

Deteksi Objek Serupa Menggunakan You Only Look Once (YOLO3.0)

Sri Desy Siswanti¹, Kharisma², Ahmad Fali³, Huda Ubaya⁴, Iman Saladin⁵, Ghuftron Mubaroq⁶

^{1,2,3,4,5} Sistem Komputer, Universitas Sriwijaya, ^{1,6} Teknik Elektro, Universitas Nurtanio

e-mail: desy0712@gmail.com, mubaroqghuftron@gmail.com

Abstrak

Saat ini deteksi objek yang digabungkan dengan sistem AI sering digunakan untuk mendeteksi (objek dalam sebuah gambar. Hal ini telah diaplikasikan dalam kehidupan sehari-hari diantaranya di bidang pertahanan, sistem pengawasan suatu kota maupun dipakai pada fitur mobil untuk meminimumkan kecelakaan. Dalam paper ini fokus membahas deteksi objek menggunakan YOLO3. Objek yang dideteksi dalam paper ini adalah truk dan bus, kedua kendaraan tersebut merupakan kendaraan beroda empat berbentuk persegi panjang yang sering ditemukan di jalan. Kedua kendaraan ini merupakan objek berbentuk mirip dan kadang keliru untuk identifikasi kedua objek tersebut. Sistem ini dimulai dari membuat dataset gambar truk dan bus, dimana dataset ini terdiri dari data training dan dataset. Kemudian dilanjutkan dengan proses ekstraksi fitur yang menggunakan metode Darknet-53 dan deteksi objek menggunakan Feature Pyramid Network (FPN), akhirnya jika dikenali maka objeknya akan diberikan bounding box. Tujuan dari proses ini mendapatkan nilai akurasi dengan objek yang memiliki bentuk yang mirip dan mencari faktor yang mempengaruhi nilai akurasi tersebut. Hasil proses deteksi objek menggunakan YOLO3 dapat menaikkan nilai akurasi pada objek dengan bentuk serupa, walaupun terdapat beberapa kelemahan. Faktor yang mempengaruhi nilai akurasi yaitu nilai threshold yang sangat mempengaruhi dalam membedakan bentuk objek yang satu dengan objek yang lain terutama jika objek tersebut memiliki bentuk yang mirip.

Kata kunci: YOLO3, objek, mirip, truk, bus.

Abstract

Currently object detection combined with AI systems is often used to detect objects in an image. This has been applied in everyday life, including in the field of defense, city surveillance systems or used in car features to minimize accidents. This paper focuses on discussing object detection using YOLO3. The objects detected in this paper are trucks and buses because the two objects have a similar shape. This system starts with creating a truck and bus image dataset, where this dataset consists of training data and dataset. Then proceed with the feature extraction process using the Darknet-53 method and object detection using the Feature Pyramid Network (FPN). Finally, if recognized, the object will be given a bounding box. The purpose of this process is to get an accuracy value with objects that have a similar shape and look for factors that affect the accuracy value. The results of the object detection process using YOLO3 can increase the accuracy value even though there are some weaknesses. The factor that affects the accuracy value is the threshold value which is very influential in distinguishing the shape of an object from other objects, especially if the object has a similar shape.

Keyword: YOLO3, object, similar, truck, bus

1. PENDAHULUAN

Persepsi visual manusia terhadap objek dalam suatu gambar, ilmu komputer visi dalam deteksi objek dapat memberikan informasi dari objek yang terdeteksi[1]. Pada *Unmanned Ariel vehicle* (UAV) atau drone untuk mengecek kondisi dari fasilitas yang di targetkan dapat menggunakan deteksi objek, jika dibandingkan pemantauan dengan secara manual, deteksi objek dapat lebih membantu, lebih aman dan menghemat waktu[2]. Salah satu aplikasi dalam dunia nyata dari deteksi objek yaitu pengawasan kota dengan monitoring CCTV yaitu orang, hewan, kendaraan[3][4]. Mobil KIA Sedona Hi_limousine telah menerapkan deteksi objek dalam fitur bantuan pengemudi (*Advance Driver Assistance System*) untuk mengurangi risiko kecelakaan. Fitur tersebut dapat deteksi objek yaitu kendaraan, pengendara sepeda dan pejalan kaki. Algoritma proses pengolahan citra [3]digunakan untuk mengekstraksi bagian objek yang diinginkan. *Image retrieval*, keamanan, medis dan pertahanan merupakan aplikasi pengolahan citra.

Melengkapi sebuah komputer dengan kemampuan dan kepintaran manusia, *Artificial Intelligence* (AI) merupakan suatu cabang dalam bidang ilmu komputer yang mengkaji tentang hal tersebut[5]. Dalam mengenali objek suatu gambar manusia dapat melihat langsung serta dapat mengenalinya. Untuk menyerupai kemampuan tersebut diperlukan sebuah algoritma yang dapat pemrosesan gambar secara cepat dan akurat sehingga dapat membantu pengguna mendapatkan informasi objek secara langsung dan nyata. Teknologi yang memiliki kemampuan seperti persepsi manusia secara efisiensi dengan kinerja yang luar biasa yaitu teknologi Deep Learning (DL)[5]. Jika menemukan objek yang memiliki bentuk fitur yang mirip sehingga sulit dikenali maka sangat dibutuhkan DL untuk mengenali objek tersebut secara akurat dan cepat. Hal ini merupakan tantangan para peneliti untuk mengolah data objek tersebut sehingga dapat mendeteksi, lokalisasi serta klasifikasi objek.

You Only Look Once (YOLO) merupakan salah satu metode DL untuk deteksi objek yang banyak digunakan oleh peneliti karena memiliki kemampuan deteksi objek dengan cepat dan akurat. YOLO3 memiliki kemampuan hampir sama bahkan 3.8x lebih cepat daripada RetinaNet yang dinyatakan dalam penelitian J Redmon [6]. Jika dibandingkan dengan algoritma Faster R-CNN, YOLO3 memiliki sensitivitas lebih baik dengan akurasi 99,07% dan YOLO3 telah banyak digunakan dalam berbagai aplikasi diantaranya untuk deteksi mobil pada UAV[7]. YOLO3 telah diimplementasikan untuk pemantauan lalu lintas berdasarkan jumlah kendaraan berdasarkan klasifikasi kendaraan (mobil, truk, bus dan sepeda motor), yang dilakukan oleh penelitian[8] dengan tingkat akurasi deteksi objek rata-rata 90,8%.

Berdasarkan hal tersebut, objek yang mirip atau serupa menjadi pokok penulis dalam melakukan penelitian di bidang deteksi objek.

2. METODE PENELITIAN

2.1 Riset Yang Relevan Dengan Penelitian Deteksi Objek

Beberapa penelitian terkait dengan deteksi objek akan menjadi perhatian utama dalam penelitian ini.

1. Pada tahun 2020, Dufan J.P.Manajang dan rekan-rekannya melakukan penelitian yang mengimplementasikan YOLO3 menggunakan Tensorflow untuk mengidentifikasi jenis kendaraan bermotor (mobil, truk, bus, dan sepeda motor). Hasil dari penelitian tersebut menunjukkan tingkat akurasi rata-rata sebesar 90,8%. Meskipun demikian, penelitian ini tidak mengulas faktor-faktor yang memengaruhi tingkat akurasi YOLO3, terutama ketika mengidentifikasi objek yang memiliki bentuk serupa [8].
2. Penelitian yang dilakukan oleh Lazaro dan timnya pada tahun 2017 bertujuan untuk mendeteksi berbagai jenis kendaraan dengan menggunakan metode konvensional Haar-Feature untuk mengklasifikasikan kendaraan ke dalam kategori kecil, sedang, dan besar. Dalam penelitian ini, metode tersebut diaplikasikan untuk mengukur kepadatan lalu lintas dengan tingkat akurasi rata-rata sebesar 77,8%, yang merupakan tingkat akurasi yang relatif rendah. Namun, penelitian ini kurang menggambarkan dengan jelas kategori kendaraan kecil, sedang, dan besar. Seharusnya, penelitian ini mencakup jenis kendaraan seperti truk/bus, sedan, sepeda motor, dan sebagainya untuk klasifikasinya.[9].
3. Pada tahun 2017, Muhammad Irfan dan rekan-rekannya melakukan penelitian mengenai pengklasifikasian kendaraan (Mobil, Bus, dan Truk) dengan menggunakan metode Multilayer Perceptron dalam konteks pemantauan lalu lintas. Hasil penelitian ini menunjukkan tingkat akurasi sebesar 92,67% [10].
4. Pada tahun 2019, Nur Fadlia dan timnya melakukan penelitian mengenai pengklasifikasian jenis kendaraan (Mobil, Motor, Sepeda) menggunakan Convolutional Neural Network (CNN) dengan tingkat akurasi sebesar 73,3%, yang dapat dianggap sebagai tingkat akurasi yang relatif rendah [11].

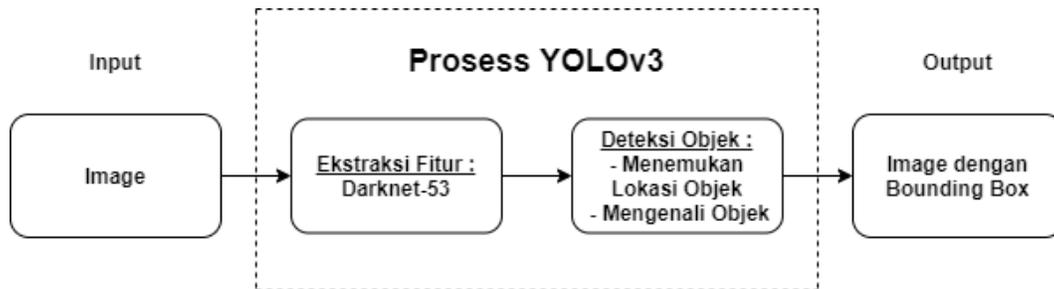
2.2 Deteksi Objek

Domain penelitian dalam bidang penglihatan komputer [12], penggunaan gambar dan video digunakan untuk mendeteksi, mengklasifikasikan, dan melacak objek atau peristiwa dalam upaya untuk memahami konteks dunia nyata. Visi komputer memberikan daya guna pada berbagai aplikasi seperti pencarian gambar, navigasi robot, analisis citra medis, manajemen foto, dan banyak lainnya. Deteksi objek digunakan untuk mengidentifikasi keberadaan serta mengidentifikasi objek dalam gambar, sementara pengenalan objek melibatkan pelatihan model menggunakan dataset yang berisi objek-objek yang berbeda, mengelompokkan objek menjadi dua kategori, yakni kelas yang mewakili objek dan kelas yang mewakili bukan objek.

Dalam konteks komputer, gambar dianggap sebagai representasi visual yang terdiri dari berbagai objek yang berbeda dan latar belakang yang dibentuk oleh elemen-elemen yang ada dalam gambar tersebut. Dalam bidang deteksi objek, prosesnya biasanya melibatkan pencarian di seluruh bagian gambar yang sudah diidentifikasi secara lokal, yang memiliki sifat geometris yang sesuai dengan objek target dalam database pelatihan. Ini dapat dicapai dengan melakukan pemindaian objek menggunakan berbagai template yang berbeda-beda dalam hal lokasi, skala, dan rotasi, dan deteksi dianggap berhasil jika terdapat tingkat kesamaan yang cukup tinggi antara objek template dan objek yang terdapat dalam gambar [12].

Sistem deteksi objek yang mirip ini melibatkan proses pengambilan data, pengolahan data, ekstraksi fitur, dan klasifikasi. Kerangka dasar dari sistem ini dapat dilihat dalam gambar 1. Tahapan-tahapan dalam deteksi dan pengenalan objek adalah sebagai berikut,

1. Pembuatan dataset melibatkan pengumpulan gambar objek, yang dalam hal ini adalah truk dan bus, dengan resolusi 450 x 300 piksel dalam format jpg. Dataset yang telah disusun terdiri dari 150 gambar, yang terbagi menjadi 120 data untuk latihan dan 30 data untuk pengujian.
2. Pengambilan fitur objek dilakukan dengan menerapkan metode Darknet-53.
3. Dalam mendeteksi objek, digunakan metode *Feature Pyramid Network* (FPN) untuk mendeteksi lokasi dan mengidentifikasi objek.
4. Objek yang berhasil terdeteksi dan diidentifikasi akan diberikan kotak pembatas (*Bounding Box*).



Gambar 1. Diagram Konsep Sistem Deteksi Objek Serupa

2.2.1 Akuisisi Citra Objek

Objek yang disoroti dalam makalah ini adalah bus dan truk, dikarenakan kedua objek tersebut memiliki kemiripan dan kesamaan dalam bentuknya. Ilustrasi beberapa gambar contoh objek yang digunakan dalam makalah ini dapat dilihat pada gambar 2 dan gambar 3.



Gambar 2. Contoh Objek Bus yang Digunakan

Bus adalah kendaraan besar berjenis bermotor untuk angkutan umum yang memiliki kapasitas penumpang besar dan berbentuk persegi panjang.

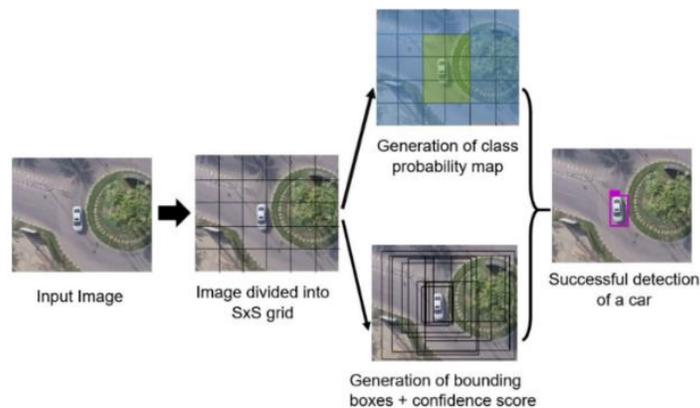


Gambar 3. Contoh Objek Truk Yang Digunakan

Truk adalah kendaraan bermotor besar yang digunakan untuk angkutan umum dengan kapasitas penumpang besar. Truk ini memiliki kemiripan dengan bus dalam bentuknya, yang juga berbentuk besar dengan dimensi persegi panjang.

2.2.2 *You Only Look Once 3 (YOLO3)*

You Only Look Once (YOLO) merupakan salah satu model jaringan saraf tiruan yang digunakan dalam deteksi objek yang berbasis pada CNN. YOLO pertama kali ditemukan dan diumumkan oleh J. Redmon pada tahun 2015. YOLO bertujuan untuk meningkatkan kecepatan dalam mendeteksi objek, yang berbeda dari sebagian besar metode lain yang cenderung menggunakan pendekatan yang lebih rumit [13]. Cara kerja YOLO3 ditunjukkan pada gambar 4.



Gambar 4. Cara Kerja YOLO3

YOLO3 adalah perkembangan lanjutan dari versi sebelumnya YOLO. Perbaikan pertama yang diperkenalkan dalam YOLO3 adalah penggunaan klasifikasi multi-label, yang berbeda dari pendekatan label eksklusif yang digunakan sebelumnya. Ini menggunakan classifier logistik untuk menghitung probabilitas bahwa suatu objek adalah label tertentu. Perbaikan kedua adalah dalam prediksi bounding box, di mana skor objektivitas 1 diberikan kepada kotak pembatas anchor yang paling cocok dengan objek yang sesuai dengan data sebenarnya, sementara anchor lain yang memiliki tumpang tindih dengan objek di bawah ambang batas diabaikan. Oleh karena itu, YOLO3 menetapkan satu kotak pembatas anchor untuk setiap objek sebenarnya. Perbaikan ketiga adalah penggunaan prediksi lintas skala dengan bantuan jaringan fitur piramida. YOLO3 membuat prediksi pada tiga skala yang berbeda dan kemudian mengekstrak fitur dari masing-masing skala tersebut. Hasil prediksi dari jaringan ini adalah tensor tiga dimensi yang mencakup informasi mengenai bounding box, skor objektivitas, dan prediksi kelas. Inilah sebabnya mengapa dimensi tensor berakhir dengan perubahan dari versi sebelumnya menjadi yang terlihat seperti ini:

$$N \times N \times (3 * (4 + 1 + C)) \quad (1)$$

Keterangan :

$N \times N$: jumlah grid pada sistem

3 : untuk mendecode fitur yang diekstraksi dari masing-masing 3 skala

4+1 : untuk mendecode bounding boxes offsets + objectness score

C : Jumlah class yang kita latih pada sistem

2.2.3 Darknet-53

Darknet adalah alat ekstraksi fitur yang dikembangkan oleh Joseph Redmon. Dalam YOLO3, Darknet-53 digunakan untuk mengambil fitur dari sebuah gambar dan menghasilkan peta fitur yang akan digunakan sebagai masukan dalam proses deteksi objek [13]. Darknet-53 adalah suatu jenis *Convolutional Neural Network* (CNN) yang terdiri dari 53 lapisan dan memanfaatkan *skip connections network* yang diilhami oleh ResNet. Darknet-53 juga menggunakan lapisan konvolusi dengan kernel 3x3 dan 1x1. Model ini telah terbukti memberikan tingkat akurasi yang baik, sambil mengurangi penggunaan operasi floating point dan meningkatkan kecepatan [7]. Arsitektur Darknet-53 dapat dilihat pada gambar 5.

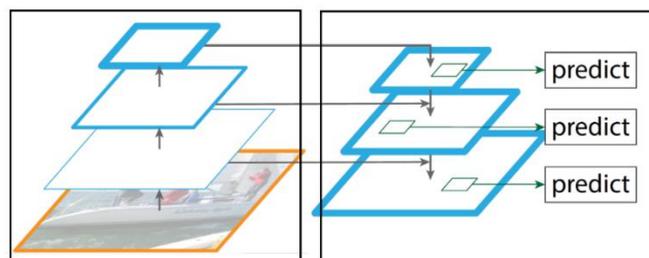
	Type	Filter	Stride	Output
	Input Image			w x h
	Convolutional	3x3x32	1	w x h
	Convolutional	3x3x64	2	w/2 x h/2
1x	Convolutional	1x1x32	1	
	Convolutional	3x3x64	1	
	Residual			w/2 x h/2
	Convolutional	3x3x128	2	w/4 x h/4

2x	Convolutional	1x1x64	1	
	Convolutional	3x3x128	1	
	Residual			w/4 x h/4
8x	Convolutional	3x3x256	2	w/8 x h/8
	Convolutional	1x1x128	1	
	Convolutional	3x3x56	1	
8x	Residual			w/8 x h/8
	Convolutional	3x3x512	2	w/16 x h/16
	Convolutional	1x1x256	1	
8x	Convolutional	3x3x512	1	
	Residual			w/16 x h/16
	Convolutional	3x3x1024	2	w/32 x h/32
4x	Convolutional	1x1x512	1	
	Convolutional	3x3x1024	1	
	Residual			w/32 x h/32

Gambar 5. Arsitektur Ekstraksi Fitur Darknet-53

2.2.4 Feature Pyramid Network (FPN)

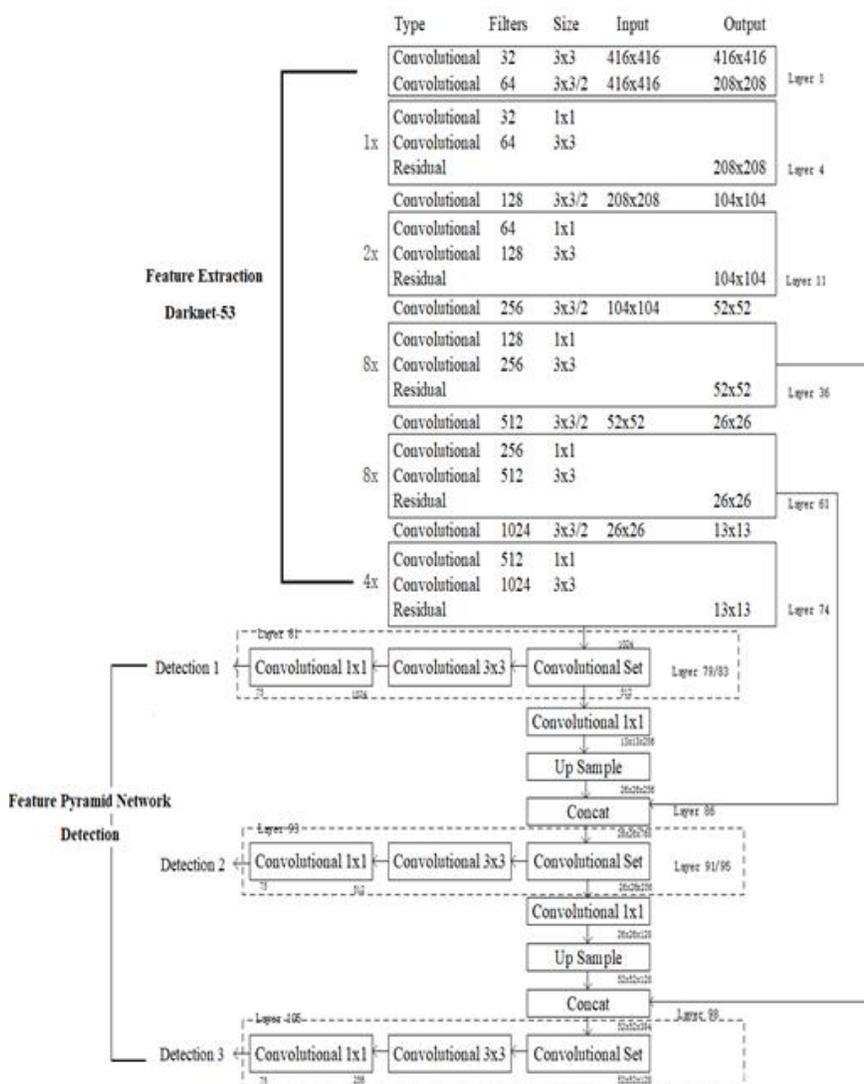
Dalam YOLO3, langkah deteksi mengadopsi algoritma *Feature Pyramid Network* (FPN). *Feature pyramid* adalah elemen kunci dalam sistem deteksi objek untuk mengenali objek pada berbagai skala. FPN adalah suatu arsitektur yang memanfaatkan arsitektur top-down dengan mengintegrasikan koneksi lateral untuk membangun peta fitur semantik tingkat tinggi pada semua skala. [14]. Pada YOLO3, FPN diterapkan hanya pada langkah deteksi, sedangkan untuk mengambil fitur, metode yang berbeda, yaitu Darknet-53, digunakan. Hal tersebut terlihat pada keseluruhan gambaran sistem deteksi objek dalam YOLO3 pada gambar 6 dan gambar 7.



Ekstraksi fitur Darknet-53

FPN

Gambar 6. Sistem YOLO3



Gambar 7. Diagram Alur YOLO3

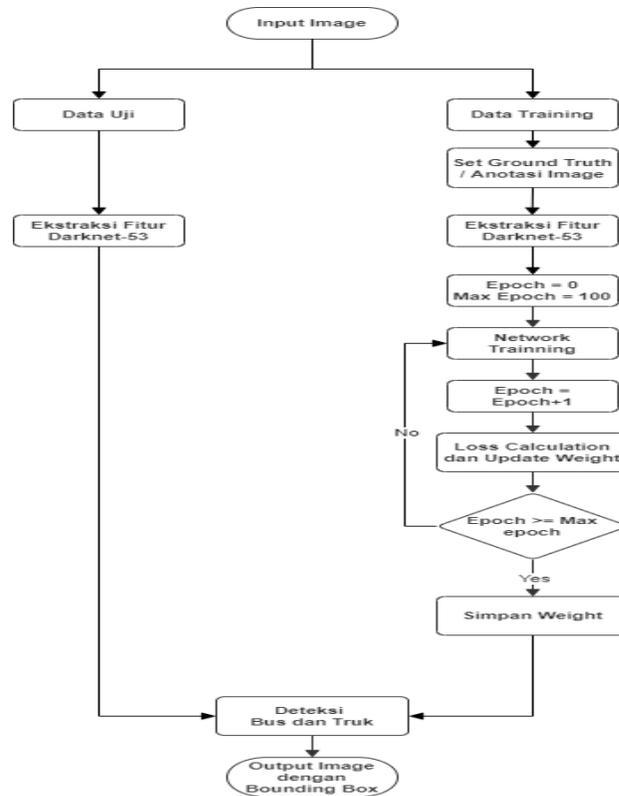
3 HASIL DAN PEMBAHASAN

Dataset objek yang digunakan dalam penelitian ini dihasilkan dengan menggunakan kamera smartphone yang memiliki resolusi 450x300 piksel. Pemilihan resolusi ini dilakukan untuk mempertahankan konsistensi dengan beberapa standar yang terdapat dalam penelitian deteksi objek yang terdokumentasi dalam referensi. Objek yang menjadi subjek penelitian adalah bus dan truk. Total jumlah objek yang digunakan dalam dataset ini adalah sebanyak 150, dan rinciannya adalah sebagai berikut :

Tabel 1. Rincian Dataset Citra Objek

	Data Traning	Data Uji	Keseluruhan
Bus	60	30	
Truk	60	30	
Jumlah	120	30	

Rancangan sistem deteksi objek yang serupa dapat dilihat dalam ilustrasi pada gambar 8.



Gambar 8. Diagram Alur Pelaksanaan Sistem Deteksi Objek Serupa

3.1 Pembuatan Data Set

Citra referensi atau citra yang telah diberi label dibuat dengan melakukan proses segmentasi secara manual menggunakan aplikasi Microsoft Visual Object Tagging Tool (VoTT). Proses pelabelan citra dilakukan secara manual selama tahap pelatihan, dengan memberikan anotasi pada objek bus dan truk sesuai dengan gambar yang akan digunakan

dalam proses pelatihan. Gambaran tentang anotasi objek bus dan truk yang diterapkan secara manual dapat dilihat pada gambar 9 dan gambar 10.



Gambar 9. Pemberian Anotasi Pada Gambar Bus.



Gambar 10. Pemberian Anotasi Pada Gambar Truk

3.2 Pengujian Ekstraksi Fitur

Dalam penelitian ini, metode yang digunakan untuk mengekstraksi fitur adalah Darknet-53, yang merupakan salah satu komponen dari YOLOv3. Darknet-53 terdiri dari dua komponen utama, yaitu konvolusi dan residual. Proses ini melibatkan 53 lapisan konvolusi, sehingga dinamai Darknet-53. Pada setiap lapisan konvolusi, terdapat penggunaan fungsi aktivasi untuk memastikan bahwa fitur yang dihasilkan memenuhi persyaratan tertentu. Dalam konteks ini, fungsi aktivasi yang diterapkan dalam Darknet-53 adalah Leaky ReLU. Hasil dari ekstraksi fitur menggunakan Darknet-53 menghasilkan peta fitur dengan tiga skala yang berbeda, yang akan digunakan dalam tahap pelatihan dan deteksi. Hasil uji coba ekstraksi fitur yang memanfaatkan Darknet-53 ditampilkan dalam gambar 11.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 416, 416, 3)	0	
conv2d_1 (Conv2D)	(None, 416, 416, 32)	896	input_1[0][0]
batch_normalization_1 (BatchNorm)	(None, 416, 416, 32)	128	conv2d_1[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 416, 416, 32)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 208, 208, 64)	18496	leaky_re_lu_1[0][0]
batch_normalization_2 (BatchNorm)	(None, 208, 208, 64)	256	conv2d_2[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 208, 208, 64)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 208, 208, 32)	2080	leaky_re_lu_2[0][0]
batch_normalization_3 (BatchNorm)	(None, 208, 208, 32)	128	conv2d_3[0][0]
leaky_re_lu_3 (LeakyReLU)	(None, 208, 208, 32)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 208, 208, 64)	18496	leaky_re_lu_3[0][0]
batch_normalization_8 (BatchNorm)	(None, 104, 104, 64)	256	conv2d_4[0][0]
leaky_re_lu_8 (LeakyReLU)	(None, 104, 104, 64)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 104, 104, 128)	73856	leaky_re_lu_8[0][0]
batch_normalization_9 (BatchNorm)	(None, 104, 104, 128)	512	conv2d_9[0][0]
leaky_re_lu_9 (LeakyReLU)	(None, 104, 104, 128)	0	batch_normalization_9[0][0]
add_3 (Add)	(None, 104, 104, 128)	0	activation_2[0][0] leaky_re_lu_9[0][0]
activation_3 (Activation)	(None, 104, 104, 128)	0	add_3[0][0]
conv2d_10 (Conv2D)	(None, 52, 52, 256)	295168	activation_3[0][0]
batch_normalization_15 (BatchNorm)	(None, 52, 52, 256)	768	conv2d_10[0][0]
activation_5 (Activation)	(None, 52, 52, 256)	0	batch_normalization_15[0][0]
conv2d_15 (Conv2D)	(None, 52, 52, 128)	32896	activation_5[0][0]
batch_normalization_15 (BatchNorm)	(None, 52, 52, 128)	512	conv2d_15[0][0]
leaky_re_lu_15 (LeakyReLU)	(None, 52, 52, 128)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 52, 52, 256)	295168	leaky_re_lu_15[0][0]
batch_normalization_16 (BatchNorm)	(None, 52, 52, 256)	1024	conv2d_16[0][0]
leaky_re_lu_16 (LeakyReLU)	(None, 52, 52, 256)	0	batch_normalization_16[0][0]
add_8 (Add)	(None, 52, 52, 256)	0	activation_5[0][0] leaky_re_lu_16[0][0]
activation_8 (Activation)	(None, 52, 52, 256)	0	add_8[0][0]
conv2d_21 (Conv2D)	(None, 52, 52, 128)	32896	activation_8[0][0]
batch_normalization_21 (BatchNorm)	(None, 52, 52, 128)	512	conv2d_21[0][0]
leaky_re_lu_21 (LeakyReLU)	(None, 52, 52, 128)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 52, 52, 256)	295168	leaky_re_lu_21[0][0]
batch_normalization_22 (BatchNorm)	(None, 52, 52, 256)	1024	conv2d_22[0][0]
leaky_re_lu_22 (LeakyReLU)	(None, 52, 52, 256)	0	batch_normalization_22[0][0]
add_9 (Add)	(None, 52, 52, 256)	0	activation_8[0][0] leaky_re_lu_22[0][0]
activation_11 (Activation)	(None, 52, 52, 256)	0	add_9[0][0]
conv2d_27 (Conv2D)	(None, 26, 26, 512)	1188160	activation_11[0][0]
batch_normalization_27 (BatchNorm)	(None, 26, 26, 512)	2048	conv2d_27[0][0]
leaky_re_lu_27 (LeakyReLU)	(None, 26, 26, 512)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 26, 26, 256)	131328	leaky_re_lu_27[0][0]
batch_normalization_28 (BatchNorm)	(None, 26, 26, 256)	1024	conv2d_28[0][0]

Gambar 11. Model Darknet-53 Menggunakan Python

3.3 Proses Training Dataset Bus dan Truk

Pelatihan dilakukan dengan menggunakan input sebanyak 60 gambar bus dan 60 gambar truk yang telah diberi label.

```
total_bbox = 122779, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.902819, GIOU: 0.900634), Class: 0.999887, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
total_bbox = 122784, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.937467, GIOU: 0.936721), Class: 0.999923, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.704110, GIOU: 0.607483), Class: 0.974480, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
total_bbox = 122789, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.930055, GIOU: 0.928591), Class: 0.999977, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.774367, GIOU: 0.773329), Class: 0.999692, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
total_bbox = 122796, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.926142, GIOU: 0.919566), Class: 0.999951, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
total_bbox = 122801, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 82 Avg (IOU: 0.877836, GIOU: 0.874966), Class: 0.998927, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 94 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
v3 (mse loss, Normalizer: (iou: 0.75, cls: 1.00) Region 106 Avg (IOU: 0.000000, GIOU: 0.000000), Class: 0.000000, Obj: 0
total_bbox = 122807, rewritten_bbox = 0.000000 %
4000: 0.027694, 0.022852 avg loss, 0.000010 rate, 9.756386 seconds, 256000 images, 0.193904 hours left
Saving weights to backup/yolov3_custom_4000.weights
Saving weights to backup/yolov3_custom_last.weights
Saving weights to backup/yolov3_custom_final.weights
If you want to train from the beginning, then use flag in the end of training command: -clear
```

Gambar 12. Proses Training YOLOv3 Menggunakan Python di Google Colab

Pada tahap pengujian, dilakukan penggunaan 30 data berupa gambar bus, gambar truk, dan gambar yang tidak dikenali dalam program. Pengujian ini dilaksanakan menggunakan bahasa Python di platform Google Colab dengan perintah sebagaimana yang ditunjukkan dalam gambar 13.



Gambar 13. Proses Pengujian YOLOv3 menggunakan Python di Google Colab

Tabel 2. Hasil Perhitungan Performa Deteksi Bus dan Truk

Pengujian Deteksi Bus dan Truk			
	Threshold = 0.25	Threshold = 0.45	Threshold = 0.85
Accuracy	93.33%	96.66%	93.33%
Precision	92.30%	96.15%	96%
Recall	100%	100%	96%

Dalam tabel 2, setelah menjalani pengujian, hasil terbaik dalam mendeteksi Bus dan Truk, ditunjukkan dengan akurasi dan presisi tertinggi, diperoleh saat nilai ambang (threshold) diatur pada 0,45. Dalam pengujian kedua, ketika ambang nilai diatur terlalu rendah, akan terjadi beberapa kesalahan dalam deteksi, karena objek yang memiliki kemiripan bentuk dengan Bus atau Truk mungkin akan terdeteksi sebagai Bus atau Truk dengan nilai confiden yang rendah, seperti yang terjadi pada mobil.

Meskipun begitu, sistem kami yang menggunakan YOLO3 menghasilkan tingkat akurasi yang lebih tinggi daripada sistem pada penelitian lainnya. Hal ini dapat dilihat dalam tabel 3, yang menunjukkan bahwa akurasi sistem kami lebih tinggi dibandingkan dengan sistem lainnya.

Tabel 3. Hasil Perbandingan Nilai Akurasi dengan Penelitian yang lain

Algoritma	Akurasi
YOLO3	96,66%
Tensorflow[8]	90,8%
Haar Feature[9]	77,8%
Multilayer Perceptron[10]	92,67%
CNN[11]	73,4%

4 KESIMPULAN

Hasil pengujian menunjukkan bahwa dalam mendeteksi Bus dan Truk, tingkat akurasi dan presisi tertinggi tercapai saat ambang (threshold) diatur pada nilai 0,45. Pada pengujian kedua, ketika ambang nilai diatur terlalu rendah, terjadi beberapa kesalahan deteksi, karena objek yang memiliki kemiripan bentuk dengan bus atau truk akan terdeteksi sebagai bus atau truk dengan nilai confidence yang rendah, serupa dengan kasus mobil. Hal ini terjadi karena beberapa sudut pandang kamera membuat bentuk kendaraan lain tampak menyerupai bus atau truk, sehingga fitur bentuk yang terdeteksi menyerupai fitur bentuk truk dan bus. Untuk mengatasi hal ini bisa dilakukan dengan meningkatkan data training, dapat juga dengan menambahkan metode praprosesing sebelumnya sehingga fitur bentuk objek lebih terlihat oleh YOLO3.

DAFTAR PUSTAKA

- [1] T. F. Ju, W. M. Lu, K. H. Chen, and J. I. Guo, "Vision-based moving objects detection for intelligent automobiles and a robustness enhancing method," *Dig. Tech. Pap. - IEEE Int. Conf. Consum. Electron.*, vol. 1, pp. 75–76, 2014.
- [2] F. Lin, X. Zheng, and Q. Wu, "Small object detection in aerial view based on improved YoloV3 neural network," *Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020*, pp. 522–525, 2020.
- [3] A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, "Object Detection Algorithms for Video Surveillance Applications," *Proc. 2018 IEEE Int. Conf. Commun. Signal Process. ICCSP 2018*, pp. 563–568, 2018.
- [4] H. V. Nguyen and L. H. Tran, "Application of graph segmentation method in thermal camera object detection," *2015 20th Int. Conf. Methods Model. Autom. Robot. MMAR 2015*, pp. 829–833, 2015.
- [5] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *J. Phys. Conf. Ser.*, vol. 1004, no. 1, 2018.
- [6] J. Redmon and A. Farhadi, "YOLO v.3," *Tech Rep.*, pp. 1–6, 2018.
- [7] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3 Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. 2019 1st Int. Conf. Unmanned Veh. Syst. UVS 2019, pp. 1–6, 2019.
- [8] D. Manajang, S. Dompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *J. Tek. Inform.*, vol. 15, no. 3, pp. 171–178, 2020.
- [9] A. Lazaro, J. L. Buliali, and B. Amaliah, "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV," *J. Tek. ITS*, vol. 6, no. 2, 2017.
- [10] M. Irfan, B. A. Ardi Sumbodo, and I. Candradewi, "Sistem Klasifikasi Kendaraan

- Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron,” *IJEIS (Indonesian J. Electron. Instrum. Syst.*, vol. 7, no. 2, p. 139, 2017.
- [11] N. Fadlia and R. Kosasih, “Klasifikasi Jenis Kendaraan Menggunakan Metode Convolutional Neural Network (Cnn),” *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 3, pp. 207–215, 2019.
- [12] F. Jalled and I. Voronkov, “Object Detection using Image Processing,” pp. 1–6, 2016.
- [13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [14] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936–944.