

Sistem Pendeteksi Sapi pada Peternakan dengan Citra dan Video UAV Menggunakan Metode Single Shot Detector

Fidra Irfan Fikri
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

fidrairfan@student.telkomuniversity.ac.id

Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

Randy Erfa Saputra
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

resaputra@telkomuniversity.ac.id

Abstrak— Indonesia adalah negara islam yang senang sekali mengkonsumsi daging sapi. Sapi memiliki dua jenis yakni sapi potong dan sapi perah. Sapi biasanya dipelihara dipeternakan dan memiliki jenis kandang yang berbeda-beda. Jika peternakan memiliki kandang yang besar pasti akan sulit dan memakan banyak waktu dalam hal pengawasan, apalagi saat sapi-sapi itu berada di ladang gembala maka akan sangat beresiko dalam hal kehilangan hewan ternak seperti pencurian hingga sapi kabur dari lokasi. Pada penelitian ini yakni membuat sebuah sistem deteksi sapi dengan citra dan video UAV menggunakan metode SSD (*Single Shot Detector*) untuk membantu peternakan dalam pengawasan hewan ternak khususnya sapi. Hasil yang telah di dapat bahwa sistem dapat mendeteksi objek dengan ditandai kotak pembatas dan diberi label sapi. Lalu rata-rata akurasi terbaik dalam hal mendeteksi objek adalah ketika drone berada di ketinggian 5 meter yakni 94.45% untuk kondisi diam dan 100% untuk kondisi berjalan.

Kata kunci— SSD (*Single Shot Detector*), Visi Komputer, Drone, Sapi.

I. PENDAHULUAN

Indonesia adalah negara islam yang besar dan suka sekali mengkonsumsi sapi, apalagi pada saat menjelang hari raya besar seperti Idul Fitri atau Idul Adha yang mana harga daging sapi meningkat. Sapi memiliki dua jenis, yakni sapi potong dan sapi perah[1]. Sapi biasanya dipelihara di peternakan yang jauh dari pemukiman warga yang tujuannya agar sapi-sapi ini tidak jenuh. Setiap peternakan sapi pasti memiliki jenis kandang yang berbeda-beda, ada yang memiliki kandang tertutup saja tapi luas, dan ada yang memiliki kandang tertutup sekaligus kandang terbuka. Peternakan sapi yang memiliki kandang yang besar pasti punya populasi hewan yang banyak, sehingga pengawasan biasa akan lebih sulit dan memakan waktu yang lama. Apalagi sapi-sapi tersebut berada di ladang gembala yang mana sangat beresiko dalam hal kehilangan hewan ternak seperti penculikan hewan hingga sapi kabur dari lokasi.

Dari permasalahan tersebut, maka penelitian ini dirancang, dibuat, dan diimplementasikan sebuah sistem untuk mendeteksi hewan ternak berupa sapi menggunakan

citra dan video objek dengan metode *Single Shot Detector* yang nantinya akan dibantu menggunakan *drone*. Penggunaan drone di sini fungsinya untuk membantu mempermudah peternak dalam pemantauan sapi dari jarak jauh, apabila sapi tersebut berada pada ladang gembala maupun di kandang sapi yang besar.

II. KAJIAN TEORI

A. Visi Komputer

Visi komputer atau bisa dikatakan computer vision memiliki dua kata, yakni computer dan vision. Vision dapat diartikan sebagai penglihatan layaknya panca indra manusia. Visi komputer atau computer vision adalah sebuah bidang ilmu yang mempelajari dan menganalisis sebuah gambar pada komputer agar memperoleh hasil informasi sebagaimana seperti panca indra penglihatan pada manusia[2]. Penggunaan visi komputer ini di desain untuk membuat komputer dapat melihat dan mengenali sebuah objek[3].

Sekarang banyak sekali implementasi visi komputer dalam hal mendeteksi objek di kehidupan sehari-hari. Objek pada visi komputer biasanya hanya mengacu pada satu elemen pada gambar, hal ini membuat klasifikasi deteksi objek semakin sulit. Oleh karena itu cara paling populer pada visi komputer dalam mendeteksi objek pada gambar, yakni menggunakan bantuan kotak pembatas[4].

B. Pengolahan Citra Digital

Pengolahan citra digital adalah merupakan disiplin ilmu yang mempelajari tentang pemrosesan atau perbaikan kualitas suatu gambar dalam bentuk dua dimensi. Citra digital sendiri merupakan gambar diam (foto) atau bisa gambar bergerak (video/gif) yang diolah oleh komputer, sehingga disimpan di dalam komputer dalam bentuk angka-angka[2].

Pengolahan citra digital terdiri dari beberapa ilmu yang berkaitan dengan perbaikan kualitas gambar (kontras, warna, dan restorasi citra), transformasi gambar (translasi, rotasi, skala, dan geometik), pemilihan ciri citra yang optimal,

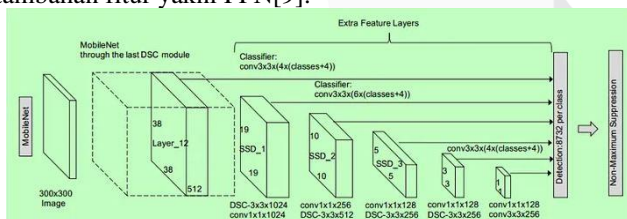
waktu proses data, transmisi data, dan melakukan penyimpanan data yang telah direduksi dan dikompres[5].

C. SSD (Single Shot Detector)

Single shot Detector atau singkatannya SSD merupakan algoritma yang bertujuan untuk mendeteksi dan mengenali sebuah objek pada gambar atau citra dengan metode single deep neural network[6]. Algoritma Single shot Detector (SSD) ini dibuat untuk mendeteksi objek secara real time. Metode Single shot Detector (SSD) mendeteksi objeknya menggunakan bantuan kotak pembatas, dan menerapkan perbaikan dengan fitur multi-skala dan kotak default agar mengurangi penurunan akurasi. Perbaikan yang dilakukan pada algoritma Single shot Detector (SSD) ini memungkinkan dapat menyamai akurasi yang dimiliki oleh algoritma Faster R-CNN dengan menggunakan resolusi gambar yang rendah dan juga membuat algoritma ini memiliki kecepatan yang lebih tinggi dalam mendeteksi objek[7].

Arsitektur SSD termasuk kedalam jenis CNN (Convolutional Neural Network), yang mana merupakan Neural Network untuk mendeteksi objek pada suatu gambar[6]. Arsitektur SSD dibangun di atas arsitektur VGG-16 (Visual Geometry Group 16), yang mana tidak sepenuhnya lapisan terhubung. Alasan menggunakan arsitektur tersebut di dalam arsitektur SSD, yakni kinerjanya yang kuat dalam tugas melakukan klasifikasi pada gambar kualitas tinggi. Hal tersebut juga sangat efektif dalam mengatasi masalah yang mana transfer learning dapat membantu meningkatkan hasil. Lapisan VGG pada arsitektur SSD terhubung dengan satu set konvolusi tambahan, mulai dari konvolusi 6 hingga konvolusi 11. Hal ini memungkinkan bertujuan untuk melakukan ekstrak fitur pada berbagai skala dan secara progresif mengurangi ukuran input ke setiap lapisan berikutnya[8].

Pada penelitian ini algoritma yang dipakai adalah SSD MobileNetV2 FPNLite yang mana arsitekturnya hampir sama seperti SSD pada umumnya. Perbedaannya yaitu pada SSD ini arsitektur dasar menggunakan MobileNetV2 dan ada tambahan fitur yakni FPN[9].



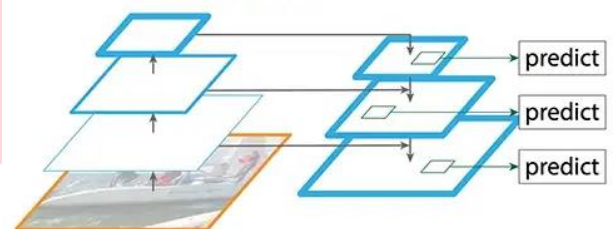
GAMBAR 1 Arsitektur SSD MobileNetV2 [9]

MobileNet merupakan sebuah model kecil dengan latensi rendah serta daya rendah yang dipakai untuk memenuhi sumber daya terbatas dari berbagai kasus penggunaan. MobileNetV2 juga dikenal sebagai fitur ekstraktor yang sangat efektif untuk mendeteksi dan segmentasi objek. Penggunaan koneksi residual tradisional memungkinkan pelatihan yang lebih cepat dan memiliki akurasi yang lebih baik. Pada arsitektur MobileNet V2 di dalamnya menggunakan depthwise dan pointwise convolution. MobileNet V2 menambahkan fitur baru pada arsitekturnya, yakni ada linear bottleneck dan shortcut connection bottleneck[10].

Di Bagian bottleneck terdapat input dan output antara model, lalu lapisan pada bagian dalam melakukan

enkapsulasi kemampuan model untuk mengubah input dengan pixel rendah ke deskriptor yang lebih tinggi yakni kategori gambar. Akhirnya arsitektur MobileNet V2 memiliki kemiripan seperti halnya koneksi residual pada CNN tradisional, yang mana shortcut atau jalan pintas antar bottleneck dapat memungkinkan training atau latihan yang lebih cepat dan akurasi yang lebih baik[10].

FPN atau bisa dikatakan Feature Pyramid Network dibuat untuk menggabungkan fitur-fitur dari berbagai tingkat jaringan konvolusi untuk mendeteksi objek yang lebih baik pada skala berbeda[11]. FPN (Feature Pyramid Network) adalah merupakan fitur ekstraktor yang di desain seperti konsep piramida dengan mempertimbangkan kecepatan dan akurasi. FPN menghasilkan beberapa lapisan peta fitur multi-skala dengan kualitas yang lebih baik dari pada fitur piramida biasa dalam mendeteksi objek. Berikut merupakan gambar piramida FPN[12].



GAMBAR 2 Arsitektur FPN[12]

FPN sendiri memiliki dua jalur, di antaranya adalah jalur bawah ke atas, dan jalur atas ke bawah. Penggunaan jalur bawah ke atas yakni untuk ekstraksi fitur pada jaringan konvolusi yang mana ketika naik, resolusi spasial akan turun. Akan tetapi semakin banyaknya struktur tingkat tinggi yang terdeteksi, maka nilai semantik untuk setiap lapisan akan meningkat. Selanjutnya, penggunaan dari jalur atas ke bawah yakni untuk membangun lapisan resolusi yang lebih tinggi dari lapisan semantik yang kaya. Agar pelatihan lebih mudah dan membantu detektor dalam memprediksi lokasi yang lebih baik, maka ada koneksi lateral antara lapisan yang direkonstruksi dan peta fitur[12].

D. Confusion Matrix

Confusion matrix adalah sebuah matrix n*n yang dipakai untuk mengukur kinerja model klasifikasi[13]. Confusion matrix sering juga disebut error matrix yang pada dasarnya memberikan informasi tentang perbandingan hasil klasifikasi yang dilakukan oleh model atau sistem dengan hasil klasifikasi sebenarnya[14].

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

GAMBAR 3 Confusion matrix

Dari gambar diatas menunjukkan bahwa di dalam tabel confusion matrix terdapat 4 simbol atau istilah hasil proses klasifikasi, di antaranya yakni True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). Supaya dapat mengetahui performa model yang telah dibuat, maka ada beberapa rumus yang sering digunakan atau yang paling populer, diantaranya yakni menghitung presisi (precision), recall, dan akurasi (accuracy). Berikut ini adalah rumus-rumus confusion matrix[14]:

1. Precision

Precision adalah tingkat keakuratan antara data yang asli dengan hasil prediksi yang keluar dari model. Berikut adalah rumusnya:

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

2. Recall

Recall adalah menggambarkan keberhasilan dari model dalam menemukan kembali sebuah informasi. Berikut adalah rumusnya:

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

3. Akurasi

Accuracy adalah menghitung seberapa akurat model yang dibuat dalam mengklasifikasi dengan benar. Berikut adalah rumusnya:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

III. METODE

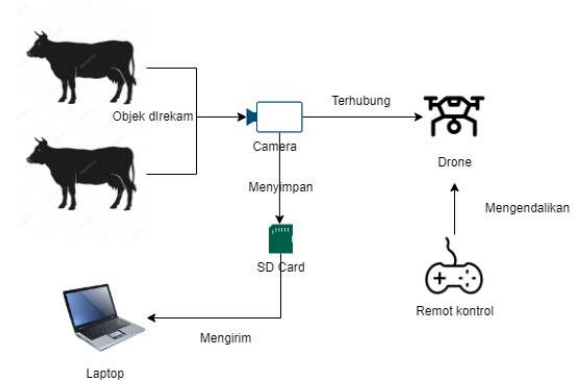
A. Desain sistem

Pada penelitian ini desain sistem tugas akhir dengan judul sistem pendeteksi sapi pada peternakan dengan citra dan video UAV menggunakan metode single shot detector (SSD) membutuhkan perancangan dan persiapan agar sesuai dengan spesifikasi Software dan hardware yang dibutuhkan.

Sebelum melakukan perancangan dan melakukan desain sistem, hal yang perlu dilakukan adalah mencari berbagai referensi terkait studi sistem deteksi objek yang menggunakan alat bantu berupa drone atau UAV. Pencarian referensi tentang sistem deteksi objek menggunakan UAV, yakni dengan membaca dan mengumpulkan berbagai paper atau jurnal penelitian sebelumnya yang serupa. Kemudian setelah membaca paper atau jurnal, selanjutnya melakukan pembuatan rangkuman terkait dengan paper atau jurnal yang sudah dibaca.

Setelah membuat rangkuman, maka selanjutnya adalah memilih satu persatu teori yang nantinya akan digunakan untuk penelitian sistem pendeteksi sapi pada peternakan dengan citra UAV. Teori-teori ini dikumpulkan dan dipakai untuk menunjang penelitian tugas akhir ini.

Selanjutnya, yakni mencari sebuah peternakan yang mana nantinya dipakai untuk mengumpulkan beberapa data tentang hewan ternak berupa sapi. Data yang telah dikumpulkan nantinya akan dipakai untuk implementasi sistem pendeteksi objek pada penelitian ini. Kemudian hal selanjutnya yakni membuat desain sistem tentang penelitian tugas akhir ini.

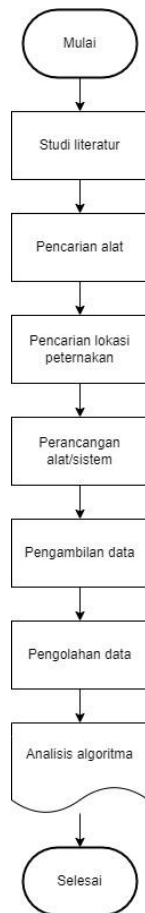


GAMBAR 4
Desain sistem penelitian

Gambar di atas adalah merupakan desain sistem dari penelitian tugas akhir ini yang berjudul sistem pendeteksi sapi pada peternakan dengan citra dan video UAV menggunakan metode single shot detector (SSD). Awalnya dilakukan dengan menerbangkan drone menuju ke arah objek, yakni berupa sapi yang telah dikeluarkan dari kandang tertutup. Lalu drone akan merekam objek tersebut menggunakan kamera yang sudah terpasang, yang kemudian hasil rekaman objek tersebut akan disimpan ke dalam micro-SD yang ada di dalam drone. Data rekaman itu kemudian dikirim ke komputer atau laptop untuk melakukan pengetesan algoritma pada penelitian tugas akhir ini. Jika data sudah berada berada di dalam tempat penyimpanan laptop, maka akan dilakukan running algoritma. Nantinya algoritma tersebut yakni SSD (single shot detector) akan mendeteksi hewan berupa sapi dan diberi label bernama sapi.

B. Diagram Alir Penelitian

Pada penelitian ini memiliki prosedur dalam hal proses mendeteksi sapi pada peternakan melalui kamera pada sebuah drone atau UAV. Prosedur penelitian ini terdiri dari studi literatur, Pencarian alat, Pencarian peternakan, Perancangan alat/sistem, Pengambilan data, Pengolahan data, dan analisis. Di sebuah penelitian pasti memerlukan sebuah diagram alir agar tahapan dari penelitian dapat dipahami. Diagram alir dapat dilihat pada gambar 5.



GAMBAR 5

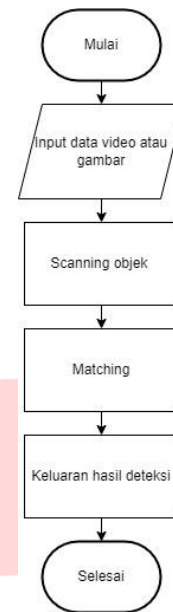
Diagram alir penelitian

Penjelasan dari diagram alir diatas adalah sebagai berikut:

1. Studi literatur, yakni melakukan studi literatur terhadap buku-buku, jurnal, artikel, dan paper terkait dengan topik pada penelitian ini, yang nantinya akan dijadikan referensi. Adapun juga internet dipakai dan digunakan untuk media pencarian referensi.
2. Pencarian alat, yakni menentukan alat berupa drone yang mana nantinya akan digunakan sebagai media pengambilan data di peternakan.
3. Pencarian lokasi peternakan, yakni mencari peternakan yang dibutuhkan untuk pengambilan data sapi pada penelitian ini.
4. Perancangan sistem, yakni merancang atau membuat sistem deteksi objek menggunakan algoritma SSD (Single Shot Detector).
5. Pengambilan data, yakni mengambil data berupa foto dan video hewan berupa sapi di peternakan menggunakan kamera pada drone.
6. Pengolahan data disini maksudnya melakukan pengujian sistem pendeteksi objek dengan algoritma SSD (Single Shot Detector) menggunakan data hewan berupa sapi pada peternakan yang sudah diambil sebelumnya.
7. Analisis algoritma adalah menghitung tingkat akurasi dari algoritma SSD (Single Shot Detector) yang digunakan pada sistem pendeteksi objek menggunakan citra UAV.

E. Diagram Alir Deteksi Objek

Selanjutnya penelitian ini memiliki diagram alir dalam hal deteksi objek. Bisa dilihat diagram alir deteksi objek pada gambar 6.



GAMBAR 6

Diagram alir deteksi objek

Penjelasan dari diagram alir diatas adalah sebagai berikut:

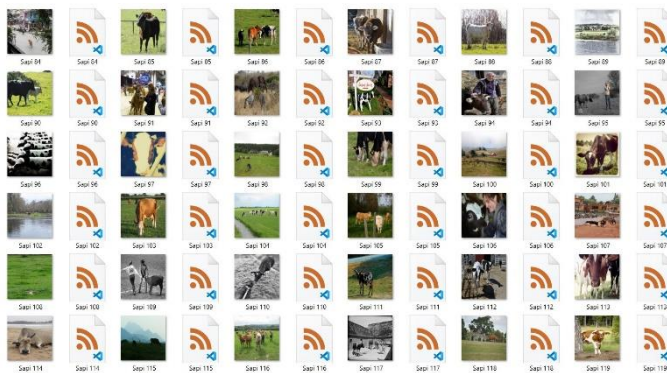
1. Data disini merupakan hasil rekaman yang telah dilakukan di peternakan menggunakan kamera drone, nantinya akan di input di dalam komputer.
2. Setelah melakukan input data maka algoritma itu akan melakukan pemindaian atau scanning dari video atau gambar.
3. Lalu algoritma tersebut akan mencocokkan atau matching objek dengan data yang sudah dilatih sebelumnya.
4. Terakhir adalah mengeluarkan hasil berupa nama objek beserta kotak pembatas pada objek yang dideteksi.

F. Fungsi dan Fitur

Pada penelitian tugas akhir ini, deteksi objek menggunakan algoritma SSD (*Single Shot Detector*). Algoritma ini berfungsi untuk menentukan atau mengidentifikasi sebuah objek, yang mana algoritma ini termasuk ke dalam sebuah teknologi kecerdasan buatan yang menjurus ke sebuah bidang ilmu lanjutan berupa visi komputer atau *computer vision*. Fitur pada penelitian ini, yakni dapat memberikan atau menunjukkan hewan sapi menggunakan bounding box atau kotak pembatas yang telah diberi label nama berupa sapi.

G. Dataset

Dataset yang dipakai untuk *training* SSD MobileNetV2 FPNLite pada penelitian ini adalah gambar berupa sapi lalu diberi label dan kotak pembatas menggunakan aplikasi LabelImg. Selain dataset gambar sapi, ada dataset gambar selain sapi antara lain ayam, kucing, dan kambing yang tidak diberi kotak pembatas atau *bounding box*. Hal tersebut bertujuan untuk menunjukkan kepada algoritma bahwa objek tersebut bukan sapi. Metode ini dinamakan metode *negative training*.



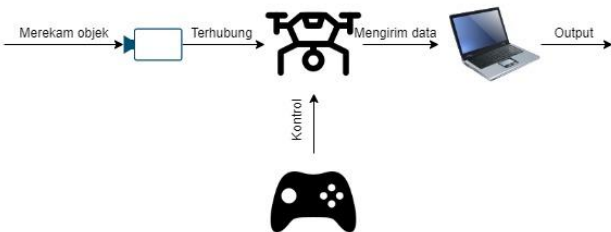
GAMBAR 7
Dataset untuk *training*

H. Metode Training SSD MobileNetV2 FPNLite

Pada penelitian ini *training* untuk algoritma SSD MobileNetV2 FPNLite dilakukan menggunakan Jupyter Notebook dan command prompt dengan dataset sapi yang telah diberi label dan kotak pembatas. Algoritma SSD MobileNetV2 FPNLite di *train* dengan menggunakan beberapa konfigurasi parameter demi mendapatkan konfigurasi yang sempurna. Parameter yang diujikan berupa training step, learning rate, batch, dan epoch yang mana akan diujikan satu-persatu dengan konfigurasi lainnya tidak berubah, contohnya seperti pengujian *batch* dengan konfigurasi *training step*, *learning rate*, dan *epoch* nilai konfigurasinya tetap atau sama.

I. Desain Perangkat keras

Sebagai penunjang keberhasilan penelitian tugas akhir ini, maka membutuhkan sebuah desain perangkat keras. Berikut ini adalah desain perangkat keras yang ditunjukkan pada gambar 8.



GAMBAR 8
Desain perangkat keras

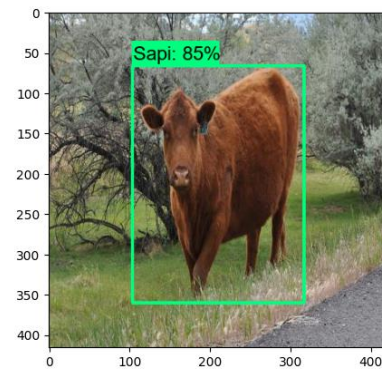
Perangkat keras yang dipakai pada penelitian ini memiliki beberapa komponen, yakni terdiri dari remot kontrol *drone* yang berfungsi sebagai alat untuk mengendalikan drone. Selanjutnya ada *drone* yang telah terkoneksi dengan kamera. Drone ini berfungsi sebagai alat untuk membantu kamera menuju objek yang ada di peternakan. Kamera sendiri berfungsi sebagai alat perekam atau pengambilan data berupa sapi di peternakan, yang nantinya data tersebut akan digunakan untuk melakukan pendeteksian pada algoritma yang dipakai penelitian tugas akhir ini. Yang terakhir adalah laptop atau komputer, yang berfungsi untuk menjalankan algoritma pada penelitian ini dan mengeluarkan hasilnya.

IV. HASIL DAN PEMBAHASAN

A. Hasil Percobaan

Dari pengujian dan percobaan yang telah dilakukan, penelitian ini menghasilkan hasil deteksi yang nantinya dapat mendeteksi sapi dengan menampilkan kotak pembatas dan diberi nama sapi pada kotak tersebut. Percobaan pada

algoritma SSD (Single Shot Detector) akan dilakukan menggunakan gambar terlebih dahulu sebelum dilakukannya tes atau pengujian menggunakan video.



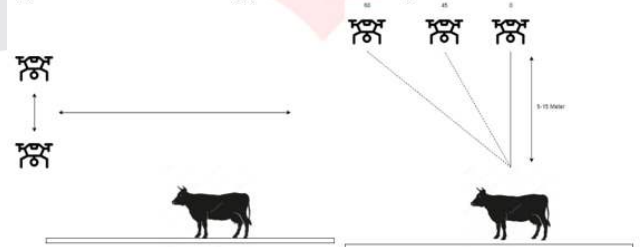
GAMBAR 9
Hasil percobaan menggunakan gambar

B. Skenario Pengujian

Skenario pengujian pertama pada penelitian ini adalah melakukan *training* dataset dengan mencari konfigurasi terbaik yang nantinya dipakai untuk pengujian di lapangan atau skenario kedua. Konfigurasi yang dicari yakni *training step*, *learning rate*, *batch size*, dan *epoch*. Nilai mAP yang bagus dan *total loss* sedikit pada *training step* akan dipilih untuk pengujian selanjutnya, yakni *learning rate*. Begitu juga nilai mAP yang bagus dan *total loss* sedikit pada *learning rate* akan dipilih untuk pengujian selanjutnya, yakni *batch size*. Jika ada nilai *batch size* yang bagus maka akan dipakai untuk pengujian *epoch*.

Skenario pengujian kedua pada penelitian ini adalah *drone* yang sudah terpasang kamera akan terbang di udara dan berjalan menuju objek. Kemudian kamera pada *drone* akan diatur sesuai dengan angle atau sudut yang diujikan, yakni dari sudut 0 derajat dengan jarak ketinggian mulai dari 5 meter hingga 15 meter. Sudut yang kedua adalah 45 derajat dengan jarak 5 meter hingga 15 meter. Kemudian yang terakhir adalah sudut 60 derajat dengan jarak 5 meter hingga 15 meter.

Skenario pengujian selanjutnya adalah *drone* yang terpasang kamera akan terbang di udara menuju objek. Kemudian *drone* itu ambil jarak ke belakang sedikit. Lalu kamera pada drone akan diatur mengarah sedikit kebawah. Setelah itu drone akan berjalan maju mengarah objek dan melewati atas objek dengan jarak ketinggian yang sudah diatur. Untuk jarak ketinggian pada drone berjalan, yakni mulai dari 5 meter hingga 15 meter di atas objek.

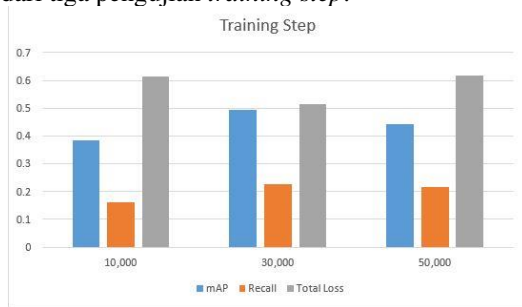


GAMBAR 10
Skenario pengujian

C. Pengujian Learning Step

Pengujian *training step* disini adalah untuk mengetahui berapa *step* yang dibutuhkan agar mendapatkan nilai mAP paling bagus dan total loss paling sedikit pada saat *training* data sapi. Untuk skenario konfigurasi *training step* adalah

10,000, 30,000, dan 50,000. Berikut ini merupakan grafik hasil dari tiga pengujian *training step*.

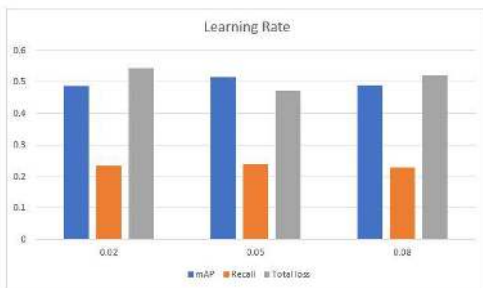


GAMBAR 11 Grafik *trainig step*

Pada grafik batang diatas bisa disimpulkan, bahwa semakin tinggi *training step* hasil tidak diperoleh dengan baik, karena pada saat *training step* 50,000 *total loss* yang dimiliki jadi bertambah dari sebelumnya dan mAPnya tidak sebaik *training step* 30,000. Pada saat *training step* 50,000 mAP yang di dapat adalah 0.444 dengan *total loss* 0.618, sedangkan *training step* 30,000 mAP yang di dapat adalah 0.494 dengan *total loss* 0.515. Untuk *learning step* 10,000 nilai mAP 0.383 dengan *total loss* 0.614.

D. Pengujian Learning Rate

Pengujian *learning rate* disini adalah untuk mengetahui berapa *learn rate* yang dibutuhkan agar mendapatkan nilai mAP paling bagus dan *total loss* paling sedikit pada saat training data sapi. Untuk skenario konfigurasi *learning rate* adalah 0.02, 0.05, dan 0.08. Berikut ini merupakan hasil dari tiga pengujian *learning rate*.



GAMBAR 12 Grafik *learning rate*

Dapat diperhatikan pada grafik batang di atas, bahwa semakin rendah *learning rate* hasil tidak diperoleh dengan baik, karena pada saat *learning rate* 0.02 *total loss* yang dimiliki jadi bertambah dari sebelumnya dan mAPnya tidak sebaik *learning rate* 0.05. Pada saat *learning rate* 0.08 mAP yang di dapat adalah 0.488 dengan *total loss* 0.519, sedangkan *learning rate* 0.05 mendapatkan hasil yang lebih baik dari pada *learning rate* sebelum maupun sesudahnya.

E. Pengujian Batch Size

Pada pengujian *batch size* hasil terbaik berada pada *batch size* 4 yang mana nilai mAP 0.516 dengan *total loss* 0.470. Untuk *batch size* 1 nilai mAP yang didapat 0.341 dengan *total loss* 0.772. yang terskhir untuk *batch size* 2 nilai mAP 0.432 dengan *total loss* 0.580.

F. Pengujian Epoch

Pada pengujian *epoch*, hasil paling baik adalah *epoch* 1 yang mana memiliki *total loss* 0.470 dengan mAP 0.516. Untuk pengujian *epoch* 10 hasil mAP 0.481 dengan *total loss* 0.507. dan yang terakhir *epoch* 100 hasil mAP yang di dapat

0.506 dengan *total loss* 0.518. Pada penelitian ini memilih untuk menggunakan *epoch* 1 karena memiliki *total loss* paling kecil, agar sistem ini dapat mendeteksi objek dengan baik.

G. Pengujian Deteksi Sapi Drone Diam

Pada percobaan ini drone akan terbang menggunakan skenario yang sudah di jelaskan pada sebelumnya. Pengujian ini memiliki 2 kali percobaan dengan sudut yang sama dan ketinggian yang sama. Berikut ini tabel hasil percobaannya.

TABEL 1 Percobaan 1 *drone* diam

Percobaan 1								
Ketinggian 5 meter			Ketinggian 10 meter			Ketinggian 15 meter		
Su dut	Lu x	Aku rasi	Su dut	Lu x	Aku rasi	Su dut	Lu x	Aku rasi
0	245	100	0	985	100	0	985	100
	15	%		56	%		56	%
45	174	100	45	774	100	45	774	100
	06	%		70	%		70	%
60	128	100	60	128	100	60	128	50%
	92	%		92	%		92	

TABEL 2

Percobaan 2 *drone* diam

Percobaan 2								
Ketinggian 5 meter			Ketinggian 10 meter			Ketinggian 15 meter		
Su dut	Lu x	Aku rasi	Su dut	Lu x	Aku rasi	Su dut	Lu x	Aku rasi
0	985	66.6	0	245	100	0	245	100
	56	7%		15	%		15	%
45	187	100	45	187	100	45	187	100
	61	%		61	%		61	%
60	314	100	60	314	50%	60	314	0%
	75	%		75			75	

TABEL 3

Rata-rata akurasi

Akurasi Rata-rata	
5 meter	94.45%
10 meter	91.67%
15 meter	75%

Dari hasil yang di dapat bahwa akurasi pada algoritma sistem ini akan menurun setiap ketinggian dan sudut bertambah. Pada saat ketinggian 5 meter nilai rata-rata akurasinya masih bagus yakni 94.45%. Ketika *drone* mulai naik di ketinggian 10 meter, nilai akurasi mulai menurun sedikit yakni 91.67%. Lalu saat *drone* naik lagi di ketinggian 15 meter, nilai akurasi jatuh atau menurun drastic menjadi 75%. Hal ini mungkin disebabkan algortma SSD MobileNet V2 FPNLite sudah tidak memungkinkan mendeteksi objek dari jarak jauh dan juga ada hal lain yang menyebabkan algoritma ini sulit mendeteksi dengan baik, seperti cahaya, dan angin.

H. Pengujian Deteksi Sapi Drone Jalan

Pada percobaan ini drone akan terbang menggunakan skenario yang sudah di jelaskan pada sebelumnya. Pengujian ini akan menggunakan beberapa ketinggian. Berikut ini adalah tabel rata-rata akurasi hasil percobaan.

TABEL 4
Hasil rata-rata akurasi *drone* berjalan

Akurasi Rata-rata	
5 meter	100%
10 meter	25%
15 meter	50%

Dari percobaan yang telah dilakukan, *drone* bergerak maju dengan kecepatan 0.1 m/s pada ketinggian mulai dari 5 meter hingga 15 meter. Hasil yang di dapat ada yang memiliki nilai rata-rata akurasi tinggi dan ada yang mendapatkan nilai rata-rata akurasi yang rendah. Pada saat 5 meter rata-rata akurasi masih 100%. Kemudian pada saat ketinggian di naikan menjadi 10 meter rata-rata akurasi jatuh ke 25%. Lalu ketika ketinggian dinaikan lagi menjadi 15 meter, rata-rata akurasi yang di dapat adalah 50%. Pada saat di ketinggian 10 meter sistem ini tidak bisa mendeteksi objek dengan baik, karena ada angin yang kencang dan cuaca yang mendung. Oleh sebab itu sistem sangat sulit mendeteksi objek dengan baik.

V. KESIMPULAN

Berikut ini merupakan kesimpulan dari hasil penelitian dan pengujian yang sudah dilakukan pada sebelumnya.

1. Algoritma SSD MobileNetV2 FPNLite dapat melakukan deteksi pada sapi yang ada di peternakan menggunakan drone pada ketinggian dan posisi sudut yang bermacam-macam setelah melakukan training pada algoritma.
2. Penggunaan algoritma SSD MobileNetV2 FPNLite ketika percobaan di lapangan yakni dapat melakukan deteksi dengan akurasi yang cukup tinggi saat *drone* diam maupun bergerak. Akurasi terbaik pada saat mendeteksi sapi adalah ketika *drone* berada pada ketinggian 5 meter dengan nilai rata-ratanya 94.45% untuk yang diam, dan untuk *drone* berjalan rata-ratanya yakni 100%. Saat ketinggian 10 meter dan 15 meter rata-rata akurasi mulai menurun. Nilai rata-rata akurasi 10 meter adalah 91.67% untuk drone diam dan 25% untuk *drone* berjalan. Untuk ketinggian 15 meter nilai rata-rata akurasinya 75% untuk *drone* diam dan 50% untuk drone berjalan.

REFERENSI

- [1] ©Badan Pusat Statistik, "Peternakan dalam Angka 2022," *Direktorat Stat. Peternakan, Perikanan, dan Kehutan.*, p. 135, 2022, [Online]. Available: <https://www.bps.go.id/publication/2022/06/30/4c014349ef2008bea02f4349/peternakan-dalam-angka-2022.html>.
- [2] D. R. Hartawan, T. W. Purboyo S.Si., S.T., M.PMat., and C. Setianingsih, S.T., M.T., "BENCANA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) DESIGN AND IMPLEMENTATION OF DISASTER VICTIMS DETECTION SYSTEM USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD," *S1 Tek. Komputer, Fak. Tek. Elektro, Univ. Telkom*, p. 8, 2019.
- [3] A. D. Gunawan, M. Rumani, and C. Setianingsih, "Dalam Pencarian Korban Bencana Alam Menggunakan Drone Berbasis Mikro Komputer Design and Implementation of Human Detection System in Search OF Natural Disaster Victims Using Drone Based on Micro Computer," *e-Proceeding Eng.*, vol. 4, no. 3, pp. 4115–4122, 2017.
- [4] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude UAV datasets: A survey," *Image Vis. Comput.*, vol. 104, 2020, doi: 10.1016/j.imavis.2020.104046.
- [5] N. Z. Munantri, H. Sofyan, and M. Yanu, "Aplikasi Pengolahan Citra Digital Untuk Mendeteksi Umur Pohon," *Telematika*, vol. 16, no. 2, pp. 97–104, 2019.
- [6] S. Fuady, N. Nehru, and G. Anggraeni, "Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera," *J. Electr. Power Control Autom.*, vol. 3, no. 2, p. 39, 2020, doi: 10.33087/jepca.v3i2.38.
- [7] N. A. H. Y.C. Wong, J.A. Lai, S.S.S. Ranjit, A.R. Syafeeza, "Convolutional Neural Network for Object Detection System for Blind People," *J. Telecommun. Electron. Comput. Eng.*, vol. 11, no. 2, pp. 1–6, 2019, doi: 10.35940/ijrte.B1129.0882S819.
- [8] A. Rohan, M. Rabah, and S. H. Kim, "Convolutional Neural Network-Based Real-Time Object Detection and Tracking for Parrot AR Drone 2," *IEEE Access*, vol. 7, pp. 69575–69584, 2019, doi: 10.1109/ACCESS.2019.2919332.
- [9] Mayank Shinghal, "Object Detection using SSD Mobilenet and Tensorflow Object Detection API : Can detect any single class from coco dataset. | by mayank singhal | Medium," Jul. 07, 2020. <https://medium.com/@techmayank2000/object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bbd0691> (accessed Feb. 04, 2023).
- [10] N. Nufus *et al.*, "Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360° Ternormalisasi," in *Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO)*, 2021, vol. 3, no. November, pp. 123–134, doi: 10.54706/senastindo.v3.2021.123.
- [11] P. Farid Hasainia, "What is Feature Pyramid Network (FPN)?," Apr. 05, 2022. <https://www.fastpath.one/blog/feature-pyramid-network-fpn> (accessed Feb. 04, 2023).
- [12] Jonathan Hui, "Understanding Feature Pyramid Networks for object detection (FPN) | by Jonathan Hui | Medium," Mar. 28, 2018. <https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c> (accessed Feb. 04, 2023).
- [13] Indhumathy Chelliah, "Confusion Matrix for Multiclass Classification by Indhumathy Chelliah | MLearning.ai | Medium|," Sep. 15, 2022. <https://medium.com/mlearning-ai/confusion-matrix-for-multiclass-classification-f25ed7173e66> (accessed Jan. 18, 2023).
- [14] Kuncahyo Setyo Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning | by Kuncahyo Setyo Nugroho | Medium," Nov. 13, 2019. <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-unsupervised-machine>

learning-bc4b1ae9ae3f (accessed Jan. 18, 2023).

