilakukan secara *real-time* dengan efisiensi dan akurasi tinggi. Algoritma YOLOv7 mempunyai akurasi 56,8% dan mempunyai kecepatan deteksi yang tinggi, mencapai 5 – 160 FPS [6] bahkan dalam situasi di mana terdapat lebih dari satu objek dalam gambar.

Penelitian ini akan melakukan analisis dan implementasi deteksi objek secara *real time* dengan menggunakan algoritma YOLOv7. Pada penelitian ini objek yang akan dideteksi adalah Pin Del dan mahasiswa IT Del. Institut Teknologi Del atau yang biasa disingkat dengan sebutan IT Del merupakan kampus yang mewajibkan mahasiswa/i untuk menggunakan Pin Del saat masuk dan keluar kampus yang berguna untuk mengetahui bahwa Mahasiswa tersebut merupakan Mahasiswa Institut Teknologi Del. Penelitian ini akan mengimplementasikan algoritma YOLOv7 dalam pendeteksian objek serta diharapkan dapat dilakukan secara *real-time* dengan efisiensi dan akurasi tinggi, serta memungkinkan sistem untuk secara cepat mengidentifikasi dan memverifikasi penggunaan Pin Del oleh mahasiswa/i sebelum memasuki area kampus bahkan dalam situasi di mana terdapat lebih dari satu objek dalam gambar.

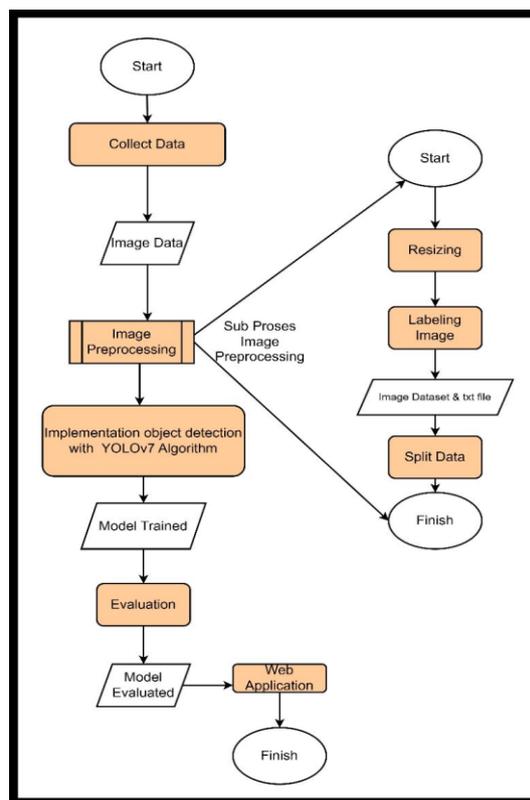
Penelitian ini juga akan melakukan analisis terhadap beberapa *hyperparameter* YOLOv7 untuk memastikan model yang dihasilkan memberikan deteksi yang optimal terhadap objek Pin Del dan mahasiswa secara bersamaan. Pengaturan *hyperparameter* akan disesuaikan dengan dataset yang digunakan dalam eksperimen guna memaksimalkan kinerja model dalam pendeteksian objek. *Hyperparameter tuning* dilakukan untuk mencari nilai akurasi model yang paling optimal dalam pendeteksian objek secara *real time*. Model dengan *hyperparameter* yang optimal akan diimplementasikan ke dalam sebuah aplikasi berbasis

website dan diharapkan dapat mengotomatisasi pendeteksian objek.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Tahapan yang dilakukan dalam mendeteksi objek secara *real-time* dengan YOLOv7 pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

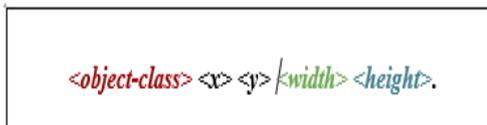
1. Pengumpulan data

Dataset yang digunakan dalam penelitian ini adalah data gambar mahasiswa IT Del beserta Pin Del. Gambar tersebut dikumpulkan melalui responden Google form yang dibagikan oleh peneliti kepada mahasiswa dan difoto secara langsung oleh peneliti. Tiga karakteristik dalam pengambilan data gambar yaitu lingkungan dan kondisi pencahayaan yang terang, gambar pin del dan mahasiswa tidak terhalang oleh objek lainnya, dan gambar tidak boleh samar atau *blur*. Pada proses pengumpulan data, data yang dihasilkan adalah data dengan format gambar .jpg, .jpeg dan .png. Variasi gambar yang dikumpulkan adalah gambar mahasiswa, gambar mahasiswa dengan pin Del dan gambar Pin Del. Hasil dari pengumpulan data diperoleh sebanyak 484 gambar dimana gambar-gambar ini akan di *preprocessing* sebelum digunakan pada model.

2. Image preprocessing

Tahap *preprocessing* gambar pada penelitian ini terdiri dari tahap *resizing*, *labelling image*, dan *split data*. Berikut penjelasan setiap tahapan dalam *preprocessing*:

- a. *Resizing* Gambar
Resizing gambar dilakukan untuk meningkatkan efisiensi waktu pemrosesan dalam pendeteksian dengan algoritma YOLOv7.
- b. Pelabelan Objek
Pelabelan objek atau disebut juga anotasi data adalah suatu tahapan yang dilakukan dengan memberikan keterangan pada setiap data yang ada untuk digunakan pada proses *training* dan pengujian. Pelabelan *image* dilakukan menggunakan tools *LabelImg*. Hasil dari pelabelan objek adalah file berupa format *.txt* seperti Gambar 2 berikut:



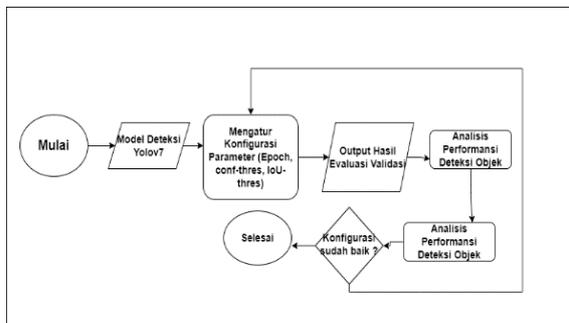
Gambar 2. Format.txt Anotasi Gambar

Keterangan :

object-class : id yang dianotasikan untuk tiap objek dan dimulai dari id 0
x dan y : titik tengah *bounding box*
width : Lebar *bounding box*
height : Tinggi *bounding box*

- c. Pembagian *Dataset*
Dataset akhir yang digunakan adalah dataset gambar dan hasil pelabelan yang dilakukan. Kemudian, dilakukan pembagian dataset yang bertujuan menghindari *overfitting* dan mengevaluasi model. Pembagian dataset pada penelitian ini dilakukan menggunakan metode *Hold-Out* yang melibatkan pembagian dataset menjadi subset yang berbeda yaitu membagi dataset menjadi data *training*, *validation* dan data *testing*.
- ### 3. Implementasi deteksi objek dengan algoritma YoloV7

Pada implementasi penelitian ini dilakukan beberapa eksperimen ataupun percobaan dalam mendeteksi objek dengan melakukan *hyperparameter tuning*. Rancangan eksperimen dapat dilihat pada Gambar 3.



Gambar 3. Rancangan Implementasi Algoritma YOLOv7

Proses eksperimen *hyperparameter* dimulai dengan tahapan *training* data hingga mendapatkan model deteksi. Setelah melakukan *training* data yang

juga melakukan eksperimen terhadap *epoch* kemudian dilanjutkan dengan proses evaluasi model dengan evaluasi pada data validasi yang telah disiapkan sebelumnya. Tahapan validasi data dilakukan dengan menentukan *hyperparameter* yang sesuai yaitu *confidence threshold* dan *IoU threshold*. Tahap *decision* atau keputusan, memperhatikan evaluasi data validasi dimana ketika hasil *mean Average Precision* (mAP) yang diperoleh sudah tinggi dan stabil maka akan dilakukan kembali evaluasi dengan nilai *hyperparameter* lain hingga menghasilkan hasil yang optimal.

4. Evaluasi dan pembangunan aplikasi

Evaluasi dilakukan dengan metrik deteksi objek, yaitu *Precision Recall* serta mAP. Evaluasi ini juga digunakan untuk menentukan nilai *hyperparameter* yang terbaik. Model dengan hasil parameter yang optimal akan diimplementasikan ke dalam aplikasi berbasis website. Pembangunan aplikasi dilakukan menggunakan *flask* berdasarkan hasil dari evaluasi model terbaik.

2.2. YOLOv7

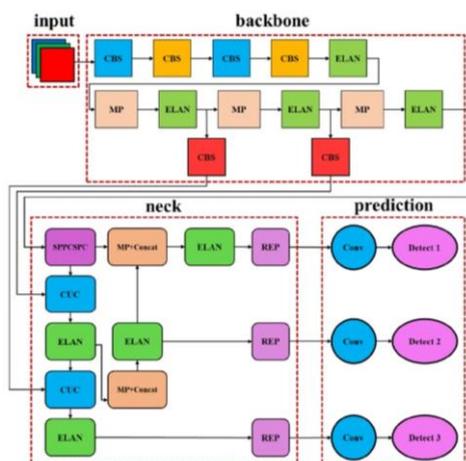
Deep learning memungkinkan komputer untuk belajar mengklasifikasikan gambar, audio, teks atau video [7]. YOLOv7 merupakan bagian dari *deep learning* dan merupakan algoritma one-stage detection yang melakukan deteksi lokasi objek dan klasifikasi objek dalam satu tahapan [2]. YOLOv7 bekerja dengan melihat seluruh area gambar sekali, lalu melewati jaringan saraf sekali langsung mendeteksi objek yang ada. Berikut adalah langkah-langkah deteksi objek dengan YOLOv7:

- a. Membagi citra masukan menjadi *grid* berukuran $S \times S$ [8].
- b. Setelah membagi citra masukan menjadi *grid*, setiap *grid* memprediksi *bounding box*, koordinat x, y, lebar, tinggi beserta skor kepercayaannya [9]. Skor kepercayaan menggambarkan keyakinan algoritma bahwa *bounding box* yang diprediksi berisi objek yang ingin di deteksi [9].
- c. Untuk *filter bounding box* yang dihasilkan, sebuah ambang batas ditentukan untuk mempertahankan deteksi objek dengan skor kepercayaan yang lebih tinggi dan mengabaikan yang di bawah ambang batas tersebut. Hanya *bounding box* dengan skor kepercayaan yang melebihi ambang batas yang dipertahankan.
- d. Selanjutnya, IoU dihitung untuk mengukur tumpang tindih antara *bounding box* prediksi dan *ground truth* [10].
- e. *Non Maximal Suppression (NMS)* kemudian diterapkan untuk mengatasi *bounding box* yang tumpang tindih. Jika terdapat beberapa *bounding box* yang memprediksi objek yang sama, algoritma YOLOv7 akan memilih *bounding box* dengan skor kepercayaan tertinggi serta menghapus *bounding box* yang diprediksi.

- f. Hasil akhir dari deteksi objek adalah *bounding box* pada objek inputan citra gambar yang menunjukkan lokasi deteksi objek beserta label sebagai identifikasi objek tersebut [11].

YOLO v7 adalah *single-stage real-time object detector* yang diperkenalkan pada Juli 2022. Meskipun YOLOv7 memiliki kemampuan untuk dengan cepat mengenali objek dalam gambar, tetapi algoritma YOLOv7 memiliki keterbatasan dan tantangan dalam melakukan lokalisasi yang akurat untuk beberapa objek, terutama objek yang berukuran kecil [12].

Arsitektur Algoritma YOLOv7 terdiri dari tiga bagian utama, yaitu: *Backbone*, *Neck* dan *Head* (*Prediction/Dense Prediction*). Gambar 4 merupakan tampilan Arsitektur YOLOv7.



Gambar 4. Arsitektur YOLOv7 [13]

1. Backbone

Backbone adalah arsitektur yang digunakan untuk mengekstrak fitur penting dari gambar. Bagian Backbone terdiri dari beberapa modul CBS, Modul E-ELAN dan modul MP [14]. Modul CBS (*Convolutional-BatchNorm-SiLU activation functions*) adalah komponen yang digunakan dalam sistem untuk mengekstraksi fitur gambar pada berbagai skala. Modul CBS terdiri dari tiga lapisan utama:

- Convolution Layer**
Lapisan ini bertanggung jawab untuk melakukan operasi konvolusi pada input gambar.
- Batch Normalization Layer (BN)**
Lapisan normalisasi batch digunakan untuk meningkatkan kestabilan dan kecepatan pelatihan jaringan.
- SiLU activation Function**
Fungsi aktivasi *Sigmoid-Weighted Linear Unit* digunakan untuk memberikan respons non-linear pada fitur-fitur yang diekstraksi oleh lapisan konvolusi. Fungsi ini akan memberikan informasi rentang 0 dan 1 sebagai probabilitas keberadaan suatu fitur dalam gambar.

E-ELAN merupakan blok komputasi pada *backbone* YOLOv7 untuk mengarah pada peningkatan kecepatan dan akurasi. Modul E-ELAN terdiri dari beberapa E-LAN diusulkan karena menggunakan *expand, shuffle, merge cardinality* serta *stack in computational block* untuk mencapai kemampuan dan terus meningkatkan kemampuan pembelajaran jaringan tanpa merusak jalur gradien asli. Strategi E-ELAN adalah menggunakan konvolusi grup untuk memperluas saluran dan kardinalitas blok komputasi. Modul MP menambahkan lapisan maxpool berdasarkan modul CBS yang membentuk cabang atas dan cabang bawah. Cabang atas membagi panjang dan lebar gambar menjadi separuh melalui *maxpool*. Lapisan CBS kedua membagi separuh panjang dan lebar menjadi separuh dan akhirnya menggunakan operasi *Concat* (menggabungkan) [15]

2. Neck

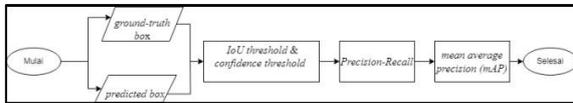
Setelah dari backbone, di *layer Neck* fitur gambar digabungkan dari berbagai level resolusi untuk kemudian diteruskan ke prediksi dengan menggunakan modul SPPCSPC (*Spatial Pyramid Pooling and Concatenation*). Proses ini bertujuan memadukan informasi dari berbagai tingkat detail dalam fitur-fitur. *Feature map* yang dihasilkan ada dua bagian. *Feature map* yang pertama mengimplementasikan tiga operasi konvolusi dan tiga operasi *max pooling* untuk memisahkan tiga set *feature map* dengan ukuran yang berbeda guna mendapatkan beberapa fitur yang beragam. Kemudian *feature map* tersebut digabungkan dan melakukan konvolusi yang kedua untuk mendapatkan *feature map* yang pertama. *Feature map* yang kedua digunakan untuk mengimplementasikan operasi konvolusi konvolusi kembali. Terdapat juga lapisan CUC (*Combining Unit of Convolution*) yang merupakan unit dasar dari penggabungan *feature map* yang mencakup konvolusi, *upsampling* dan concat [13].

3. Head

Head atau disebut juga *dense prediction* mengambil fitur dari *Neck* dan kemudian melakukan lokalisasi dan klasifikasi dalam waktu yang sama. *Dense prediction* adalah prediksi akhir yang terdiri atas skor kepercayaan prediksi, kelas probabilitas, vektor yang berisikan koordinat *bounding box* yang diprediksi (titik tengah, tinggi dan lebar). Bagian *Head* menyesuaikan jumlah saluran gambar untuk tiga skala berbeda dari keluaran *Neck* melalui struktur RepVGG Block (REP), kemudian melalui konvolusi 1×1 digunakan untuk memprediksi kepercayaan, kategori dan *bounding box*. Hasil *output feature map* ada 3 ukuran yang dapat digunakan untuk melakukan deteksi untuk objek yang memiliki variasi ukuran [15].

2.3. Matrix Evaluasi

Tahapan evaluasi hasil deteksi penggunaan PinDel oleh Mahasiswa dapat dilihat pada Gambar 5 berikut:



Gambar 5. Rancangan Implementasi Algoritma YOLOv7

Berdasarkan Gambar 5, hasil implementasi deteksi dengan algoritma YOLOv7, dengan memanfaatkan *ground-truth box* yang merupakan hasil anotasi dan hasil deteksi yang disebut *predicted box* maka dilakukan evaluasi deteksi objek dengan menentukan *IoU threshold* beserta *confidence threshold*. Evaluasi dilakukan dengan menghitung nilai *Precision-Recall* dan *mean average precision* (mAP).

Berikut matriks evaluasi yang digunakan dalam penelitian ini [16]:

a. *Precision*: digunakan untuk menghitung hasil yang diperkirakan positif, berapa banyak yang benar-benar positif.

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

b. *Recall*: penentuan model seberapa baiknya dalam mencari nilai yang positif.

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

c. *mean Average Precision*: rata-rata dari average precision pada setiap kelas objek yang dideteksi.

$$mAP = \sum_{i=1}^n \frac{AP(i)}{N} \times 100\% \quad (3)$$

2.4. Hyperparameter

Pada penelitian ini terdapat *hyperparameter* yang harus ditentukan sebelum sebuah model akan di *training* untuk mendeteksi objek. Untuk mendapatkan nilai *hyperparameter* yang optimal, diperlukan melakukan eksperimen pada beberapa kombinasi nilai *hyperparameter*. Berikut *hyperparameter* yang digunakan dalam YOLOv7 untuk penelitian ini:

a. **Batch Size**

Batch size adalah *hyperparameter* yang digunakan sebagai jumlah contoh perdata yang digunakan dalam satu iterasi (step) pelatihan [17]. Dalam penelitian ini, *batch size* yang akan digunakan dalam tiap eksperimen adalah nilai *default* 16.

b. **Epoch**

Epoch merupakan *hyperparameter* yang menentukan berapa kali keseluruhan dataset training akan dijalankan [17].

c. **Confidence Threshold**

Confidence threshold (ambang kepercayaan) adalah batas atau ambang nilai yang digunakan untuk

menyaring deteksi objek berdasarkan tingkat kepercayaan (*confidence*) oleh model deteksi YOLOv7 [5].

d. **IoU Threshold**

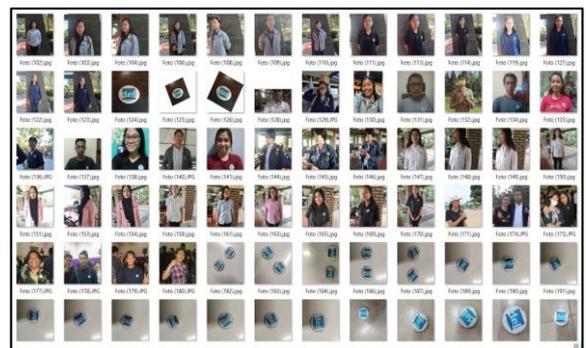
IOU threshold (ambang IoU) menentukan batas *bounding box* dianggap sebagai deteksi yang benar. Nilai *default threshold* pada algoritma YOLOv7 adalah 0,5 [5]. Namun, nilai *threshold* yang optimal berbeda-beda tergantung pada kebutuhan aplikasi dan karakteristik dataset yang digunakan untuk pelatihan.

3. HASIL DAN PEMBAHASAN

Pada bagian ini diuraikan mengenai hasil dari penelitian beserta pengujian yang telah dilakukan.

3.1. Dataset

Dataset gambar yang terkumpul dari google form dan di foto secara langsung berjumlah 484 gambar yang dalam gambar tersebut terdiri dari 242 *class* objek pin Del dan 242 *class* objek mahasiswa. Contoh dataset gambar dapat dilihat pada Gambar 6.



Gambar 6. Dataset

3.2. Image Preprocessing

Sebelum melakukan *training data*, perlu dilakukan *image processing*. *Image processing* mencakup *resizing*, pelabelan, serta pembagian *dataset*.

1. **Resizing gambar**

Data gambar yang dikumpulkan akan di *resize* supaya seragam dan memenuhi kebutuhan *input-an* gambar pada YOLOv7 menjadi ukuran 640 x 640 piksel. Proses ini juga untuk mempercepat waktu komputasi pada proses deteksi objek.

2. **Pelabelan Objek**

Hasil gambar yang sudah di-*resize* ukurannya menjadi 640 x 640 piksel, kemudian dilabeli menggunakan LabelImg. Terdapat 2 objek yang diberi label pada penelitian ini. Objek Pin Del dengan ID class 0 dan Objek Mahasiswa dengan ID class 1. Pelabelan ini dilakukan dengan menandai objek PinDel dan Mahasiswa dengan kotak persegi panjang yang dapat dilihat seperti pada Gambar 7.



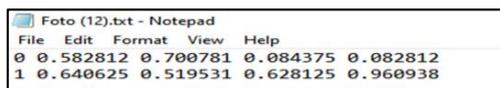
Gambar 7. Proses Pelabelan Gambar

Hasil dari pelabelan dataset adalah file *classes.txt* dan *file.txt*. File *classes.txt* yang dapat dilihat pada Gambar 8 berisi daftar kelas objek yang akan digunakan dalam proses *training*.



Gambar 8. Daftar Class

File.txt hasil pelabelan tiap gambar yang dapat dilihat pada Gambar 9.



Gambar 9. Hasil Pelabelan Gambar

Hasil proses pelabelan ini akan digunakan sebagai dataset. Dataset yang digunakan pada penelitian ini adalah data gambar format .jpg, .jpeg dan .png, serta data hasil *labeling* file format .txt.

3. Pembagian Dataset

Split dataset yang dilakukan berguna untuk membagi dataset menjadi 3 bagian, yaitu data *training*, data *validation* serta data *test*. Pada penelitian ini, pembagian dataset yang digunakan adalah 70 : 20 : 10, yang merupakan rasio pembagian dataset yang optimal sesuai dengan penelitian yang pernah dilakukan, dengan hasil Accuracy (mAP) 0.9951 dan hasil training time 15 menit 55 detik [18]. Pembagian dataset pada penelitian ini yang menghasilkan pembagian dataset seperti pada Tabel 1.

Tabel 1. Distribusi Pembagian Dataset

Pembagian	Jumlah Data Gambar	Jumlah Class	
		Pin Del	Mahasiswa
Train 70%	338	294	294
Val 20%	96	108	73
Test 10%	50	63	22

3.3. Evaluasi Model

Dalam menentukan nilai untuk beberapa *hyperparameter* yang akan digunakan, dilakukan beberapa percobaan pembentukan model dengan

beberapa nilai *epoch* seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Hasil Percobaan Epoch

Epoch	Evaluasi Model		
	Precision	Recall	mAP@.5
50	0,9503	0,9603	0,9771
100	0,9987	0,9521	0,9851
150	0,9681	0,9931	0,9913
200	0,9812	0,9611	0,9861
250	0,991	0,9406	0,9856
300	0,9881	0,9625	0,9933

Hasil evaluasi model menggunakan *precision*, *recall*, dan *mAP@.5*, diperoleh nilai epoch terbaik adalah 300. Pada *Epoch* 300, diperoleh *mAP@.5* adalah 0,9933. Nilai *mAP@.5* menunjukkan nilai yang IoU gunakan yaitu 0,5 yang merupakan nilai *default* IoU. Oleh karena itu, dalam penelitian ini digunakan *epoch* 300 dalam pembentukan model (*training*) dengan YOLOv7.

Setelah melakukan *training* dan mendapatkan hasil model yang paling sesuai, selanjutnya dilakukan evaluasi model pada data validasi yang sudah disiapkan sebelumnya. Dalam evaluasi model ini, dilakukan pengaturan *hyperparameter* berupa IoU dan *Confidence Threshold*. Merujuk pada Tabel 1 data validasi yang digunakan berjumlah 96. Hasil deteksi pada data validasi masih diperoleh kesalahan deteksi berupa False Positive yang mendeteksi background menjadi PinDel seperti ditunjukkan pada Gambar 10 ketika tidak memberikan konfigurasi pada deteksi objek.



Gambar 10. Kesalahan Deteksi Data Validasi

Hasil deteksi kesalahan yang ditemukan memiliki *confidence score* 0,40 dan 0,21. Oleh karena itu, digunakan *confidence threshold* 0,5 yang berarti objek yang akan terdeteksi hanyalah objek yang memiliki *confidence* lebih tinggi atau sama dengan 0,5, selebihnya diabaikan. Sehingga, dengan menggunakan *hyperparameter epoch* 300, IoU dan *confidence threshold* bernilai 0,5 diperoleh hasil evaluasi yang dapat dilihat dalam Tabel 3 untuk tiap objek.

Tabel 3. Hasil Evaluasi Data Validasi

Class	Precision	Recall	mAP@.5
Pin Del	0,991	0,991	0,995
Mahasiswa	0,946	0,959	0,959

Berdasarkan Tabel 3 untuk kedua objek yaitu mahasiswa dan PinDel diperoleh *precision* sebesar 0.968, *Recall* 0.975, *mAP@.5* mencapai 0,977.

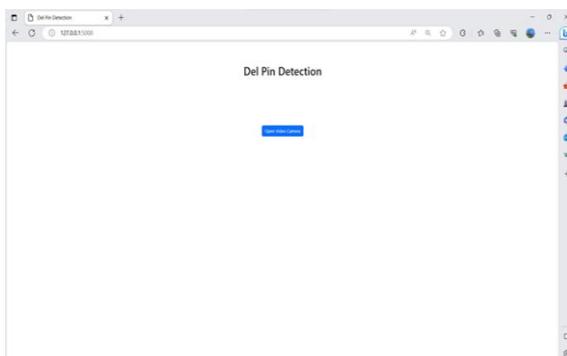
Setelah melakukan eksperimen terhadap *hyperparameter* di data validasi, nilai *hyperparameter* yang sudah mencapai batas optimum digunakan pada *testing*. Dengan menggunakan *epoch* 300 serta *IoU-threshold* dan *confidence threshold* sebesar 0,5, berikut hasil *testing* model dapat dilihat pada Gambar 11.



Gambar 11. Hasil *Testing* Deteksi Objek dengan YOLOv7

3.4. Hasil Implementasi aplikasi

Pada pengembangan aplikasi, aplikasi dikembangkan dalam bentuk website dengan menggunakan *framework Flask* dikarenakan konvensi dalam pemanggilan perintah pada sebuah aplikasi web terhadap server ditulis dalam bahasa pemrograman python [19]. Penggunaan *flask* juga memudahkan pengembang dalam mengatur *behaviour* dari suatu web serta membuat web menjadi lebih terstruktur [20]. Pada aplikasi, terdapat halaman home/beranda seperti Gambar 12.



Gambar 12. Halaman Beranda Aplikasi

Halaman tersebut terdapat sebuah *title* Del Pin Detection dan sebuah *button* Open Video Camera. *Button* tersebut akan mengarah ke arah kamera yang dapat mendeteksi Objek Mahasiswa dan Objek Pin Del beserta *confidence score* dan nilai FPS(*Frame Per Second*) seperti Gambar 13.



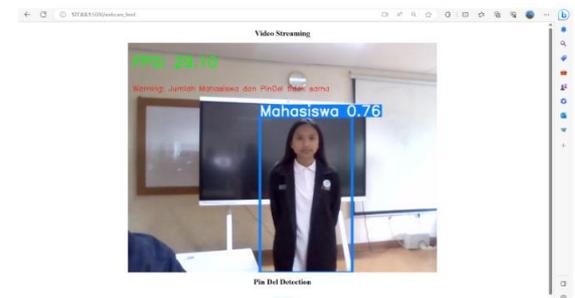
Gambar 13. Hasil Deteksi Objek secara Real-time

Gambar 13 menampilkan hasil deteksi objek yang dilakukan secara *real-time*. Terdapat 2 objek yang terdeteksi Pin Del dan 2 objek Mahasiswa yang terdeteksi. Tiap *bounding box* pada objek yang terdeteksi, terdapat nama objek dan nilai *confidence score*, contohnya pada objek mahasiswa memiliki *confidence score* 0.95 dan PinDel 0.82 beserta nilai fps dari deteksi yang dilakukan 26.61 fps.

Pada aplikasi tersebut, jumlah objek di antara Mahasiswa dan PinDel haruslah sama, Apabila jumlah objek tersebut tidak sama, maka akan menimbulkan alarm dan warning yang menandakan bahwa terdapat mahasiswa tidak memakai pin del atau pin del yang terdeteksi namun tidak dikenakan oleh mahasiswa seperti Gambar 14.



Gambar 14. Hasil Deteksi dengan Warning



Gambar 15. Hasil Deteksi *Real-time* Jarak Jauh.

4. DISKUSI

Pada Penelitian yang dilakukan sebelumnya [6] tingkat akurasi deteksi objek dengan YOLOv7 mencapai 56,8% serta mempunyai kecepatan deteksi yang mencapai 5 - 160 FPS. Pada penelitian ini ada beberapa tahapan yang dilakukan dalam mendeteksi objek menggunakan YOLOv7. Sebelum melakukan proses implementasi deteksi objek menggunakan

algoritma YOLOv7, Dataset yang digunakan berjumlah 484 gambar. Dataset tersebut dilakukan *resize* dengan ukuran 640 x 640 piksel. Setelah melakukan *resize* gambar, proses *labelling* terhadap objek gambar akan berupa *output* dalam file txt, yang berisi id terhadap label Mahasiswa yaitu 0 sedangkan id terhadap PinDel yaitu 1, sedangkan untuk nilai *x*, *y*, *weight*, dan *height*, tergantung pada letak, tinggi, dan lebar *bounding box label*. Setelah melakukan pelabelan pada 484 data gambar dihasilkan 465 label untuk *class* PinDel dan 389 label untuk *class* Mahasiswa sehingga dataset yang diperoleh adalah 484 data gambar berformat .png, .jpg dan .jpeg serta 854 data hasil label yang sudah diberi *class*. Setelah gambar tersebut di label, data akan di-*split* dengan perbandingan *Training:Validation:Testing* bernilai 70:20:10.

Dalam proses mencari model yang optimal, dilakukan dengan beberapa percobaan terhadap *hyperparameter*. Percobaan *hyperparameter* yang telah dilakukan dalam penelitian ini terdiri dari: *Epoch*, *Confidence threshold*, serta *IoU threshold* yang bernilai, *epoch*: 300, *confidence threshold* dan *IoU threshold* bernilai 0.5. Hasil evaluasi deteksi yang dilakukan didapatkan nilai mAP@.5 PinDel 0.995 dan mAP@.5 Mahasiswa 0.959, serta kecepatan deteksi secara real-time yang didapatkan mencapai 7 hingga 40 FPS. Hasil model yang optimal tersebut digunakan untuk pengembangan aplikasi dalam bentuk website yang dapat membantu untuk mendeteksi objek PinDel dan Mahasiswa. Adanya peningkatan terhadap nilai akurasi tersebut dikarenakan adanya evaluasi matrix yang dilakukan untuk mencari model yang optimal, serta nilai kecepatan deteksi masih berada dalam rentang nilai kecepatan dari penelitian sebelumnya [6]. Berdasarkan hasil penelitian ini kelebihan algoritma YOLOv7 dalam mendeteksi objek Pin Del memiliki kecepatan komputasi yang baik dengan mampu mendeteksi objek hingga 40 FPS. Namun, pada penelitian ini YOLOv7 memiliki kekurangan yaitu memiliki keterbatasan terhadap jarak, ukuran, resolusi gambar yang diinputkan pada kamera. Pada Gambar 15 dapat dilihat bahwa objek mahasiswa yang cenderung lebih besar lebih mudah dikenali dan dilakukan deteksi dibandingkan Pin Del yang cenderung lebih kecil.

5. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan Algoritma YOLOv7 untuk mendeteksi secara *real-time* objek Pin Del dan Mahasiswa di IT Del. Melalui eksperimen, *hyperparameter* dievaluasi untuk mencari nilai optimal, dan hasil evaluasi model menunjukkan *Precision* sebesar 0.946, *Recall* sebesar 0.959, dan mAP@.5 sebesar 0.977 dengan pengaturan *epoch* sebanyak 300 serta *confidence threshold* dan *IoU threshold* bernilai 0.5. Selain itu, diperoleh kecepatan deteksi antara 7 hingga 40 FPS dimana hal ini

menunjukkan respons yang cepat dalam mendeteksi objek secara *real-time*. Pada penelitian selanjutnya dapat melakukan pengembangan dengan menggunakan yolo versi terbaru untuk meningkatkan akurasi dalam mendeteksi objek. Hal itu dikarenakan model YOLOv8 lebih baik dalam mendeteksi benda kecil dibandingkan YOLOv7.

DAFTAR PUSTAKA

- [1] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [2] S. N, A. M. P, and H. P. V, "Object Detection using YOLO And Mobilenet SSD: A Comparative Study," *Int. J. Eng. Res. Technol.*, vol. 11, no. 6, pp. 134–138, 2022.
- [3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv Comput. Vis. Pattern Recognit.*, 2020.
- [4] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2021.
- [5] D. W. Effendi, "Perbandingan Optimasi Sgd Dan Adam Pada Arsitektur Yolov5 (You Only Look Once) Untuk Deteksi Alat Pelindung Diri," *UPN "Veteran" Jawa Timur*, no. 69, pp. 5–24, 2022.
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1–15, Jul. 2022.
- [7] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) pada Ekspresi Manusia," *Algor*, vol. 1, pp. 12–20, 2020.
- [8] A. Amwin, "Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO)," Universitas Islam Indonesia, 2021.
- [9] M. . Zayed, A. Amin, and S. Rahman, "Real-time Detection and Recognition of Traffic Signs in Bangladesh using YOLOv3 Detector," 2021.
- [10] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021.
- [11] C. Gerald and C. Lubis, "Pendeteksian Dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once Dan Convolutional Neural Network," *J. Ilmu*

- Komput. dan Sist. Inf.*, vol. 8, no. 2, p. 197, 2020.
- [12] P. D. Varagnolo, "Computer Vision-based detection and tracking of fish," 2022.
- [13] B. R. Chang, H. Tsai, and C. Hsieh, "Accelerating the Response of Self-Driving Control by Using Rapid Object Detection and Steering Angle Prediction," *ELectronics*, vol. 12, p. 2161, 2023.
- [14] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," *arXiv Comput. Vis. Pattern Recognit.*, 2022.
- [15] Z. Qiu, H. Bai, and T. Chen, "Special Vehicle Detection from UAV Perspective via YOLO-GNS Based Deep Learning Network," *Drones*, vol. 7, no. 2, p. 117, 2023.
- [16] H. Hammam, A. Asyhar¹, S. A. Wibowo², and G. Budiman³, "Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video Implementation and Performance Analysis of You Only Look Once (Yolo) Method As Porn Censorship in Video," *e-Proceeding Eng.*, vol. 7, no. 2, pp. 3631–3638, 2020.
- [17] J. Brownlee, "What is the Difference Between a Batch and an Epoch in a Neural Network?," *Mach. Learn. Mastery*, no. July, pp. 3–4, 2018.
- [18] Y. C. How, A. F. Ab. Nasir, K. F. Muhammad, A. P. P. Abdul Majeed, M. A. Mohd Razman, and M. A. Zakaria, "Glove Defect Detection Via YOLO V5," *Mekatronika*, vol. 3, no. 2, pp. 25–30, 2022.
- [19] A. Kumar Siliwangi and D. Prabowo, "Pencarian Informasi Berbasis Teks dalam Komik Digital Menggunakan OCR," vol. 8, no. 2, 2022.
- [20] J. A. Saputra and Y. A. Susetyo, "DEVELOPMENT OF PRICE DISTRIBUTION MODULE ON MERCHANDISE APPLICATION USING FLASK FRAMEWORK AT PT XYZ ", *J. Tek. Inform. (JUTIF)*, vol. 3, no. 5, pp. 1293-1300, Oct. 2022.