

# Pendeteksi Pola Candlestick Chart Pada Saham Menggunakan Algoritma YOLO

1<sup>st</sup> Gerin Aryo Prasetya  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

gerinaryoprasetya@student.telkomuniversity.ac.id

2<sup>nd</sup> Imelda Atastina  
Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

imelda@telkomuniversity.ac.id

**Abstrak**—Analisis teknikal merupakan salah satu bentuk analisis yang dilakukan oleh seorang investor untuk menganalisis pergerakan harga saham. Analisis teknikal memanfaatkan beberapa informasi penting dalam saham seperti harga, *volume* dan *open interest* yang melekat pada saham yang selanjutnya dapat membentuk sebuah *candlestick*. Salah satu metode yang digunakan pada analisis teknikal adalah mengamati pola yang terbentuk dari kumpulan *candlestick*, pola yang terbentuk ini akan digunakan oleh seorang investor memprediksi apakah harga saham akan bergerak naik atau turun. Cara yang digunakan oleh seorang investor untuk menemukan pola ini adalah mengamati secara langsung pada chart sebagai data berupa gambar. Karena data yang diamati berupa gambar, maka untuk mendeteksi pola *candlestick* dapat dibantu dengan beberapa algoritma *deep learning*, salah satu algoritma *deep learning* yang dapat digunakan untuk mendeteksi objek adalah YOLO (*You Only Look Once*). Penelitian ini menganalisa performansi dari algoritma YOLO dengan versi YOLOv5 yang digunakan pada *datasets candlestick chart* untuk mendeteksi pola *double top* dan *double bottom* pada *candlestick chart* suatu saham. Hasilnya semakin banyak jumlah *datasets* yang digunakan pada saat *training model* YOLOv5, akurasi yang didapatkan menjadi semakin baik yaitu dengan akurasi maksimal sebesar 75,9%.

**Kata kunci**—*yolo, candlestick chart, deep learning, double top, double bottom, saham*

**Abstract**—*Technical analysis is a form of analysis performed by an investor to analyze stock price movements. Technical analysis utilizes some important information on stocks such as price, volume and open interest attached to stocks which can then form a candlestick. One of the methods used in technical analysis is to observe the pattern formed from a collection of candlesticks, this pattern formed will be used by an investor to predict whether the stock price will move up or down. The method used by an investor to find this pattern is to observe directly on the chart as data in the form of images. Because the observed data is in the form of images, several deep learning algorithms can help detect candlestick patterns. One of the deep learning algorithms that can be used to detect objects is YOLO (You Only Look Once). This study analyzes the performance of the YOLO algorithm with the YOLOv5 version used in candlestick chart datasets to detect double top and double bottom patterns on a stock candlestick chart. The result is that the greater the number of datasets used when training the*

*YOLOv5 model, the better the accuracy obtained, with a maximum accuracy of 75.9%.*

**Keywords**—*yolo, candlestick chart, deep learning, double top, double bottom, stocks*

## I. PENDAHULUAN

### A. Latar Belakang

Kemampuan seorang investor dalam menganalisis suatu pergerakan harga saham baik melalui analisis secara teknikal atau fundamental secara relatif akurat dapat memungkinkan investor untuk mendapatkan keuntungan yang cukup besar.[1] Salah satu analisis teknikal yang biasa digunakan oleh investor adalah menganalisis pola yang terbentuk dari kumpulan *candlestick* hasil perubahan harga saham dari rentang waktu tertentu. Seorang investor dapat lebih mudah untuk menafsirkan situasi nilai pasar saat ini atau yang lalu menggunakan data visual, serta melihat riwayat pergerakan harga saham pada periode waktu tertentu dan menemukan pola.[2] Contoh pola dari *candlestick* yang dapat ditemukan adalah pola *double top* dan *double bottom* yang pada penelitian ini kedepannya akan penulis coba deteksi menggunakan algoritma YOLO. Namun, tidak semua pola dapat ditentukan atau ditemukan pada periode waktu tertentu dan pola lebih cenderung muncul secara acak dengan periode waktu yang bervariasi.

Data visual yang dihasilkan dari pola yang terbentuk dapat diklasifikasikan menggunakan bantuan algoritma *deep learning*. Klasifikasi berbasis *deep learning* ini terbukti memiliki kinerja sangat baik di bidang pengolahan gambar, video dan audio selama beberapa tahun terakhir[3]. Beberapa algoritma *deep learning* yang dapat digunakan untuk melakukan klasifikasi data visual yaitu *Convolutional Neural Network* (CNN) dan *You Only Look Once* (YOLO).

Beberapa penelitian menyatakan bahwa algoritma CNN dapat dengan baik melakukan klasifikasi pada pola yang ditemukan. Pada penelitian [4] dilakukan klasifikasi pola pada *Istanbul Stock Exchange* and *NASDAQ Stock Exchange* dan mendapatkan rata-rata akurasi sebesar 92.03% untuk jumlah training dataset sebanyak 10,30.50,70 dan 80. Namun, algoritma CNN memiliki masalah terhadap waktu response yang dibutuhkan, dalam hal ini algoritma YOLO dapat mengalahkan CNN dalam hal waktu response secara real time.[5]

Berdasarkan latar belakang tersebut, penelitian ini akan melakukan pendeteksian objek berupa pola *candlestick chart* pada saham dan menganalisa performa dari algoritma YOLO untuk melakukan pendeteksian objek pada *candlestick chart* di saham.

#### B. Topik dan Batasannya

Penelitian ini membahas performansi dari algoritma YOLO untuk mendeteksi objek pola *candlestick* serta melakukan perhitungan dari jumlah pola yang ditemukan. Pola yang akan dideteksi antara lain adalah *double top* dan *double bottom*

#### C. Tujuan

Tujuan dilakukannya penelitian ini adalah untuk mengetahui performansi dari algoritma YOLO untuk mendeteksi pola *candlestick* pada saham serta mengetahui jumlah pola yang ditemukan pada suatu saham menggunakan algoritma YOLO.

#### D. Organisasi Tulisan

Pada laporan penelitian ini, akan dibahas studi literatur yang terkait dan berhubungan serta studi yang menjadi acuan penelitian pada Bab 2. Kemudian sistem yang dibangun pada penelitian ini dapat dilihat pada Bab 3. Evaluasi hasil uji dari penelitian yang telah dilakukan tertera pada Bab 4. Serta kesimpulan dari penelitian dapat dilihat pada Bab 5.

## II. STUDI TERKAIT

Penelitian terhadap klasifikasi pattern pada *candlestick* telah dilakukan oleh beberapa peneliti dengan menggunakan beberapa metode object detection yang berbeda. Kaya, dkk [4] melakukan pengelompokan pola chart menggunakan metode *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM), dan didapatkan hasil bahwa CNN memiliki performa yang paling baik diantara metode lainnya yaitu rata-rata akurasi sebesar 92,03% sedangkan untuk RNN sebesar 69,70% dan LSTM sebesar 69,53% dengan pola *candlestick* yang diamati adalah “*double bottom*” dan “*double top*” menggunakan 3242 datasets contoh kelas, 1542 untuk kelas “*double bottom*” dan 1697 untuk kelas “*double top*”. Skuratov, dkk [2] menggunakan metode CNN saja untuk mendeteksi pola yang terbentuk, dan pola yang diamati adalah “*double bottom*” dan “*rising triangle*” dan mendapati akurasi sebesar 98,6% menggunakan metode CNN dengan kecepatan sekitar 0,65 detik setiap 1000 contoh data. Birogul, dkk [3] melakukan rekognisi untuk menentukan jual atau beli pada saham menggunakan algoritma YOLO, dengan bantuan berupa gambar dari chart saham pada periode waktu mulai dari tahun 2018 hingga tahun 2019, dan mendapati 112 pola yang ditemukan, dan memberikan total profit sebesar 102,68%. Sedangkan Velay, dkk [6] melakukan rekognisi pola pada chart menggunakan algoritma CNN dan LSTM untuk menemukan pola *bearish flag* dalam rentang waktu

satu tahun menggunakan periode waktu 30 menit, dan berhasil menemukan sekitar 3000 pola *bearish flag*.

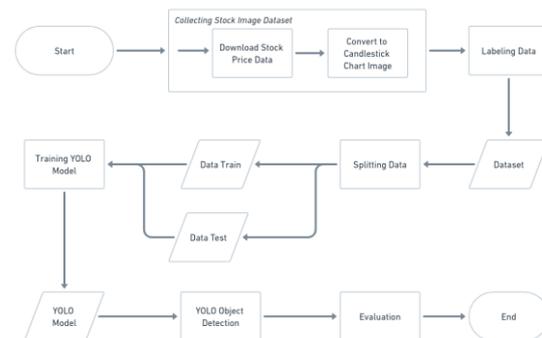
Selain mendeteksi kumpulan *candlestick* yang membentuk sebuah pola, terdapat pula penelitian yang mendeteksi pada satu buah *candlestick*. Pada penilitan Aditya, dkk. [7] melakukan rekognisi pada pola satuan *candlestick* menggunakan gabungan antaran LSTM dan CNN dan mendapati rata-rata akurasi yang didapatkan sebesar 85% untuk predeksi yang ditemukan.

Beberapa penilitain pun telah memberikan hasil yang baik untuk mendeteksi objek menggunakan algoritma YOLO. Penilitain Yang, dkk [8] melakukan deteksi objek untuk wajah secara *real time*, dan terbukti bahwa YOLO dapat mendeteksi objek secara sangat cepat, pada penilitan tersebut YOLO berhasil mendeteksi wajah dengan kecepatan rata-rata 0,028 detik. Akurasi yang sama dengan SSD namun lebih cepat hingga 3 kali lipat menggunakan YOLO. Sedangkan Wang, dkk [9] berusaha untuk medeteksi objek insulators yang terdapat pada gambar menggunakan algoritma YOLO dengan jumlah gambar yang diuji sebanyak 200 gambar dengan akurasi yang didapatkan adalah sebesar 83,5%.

Berdasarkan studi-studi tersebut, dalam hal mendeteksi pola pada kumpulan *candlestick* chart saham, dapat dibantu menggunakan beberapa algoritma deep learning. Salah satu algoritma deep learning yang bisa digunakan adalah YOLO, dan YOLO sendiri memiliki performa untuk akurasi dan kecepatan yang baik dalam kebutuhan untuk mendeteksi objek pada sebuah gambar.

## III. SISTEM YANG DIBANGUN

Model pendeteksi objek yang akan dibangun pada sistem ini bertujuan untuk menemukan pola *double top* dan *double bottom* pada *candlestick* chart suatu saham. Sedangkan algoritma yang digunakan untuk melakukan pendeteksian objek adalah YOLO (*You Only Look Once*). Secara umum, rancangan sistem yang akan dibangun adalah sebagai yang dapat dilihat pada Gambar 1.



GAMBAR 1. FLOW CHART RANCANGAN SISTEM

#### A. Collecting Stock Image Dataset

Pada penelitian ini, dataset yang digunakan untuk melakukan *training* model YOLO merupakan

kumpulan gambar *candlestick* chart serta terdapat pola *double top* atau *double bottom* didalamnya. Kumpulan gambar tersebut diambil dari saham-saham yang terdapat pada *Nasdaq Stocks Exchange* dalam rentang waktu 6 bulan dan 3 bulan, serta timeframe yang digunakan adalah 4 jam dan 1 jam. Pada tahap ini, data yang berhasil dikumpulkan sebanyak 8091 data gambar *candlestick chart* namun pada gambar-gambar tersebut tidak pasti terdapat pola *double top* dan/atau *double bottom*.

### 1. Download Stock Price Data

Tahap pertama dalam mengumpulkan *datasets* adalah mengunduh informasi harga saham-saham yang akan digunakan sebagai *datasets* dengan format CSV (*Comma-Separated Values*). Pada dokumen tersebut terdapat informasi mengenai harga buka, harga tutup, harga tertinggi dan harga terendah.

### 2. Convert to Candlestick Chart Image

Setelah dokumen harga saham berformat CSV berhasil diunduh, selanjutnya adalah memvisualisasikannya ke dalam bentuk gambar *candlestick* chart menggunakan bahasa pemrograman *python* dengan library *mplfinance*[10] untuk merubah data pada CSV menjadi *candlestick chart*. Contoh dari *dataset* yang telah dikonversi dapat dilihat pada Gambar 2.



GAMBAR 2. DATASET CANDLESTICK CHART

### B. Labelling Data

Hasil dari mengumpulkan *dataset* gambar *candlestick* chart kemudian melalui tahap memberikan label atau anotasi terhadap gambar secara manual dimana data dibagi menjadi 2 kelas atau label yaitu *double top* dan *double bottom*. Pemberian label dilakukan menggunakan bantuan program *LabelImg*[11]. *LabelImg* merupakan program open source yang dapat digunakan untuk memberikan label pada gambar dalam format YOLO. Pemberian label dilakukan dengan memperhatikan gambar *candlestick* chart yang sudah didapat, apabila pada gambar *candlestick* chart tersebut terdapat pola *double top* atau *double bottom* maka akan diberikan label dengan cara mengkotaki pola yang terdapat pada gambar tersebut. Contoh pola *double top* dan *double bottom* yang telah diberi label ditunjukkan pada Gambar 3 dan Gambar 4.



GAMBAR 3. LABEL POLA DOUBLE TOP

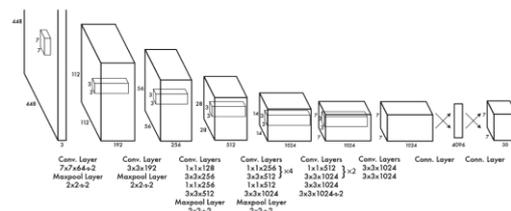


GAMBAR 4. LABEL POLA DOUBLE BOTTOM

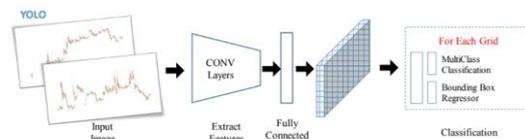
Tidak semua data gambar yang terkumpul yaitu sebanyak 8091 gambar terdapat pola *double top* atau *double bottom* di dalamnya. Pada penelitian ini, total *dataset* gambar yang terdapat pola *double top* dan/atau *double bottom* sebanyak 926 gambar dari 8091 gambar yang telah dikumpulkan, dengan total kelas *double bottom* sebanyak 619 label dan *double top* sebanyak 705 label.

### C. YOLO Object Detection

YOLO (*You Only Look Once*) merupakan pengembangan dari CNN dan memiliki arsitektur yang berbasis CNN, yang arsitektur dan diagram umumnya disajikan pada Gambar 5 dan Gambar 6. YOLO telah menjadi salah satu algoritma deep learning yang paling populer yang digunakan dalam bidang klasifikasi objek belakangan ini. YOLO adalah struktur yang lebih cepat dan lebih sukses dalam hal pengenalan objek jika dibandingkan dengan pesaingnya seperti CNN.



GAMBAR 5. ARSITEKTUR YOLO



GAMBAR 6. DIAGRAM UMUM YOLO

YOLO sedikit berbeda dengan metode tradisional, dimana pada YOLO input data akan diubah menjadi sebuah kotak dan mengestimasi koordinat serta prediksi kelas secara bersamaan. Pada tahap pertama,

input data akan dibagi menjadi SxS kisi. Selanjutnya setelah input data melewati *neural network*, kisi ini sendiri memiliki tanggung jawab untuk mencari apakah terdapat objek di dalamnya atau tidak, jika terdapat objek didalamnya maka akan ditentukan apakah objek sudah berada di titik tengah, tinggi dan lebarnya dari titik tengah. YOLO memisah vektor prediksi pada masing-masing kisi. Pada setiap kisi ini menggambarkan *Confidence Score*,  $b_x, b_y, b_w, b_h$  dan nilai probabilitas kelas.

*Confidence Score* menunjukkan seberapa pasti objek berada pada kisi yang benar (0 jika pasti tidak terdapat objek dan 1 jika pasti ditemukan objek yang tepat). Dengan kata lain,  $b_x$  pada rumus 1 menghitung koordinat  $x$  titik tengah objek,  $b_y$  pada rumus 2 menghitung koordinat  $y$  titik tengah objek,  $b_w$  pada rumus 3 menghitung lebar dari objek dan  $b_h$  pada rumus 4 menghitung tinggi dari objek. Jaringan memprediksi 4 titik koordinat untuk setiap kotak  $t_x, t_y, t_w, t_h$ . Bila *cell* melewati pojok kiri atas dari gambar  $(c_x, c_y)$  dan kotak memiliki lebar dan tinggi  $p_w, p_h$  maka prediksi mengacu pada [3] :

$$b_x = \sigma(t_x) + c_x \quad (1)$$

$$b_y = \sigma(t_y) + c_y \quad (2)$$

$$b_w = p_w e^{t_w} \quad (3)$$

$$b_h = p_h e^{t_h} \quad (4)$$

### 1. YOLOv3

YOLOv3 melakukan beberapa perbaikan jika dibandingkan dengan seri YOLO sebelumnya, termasuk: menyesuaikan struktur jaringan; menggunakan fitur multi-skala untuk deteksi objek; dan mengklasifikasikan objek menggunakan *Logistic*. YOLOv3 menggunakan jaringan *Darknet* terbaru yaitu *Darknet53* dan menambahkan modul sisa ke jaringan, yang mirip dengan gagasan Resnet[12]. Hal tersebut membantu untuk memecahkan masalah gradien jaringan dalam. Setiap modul residu terdiri dari dua lapisan konvolusional dan sebuah tautan pintasan atau *shortcut link*. *Darknet53* terdiri dari sebuah struktur konvolusional dan modul residual. Arsitektur dari jaringan YOLOv3 ditunjukkan pada Gambar 7.

	Type	Filters	Size	Output
	Convolutional	32	3 3	256 256
	Convolutional	64	3 3/2	128 128
1	Convolutional	32	1 1	
	Convolutional	64	3 3	
	Residual			128 128
2	Convolutional	128	3 3/2	64 64
	Convolutional	64	1 1	
	Convolutional	128	3 3	
8	Residual			64 64
	Convolutional	256	3 3/2	32 32
	Convolutional	128	1 1	
8	Convolutional	256	3 3	
	Residual			32 32
	Convolutional	512	3 3/2	16 16
8	Convolutional	256	1 1	
	Convolutional	512	3 3	
	Residual			16 16
4	Convolutional	1024	3 3/2	8 8
	Convolutional	512	1 1	
	Convolutional	1024	3 3	
	Residual			8 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

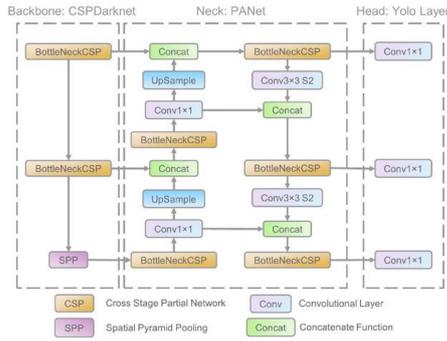
GAMBAR 7. STRUKTUR JARINGAN *DARKNET53*

Pada seluruh jaringan *backbone* dari YOLOv3, tidak ada *pooling layer* dan *fully connected layer*. Dengan mengatur langkah parameter konvolusi menjadi 2, *down sampling* dari jaringan akan dilakukan. Ukuran gambar diperkecil menjadi setengah dari ukuran aslinya setelah melewati lapisan konvolusional. Setiap implementasi Convolutional termasuk konvolusi, BN (Batch Normalization Layer), dan Fungsi aktivasi *Leaky RELU*[13].

### 2. YOLOv5

Pada penelitian ini versi YOLO yang digunakan adalah versi 5 atau YOLOv5[14]. Jocher meluncurkan YOLOv5 secara publik pada 9 Juni 2020 dan masih diperbarui hingga sekarang. Perilisan YOLOv5 mencakup empat ukuran model utama yang berbeda, yaitu YOLOv5s, yang terkecil; YOLOv5m, sedang; Dan YOLOv5l, besar; dan YOLOv5x, yang terbesar. Saat dirilis, YOLOv5 awalnya hanya ditujukan untuk ukuran gambar 640 piksel, tetapi sekarang juga menawarkan ukuran 1280 piksel.

Selain itu, arsitektur YOLOv5 memiliki *Cross Stage Partial Connection* (CSP) *backbone* dan *neck* PANET, seperti YOLOv4. Namun, YOLOv5 menggunakan *PyTorch* alih-alih menggunakan *Darknet* seperti yang digunakan pada versi asli YOLO. Peningkatan signifikan dalam YOLOv5 meliputi augmentasi data mosaik dan pembelajaran otomatis pada *bounding box anchors*. [15] Arsitektur dari jaringan YOLOv5 ditunjukkan pada Gambar 8.



GAMBAR 8. ARSITEKTUR YOLOV5

#### D. Evaluasi

Pengukuran evaluasi yang digunakan meliputi metrik *Precision* (P), *Recall* (R), dan *Average Precision* (AP). AP mengukur kombinasi *Recall* dan *Precision* untuk hasil pengambilan peringkat dan merupakan *average precision* pada berbagai nilai *recall*. [15] Rumus untuk menghitung *Precision* dan *Recall* mengacu pada :

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

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

Dimana :

1. TP : True Positive
2. FP : False Positive
3. FN : False Negative

Nilai *Average Precision* (AP) biasanya digunakan sebagai indeks performa untuk deteksi objek. Ini mewakili akurasi model dalam kategori tertentu, yang dapat dihitung dengan area di bawah kurva *Precision-Recall* (P-R). [16] Persamaan untuk *Average Precision* mengacu pada :

$$AP = \int_0^1 P(R) dR \quad (3)$$

#### E. Pengaturan Percobaan

Pada Penelitian ini, *training* model YOLOv5 dilakukan pada sistem yang menggunakan GPU NVIDIA RTX 3070 8GB VRAM, 16 GB RAM DDR4 3200 mHz, Ryzen 5 5600X (6 Cores 12 Threads 6 Clock Speed up to 4.6 GHz). Model YOLOv5 yang digunakan adalah YOLOv5m *medium*. Parameter *training* yang digunakan dapat dilihat pada Tabel 1.

Parameter	Nilai
Class	2
Batch Size	16
Epoch	150
Image Size	640x640

Konfigurasi yang digunakan untuk *batch size* adalah sebesar 16 karena Ketika menggunakan *batch*

*size* yang lebih besar yaitu 32 atau 64, sistem yang digunakan tidak dapat melakukan *training* karena terjadi *error* “*Cuda Out of Memory*” pada GPU RTX 3070 8GB yang digunakan. Sama halnya dengan model *pre-trained* yang digunakan, Ketika menggunakan model *pre-trained* YOLOv5 yang lebih besar seperti YOLOv5l atau YOLOv5x.

*Batch size* yang digunakan adalah 16 karena keterbatasan sistem yang digunakan, dengan menggunakan *batch size* yang lebih besar seperti 32 atau 64 membutuhkan memori VRAM yang lebih besar. Begitu pula dengan *image size* yang digunakan, karena dengan *image size* yang lebih besar.

#### IV. EVALUASI

##### A. Data

Data yang digunakan pada penelitian ini menggunakan total 926 data gambar *candlestick chart* yang diambil dari saham yang terdaftar pada NASDAQ dan telah diberikan label *double top* atau *double bottom*, yang selanjutnya akan dibagi menjadi 3 skenario pengujian. Pada dataset persebaran data untuk label *double top* dan *double bottom* ditunjukkan pada Tabel 2.

Pola	Jumlah Label
Double Top	705
Double Bottom	619

##### B. Perbandingan Akurasi YOLOv3 dan YOLOv5

Pada *datasets candlestick chart* yang digunakan pada penelitian ini, versi YOLOv5 memiliki performansi yang lebih baik jika dibandingkan dengan versi YOLOv3. Rata-rata perbandingan performansi yang didapatkan pada *datasets candlestick chart* untuk mendeteksi objek pola *double top* dan *double bottom* dapat dilihat pada Tabel 3.

Versi YOLO	mAP <sup>50</sup> (%)
YOLOv3	50,47
YOLOv5	75,9

YOLOv5 memiliki akurasi yang lebih tinggi yaitu 75,9% jika dibandingkan dengan YOLOv3 yaitu 50,47%, sehingga secara keseluruhan YOLOv5 memiliki performansi yang lebih baik pada *datasets candlestick chart*.

##### C. Skenario dan Hasil Uji

Model YOLOv5m yang dihasilkan dari hasil *training* sebelumnya akan diuji coba pada 10 saham yang berbeda dengan rentang waktu 1 tahun, sejak 25 Mei 2022 hingga 25 Mei 2023. Pada penelitian ini pengujian dilakukan dengan 3 skenario, yaitu :

1. Pengujian performansi model YOLOv5m menggunakan 186 data *train* dan 47 data *test*.
2. Pengujian performansi model YOLOv5m menggunakan 372 data *train* dan 93 data *test*.

3. Pengujian performansi model YOLOv5m menggunakan 741 data *train* dan 185 data *test*.

#### 1. Skenario Pengujian Pertama

Skenario pertama dilakukan dengan mengukur performansi algoritma YOLO pada model YOLOv5m yang sudah *ditraining* sebelumnya menggunakan model dengan 186 data *train* dan 47 data *test* yang digunakan untuk *training*. Nilai akurasi  $mAP^{50}$ , *precision* dan *recall* pada skenario pertama dapat dilihat pada Tabel 4.

TABEL 4. HASIL PENGUJIAN SKENARIO PERTAMA

<i>Class</i>	<i>Precision</i> (%)	<i>Recall</i> (%)	$mAP^{50}$ (%)
<i>All</i>	<b>77</b>	<b>59,5</b>	<b>64,3</b>
<i>Double Top</i>	63,5	51,3	53,4
<i>Double Bottom</i>	90,4	67,6	75,2

Lalu model hasil training tersebut akan diujicoba untuk mendeteksi pola *double top* atau *double bottom* pada 10 saham yang berbeda. Hasil uji coba untuk pengujian pertama dapat dilihat pada tabel 5.

TABEL 5. UJI COBA MODEL PENGUJIAN SKENARIO PERTAMA

<i>Double Top</i>	70,1	74,3	71
<i>Double Bottom</i>	86,3	60,9	69,1

Lalu model hasil training tersebut akan diujicoba untuk mendeteksi pola *double top* atau *double bottom* pada 10 saham yang berbeda. Hasil uji coba untuk pengujian pertama dapat dilihat pada tabel 7.

TABEL 7. UJI COBA MODEL PENGUJIAN SKENARIO KEDUA

<i>Saham</i>	<i>Double Bottom</i>	<i>Double Top</i>
ACES	1	1
ALTO	1	1
BBCA	1	0
BEST	0	1
BIPI	0	1
BTEK	1	2
CENT	1	1
CINT	1	1
CTRA	1	2
GOLD	1	2

#### 3. Skenario Pengujian Ketiga

Skenario pertama dilakukan dengan mengukur performansi algoritma YOLO pada model YOLOv5m yang sudah *ditraining* sebelumnya menggunakan model dengan 741 data *train* dan 185 data *test* yang digunakan untuk *training*. Nilai akurasi  $mAP^{50}$ , *precision* dan *recall* pada skenario pertama dapat dilihat pada Tabel 8.

<i>Saham</i>	<i>Double Bottom</i>	<i>Double Top</i>
ACES	2	0
ALTO	1	0
BBCA	1	1
BEST	1	1
BIPI	0	1
BTEK	1	1
CENT	2	1
CINT	0	1
CTRA	0	1
GOLD	0	2

#### 2. Skenario Pengujian Kedua

Skenario pertama dilakukan dengan mengukur performansi algoritma YOLO pada model YOLOv5m yang sudah *ditraining* sebelumnya menggunakan model dengan 372 data *train* dan 93 data *test* yang digunakan untuk *training*. Nilai akurasi  $mAP^{50}$ , *precision* dan *recall* pada skenario pertama dapat dilihat pada Tabel 6.

TABEL 6. HASIL PENGUJIAN SKENARIO KEDUA

<i>Class</i>	<i>Precision</i> (%)	<i>Recall</i> (%)	$mAP^{50}$ (%)
<i>All</i>	<b>78,5</b>	<b>67,6</b>	<b>70,1</b>

TABEL 8. HASIL PENGUJIAN SKENARIO KETIGA

<i>Class</i>	<i>Precision</i> (%)	<i>Recall</i> (%)	$mAP^{50}$ (%)
<i>All</i>	<b>72,3</b>	<b>69,8</b>	<b>75,9</b>
<i>Double Top</i>	73,8	66,9	75,5
<i>Double Bottom</i>	70,8	72,6	76,3

Lalu model hasil training tersebut akan diujicoba untuk mendeteksi pola *double top* atau *double bottom* pada 10 saham yang berbeda. Hasil uji coba untuk pengujian pertama dapat dilihat pada tabel 9.

TABEL 9. UJI COBA MODEL PENGUJIAN SKENARIO KETIGA

<i>Saham</i>	<i>Double Bottom</i>	<i>Double Top</i>
ACES	1	1
ALTO	2	0
BBCA	2	1
BEST	1	1
BIPI	1	0
BTEK	2	0
CENT	3	0
CINT	1	0
CTRA	1	2
GOLD	1	2

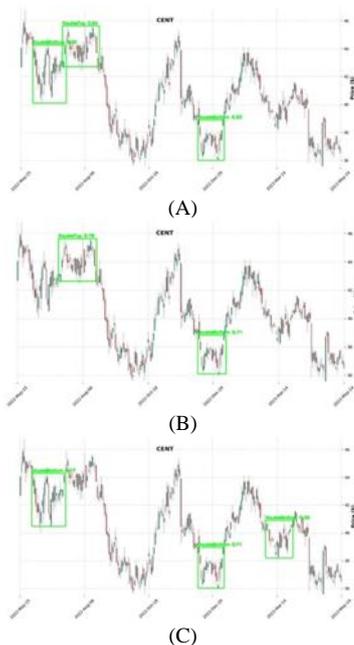
#### D. Analisis

YOLOv5 memiliki performansi akurasi yang lebih baik jika dibandingkan dengan YOLOv3, karena pada YOLOv5 menggunakan algoritma *auto*

*learning bounding boxes*[17], sehingga dapat meningkatkan akurasi secara keseluruhan pada algoritma. Dan pada YOLOv3 menggunakan *backbone* Darknet53, yang dimana memiliki kesulitan untuk mendeteksi objek yang kecil, sedangkan YOLOv5 menggunakan CSPdarknet53 yang dapat meningkatkan akurasi secara signifikan pada *datasets candlestick chart*.

Pada penelitian ini juga dilakukan uji coba pada jumlah *datasets* yang berbeda untuk YOLOv5, dan didapatkan hasil bahwa semakin banyak *datasets* yang digunakan saat melakukan *training* pada YOLOv5, akurasi dari model yang dihasilkan cenderung memiliki akurasi yang lebih baik, hal ini dikarenakan semakin banyak *datasets* yang digunakan, maka data yang digunakan akan semakin beragam yang memungkinkan model untuk melihat lebih banyak variasi objek yang dapat dipelajari.

Lalu, model diuji coba pada 10 saham yang berbeda menggunakan tiga skenario berbeda pada jumlah *datasets* yang digunakan untuk melakukan *training*. Hasil yang didapatkan bahwa ketiga model tersebut dapat mendeteksi pola yang terdapat pada *candlestick chart* suatu saham, namun dengan hasil yang dapat berbeda dari masing-masing model yang digunakan pada suatu gambar *candlestick chart* seperti contoh yang ditampilkan pada Gambar 9.



GAMBAR 9. *OUTPUT DETECTION IMAGES SAHAM CENT*  
(A) SKENARIO PENGUJIAN PERTAMA (B) SKENARIO  
PENGUJIAN KEDUA (C) SKENARIO PENGUJIAN KETIGA

Seperti yang dapat dilihat pada Gambar 9, untuk contoh kasus pada saham CENT terdapat perbedaan untuk objek pola *candlestick* yang terdeteksi pada masing-masing skenario pengujian.

## V. KESIMPULAN

Dari perbandingan akurasi versi YOLO yang ditampilkan pada Tabel 1, akurasi dari versi YOLOv5 memiliki performansi yang lebih baik jika dibandingkan dengan versi terdahulunya yaitu YOLOv3 pada *datasets candlestick chart* saham. Algoritma YOLO dengan menggunakan YOLOv5 dan *pre-trained model* YOLOv5m memiliki performansi rata-rata sebesar 64,3% pada 186 data *train* dan 47 data *test*, 70,1% pada 372 data *train* dan 93 data *test*, 75,9% pada 741 data *train* dan 185 data *test*. Berdasarkan hasil tersebut performansi akurasi dari YOLO meningkat sesuai dengan bertambahnya jumlah dari *datasets* yang digunakan pada saat melakukan *training model* dari YOLO. Model dari YOLOv5 yang dibuat juga dapat digunakan untuk mendeteksi pola dari *double bottom* dan *double bottom* pada suatu saham.

Pada penelitian selanjutnya dapat dilakukan pengujian dengan terhadap pola lainnya seperti *bearish flag*, *head and shoulder*, dll. Juga dilakukan menggunakan versi *pre-trained* YOLOv5 lainnya seperti YOLOv5l atau YOLOv5x. Dan juga dapat dilakukan menggunakan versi YOLO terbaru lainnya seperti YOLOv7 atau YOLOv8 dengan menggunakan spesifikasi sistem untuk *training* yang lebih besar.

## REFERENSI

- [1] Haoxuan Qiu and Fanzhuoqun Liu, "Candlestick Analysis in Forecasting U.S. Stock Market: Are They Informative and Effective," *IEEE International Conference on Big Data Analytics*, 2019.
- [2] V. Skuratov, K. Kuzmin, I. Nelin, and M. Sedankin, "APPLICATION OF A CONVOLUTIONAL NEURAL NETWORK TO CREATE A DETECTOR OF TECHNICAL ANALYSIS FIGURES ON EXCHANGE QUOTES CHARTS," *EUREKA: Physics and Engineering*, vol. 6, pp. 50–56, Nov. 2019, doi: 10.21303/2461-4262.2019.001055.
- [3] S. Birogul, G. Temur, and U. Kose, "YOLO Object Recognition Algorithm and 'Buy-Sell Decision' Model over 2D Candlestick Charts," *IEEE Access*, vol. 8, pp. 91894–91915, 2020, doi: 10.1109/ACCESS.2020.2994282.
- [4] C. B. Kaya, A. Yılmaz, G. N. Uzun, and Z. H. Kilimci, "Stock Pattern Classification from Charts using Deep Learning Algorithms," *Academic Perspective Procedia*, vol. 3, no. 1, pp. 445–454, Oct. 2020, doi: 10.33793/acperpro.03.01.89.
- [5] A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, "The Real-Time Detection of Traffic Participants Using YOLO Algorithm," in *2018 26th Telecommunications Forum (TELFOR)*,

- IEEE, Nov. 2018, pp. 1–4. doi: 10.1109/TELFOR.2018.8611986.
- [6] M. Velay and F. Daniel, “Stock Chart Pattern recognition with Deep Learning,” 2018. [Online]. Available: <http://www.lusis.fr>
- [7] A. Ramadhan, I. Palupi, and B. A. Wahyudi, “Candlestick Patterns Recognition using CNN-LSTM Model to Predict Financial Trading Position in Stock Market,” 2022, doi: 10.47065/josyc.v3i4.2133.
- [8] W. Yang and Z. Jiachun, “Real-time face detection based on YOLO,” in *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, IEEE, Jul. 2018, pp. 221–224. doi: 10.1109/ICKII.2018.8569109.
- [9] Shenghui Wang, Leilei Niu, and Nan Li, “Research on Image Recognition of Insulators Based on YOLO Algorithm,” *International Conference on Power System Technology*, 2018.
- [10] Eryk Lewinson, “Matplotlib/mplfinance: Financial Markets Data Visualization using matplotlib.”
- [11] Tzutalin, “LabelImg is a graphical image annotation tool.,” <https://github.com/heartexlabs/labelImg>.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [13] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” Feb. 2015.
- [14] “Jocher, G. yolov5. 2021. Available online: <https://github.com/ultralytics/yolov5> (accessed on 31 January 2022).”
- [15] A. J. Mantau, I. W. Widayat, J.-S. Leu, and M. Köppen, “A Human-Detection Method Based on YOLOv5 and Transfer Learning Using Thermal Image Data from UAV Perspective for Surveillance System,” *Drones*, vol. 6, no. 10, 2022, doi: 10.3390/drones6100290.
- [16] L. Chang, Y.-T. Chen, J.-H. Wang, and Y.-L. Chang, “Modified Yolov3 for Ship Detection with Visible and Infrared Images,” *Electronics (Basel)*, vol. 11, no. 5, p. 739, Feb. 2022, doi: 10.3390/electronics11050739.
- [17] W. Soudiene Mseddi, M. A. Sedrine, and R. Attia, “YOLOv5 Based Visual Localization For Autonomous Vehicles,” in *2021 29th European Signal Processing Conference (EUSIPCO)*, IEEE, Aug. 2021, pp. 746–750. doi: 10.23919/EUSIPCO54536.2021.9616354.