

Sistem Cerdas Pendeteksi Dan Penghitung Jumlah Korban Bencana Alam Menggunakan Algoritma *Deep Learning*

M Adamu Islam¹, Moch. Zen Samsono Hadi², Rahardhita Widyatra³

^{1,2,3} Politeknik Elektronika Negeri Surabaya, Jl. Raya ITS, Kampus PENS, Sukolilo, Surabaya, Indonesia

Email: adamuislam16@gmail.com¹, zenhadi@pens.ac.id², widi@pens.ac.id³

Abstrack – The speed of searching for victims of natural disasters is an important factor that affects the chances of survivors. Most of the locations affected by natural disasters will be difficult to access, and currently, the rescue team is still using heavy equipment to open the access. This requires a long time to go to the location and search for victims. In this research, we propose a smart device to help the National Search and Rescue Agency (BASARNAS) use a drone equipped with cameras to search for victims of natural disasters in real time. By using this smart device, the search will be more effective, because it can speed up the rescue team in searching for victims at the disaster site. The process of detecting and calculating the number of victims is carried out on the camera using the Convolutional Neural Network algorithm and the You Only Look Once version 4 (YOLOv4) architecture. The results of this study are that the Convolutional Neural Network algorithm has a fairly high accuracy of 92.29% in detecting the victim's condition. Besides that, the height of the camera and the position of the camera also affect the accuracy obtained, whereas the results obtained with a static camera have optimal accuracy, which is 95% and the higher the camera, the accuracy will decrease, which is 67%.

Keywords – Bencana alam, *Convolutional Neural Network*, YOLOv4

Intisari – Proses kecepatan pencarian korban bencana alam merupakan faktor penting yang memengaruhi peluang korban untuk bertahan hidup. Sebagian besar lokasi yang terkena bencana alam akan sulit untuk diakses, dan saat ini tim penyelamat masih menggunakan alat berat untuk membuka akses tersebut. Hal ini membutuhkan waktu yang lama untuk menuju ke lokasi dan mencari korban. Pada penelitian ini, kami mengusulkan sebuah perangkat cerdas untuk membantu Badan Nasional Pencarian dan Pertolongan (BASARNAS) menggunakan drone yang dilengkapi dengan kamera untuk mencari korban bencana alam secara *real-time*. Dengan menggunakan perangkat cerdas yang dibuat ini, pencarian akan lebih efektif, karena ini dapat mempercepat tim penyelamat dalam mencari korban di lokasi bencana. Proses pendeteksian dan perhitungan jumlah korban dilakukan pada kamera menggunakan algoritma *Convolutional Neural Network* dan arsitektur *You Only Look Once* versi 4 (YOLOv4). Hasil dari penelitian ini adalah algoritma *Convolutional Neural Network* memiliki akurasi yang cukup tinggi yaitu sebesar 92,29% dalam mendeteksi kondisi korban. Selain itu tinggi kamera dan posisi dari kamera juga memengaruhi akurasi yang didapat, dimana hasil yang didapat dengan kamera statis memiliki akurasi yang cukup optimal yaitu sebesar 95%. Namun semakin tinggi kamera, akurasi akan semakin turun yaitu sebesar 67%.

Kata Kunci – Bencana alam, *Convolutional Neural Network*, YOLOv4

I. PENDAHULUAN

Posisi geografis dan geodinamik Indonesia telah menempatkan tanah air kita sebagai salah satu wilayah yang rawan bencana alam (*natural disaster-prone region*). Indonesia merupakan negara kepulauan dimana tiga lempeng besar dunia bertemu, yaitu Lempeng Indo-Australia, Lempeng Eurasia, dan Lempeng Pasifik. Interaksi antar lempeng-lempeng tersebut

menempatkan Indonesia sebagai wilayah yang memiliki aktivitas kegunungapian dan kegempaan yang cukup tinggi [1]. Dampak yang bisa ditimbulkan dari bencana alam yaitu hancurnya keseimbangan alam, lingkungan menjadi rusak, korban nyawa manusia, dan korban harta benda. Sedangkan untuk efek psikologis, korban bencana alam akan mengalami gangguan kurang tidur, mimpi buruk, dan kehilangan kekeluargaan beraktifitas [2].

Dampak yang ditimbulkan dari permasalahan diatas dapat diantisipasi dan diminimalisir jika ditangani dengan cepat, tepat, dan juga seksama. Oleh karena itu, kehadiran Tim Pencari dan Penyelamat sangat dibutuhkan jika terjadi musibah. Tugas Badan Nasional Pencarian dan Pertolongan (BASARNAS) akan dapat terlaksana dengan baik jika didukung dengan ketersediaan alat pendeteksi. Penelitian dan riset mengenai pendeteksian korban bencana alam telah dilakukan Agiel Fahreza Aliek, dkk [3] yaitu melakukan pendeteksian korban bencana alam menggunakan naïve bayes classifier. Penelitian tersebut melakukan pendeteksian menggunakan sensor thermal, dimana sensor tersebut akan mendeteksi suhu tubuh dan mendapatkan hasil yang kurang maksimal dalam mendeteksi korban yaitu hanya dapat mendeteksi dengan jarak 100 cm. Untuk mendeteksi korban bencana alam selain menggunakan sensor thermal adalah dengan menggunakan kamera berbasis *Deep Learning*. *Deep Learning* merupakan subkelas dari *machine learning*. Dibandingkan pembelajaran mesin tradisional, *Deep Learning* lebih cocok untuk pemrosesan data besar, kinerja algoritma meningkat dengan meningkatnya volume data [4]. Pada dasarnya model *Deep Learning* jika diimplementasikan ke hardware akan cocok digunakan sesuai dengan kebutuhannya.

Dengan sulitnya akses menuju lokasi bencana alam, alat yang sudah ada saat ini akan sulit untuk digunakan jika belum berada pada lokasi. Hal ini dapat mengurangi waktu bertahan hidup korban bencana alam, karena kurang cepatnya tim evakuasi menuju ke titik lokasi bencana alam. Sehingga dibutuhkan sebuah alat yang efisien untuk mendeteksi adanya korban setelah bencana terjadi. Dengan permasalahan yang ada, maka kami mengusulkan sebuah perangkat cerdas untuk mengetahui kondisi korban dan jumlahnya serta dapat berkomunikasi dari jarak jauh pada lokasi bencana yang sulit diakses menggunakan komunikasi lora. Dimana perangkat ini berupa modul mikrokontroler yang dilengkapi dengan sebuah webcam yang diangkat oleh drone. Webcam ini sudah dilatih menggunakan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur YOLOv4, dimana arsitektur ini jika diterapkan pada hardware akan dapat melakukan deteksi secara cepat dengan akurasi yang cukup optimal. Keunggulan dari penelitian ini dibandingkan dengan sebelumnya adalah perangkat cerdas yang dibuat mampu mendeteksi dan mengklasifikasi kondisi korban serta menghitungnya. Namun kami masih membatasi implementasinya di perangkat hardware dengan pengujian statis (tanpa drone) dengan maksud untuk menguji kehandalan unjuk kerja algoritma di hardware.

II. SIGNIFIKANSI STUDI

Pada bab ini, penulis menjelaskan mengenai penelitian terkait, metode yang digunakan penelitian, langkah-langkah membangun alat yang diusulkan, dan merancang alat yang diusulkan.

A. Penelitian Terkait

M. B. Berjiga, dkk [5] melakukan penelitian, dimana penulis mempresentasikan metode untuk mendukung operasi SAR dalam longsor menggunakan UAV yang dilengkapi dengan kamera untuk memindai puing-puing longsor. Gambar yang diperoleh kemudian diproses oleh sistem yang terdiri dari CNN yang telah dilatih sebelumnya untuk menghasilkan deskripsi gambar yang sesuai dan pengklasifikasi SVM linier untuk membedakan antara objek dan latar belakang. Dan hasil yang didapat menunjukkan bahwa pemrosesan pada resolusi yang lebih tinggi dapat meningkatkan akurasi deteksi tetapi dengan peningkatan waktu pemrosesan. Kemudian metode pre-processing dapat digunakan untuk mendeteksi ubin yang memiliki objek dan tidak memiliki objek.

I. A. Sulistijono, dkk [6], melakukan sebuah penelitian tentang *multilayer neural network* untuk deteksi korban bencana pada latar belakang yang sangat berantakan dan diuji menggunakan lembar data IDV-50 yang baru. Kemudian dilakukan pra pelatihan menggunakan datasheet diikuti dengan *finetuning* hanya pada lapisan terakhir menggunakan datasheet spesifik. Lembar data korban bencana Indonesia yang baru juga disediakan. Eksperimen menunjukkan hasil yang didapat akan bermanfaat bagi tim SAR untuk menemukan korban dengan cepat. Akan tetapi pada penelitian ini hanya dilakukan simulasi saja dalam pendeteksiannya.

I. A. Sulistijono, dkk [7], mengembangkan kerangka deteksi korban mulai dari perolehan citra menggunakan drone dan pemrosesan lebih lanjut menggunakan *convolutional neural network* (CNN) untuk menemukan korban pada pascabencana. Gambar yang didapat kemudian dikirim ke server dan diproses lebih lanjut untuk menemukan kemungkinan adanya korban. Proses pendeteksian korban berbeda dengan pendeteksian manusia biasa, beberapa tahapan *pre-processing* diterapkan untuk meningkatkan variasi dari dataset yang telah ada. Dan digunakan alat tambahan untuk mengambil gambar dan data sensor, seperti lokasi dan waktu yang menggunakan Sistem Satelit Navigasi Global. Untuk hasil dari penelitian ini masih kurang maksimal, karena gambar dikirimkan secara streaming sehingga dapat menghabiskan baterai dan tempat penyimpanan.

D. Zhang, dkk [8], mengusulkan sistem penyelamatan manusia dilokasi bencana menggunakan tiga jenis sensor yaitu sensor CO₂, kamera termal, dan mikrofon. Sistem ini digunakan untuk mendeteksi korban yang masih hidup di bawah reruntuhan. Sensor CO₂ dapat memberikan informasi yang berguna untuk menemukan korban, sementara kamera thermal kurang efisien karena adanya hambatan akses di beberapa area. Selain itu, algoritma pengenalan suara berbasis SVM yang menggunakan mikrofon kurang optimal mendeteksi korban apabila korban tidak bersuara.

M. Zacharie, dkk [9], melakukan penelitian untuk menyelamatkan nyawa manusia saat terjadi bencana alam, seperti gempa bumi. Penelitian ini memperkenalkan pendeteksian tubuh manusia menggunakan pengolahan citra dari kamera UAV. Warna kulit manusia diekstraksi terlebih dahulu dalam RGB kemudian diubah menjadi HSV. Pengujian dilakukan baik di dalam maupun di luar ruangan, di mana seorang manusia merepresentasikan objek yang di posisikan secara dekat dan jauh dari kamera untuk memeriksa kemampuan deteksi dalam kedua kasus tersebut. Hasil eksperimen menunjukkan bahwa dekat atau jauh objek dari kamera, kamera dapat dengan jelas mendeteksi tubuh manusia dan bagian tubuh manusia seluruhnya. Akan tetapi, karena pada penelitian ini menggunakan RGB, maka apabila kurang penerangan maka korban tidak terlihat, dan juga warna kulit akan berpengaruh pada hasil pendeteksian.

P. Malhotra, dkk [10], melakukan sebuah penelitian untuk membandingkan algoritma dari *Deep Learning* untuk mendeteksi objek. Algoritma *Deep Learning* yang dibandingkan yaitu R-CNN, Fast R-CNN dan YOLO. Algoritma tersebut akan dibandingkan dengan beberapa perbedaan yaitu perbedaan dari dataset dan perbedaan dari teknik mendeteksi objek. Pada penelitian ini masih belum diimplementasikan secara langsung untuk mendeteksi adanya korban bencana alam.

M. Adam, dkk [11], melakukan sebuah penelitian pada beberapa arsitektur CNN, dimana arsitektur yang digunakan adalah arsitektur dari Mobilenet, VGG-16, YOLO, dan custom layer. Algoritma ini diterapkan untuk mendeteksi korban bencana alam, dan hasil yang didapat, arsitektur dari Mobilenet dan YOLO terbaik dalam mendeteksi korban bencana alam, dan penelitian ini hanya dilakukan pada software saja dan belum menggunakan hardware untuk pendeteksiannya.

Perbandingan antara penelitian-penelitian maupun sistem yang telah ada dengan sistem yang diajukan dapat dijabarkan sebagai berikut:

TABEL I
PERBANDINGAN PENELITIAN SEBELUMNYA

No	Peneliti	Aplikasi Kebencanaan	Drone	CNN	Hardware	Aplikasi
1	M. B. Berjiga, dkk [5]	√	√	√	-	Deteksi adanya objek pada longsor salju
2	I. A. Sulistijono, dkk [6]	√	√	√	-	Deteksi korban dengan latar belakang yang berantakan
3	I.A. Sulistijono, dkk [7]	√	√	√	Raspberry Pi 3, Pi Camera	Deteksi korban menggunakan framework pada software
4	D. Zhang, dkk [8]	√	-	-	Sensor CO2, kamera Thermal	Deteksi korban menggunakan robot dengan pendeteksian suara
5	M. Zacharine, dkk [9]	√	√	√	-	Deteksi korban dengan mengkonversi RGB menjadi HSV
6	P. Malhotra, dkk [10]	-	-	√	-	Membandingkan algoritma Deep Learning yaitu R-CNN, Fast R-CNN, dan YOLO
7	M. Adamu, dkk [11]	√	-	√	-	Membandingkan arsitektur CNN dengan arsitektur Mobilenet, VGG-16, dan YOLO
8	Sistem yang diajukan	√	√	√	Raspberry Pi 4, Pi Camera, LoRa	Deteksi klasifikasi dan menghitung jumlah korban menggunakan YOLOv4

Dari tabel diatas, dapat dilihat perbandingan sistem yang kami usulkan yaitu aplikasi kebencanaan menggunakan drone yang dilengkapi dengan komponen tambahan. Dimana komponen tambahan ini terdiri dari CNN dan alat komunikasi menggunakan modul LoRa. CNN digunakan untuk mendeteksi objek yang menggunakan modul Raspberry pi 4 model B. Pada penelitian ini memiliki keunggulan yakni drone akan mengirimkan teks berupa posisi objek dan jumlah objek yang terdeteksi, kemudian hanya akan dilakukan komunikasi ke server apabila objek terdeteksi, yang berguna untuk penghematan daya baterai. Dan pada penelitian ini kami masih berfokus pada proses pendeteksian dengan Raspberry Pi dan kamera, untuk penggunaan drone dan LoRa akan kami lakukan pada penelitian selanjutnya.

B. Deep Learning

Deep Learning adalah bidang baru dalam penelitian pembelajaran mesin. Pembuatannya termotivasi pada pembentukan jaringan saraf yang mensimulasikan otak manusia untuk pembelajaran secara analitis. *Deep Learning* meniru mekanisme otak manusia untuk menginterpretasikan data seperti gambar, suara dan teks [12].

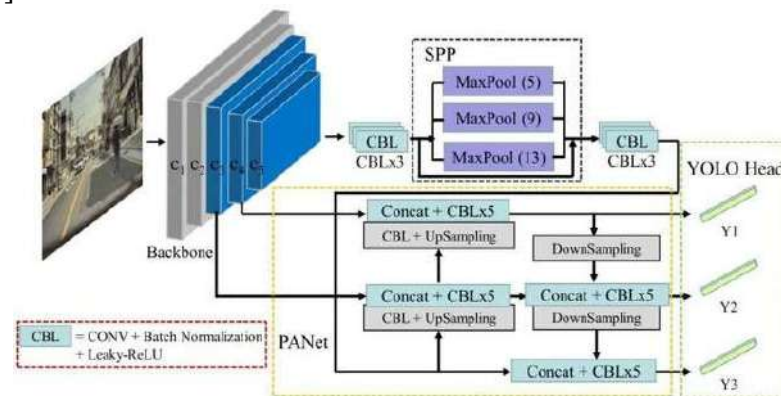
Konsep *Deep Learning* diusulkan oleh Hinton et al. [13] berdasarkan *Deep Belief Network* (DBN), di mana sebuah algoritma pelatihan unsupervised lapis layer demi layer pengawasan diusulkan yang memberikan harapan untuk memecahkan optimasi masalah struktur dalam. Selain itu, jaringan saraf konvolusi yang diusulkan oleh Lecun et al. [14] adalah algoritma pertama yang diusulkan untuk pembelajaran struktur multi-layer yang menggunakan hubungan ruang relatif untuk mengurangi jumlah parameter untuk meningkatkan kinerja pelatihan.

Deep Learning adalah metode pembelajaran mesin berdasarkan karakterisasi pembelajaran data. Pengamatan seperti gambar, dapat diekspresikan dalam berbagai cara, seperti vector dari setiap nilai intensitas piksel, dan lain-lain. Sama halnya dengan metode *Machine Learning*, metode *Deep Learning* juga memiliki *supervised learning* dan *unsupervised learning*. Dan

manfaat *Deep Learning* adalah penggunaan fitur *unsupervised* atau *semi-supervised learning* dan ekstraksi fitur hierarkis ke efisien mengganti fitur secara manual [15].

C. You Only Look Once (YOLO)

YOLO adalah sebuah metode untuk mendeteksi objek. YOLO memproses gambar secara real-time pada empat puluh lima (45) frames per second. YOLO dapat merepresentasi objek yang sangat umum [16], [17]. YOLO mendeteksi dengan sebuah *single convolutional network* dan memprediksi beberapa bounding boxes. Pertama yang dilakukan system YOLO membagi citra input ke dalam grid $S \times S$. Setiap sel grid memprediksi *bounding boxes* dan memprediksi nilai dari setiap *bounding box* tersebut. Nilai ini digunakan untuk mengetahui seberapa akurat model yang terdapat dalam sebuah objek didalam kotak tersebut. Setiap *bounding box* terdiri dari 5 prediksi: x , y , w , h , dan *confidence*. Koordinat (x, y) mewakili pusat dari kotak relatif ke batas sel grid. (w, h) atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Terakhir adalah *confidence* yang mewakili *Intersection over Union* (IoU) antara kotak prediksi dan kotak *groundtruth* [18].



Gambar 2. Arsitektur YOLO

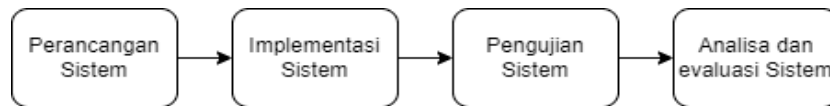
Untuk arsitektur YOLOv4 itu sendiri adalah menggunakan *Backbone*, *Neck*, dan *Head*. Pada *Backbone*, kami menggunakan *ResNet* untuk mengekstrak fiturnya. Kemudian diterapkan beberapa layer untuk mengekstrak fiturnya, lebih banyak layer pada arsitektur dapat menghasilkan level tingkatan fitur yang berbeda dengan semantic yang lebih tinggi [19], dan untuk neck berfungsi sebagai lapisan ekstra yang digunakan untuk mengekstrak feature maps dari berbagai stages pada *backbone*. Untuk head digunakan untuk mendeteksi objek dan untuk *bounding box*. Pada *head* biasanya diterapkan beberapa *hyperparameter* sesuai dengan kebutuhan penulis untuk berapa kelas yang akan digunakan. YOLOv4 ini memiliki keunggulan dalam proses *Convolution* yang cepat, sehingga kami menggunakan sistem ini untuk proses deteksi yang dilakukan pada drone.

D. Software

Pada penelitian ini, kami menggunakan service *Google Colaboratory* untuk men-training dataset-nya sebelum diterapkan pada hardware. Software ini menyediakan “Notebook Jupyter” tanpa server untuk pengembangan interaktif dan diberikan secara gratis [20]. Infrastruktur Google Colab memiliki spesifikasi CPU single core hyperthread Xeon Processor@2,3GHz, GPU yang ditenagai dengan Nvidia Tesla K80 296 CUDA cores, 12GB DDR5 VRAM, RAM 12,6GB dan Disk 29GB. Selain itu kami juga menggunakan VNC Viewer untuk Raspberry [21].

E. Desain dan Implementasi Sistem

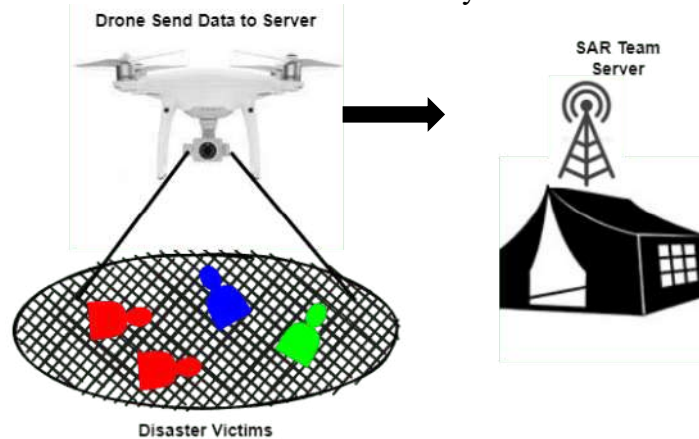
Berikut adalah proses bagaimana sistem akan bekerja. Langkah pertama pada desain sistem adalah membuat suatu blok diagram sebagai acuan dimana setiap blok mempunyai fungsi tertentu dan saling terkait sehingga membentuk sistem dari alat yang dibuat.



Gambar 3. Pemodelan Sistem

1. Perancangan Sistem

Sistem yang akan digunakan terdiri dari drone yang sudah dilengkapi dengan alat tambahan yang akan mengirimkan data ke server menggunakan sistem komunikasi wireless. Drone akan diterbangkan di daerah bencana untuk mendeteksi adanya korban dan letak posisi korban.



Gambar 4. Rancangan Sistem

Dari gambar diatas, dapat dilihat bahwa sistem terbagi menjadi 2, yakni di lokasi terdampak dan lokasi untuk penempatan server. Daerah terdampak akan digunakan untuk menerbangkan drone yang akan mengirimkan informasi posisi GPS dan jumlah objek yang terdeteksi. Sedangkan server terletak di daerah yang tidak jauh dari bencana akan tetapi aman dari bencana. Dan ketika drone mendeteksi adanya objek maka LoRa akan mengirimkan data ke server, jika tidak mendeteksi objek maka tidak akan mengirimkan data ke server. Hal ini mengakibatkan efisiensi energi dari baterai. Dan pada penelitian ini kami masih mengimplementasikan pada hardware Raspberry Pi, dan belum diuji di drone dan LoRa.



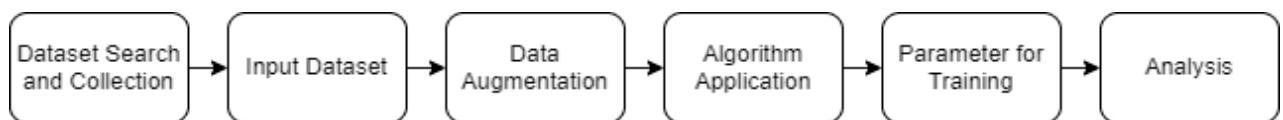
Gambar 5. Alat yang digunakan

Pada alat yang akan kami gunakan yaitu terdiri dari Raspberry Pi 4 4GB dan Raspberry Pi Camera 16MP IMX519. Raspberry Pi 4 4GB yang kami pilih ini memiliki spesifikasi yang paling tinggi, dimana akan mempercepat proses deteksi. Selain itu, Raspberry Pi Camera 16MP akan digunakan sebagai input awal yang akan diproses ke algoritma yang sudah dilatih. Pemilihan ini untuk mempertimbangkan kualitas gambar dan berat dari hardware yang akan dipasang di drone. Alat ini akan diintegrasikan dengan drone dengan cara memasangkan alat dibagian bawah drone sebagaimana pada gambar 4. Sedangkan pada server terdiri dari laptop untuk memonitoring hasilnya. Data yang dikirimkan yaitu berupa data deteksi korban dan jumlah korban yang terdeteksi.

F. Sistem Metodologi untuk Pendeteksian Korban

Sistem ini akan melakukan pengklasifikasian korban bencana alam dengan menggunakan algoritma CNN dengan arsitektur YOLOv4. Ada beberapa langkah sistem agar berkerja dengan baik. Pertama dilakukan pencarian dan pengumpulan dataset korban bencana alam, disini kami menggunakan custom dataset. Input dataset ini akan dibagi menjadi 80% untuk train set dan 20% untuk test set, hal ini mengikuti aturan umum yang biasa dilakukan, sehingga dengan pemagian tersebut kita dapat melihat hasil akurasi yang didapat dari test set.

Kemudian dilakukan augmentasi yang terdiri dari rotasi dan sharing. Augmentasi ini akan digunakan oleh setiap data untuk membuat 2 data baru. Sehingga semakin banyak dataset yang digunakan, akurasi yang didapat akan semakin tinggi. Setelah dilakukan augmentasi, maka diterapkan algoritma dengan menggunakan arsitektur YOLOv4 dan diterapkan beberapa parameter untuk training. Sehingga akan menghasilkan akurasi data yang dapat kita analisa dan juga mendapatkan file training yang paling bagus. Untuk keseluruhan layout sistem metodologi dalam penelitian ini ditunjukkan pada gambar berikut.



Gambar 6. Sistem Metodologi

G. Dataset

Dataset yang kami gunakan dalam penelitian ini adalah dataset korban bencana alam. Dataset ini didapat dari IEEE Dataset, kemudian kami gabungkan dengan custom dataset yang kami dapat. Dataset terdiri dari 2 kelas yang berbeda yaitu *Emergency* dan *Need Help*. Dimana pada kelas *Emergency*, korban dalam posisi berbaring dan kami definisikan bahwa korban membutuhkan pertolongan segera dan darurat, sedangkan pada kelas *Need Help*, korban dalam posisi duduk dan berdiri, dimana kami definisikan bahwa korban membutuhkan pertolongan. Dataset yang digunakan untuk eksperimen dapat dilihat pada table dibawah ini.

TABEL II
DATASET KORBAN BENCANA ALAM

Nama Label	Train set	Test set
<i>Emergency</i>	692	173
<i>Need Help</i>	692	173

Kemudian Proses training pada uji coba dilakukan dengan beberapa hyperparameter, proses training dengan hyperparameter ini akan digunakan pada algortima yang akan diterapkan. Untuk beberapa hyperparameter yang kami gunakan adalah sebagai berikut:

TABEL III
HYPERPARAMETER UNTUK TRAINING

Hyperparameter	Value
Learning Rate	0,0001
Epoch	6
Max Batch	6000
Step	1000
Input Size	224, 224

Dari *hyperparameter* diatas, kami menggunakan learning rate 0,0001, karena semakin kecil learning rate maka hasil training akan lebih maksimal. Kemudian kami menggunakan max

batch 6000 dengan step 1000 dan epoch 6 yang akan menghasilkan sebuah data hasil training setiap 1000 iterasi. Dikarenakan menggunakan 6 epoch maka dalam 1 epoch akan dilakukan 1000 iterasi sehingga data training yang dihasilkan ada 6 data. Nantinya dipilih data hasil training yang terbaik. Untuk input size adalah input dari gambar yang akan dilakukan training.

H. Evaluasi Model

Evaluasi model dapat kita lakukan untuk melihat hasil dari training yang sudah dilakukan, jika hasil yang didapatkan belum sesuai dengan keinginan, kita bisa merubah *hyperparameter* yang sudah dilakukan sebelumnya. Dan evaluasi model klasifikasi *Convolutional Neural Network (CNN)* dalam penelitian ini menggunakan pendekatan *confussion matrix*. Dimana pendekatan ini menampilkan hasil precision, accuracy, recall, F1-score dan mAP.

Untuk pengertian *precision* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Presisi ini harus memiliki nilai setinggi mungkin, sedangkan accuracy itu sendiri adalah rasio prediksi benar dengan keseluruhan data. Nilai ini nantinya jug akan dapat kita analisa untuk mengetahui algoritma mana yang memiliki akurasi yang paling optimal. Kemudian recall bisa dikatakan sebagai sensitifitas, dimana ini merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. F1-score merupakan perbandingan rata-rata presisi dan *recall*. Dan yang terakhir mAP *metrics* yang biasa digunakan untuk melihat performa model object detection. mAP menggambarkan seberapa presisi model kita memprediksi setiap kemungkinan yang ada pada data test.

III. HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai hasil pengujian. Hasil pengujian yang kami lakukan yaitu mendapatkan hasil training dengan arsitektur YOLOv4. Dimana hasil training ini akan dimasukkan ke sebuah program untuk melihat hasil deteksi menggunakan input dari video secara realtime dengan webcam. Kemudian pengujian deteksi korban dari webcam dan perhitungan jumlah korbannya dilakukan dengan mengubah tinggi kamera dan posisi objek ke kamera. Dan yang terakhir adalah perhitungan jumlah objek yang terdeteksi dengan objek lebih dari satu.

A. Hasil training dengan arsitektur YOLOv4

Training ini kami lakukan pada *Google Colab* dengan beberapa *hyperparameter* sebagaimana di Tabel IV. Dari beberapa hasil iterasi yang kami dapatkan, ada iterasi yang memiliki hasil terbaik. Iterasi terbaik inilah yang nantinya akan dimasukkan ke dalam program untuk mendeteksi korban secara realtime. Hal ini dilakukan pada *Google Colab* supaya tidak memberatkan pada raspberry pi 4 jika training dilakukan pada perangkat hardware. Hasil training yang kami dapatkan adalah sebagai berikut:

TABEL IV
HASIL EVALUASI DARI MODEL YOLOV4

Accuracy	92,29%
Precision	86%
Recall	88%
F1-Score	87%
mAP	92,20%

B. Pengujian dengan input sebuah video

Pengujian ini dilakukan untuk melihat hasil deteksi dan perhitungan dari sebuah masukan berupa video. Video ini memperlihatkan orang yang sedang berjalan yang terekam oleh CCTV seperti yang ditunjukkan pada gambar 16. Dan dari video ini akan kami lakukan analisa, apakah program sudah bisa mendeteksi dan menghitung jumlah orang terdeteksi.

Dan berdasarkan data dalam Tabel V, kami dapat menghitung akurasi dalam pengujian dengan input video. Hasil akurasi yang kami dapatkan yaitu sebesar 76%. Dimana pada kondisi tertentu seperti orang yang sedang membawa anak kecil dan orang yang setengah badannya tidak terlihat pada kamera membuat sistem tidak bekerja dengan optimal.



Gambar 6. Hasil deteksi dengan input video

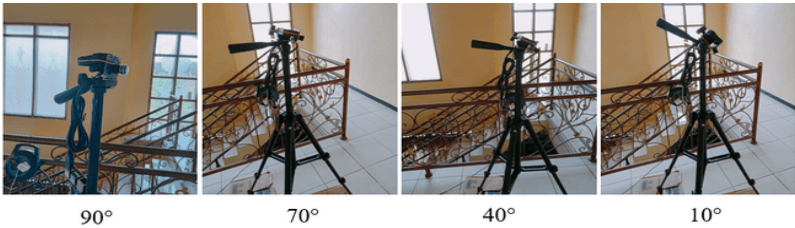
TABEL V
HASIL EVALUASI DARI MODEL YOLOV4

No	Jumlah Aktual	Jumlah Terdeteksi	TP	FP
1	10	9	9	1
2	9	9	9	-
3	13	9	9	4
4	12	8	8	4
5	11	9	9	2
6	13	10	10	3
7	14	10	10	4
8	15	9	9	6
9	14	11	11	3
10	14	11	11	3
Total			95	30

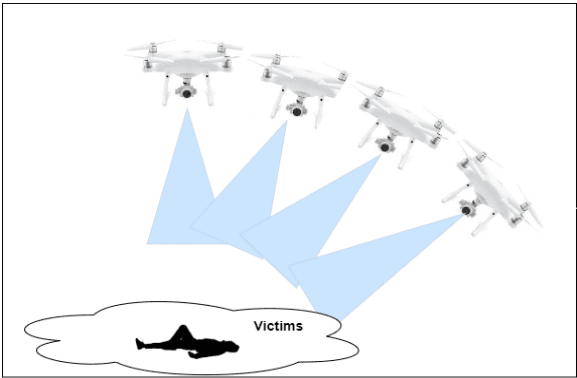
- $Recall = \frac{TP}{TP+FN} = \frac{95}{95} = 1$
- $Precision = \frac{TP}{TP+FP} = \frac{95}{125} = 0,76$
- $F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = \frac{1,52}{1,76} = 0,86$
- $Accuracy = \frac{TP}{TP+FN+FP} \times 100\% = \frac{95}{125} \times 100\% = 76\%$

C. Pengujian deteksi dengan kamera secara realtime

Pada pengujian kali ini kami melakukan deteksi korban secara realtime menggunakan Raspberry Pi 4 dan webcam. Pengujian ini kami lakukan dengan posisi kamera statis. Dan kami akan membandingkan akurasi dari berbagai tinggi kamera dan posisi sudut kamera. Untuk tinggi kamera kami menggunakan tinggi 1,5 meter, 2 meter, dan 3 meter. Dan untuk sudut kamera kami menggunakan sudut 90°, 70°, 40°, dan 10° seperti yang terlihat pada gambar 7. Untuk jarak korban dengan kamera yaitu 2 meter. Parameter pengujian ini yang meliputi variasi ketinggian dan sudut kamera serta jarak, nantinya dapat digunakan sebagai referensi untuk menguji di perangkat drone yang sesungguhnya.







Gambar 7. Sudut kamera dari 90° kearah bawah sampai 10°






Gambar 8. Skenario pengujian dengan kamera statis







Dan dari Gambar 7 tersebut kami akan menguji deteksi dengan beberapa posisi korban. Untuk hasil yang didapatkan ditunjukkan pada table dibawah ini:


TABEL VI
HASIL PENGUJIAN DENGAN TINGGI KAMERA 1,5 METER

Posisi Korban	Posisi Kamera							
	90°		70°		40°		10°	
	Detect	Tidak	Detect	Tidak	Detect	Tidak	Detect	Tidak
 Need Help	√ Akurasi: 98%	-	√ Akurasi: 98%	-	√ Akurasi: 98%	-	√ Akurasi: 84%	-
 Need Help	√ Akurasi: 99%	-	√ Akurasi: 94%	-	√ Akurasi: 83%	-	-	√
 Need Help	√ Akurasi: 97%	-	√ Akurasi: 98%	-	√ Akurasi: 92%	-	√ Akurasi: 94%	-
 Need Help	√ Akurasi: 92%	-	√ Akurasi: 99%	-	√ Akurasi: 96%	-	-	√






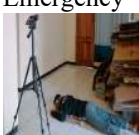

	✓ Akurasi: 83%	-	✓ Akurasi: 87%	-	✓ Akurasi: 98%	-	✓ Akurasi: 77%	-
Emergency								
	✓ Akurasi: 90%	-	✓ Akurasi: 88%	-	✓ Akurasi: 88%	-	✓ Akurasi: 93%	-
Emergency								
	✓ Akurasi: 97%	-	✓ Akurasi: 95%	-	✓ Akurasi: 62%	-	✓ Akurasi: 76%	-
Emergency								
Avg Akurasi	93%		94%		88%		61%	

TABEL VII
HASIL PENGUJIAN DENGAN TINGGI KAMERA 2 METER

Posisi Korban	Posisi Kamera							
	90°		70°		40°		10°	
	Detect	Tidak	Detect	Tidak	Detect	Tidak	Detect	Tidak
	✓ Akurasi: 98%	-	✓ Akurasi: 98%	-	✓ Akurasi: 97%	-	✓ Akurasi: 93%	-
Need Help								
	✓ Akurasi: 99%	-	✓ Akurasi: 95%	-	✓ Akurasi: 92%	-	✓ Akurasi: 96%	-
Need Help								
	✓ Akurasi: 80%	-	✓ Akurasi: 98%	-	✓ Akurasi: 61%	-	-	✓
Need Help								
	✓ Akurasi: 95%	-	✓ Akurasi: 96%	-	✓ Akurasi: 96%	-	✓ Akurasi: 69%	-
Need Help								
	✓ Akurasi: 98%	-	✓ Akurasi: 96%	-	✓ Akurasi: 96%	-	✓ Akurasi: 97%	-
Emergency								
	✓ Akurasi: 92%	-	✓ Akurasi: 89%	-	✓ Akurasi: 93%	-	✓ Akurasi: 92%	-
Emergency								

	✓ Akurasi: 98%	-	✓ Akurasi: 95%	-	✓ Akurasi: 94%	-	✓ Akurasi: 92%	-
Emergency								
Avg Akurasi	94%		95%		90%		77%	

TABEL VIII
HASIL PENGUJIAN DENGAN TINGGI KAMERA 3 METER

Posisi Korban	Posisi Kamera							
	90°		70°		40°		10°	
	Detect	Tidak	Detect	Tidak	Detect	Tidak	Detect	Tidak
	✓ Akurasi: 95%	-	✓ Akurasi: 97%	-	✓ Akurasi: 97%	-	✓ Akurasi: 56%	-
Need Help								
	✓ Akurasi: 88%	-	✓ Akurasi: 95%	-	✓ Akurasi: 87%	-	-	✓
Need Help								
	✓ Akurasi: 94%	-	✓ Akurasi: 98%	-	✓ Akurasi: 98%	-	✓ Akurasi: 95%	-
Need Help								
	✓ Akurasi: 86%	-	✓ Akurasi: 96%	-	✓ Akurasi: 97%	-	✓ Akurasi: 94%	-
Need Help								
	✓ Akurasi: 51%	-	✓ Akurasi: 89%	-	✓ Akurasi: 97%	-	✓ Akurasi: 78%	-
Emergency								
	-	✓	✓ Akurasi: 77%	-	✓ Akurasi: 92%	-	✓ Akurasi: 68%	-
Emergency								
	✓ Akurasi: 78%	-	✓ Akurasi: 91%	-	✓ Akurasi: 81%	-	✓ Akurasi: 81%	-
Emergency								
Avg Akurasi	70%		92%		92%		67%	

Dari hasil pengujian diatas, terlihat bahwa akurasi yang didapat pada ketinggian kamera 2-meter dan posisi sudut kamera 70° ke arah bawah mendapatkan akurasi paling tinggi yaitu sebesar 95%. Kemudian hasil yang kurang optimal ketika kamera dalam sudut 10° ke arah

bawah yaitu dengan rata-rata akurasi yaitu 68%, karena pada sudut tersebut korban berada hampir tegak lurus dengan kamera, sehingga sistem kurang dalam melakukan deteksi.

1. Pengujian Perhitungan Jumlah Objek Lebih Dari Satu

Pengujian ini dilakukan untuk mengetahui apakah sistem bisa mendeteksi objek lebih dari satu secara *realtime* menggunakan Raspberry Pi 4 dan webcam. Untuk posisi kamera itu sendiri kami tempatkan sesuai dengan hasil pengujian sebelumnya, yaitu pada tinggi kamera 2-meter dan posisi sudut kamera pada 70° kearah bawah. Kemudian kami akan memposisikan objek dengan beberapa pose dan beberapa jarak korban ke kamera. Untuk hasil pengujian terlihat bahwa sistem dapat bekerja dengan baik dan semua pose terdeteksi dengan benar.

TABEL IX
HASIL PENGUJIAN DENGAN OBJEK LEBIH DARI SATU

Uji ke-	Jumlah Korban dalam Frame	Jumlah Korban Hasil Deteksi Objek			
		Terdeteksi Emergency (TP)	Terdeteksi Need Help (TP)	Tidak Terdeteksi (FP)	Salah Mendeteksi (FN)
1	2 NH	0	2	0	0
2	1 NH, 1 EM	1	1	0	0
3	2 EM	2	0	0	0
4	3 NH	0	3	0	0
5	2 NH, 1 EM	1	1	0	1
6	1 NH, 2 EM	2	1	0	0
7	5 NH	0	2	3	0
8	4 NH, 1 EM	0	3	2	0
9	4 NH, 1 EM	1	3	1	0
10	4 NH, 1 EM	1	3	1	0
Total		8	19	7	1

Catatan: EM = *Emergency*, NH = *Need Help*

- $Recall = \frac{TP}{TP+FN} = \frac{27}{28} = 0,95$
- $Precision = \frac{TP}{TP+FP} = \frac{27}{35} = 0,77$
- $F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} = \frac{1,46}{1,72} = 0,85$
- $Accuracy = \frac{TP}{TP+FN+FP} \times 100\% = \frac{27}{36} \times 100\% = 75\%$

Dari hasil yang didapatkan dengan pengujian menggunakan webcam terlihat bahwa *accuracy* sistem sebesar 75%. Dimana dari hasil deteksi yang didapat, kamera tidak mampu mendeteksi jika korban terhalang dengan korban lainnya, sehingga *accuracy* yang didapatkan masih belum optimal.

IV. KESIMPULAN

Pada lokasi bencana alam, drone akan digunakan sebagai deteksi korban jika lokasi bencana sulit untuk diakses. Untuk dapat mendeteksi korban, drone akan dipasang alat tambahan yang sudah dilatih menggunakan algoritma *Concolutional Neural Network* (CNN) dan YOLOv4. Permasalahan yang dihadapi adalah bagaimana mendeteksi korban dengan akurasi yang baik dan menghitung jumlah korban dengan tepat. Pengujian pada penelitian ini belum menggunakan drone dan masih menggunakan kamera statis dengan berbagai scenario

pengujian yaitu variasi ketinggian kamera dan sudutnya, posisi korban, dan jarak kamera dengan korban untuk menguji kehandalan algoritma CNN pada perangkat hardware. Dalam melakukan deteksi pada objek menggunakan drone, hal yang paling memengaruhi adalah tinggi dari drone tersebut. Sehingga pada saat melakukan deteksi dengan drone yang lebih tinggi, akurasi dari deteksi akan semakin menurun. Dari hal tersebut kami melakukan pengujian deteksi korban dengan kamera statis terlebih dahulu untuk memvalidasi hal tersebut. Dari hasil pengujian yang sudah kami lakukan dengan memperhatikan tinggi kamera dan posisi dari kamera yaitu tinggi kamera 2-meter dan posisi kamera 70° ke arah bawah menghasilkan akurasi yang cukup optimal yakni sebesar 95%. Selain itu juga jika kamera dalam sudut 10° ke arah bawah, sistem akan kurang dalam melakukan deteksinya yaitu menghasilkan rata-rata akurasi sebesar 67%, hal ini karena objek dan kamera hampir tegak lurus posisinya, dan akurasi menurun dikarenakan dataset yang kami gunakan hampir tidak ada objek yang tegak lurus dengan kamera, sehingga hal ini memengaruhi hasil akurasi dari pengujian.

REFERENSI

- [1] Imam A. Sadisun, Dr. Eng, *"Pemahaman karakteristik bencana: Aspek fundamental dalam upaya mitigasi dan penanganan tanggap darurat bencana"*, Pusat Mitigasi Bencana – Institut Teknologi Bandung (ITB), Oktober 2018.
- [2] Didik Agys SP, *Bencana Alam, Bencana Teknologi, Racun Dan Polusi Udara; Sebuah Tinjauan Psikologi Lingkungan*, Buletin Psikologi, Vol. 13, No. 1, Juni 2018.
- [3] A. F. Aliek, M. Z. S. Hadi, N. R. Muhtadai, and A. Zainudin, "Intelligent System of Natural Disaster Victim Detection using Naïve Bayes Classifier", International Electronics Symposium (IES), 2021.
- [4] D. Yu-Nan and L. Guang-sheng, "Research and Discussion on Image Recognition and Classification Algorithm Based on Deep Learning", International Conference Machine Learning, Big Data and Business Intelligence (MLBDBI), November 2019.
- [5] M. B. Berjiga, A. Zeggada, F. Melgani, "Convolutional Neural Network for Near Real-time Object Detection from UAV Imagery in Avalanche Search and Rescue Operations", International Geoscience and Remote Sensing Symposium (IGARSS), 2017.
- [6] I. A. Sulistijono, and A. Risnumawan, "From Concrete to Abstract: Multilayer Neural Networks for Disaster Victims Detection", International Electronics Symposium (IES), pp. 93-98, 2017.
- [7] I. A. Sulistijono, T. Imansyah, M. muhajir, E. Sutoyo, M. K. Anwar, E. Satriyanto, A. Basuki, and A. Risnumawan, "Implementation of Victims Detection Framework on Post Disaster Scenario", International Electronics Symposium on Engineering Technology and Application (IES-ETA), 2018.
- [8] D. Zhang, S. Sessa, R. Kasai, S. Cosentino, C. Giacomo, Y. Mochida, H. Yamada, M. Guarnieri, and A. Takanishi, "Evaluation of a Sensor System for Detecting Humans Trapped under Rubble: Pilot Study", Multidisciplin Digital Publishing Institute (MDPI), 2018.
- [9] M. Zacharie, S. Fuji, and S. Minori, "Rapid Humnas Body Detection in Disaster Sites using Image Processing from Unmanned Aerial Vehicle (UAV) Cameras", International Conference on Intelligent Informatics and Biomedical Science (ICIIBMS), 2018.
- [10] P. Malhotra and E. Garg, "Object Detection Techniques: A Comparison", International Conference on Smart Structures and Systems (ICSSS), 2020.
- [11] M. Adamu, M. Zen, R. Widyatra, P. Kristalina, and A. Pratiarso, "Smart Victims Detection in Natural Disaster Using Deep Learning", International Electronics Symposium (IES), 2022.
- [12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep Learning", Nature Journal, vol. 521, no. 7553, pp. 436-444, 2017.

- [13] G. E. Hinton, "Deep Belief Networks", Sholarpedia, vol. 4, no. 6, pp. 5947, 2019.
- [14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 2015.
- [15] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity", IEEE Access, vol. 20, no. 6, pp. 2169-3536, 2017.
- [16] J. S. W. Hutaaruk, T. Matulatan, and N. Hayaty, "Vehicle detection in real time using the Android-based YOLO method", vol. 9, no. 1, pp. 8–14, 2020.
- [17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." 2018.
- [18] Khairunnas, E. Mulyanto, dan A. Zaini, "Making a Human Object Detection Module Using the YOLO Method for Mobile Robots", Jurnal Teknik ITS, Vol.10, 2021.
- [19] A. K. Lopian, S. Sompie, and P. Manembu, "You Only Look Once (YOLO) Implementation for Signature Pattern Classification", Jurnal Teknik Informatika, vol. 16, n0. 3, pp. 337-346, 2021.
- [20] E. Bisong, "Training a Neural Network", 2019.
- [21] S. L. Rohit and B. V. Tank, "IoT Based Health Monitoring System Using Raspberry Pi – Review", Internrtional Conference on Inventive Communication and Computational Tehnologies (ICICCT), 2018.