

Penggunaan Metode *You Only Look Once* dalam Penentu Pindah Tanaman Cabai Besar Ternotifikasi Telegram

Feli Ramasari^{1*}, Firdaus², Sri Nita³ dan Kartika⁴

¹²³ Jurusan Teknik Elektro Politeknik Negeri Padang, Jl. Limau Manih Padang, 25164, Indonesia

⁴ Jurusan Teknik Elektro Universitas Malikussaleh, Jl. Tengku Chik Ditiro Lancang Garam, 24351, Indonesia

*email:ramasarifeli@gmail.com, mrdauz@yahoo.com

Abstrak— Tanaman cabai besar (*Capsicum Annuum L.*) adalah salah satu komoditi sayuran yang mempunyai nilai ekonomi tinggi dalam pemenuhan kebutuhan domestik Indonesia dengan luas panen cabai sekitar 11.400 ha per bulannya. Salah satu tahapan dalam budidaya tanaman cabai adalah proses pindah tanam setelah penyemaian yaitu kondisi bibitnya telah memiliki 4 – 5 helai daun permanen. Dampak apabila tidak segera dipindahkan adalah tanaman tidak memiliki cukup masa adaptasi dalam menyelesaikan pertumbuhan vegetatifnya sehingga tanaman cepat menua dan memasuki stadia generatif. Pengolahan citra digital dapat diaplikasikan ke dalam sistem identifikasi dan perhitungan daun permanen tanaman cabai besar dengan pendekatan *deep learning*. Salah satu metode yang dapat digunakan adalah *You Only Look Once* versi 4 yaitu pengembangan jaringan saraf konvolusional dalam pendeteksian objek berupa hasil prediksi kotak pembatas dengan *backbone CSPdarknet53*. Model YOLOv4 diintegrasikan dengan mini komputer *Raspberry Pi 4 model B* dan pengiriman notifikasi melalui fitur Bot pada aplikasi Telegram. Dataset citra yang digunakan adalah 135 buah dengan 1409 sampel kelas permanen dan 1945 sampel untuk kelas muda. Sistem ini berhasil diterapkan dengan nilai keakuratan identifikasi dan perhitungan yaitu 92,85% serta nilai *inference time* pendeteksian satu buah tanaman adalah 11,88 detik dan pada pengiriman notifikasi adalah 25,29 detik.

Kata Kunci: YOLOv4, *Raspberry Pi*, Telegram, kotak pembatas, daun permanen

Abstract— *Capsicum Annuum L.* is one of the vegetable commodities that have high economic value in fulfilling Indonesia's domestic needs with an area of chili harvest of around 11,400 ha per month. One of the stages in the cultivation of chili plants is the process of transplanting after sowing, namely the condition of the seeds having 4-5 permanent leaves. The impact if it is not immediately transferred is that the plant does not have enough adaptation period in completing its vegetative growth so that the plant quickly ages and enters the generative stage. Digital image processing can be applied to the identification and calculation of permanent leaves of large chili plants with a deep learning approach. One method that can be used is *You Only Look Once* version 4, which is the development of a convolutional neural network in object detection in the form of boundary box prediction results with a *CSPdarknet53* backbone. The YOLOv4 model is integrated with the *Raspberry Pi 4 Model B* mini computer and sends notifications via the Bot feature in the Telegram application. The image dataset used is 135 pieces with 1409 samples for permanent class and 1945 samples for the young class. This system was successfully implemented with the identification and calculation accuracy value of 92.85% and the inference time value for the detection of one plant was 11.88 seconds and for sending notifications was 25.29 seconds.

Keywords: YOLOv4, *Raspberry Pi*, Telegram, bounding boxes, permanent leaves

© 2021 Elektron Jurnal Ilmiah

I. PENDAHULUAN

Metode *You Only Look Once* (YOLO) adalah salah satu metode yang digunakan untuk memprediksi jenis dan letak objek pada sebuah gambar secara *real-time* dengan menggunakan jaringan konvolusi (*convolution network*). Metode ini dapat memprediksi memprediksi kotak-kotak pembatas (*bounding boxes*) dan kelas probabilitas untuk setiap kotak sehingga dapat menghasilkan prediksi objek yang akurat [1]. Metode ini merupakan hasil dari perkembangan teknologi dalam bidang *computer vision* sehingga memungkinkan perangkat untuk mendeteksi, mengolompokkan dan menghitung sendiri objek pada gambar. Selanjutnya, data hasil dapat dimanfaatkan diberbagai bidang

kehidupan dan bisa dikombinasikan dengan sistem tertanam untuk pengaplikasian secara langsung.

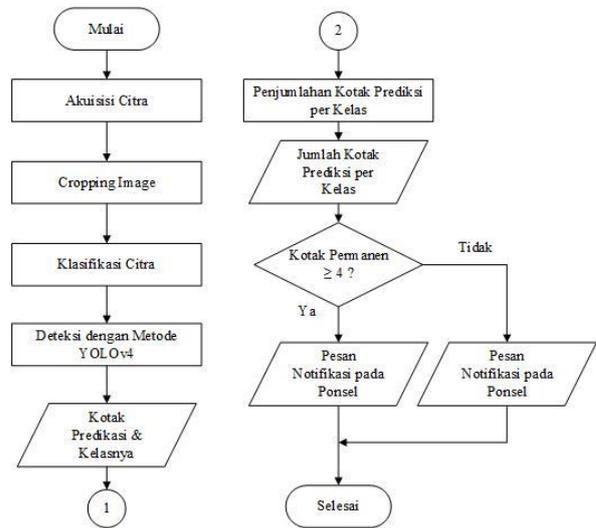
Penggunaan metode YOLO telah banyak diimplementasikan pada beberapa bidang contohnya pada tahun 2018, C. S. Asha dan A. V. Narasimhadhan membuat sistem penghitung kendaraan untuk mengestimasi trafik jalanan secara *real-time* [2]. Masih pada bidang yang sama, pada tahun 2019, Huansheng Song et al membuat sistem deteksi dan penghitungan kendaraan menggunakan YOLO versi 3 yang hasil eksperimennya membuktikan bahwa metode deteksi dan pelacakan kendaraan yang diusulkan untuk adegan video pengawasan jalan raya memiliki kinerja dan kepraktisan yang baik dibandingkan dengan metode tradisional dalam memantau lalu lintas kendaraan

dengan perangkat keras [3]. Pada bidang peternakan ikan juga telah dilakukan pengimplementasian YOLO yang digabungkan dengan algoritma *Multi-Scale Retinex* (MSR) tahun 2020 yang mencapai akurasi maksimum untuk mendeteksi ukuran ikan, menghitung jumlah dan fitur gerakan pada kolam ikan dalam air keruh [4]. Pada tahun yang sama, Kenta Itakura dan Fumiki Hosoi mengaplikasikan metode YOLO versi 2 dalam pengukuran pohon menggunakan kamera 360° yang menghasilkan perkiraan otomatis beberapa parameter struktur pohon dan berkontribusi pada pengukuran pohon yang lebih efisien [5]. Tidak hanya pada tanaman pohon, tanaman *Arabidopsis Thaliana* pun dijadikan sebagai objek penelitian pada tahun 2020 dengan menciptakan perangkat robotik darat otonom yang mampu mendeteksi dan menghitung daun hampir secara *real-time* menggunakan metode Tiny-YOLO versi 3 dan Open CV [6]. Berdasarkan beberapa implementasi diatas, YOLO banyak digunakan untuk pendeteksian, pengukuran dan perhitungan objek dengan data uji berupa video atau gambar dengan beberapa metode atau algoritma pendukung yang sesuai.

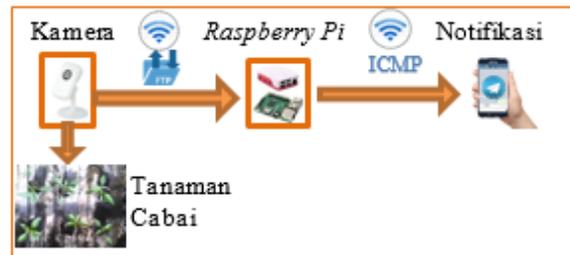
Metode *You Only Look Once* dengan keunggulannya dapat digunakan dalam bidang pertanian dengan harapan bahwa hal ini dapat membuat peningkatan keefektifan dan hasil panen nantinya. Salah satu komoditi sayuran pada bidang pertanian yang mempunyai nilai ekonomi tinggi karena peranannya yang cukup besar dalam pemenuhan kebutuhan masyarakat domestik sebagai bahan makanan maupun bahan baku olahan adalah tanaman cabai besar (*Capsicum Annuum L.*) [7]. Pemenuhan kebutuhan masyarakat Indonesia terhadap tanaman cabai besar diperlukan luas panen cabai sekitar 11.400 ha per bulannya [7]. Pada proses budidaya tanaman cabai diperlukan tahap penyemaian agar mengurangi risiko kematian bibit pada lahan pertanian [8] yang tentunya akan mengurangi produktivitas panen. Proses setelah penyemaian adalah pemindahan bibit cabai dari medium penyemaian ke lahan pertanian terbuka setelah bibit berumur sekitar 1 bulan hingga bibit memiliki 4 – 5 helai daun permanen. Proses pemindahan ini harus segera dilakukan setelah tanaman memasuki umur pindah tanamnya karena akan mempercepat adaptasi tanaman terhadap lingkungan, sehingga pertumbuhan tanaman tidak terhambat dan dapat menghasilkan bagian vegetatif yang lebih baik. Apabila proses pemindahan telambat dari umur pindah yang dapat disebabkan oleh faktor kelalaian manusia maka tanaman tidak memiliki masa adaptasi yang cukup dan menyelesaikan pertumbuhan vegetatifnya, sehingga tanaman menjadi cepat menua dan cepat memasuki stadia generatif [8]. Oleh sebab itu diperlukannya ketepatan waktu pindah untuk menghilangkan risiko pendeknya umur panen tanaman cabai karena tanaman menjadi cepat menua yang akan berefek terhadap jumlah panennya.

Pentingnya memperhatikan umur pindah pada tanaman cabai (*Capsicum Annuum L.*) diperlukan

pemantauan kondisi bibit siap pindah ke lahan yaitu telah memiliki 4 – 5 helai daun permanen dan pemilihan tanaman yang akan dipindahkan [8]. Pada kasus ini, pengoptimalan pemantauan untuk mengurangi faktor kelalaian manusia dapat dilakukan dengan mengkombinasikan keunggulan dari metode *You Only Look Once* dan fasilitas komunikasi data nirkabel sebagai penyampaian pesan hasil pemantauan.



Gambar 1. Flow Chart Sistem



Gambar 2. Blok Diagram Sistem

II. METODE

A. Konsep Perancangan

Sistem alat yang dibuat adalah sistem notifikasi pendeteksian umur pindah tanaman cabai besar dengan *flow chart* dan blok diagram sistem dapat dilihat pada gambar 1 dan 2. Pada kedua gambar tersebut dapat dilihat bahwa sistem dibuat dengan menggunakan mini komputer *Raspberry Pi 4 model B* sebagai mikrokontroler yang memproses program pengambilan citra tanaman cabai menggunakan *DCS-930L Wi-Fi Day Camera* dengan ukuran citra 640×480 px. Selanjutnya citra diolah dengan beberapa tahapan yaitu pemotongan citra gambar menjadi 6 bagian dengan ukuran sama, pendeteksian dan penjumlahan jumlah daun menggunakan metode *You Only Look Once* (YOLO) serta pengiriman notifikasi tanaman yang dapat segera di pindah tanam melalui aplikasi Telegram pada ponsel pengguna.

Dataset citra yang digunakan adalah sekumpulan citra daun tanaman cabai besar jenis cabe keriting Andaleh yang biasa ditanam di daerah Kecamatan X Koto, Kabupaten Tanah Datar dan sekitarnya yang masih dalam proses pembibitan dengan rentang usia

sekitar 2 – 4 minggu baik yang sudah memiliki jumlah daun 4 – 5 helai daun permanen maupun tidak. Proses pengambilan citra dilakukan dengan memotret daun dan kumpulan daun tanaman cabai besar menggunakan *DCS-930L Wi-Fi Day Camera*. Dataset tersebut kemudian dilakukan anotasi citra yang selanjutnya akan di proses pada bagian deteksi objek dengan melakukan pelatihan (*training*) menggunakan salah satu algoritme *deep learning* yaitu *You Only Look Once* (YOLO) versi 4. Hasil dari pelatihan adalah berupa file bobot dengan ekstensi *.weight yang selanjutnya dipergunakan untuk proses tes prediksi dengan file konfigurasi (.cfg) yang sudah dimodifikasi sesuai jumlah kelas dan spesifikasi penggunaannya. Proses perhitungan jumlah daun dari tanaman menggunakan pustaka perangkat lunak yang bersifat terbuka yaitu TensorFlow dengan menghitung kotak pembatas pada setiap kelas klasifikasi yang diberikan yaitu daun permanen dan daun muda. Pada tahap akhir, semua pemrograman yang berhasil mencapai tujuan akan diimplementasikan ke *Raspberry Pi* dengan bobot hasil pemodelan menggunakan metode YOLO versi 4 dikonversi terlebih dahulu ke TensorFlow Lite, sehingga *Raspberry Pi* dan kamera *DCS-930L Wi-Fi day Camera* menjadi satu kesatuan sistem tertanam (*embedded system*).

B. Akuisisi Citra

Akuisisi citra adalah proses pengambilan citra dari kamera sebagai input untuk perangkat pemrosesan yaitu *Raspberry Pi*. Proses pengiriman data citra dari kamera ke *Raspberry Pi* menggunakan protokol *File Transfer Protocol (FTP)* dengan *Raspberry Pi* sebagai *FTP-Server* yaitu tempat pelayanan yang berperan dalam tempat penyimpanan data dan transfer data ke *FTP-Client*. Sementara itu, *DCS-930L Wi-Fi Day Camera* berperan sebagai *FTP-Client* yaitu melakukan transfer data citra yang diambil ke *Raspberry Pi* atau dengan kata lain melakukan proses *upload* ke direktori yang sudah ditentukan pada di *Raspberry Pi*.

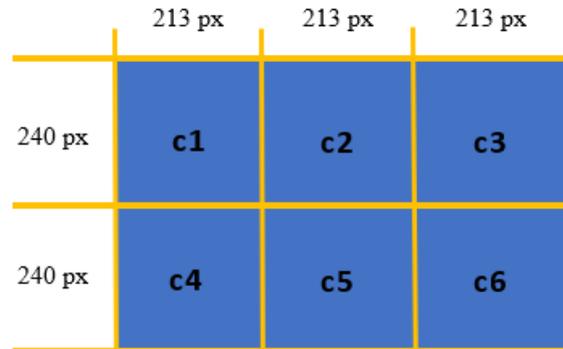
C. Cropping Image

Cropping image merupakan kegiatan memotong citra menjadi beberapa bagian yang diinginkan. Pemotongan berdasarkan pembagian ukuran lebar dan panjang citra dan dibagi sama besar dengan satuan *pixel*. Setiap bagian hasil pemotongan disimpan menjadi sebuah citra baru kemudian citra tersebut akan menjadi input pada tahapan deteksi dan perhitungan jumlah daun nantinya. Proses *cropping image* secara interaktif dapat dilihat pada gambar 3. Variabel *c* adalah citra hasil pemotongan tiap daerah yang ditandai dengan nilai tiap *c* adalah 213×240 px

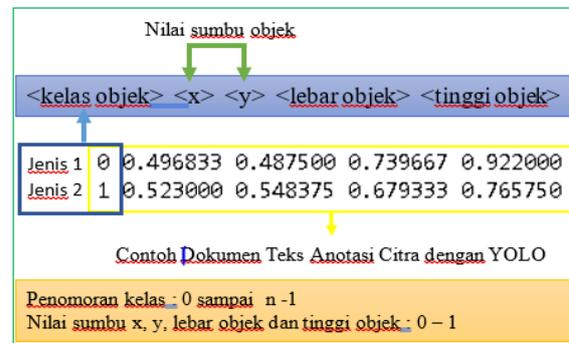
D. Anotasi Citra

Anotasi citra merupakan tahapan pemberian kotak batas (*bounding box*) dan label kelas pada citra yang dijadikan sebagai objek penelitian. Keluaran dari proses anotasi citra untuk YOLO adalah dokumen teks standar yang tidak diformat dengan ekstensi *.txt. Isi

pada dokumen teksnya dijelaskan pada gambar 4, Proses anotasi citra dilakukan dengan menggunakan master *labellmg*. Proses instalasi master *labellmg* dapat dilakukan dengan berbagai cara tergantung dengan sistem operasi pada *personal computer* seperti Ubuntu, Windows atau Mac-OS.



Gambar 3. Prose *Cropping Image* menjadi 6 bagian



Gambar 4. Penjelasan Isi Dokumen Hasil Anotasi Citra

Tabel 1. Konfigurasi YOLOv4 dalam proses *training*

Jenis Konfigurasi	Keterangan
Batch	64
Subdivision	16
Max_batches	6000
Steps	4800, 5400
Network size width	416
Network size height	416
Classes	2
Filters pada lapisan konvolusi terakhir	21

Tabel 2. Tabel *Confusion Matrix*

<i>Confusion Matrix</i>	Kebenaran	
	Positive	Negative
Prediksi	True Positive	False Positive
	False Negative	True Negative

E. Pelatihan Model

Proses pelatihan / *training* model ini dilakukan di *Google Colaboratory* atau disingkat menjadi “Colab” yang menyediakan fasilitas notebook jupyter gratis berbasis Python yang dapat dijalankan di *browser* penggunaannya. Fasilitas lainnya adalah kesempatan akses gratis GPU (*Graphics Processing Unit*) dan TPU (*Tensor Processing Unit*) berupa *virtual machine* dari *Google*. Fasilitas yang diperlukan pada proses *training* model ini adalah fasilitas GPU. *Framework* yang digunakan adalah *Darknet* dengan *backbone* yaitu

CSPDarknet53. Pada tahapan awal pelatihan, dataset dibagi menjadi dua bagian yaitu 90% untuk proses *training* dan 10% untuk proses *test*. Selain pemisahan dataset, hal lain yang harus dilakukan adalah melakukan konfigurasi awal pada beberapa aspek penting seperti *file Make* dengan mengaktifkan penggunaan pada OpenCV, GPU (*Graphics Processing Unit*), CUDNN, CUDNN_HALF, dan LIBSO. Konfigurasi selanjutnya adalah pemberian nilai parameter yang sesuai pada table 1. Hasil pelatihan model ini adalah *file* dengan format *.weights* yaitu *file* yang berisi urutan bobot yang berkaitan dengan *Sequence Alignment and Modeling*.

F. Evaluasi Model

Pengujian performansi dari model yang dipilih menggunakan metode matrix dasar untuk algoritma deteksi objek yaitu *Confusion Matrix* yang disebut juga sebagai *error matrix* dengan pertimbangan nilai parameter *Intersection over Union* (IoU). IoU merepresentasikan persentase irisan antara kotak batas yang sebenarnya dengan kotak batas hasil prediksi. Nilai IoU dapat dilihat pada hasil prediksi dengan rentang 0 sampai dengan 1. Semakin besar nilai IoU nya maka semakin akurat objek yang dideteksi, dengan IoU dikalkulasikan berdasarkan persamaan 1:

$$IoU = \frac{Irisan}{Gabungan} = \frac{Aktual \cap Prediksi}{Aktual \cup Prediksi} \quad (1)$$

Nilai IoU sebesar 0.5 dijadikan sebagai nilai ambang batas pada penentuan kriteria *Confusion Matrix* [9] seperti tampak pada tabel 2 yaitu *True Positive*, *False Positive*, *True Negative* dan *False Negative* sebagai berikut :

1. *True Positive* (TP), model memprediksi objek dengan benar dan nilai $IoU \geq 0.5$.
2. *False Positive* (FP), model salah prediksi dengan memprediksi objek lain diluar kelas (latar belakang) atau memiliki nilai $IoU < 0.5$
3. *False Negative* (FN), model tidak mendeteksi objek yang seharusnya terdeteksi pada data uji.
4. *True Negative* (TN), model tidak melakukan deteksi pada data uji yang seharusnya memang tidak perlu dideteksi.

Hasil dari *Confusion Matrix* ini menjadi input pada perhitungan nilai *precision*, *recall*, *F-score* dan *accuracy*. *Precision* mengukur akurasi model dalam mengklasifikasikan sampel sebagai positif yang mana persamaan dari *precision* adalah sebagai berikut :

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall mengukur kemampuan model untuk mendeteksi sampel positif. Semakin tinggi *recall*, semakin banyak sampel positif yang terdeteksi dengan persamaan dari *recall* adalah sebagai berikut :

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F-score dapat diartikan sebagai rata-rata tertimbang dari *precision* dan *recall* dengan nilai terbaik adalah 1 dan nilai terburuk adalah 0. Persamaan

untuk menghitung *F-score* terdapat pada persamaan 4 [10]

$$F - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Perhitungan selanjutnya adalah perhitungan akurasi dari model YOLO dengan dua jenis perhitungan akurasi yaitu *Detection Accuracy* (DA) yang memuat nilai seberapa akurat metode YOLO dapat mendeteksi suatu objek dan *Recognition Accuracy* (RA) yang memuat nilai seberapa akurat metode YOLO dalam mengenai suatu objek. Kedua nilai akurasi ini dapat dilihat melalui persamaan 5 dan 6 dengan melakukan perbandingan antara label yang tepat dan sesuai dengan kelasnya [11].

$$DA(\%) = \frac{Jumlah\ objek\ yang\ terdeteksi}{Total\ objek} \quad (5)$$

$$RA(\%) = \frac{Jumlah\ deteksi\ yang\ benar}{Total\ objek} \quad (6)$$

G. Pembuatan Sistem Notifikasi Telegram

Sistem notifikasi yang dibuat menggunakan fitur BotFather, fitur ini merupakan aplikasi pihak ketiga yang memungkinkan pengguna dapat berinteraksi dengan Bot dengan mengirim pesan permintaan sebaris. Permintaan tersebut akan dibalas oleh Bot secara otomatis yang disesuaikan dengan program atau pengaturannya, contohnya pada sistem pemberitahuan berita terkini, pencarian otomatis, dan lain sebagainya. BotFather akan mengirimkan token untuk mengakses HTTP API dari bot telegram yang digunakan untuk pengembangan bot sehingga dapat diatur untuk melakukan pengiriman pesan notifikasi pesan pemindahan tanaman ke pengguna.

H. Implementasi ke Raspberry Pi

Sebelum diimplementasikan ke *Raspberry Pi*, model YOLOv4 harus dikonversikan ke *framework* TensorFlow Lite karena memiliki performa tinggi dengan akselerasi perangkat keras dan optimalisasi model sehingga dapat diterapkan ke *Raspberry Pi*. Proses konversi ini masih menggunakan arsitektur YOLOv4, dengan *backbone* CSPdarknet53 yang berubah hanya pada bagian *framework*-nya yaitu *tf.lite*. Selanjutnya dilakukan pemasangan paket perangkat lunak pendukung yaitu; *OpenCV-python*, *Tensorflow untuk tipe Raspberry Pi*, *Python-Telegram-Bot*, *absl*, *numpy*, dan *coloursys*.

III. HASIL DAN PEMBAHASAN

A. Evaluasi Model Citra Daun

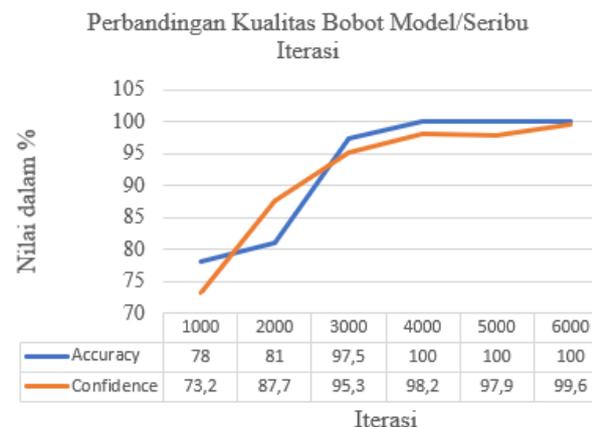
Hasil pelatihan (*training*) model YOLO yang dilakukan sesuai dengan konfigurasi pada tabel 2 memiliki proses iterasi sebanyak 6000 yang berasal dari pengaturan nilai *max_batches* yaitu banyaknya iterasi yang dilakukan. Pemilihan jumlah 6000 iterasi adalah untuk mengetahui nilai iterasi yang sesuai untuk mendapatkan hasil deteksi model terbaik. Hubungan antara jumlah iterasi dengan hasil deteksi model dapat diketahui dengan membandingkan hasil nilai *accuracy* dan nilai *confidence*. Pada bobot dengan iterasi 1000,

2000, 3000, 4000, 5000 dan 6000 pada sampel gambar yang sama. Berdasarkan pengujian yang dilakukan didapatkan hasil nilai rata-rata dari *accuracy* pada iterasi 4000 sampai dengan 6000 adalah 100 %, sementara itu nilai rata-rata dari *accuracy* pada iterasi 1000 sampai dengan 3000 mengalami kenaikan nilai.

Tabel 3. Perbandingan Hasil Deteksi pada Setiap Model per Seribu Iterasi

No	Citra Awal	Iterasi Ke-	Deteksi Citra	Deskripsi
1.		1000		<u>Permanen</u> 4 TP & 1 FN Accuracy:80 % <u>Muda</u> 1 TP Accuracy:100%
		2000		<u>Permanen</u> 5 TP Accuracy:100% <u>Muda</u> 1 TP Accuracy:100%
	3000		<u>Permanen</u> 5 TP Accuracy:100% <u>Muda</u> 1 TP Accuracy:100 %	
	4000		<u>Permanen</u> 5 TP Accuracy:100% <u>Muda</u> 1 TP Accuracy:100 %	
	5000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100%	
	6000		<u>Permanen</u> 5 TP Accuracy:100% <u>Muda</u> 1 TP Accuracy:100 %	
2.		1000		<u>Permanen</u> 3 TP & 1 FN Accuracy:75% <u>Muda</u> 2 TP 1 FP Accuracy:60%
		2000		<u>Permanen</u> 3 TP & 1 FN Accuracy:75% <u>Muda</u> 2 TP & 1 FP Accuracy:60%
		3000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
		4000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
3.		1000		<u>Permanen</u> 4 TP & 1 FN Accuracy:80 % <u>Muda</u> 3 TP & 1 FN Accuracy:75%
		2000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 4 TP Accuracy:100 %
		3000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 4 TP Accuracy:100 %
		4000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 4 TP Accuracy:100 %
		5000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 4 TP Accuracy:100 %
		6000		<u>Permanen</u> 5 TP Accuracy:100 % <u>Muda</u> 4 TP Accuracy:100 %
4.		1000		<u>Permanen</u> 2 TP & 2 FN Accuracy:50% <u>Muda</u> 2 TP & 1 FP Accuracy:60 %
		2000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
		3000		<u>Permanen</u> 3 TP & 1 FN Accuracy:75 % <u>Muda</u> 2 TP Accuracy:100%
		4000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
5000		5000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
		6000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %

5.	5000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
	6000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 2 TP Accuracy:100 %
	1000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100 %
	2000		<u>Permanen</u> 2 TP & 2 FN Accuracy:50 % <u>Muda</u> 1 TP & 3 FP Accuracy:25%
	3000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100 %
	4000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100 %
	5000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100 %
	6000		<u>Permanen</u> 4 TP Accuracy:100 % <u>Muda</u> 1 TP Accuracy:100 %



Gambar 5. Perbandingan Kualitas Bobot Model per Seribu Iterasi

Tabel 3 merupakan perbandingan hasil deteksi pada setiap model per seribu iterasi dengan menggunakan data lima buah gambar berbeda yang didapatkan dari proses pembagian dataset sebelumnya. Pada tabel dapat

dilihat perbandingan nilai *confidence* setiap gambar dengan total objek terdeteksi adalah 32 buah, memiliki perbandingan nilai rata-rata *confidence* per tiap seribu iterasi adalah 0.732, 0.877, 0.953, 0.982, 0.979, dan 0,996. Nilai *confidence* yang mendekati nilai 1 menunjukkan hasil prediksi dengan keakuratan tinggi sehingga objek yang diprediksi adalah benar dan sesuai dengan kelas objeknya. Nilai *accuracy* dan nilai *confidence* per seribu bobot model iterasi menunjukkan kenaikan nilai yang dapat dilihat pada grafik di gambar 5. Hal ini dikarenakan semakin banyak iterasi maka semakin banyak langkah untuk mencapai minimal error dari prediksi objek yang lebih akurat. Sehingga berdasarkan data tersebut, model yang sebaiknya digunakan adalah bobot model dengan iterasi 6000 iterasi yang memiliki nilai *accuracy* dan *confidence* lebih baik dibandingkan dengan bobot model dengan nilai iterasi dibawahnya.

Langkah evaluasi model selanjutnya adalah dengan menggunakan *Confusion Matrix* pada bobot model deteksi yang sama yaitu bobot model deteksi yang menghasilkan *accuracy* dan *confidence* dengan nilai terbaik. Data pengujian menggunakan 25 gambar dengan setiap gambar dibagi menjadi dua kelas deteksi yaitu kelas permanen dan kelas muda. *Confusion Matrik* pada masing-masing kelas dapat dilihat pada tabel 4 dan tabel 5. Hasil dari penggunaan *Confusion Matrix* ini digunakan untuk memperhitungkan nilai parameter-parameter evaluasi sebelumnya dengan hasil yang diperlihatkan pada tabel 6. Hasil nilai pada setiap parameter untuk kelas permanen memiliki angka yang lebih tinggi dibandingkan dengan kelas muda. Pada tabel 6 objek permanen dapat dikenali dengan akurasi 92,85%. Hal ini menunjukkan bahwa model YOLO dapat mengenali objek kelas permanen dengan baik yang mana kelas ini menjadi penentu pindah tanam bibit tanaman cabai. Sehingga dapat dilakukan penjumlahan kotak prediksi daun permanen sebagai angka yang menunjukkan jumlah daun permanen pada suatu tanaman. Apabila jumlah besar sama dari 4 buah maka, tanaman siap dipindahkan.

Tabel 4. *Confusion Matrix* Kelas Permanen

Confusion Matrix Kelas Permanen	Kebenaran	
	Positive	Negative
Prediksi Positive	78	4
Prediksi Negative	2	0

Tabel 5. *Confusion Matrix* Kelas Muda

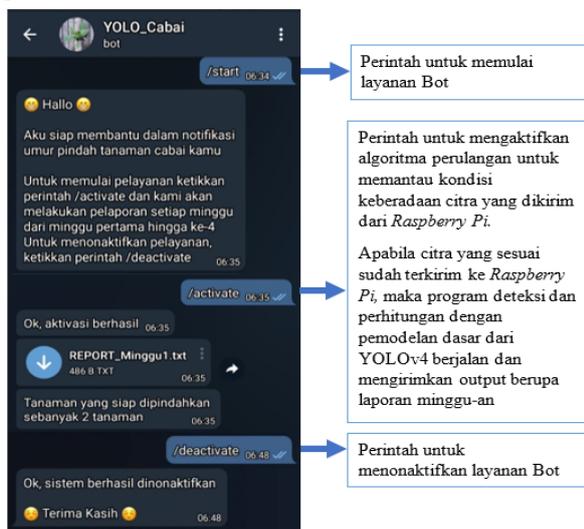
Confusion Matrix Kelas Muda	Kebenaran	
	Positive	Negative
Prediksi Positive	41	0
Prediksi Negative	37	0

Tabel 6. Perbandingan Parameter Evaluasi per Kelas

Parameter	Permanen	Muda
Precision	0,95121	1
Recall	0,975	0,525641
F-Score	0,96295	0,689075
Detection Accuracy (%)	97,61904	52,5641
Recognition Accuracy (%)	92,85714	52,5641

B. Evaluasi Sistem Notifikasi Telegram

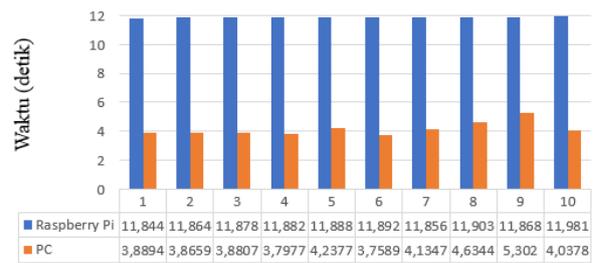
Pada bagian sistem notifikasi Telegram yang diimplementasikan ke *Raspberry Pi* dapat berkomunikasi dengan baik dengan bot Telegram yang dibangun untuk mengirim pesan notifikasi dan menjalankan program deteksi. *Request* dan *Respond* dari *Raspberry Pi* berhasil mengikuti skema pada gambar 6. Skema ini dibuat sesuai dengan tujuan penelitian yaitu pemberian notifikasi pindah tanam pada bibit tanaman cabai dengan jenis cabe kriting andaleh yang biasa ditanam di daerah Kecamatan X Koto, Kabupaten Tanah Datar sekitarnya. Pada bagian dokumen *text* yang dikirim berisikan rincian tanaman yang dapat dipindah tanam dan jumlah daun permanen serta daun mudanya. Selain itu, apabila tidak ditemukan identifikasi daun pada gambar, maka status rincian pada tanaman tersebut adalah tidak ada tanaman.



Gambar 6. Alur Pesan pada Telegram

Pengukuran *inference time* pada pengimplementasian model YOLOv4 dilakukan menjadi dua buah kondisi yaitu pada saat proses model sedang melakukan deteksi untuk satu buah tanaman dan keadaan kedua adalah pada saat proses deteksi hingga pengiriman pesan notifikasi melalui aplikasi Telegram. Rata-rata *inference time* pada kondisi pertama adalah 11,8855965 detik, dan pada kondisi kedua memiliki rata-rata *inference time* adalah 25,294732 detik. Pada gambar 7 terlihat perbandingan *inference time* pada saat menggunakan *Raspberry Pi* dan pada saat menggunakan *Personal Computer* dengan spesifikasi processor intel core i5 yang juga sama-sama dijalankan dengan CPU mendapatkan hasil lebih cepat untuk menjalankan model pemogramannya. Hal ini karena *processor* yang berfungsi dalam pengolahan datanya memiliki spesifikasi yang lebih tinggi.

Perbandingan Inference Time



Gambar 7. Perbandingan Nilai *Inference Time*

IV. KESIMPULAN

Sistem notifikasi Telegram dalam penentu pindah tanaman cabai besar dapat dibuat dengan menerapkan metode *You Only Look Once* versi 4 dengan *backbone CSPdarknet53*. Pada proses pengimplementasian ke *Raspberry Pi 4 model B*, model YOLOv4 perlu di konversi ke *framework TensorFlow Lite* dengan hasil pengukuran rata-rata *inference time* pada saat menjalankan satu deteksi pemodelan adalah 11,8855965 detik. Hal ini dapat terjadi karena penggunaan TensorFlow Lite dapat memperkecil ukuran model (*small code footprint*) dan *inference time* yang lebih cepat sehingga memungkinkan TensorFlow Lite untuk mengeksekusi secara efisien pada perangkat dengan sumber daya komputasi dan memori yang terbatas seperti *Raspberry Pi*. Hasil perhitungan nilai parameter akurasi pengenalan dan perhitungan objek daun permanennya adalah 92.8257% yang menunjukkan bahwa objek dapat diidentifikasi sebagai kelas permanen dengan baik karena banyaknya objek yang dikenali bernilai *positive*.

Saran untuk pengembangan selanjutnya adalah memperbanyak data *train* untuk dataset citra sehingga dapat mengoptimalkan akurasi dari model YOLOv4, menggunakan kamera dengan resolusi tinggi sebagai perangkat akuisisi citra digitalnya untuk penerapan di wilayah yang lebih luas, menggunakan jenis komputer mini yang memiliki spesifikasi komputasi yang lebih tinggi seperti Jetson Nano sebagai pengganti *Raspberry Pi* sehingga mempercepat *inference time* pada saat menjalankan pemograman. Serta, konsep pengimplementasian pada penelitian ini dapat dilakukan ke bidang lainnya.

REFERENSI

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [2] S. Shinde, A. Kothari, and V. Gupta, "YOLO based Human Action Recognition and Localization," *Procedia Comput. Sci.*, vol. 133, no. 2018, pp. 831–838, 2018, doi: 10.1016/j.procs.2018.07.112.
- [3] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *Eur. Transp. Res. Rev.*, vol. 11, no. 1, 2019, doi: 10.1186/s12544-019-0390-4.
- [4] H. E. D. Mohamed *et al.*, "MSR-YOLO: Method to Enhance

- Fish Detection and Tracking in Fish Farms,” *Procedia Comput. Sci.*, vol. 170, no. 2019, pp. 539–546, 2020, doi: 10.1016/j.procs.2020.03.123.
- [5] K. Itakura and F. Hosoi, “Automatic tree detection from three-dimensional images reconstructed from 360 spherical camera using YOLO v2,” *Remote Sens.*, vol. 12, no. 6, 2020, doi: 10.3390/rs12060988.
- [6] M. Buzzy, V. Thesma, M. Davoodi, and J. M. Velni, “Real-time plant leaf counting using deep object detection networks,” *Sensors (Switzerland)*, vol. 20, no. 23, pp. 1–14, 2020, doi: 10.3390/s20236896.
- [7] R. Suryani, *Outlook cabai*. Indonesia : Pusat Data dan Sistem Informasi Pertanian Sekretariat Jendral Kementrian Pertanian. 2018.
- [8] S. Swastika, D. Pratama, T. Hidayat, and K. B. Andri, *Buku Petunjuk Teknis Teknologi Budidaya Cabai Merah*. 2017.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [10] A. Hamidisepehr, S. V. Mirnezami, and J. K. Ward, “Comparison of object detection methods for corn damage assessment using deep learning,” *Trans. ASABE*, vol. 63, no. 6, pp. 1969–1980, 2020, doi: 10.13031/TRANS.13791.
- [11] Y. Umar, Hanafi, S. Mardi, Nugroho, Susiki, and R. F. Rachmadi, “Deteksi Penggunaan Helm Pada Pengendara Bermotor Berbasis Deep Learning,” 2020.