

# Deteksi Tanda-Tanda Kehidupan Pada Korban Bencana Alam Dengan Bantuan UAV Menggunakan Algoritma YOLO Dan *MoveNet Thunder*

1<sup>st</sup> Figo Plambudi Dwigantara  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

figorm@student.telkomuniversity.ac.id

2<sup>nd</sup> Casi Setianingsih  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3<sup>rd</sup> Meta Kallista  
Fakultas Teknik Elektro  
Universitas Telkom  
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

**Abstrak**— Bencana alam merupakan peristiwa alam yang mengakibatkan dampak besar bagi kehidupan manusia yang datang secara tiba-tiba dan tidak dapat dihindari oleh siapapun. Bencana alam juga dapat merusak dan menghancurkan bangunan sehingga pada pasca bencana alam seperti gempa bumi, tanah longsor, tsunami dan lain-lain, pencarian jalur darat terhambat karena permukaan daratan yang tidak stabil dan medan yang sulit dijangkau oleh Tim SAR (search and rescue). Pada penelitian ini menghasilkan output sistem pendeteksi korban bencana alam yang masih hidup atau tidak. Dalam pelaksanaannya, metode YOLO digunakan dalam penelitian ini untuk mengidentifikasi korban bencana alam. Dataset yang digunakan untuk mendeteksi korban bencana adalah dataset custom yang berisikan kurang lebih 400 gambar kelas manusia dengan berbagai pose. Algoritma yang digunakan untuk mendeteksi adanya korban atau tidaknya menggunakan metode You Only Look Once (YOLO) versi 5 yang dimana memiliki nilai cukup tinggi 90,75% untuk dapat mendeteksi objek manusia. Setelah dapat dideteksi manusia kemudian sistem akan melanjutkan pendeteksian hidup atau mati melalui titik skeleton yang ditentukan menggunakan Tensorflow MoveNet dengan akurasi 100% dari 14 video yang dicoba.

**Kata kunci**— Bencana Alam , Object Detection, Computer Vision, You Only Look Once (YOLO), Tensorflow, MoveNet.

## I. PENDAHULUAN

Hingga saat ini bencana alam masih sangat menakutkan dan menimbulkan banyak korban jiwa yang kehilangan nyawa. Bencana alam adalah bencana yang diakibatkan oleh peristiwa alam yang bersifat merusak dan mengganggu kehidupan manusia. Pada saat terjadinya bencana alam, evakuasi korban selamat yang cepat akan mengurangi kerugian nyawa dalam bencana alam. Bencana alam yang sangat besar seperti Tsunami Aceh, Gempa Palu memerlukan usaha pengolahan dalam hal tenaga, waktu dan biaya. Kemudian prosedur bantuan dan rehabilitasi untuk daerah yang terdegradasi. Dana yang diberikan negara dan diperoleh dari pembayaran bersama masih belum cukup untuk mengatasi masalah tersebut [1]. Tak jarang korbannya mencapai angka yang sangat signifikan. Akan tetapi, kondisi lokasi kejadian bencana memiliki tempat yang sulit dijangkau oleh tim SAR (search and rescue). Untuk melacak korban

bencana alam sehingga membutuhkan waktu yang lama dalam evakuasi.

UAV atau Unmanned Aerial Vehicle adalah pesawat yang diterbangkan tanpa awak atau unmanned. Global Navigation Satellite System (GLONASS) adalah sistem navigasi satelit yang dioperasikan dan digunakan oleh Rusia. UAV atau bisa dibidang drone merupakan salah satu sistem yang bisa dibidang sangat dapat membantu secara lebih efisien dalam hal pencarian korban bencana alam. Dengan terciptanya sebuah UAV (drone) yang diberikan kecerdasan buatan ini diharapkan mampu untuk membantu mencari ataupun mendeteksi korban bencana alam yang lebih efisiensi.

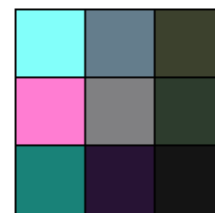
## II. KAJIAN TEORI

### A. Bencana Alam

Bencana alam merupakan suatu rangkaian peristiwa bencana yang diakibatkan oleh faktor alam, salah satu contoh bencana alam adalah gempa, tsunami, gunung Meletus, dll [2]. Bencana alam sangat merugikan baik bagi korban maupun keluarga yang terdampak. Oleh karena itu pasca bencana sangat penting untuk diperhatikan dikarenakan dengan penanganan yang tepat dan cepat akan mengurangi penderitaan akibat bencana alam yang terjadi.

### B. Pengolahan Citra

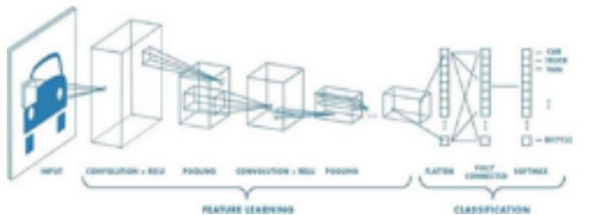
Pengolahan citra menggunakan perhitungan matematis pada objek secara piksel ataupun geometris sehingga memiliki nilai perbedaan yang dapat diperhitungkan dan dapat menunjukkan ciri yang berbeda antara objek satu dengan objek yang lainnya [3].



Gambar 1  
Potongan Citra Sampel.

C. Convolutional Neural Network

CNN (*Convolutional Neural Network*) adalah jaringan yang terkenal dalam jaringan saraf yang telah membuktikan diri area yang berbeda (pengenalan/pelokalan gambar, bahasa pengakuan, terjemahan bahasa, dll) kemudian pemrosesan CNN didasarkan pada perkalian kernel bobot 3D dan termasuk input 3D yang memiliki fitur dan keunggulan daya komputer dan bandwidth tinggi dengan mengandalkan konvolusi [4][5]. Konvolusi adalah operasi matematis yang bertujuan untuk mendapatkan suatu fungsi sehingga diperoleh gambaran rupa fungsi yang baru dari hasil perubahan awal yang sebelumnya [6].

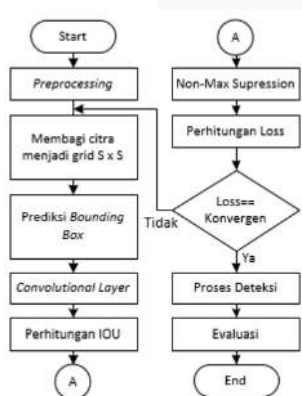


Gambar 2  
Ilustrasi Arsitektur CNN [7]

D. *You Only You Once* (YOLO)

YOLO adalah suatu metode pengenalan objek yang berbasis dan pengembangan pada CNN. Algoritma YOLO yaitu suatu algoritma yang dapat mendeteksi objek menggunakan citra yang dibagi menjadi beberapa grid yang mempunyai kinerja sangat baik dan akurasi tinggi [5].

Implementasi YOLO dapat digunakan diberbagai penelitian, salah satunya adalah penelitian yang berbasiskan objek dan yang pastinya memiliki tingkat akurasi yang sangat baik.



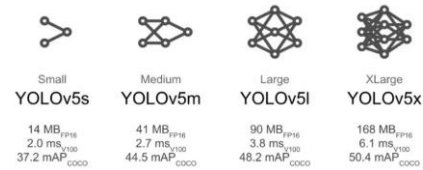
Gambar 3  
Diagram Alir Deteksi Yolo

E. YOLOv5

YOLOv5 merupakan salah satu pengembangan terbaru dari algoritma YOLO yang dirilis pada tahun 2020 sekaligus membawa banyak perbedaan daripada versi YOLO sebelumnya. YOLOv5 pada umumnya sangat baik digunakan untuk pendeteksian objek terutama objek yang terlihat jelas.

Arsitektur YOLOv5 berbeda dari rilis YOLO versi sebelumnya. Pada YOLOv5 menggunakan Pytorch dan CSPDarknet 53 untuk sebagai *backbone*. *Backbone* pada YOLOv5 mengintegrasikan perubahan gradien ke dalam fitur yang dapat mengurangi kecepatan inferensi, meningkatkan akurasi, dan mengurangi ukuran model menggunakan

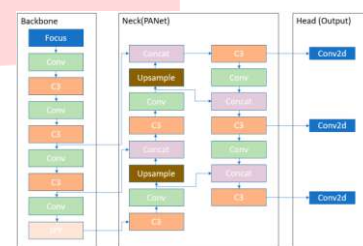
parameter ini. YOLOv5 mengadopsi PANet pada neck yang dapat meningkatkan propagasi tingkat rendah fitur dalam model, selain itu PANet dapat meningkatkan akurasi lokalisasi objek [8].



Gambar 4  
Jenis YOLOv5

	YOLOv3	YOLOv4	YOLOv5
Neural Network Type	Fully convolution	Fully convolution	Fully convolution
Backbone Feature Extractor	Darknet-53	CSPDarknet53	CSPDarknet53
Loss Function	Binary cross entropy	Binary cross entropy	Binary cross entropy and Logits loss function
Neck	FPN	SSP and PANet	PANet
Head	YOLO layer	YOLO layer	YOLO layer

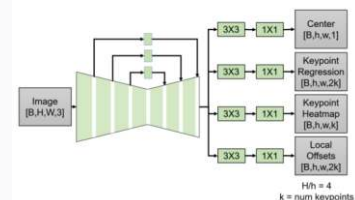
Gambar 5  
Perbandingan Arsitektur YOLO [8]



Gambar 6  
Arsitektur YOLOv5 [8]

F. *MoveNet*

*MoveNet* adalah model yang sangat cepat dan akurat yang mengenali 17 titik kunci pada tubuh. *MoveNet* merupakan pengembangan dari *Google Tensorflow* yang dapat mendeteksi *keypoint* pada manusia secara akurat [9].



Gambar 7  
Aliran dari *MoveNet*

1. *MoveNet Thunder*

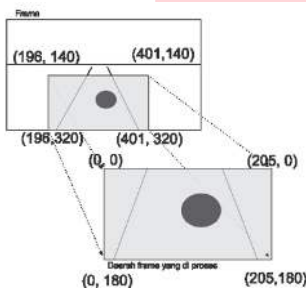
*MoveNet Thunder* adalah versi yang lebih akurat tetapi juga lebih besar dan lebih lambat dari *Lightning*. Ini berguna untuk kasus penggunaan yang membutuhkan akurasi lebih tinggi. Ekstraksi gambar *MobileNetV2* dengan dekoder Jaringan Piramida Fitur yang diikuti oleh header prediksi *CenterNet* dengan logika pasca-pemrosesan khusus. *Thunder* menggunakan faktor kedalaman 1,75 dengan keluaran *float32* bentuk tensor {1, 1, 17, 3}.



Gambar 8  
Hasil deteksi MoveNet Thunder

G. Region of Interest (ROI)

Region of Interest adalah bagian dari citra yang dipilih untuk diproses. ROI mempunyai fungsi untuk menandai suatu objek yang sedang bergerak. Algoritma deteksi objek dilakukan dengan menempatkan daerah ROI pada area jalan yang dilalui oleh kendaraan [10].



Gambar 9  
Proyeksi ROI

H. Confusion Matrix

Confusion Matrix adalah sebuah tabel yang menggambarkan performa dari sebuah model atau algoritma secara spesifik. Setiap baris dari matrix tersebut merepresentasikan class aktual dari data, dan setiap kolom merepresentasikan kelas prediksi dari suatu data [11]

Tabel 1

Confusion Matrix

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

1. True Positive = Banyaknya data aktual yang bernilai positif dan model juga memprediksi positif.
2. True Negative = Banyaknya data aktual yang bernilai negative dan model juga memprediksi negative.
3. False Positive = Banyaknya data aktual yang bernilai negatif, tetapi model memprediksi positif.
4. False Positive = Banyaknya data aktual yang bernilai positif, tetapi model memprediksi negative.

Dari 4 data tersebut, dapat diperoleh data lain yang

diperlukan untuk mengukur sebuah performa model, antara lain:

1. Presisi

Ketika model memprediksi positif, seberapa sering prediksi itu benar. Persamaan presisi dapat ditulis menggunakan persamaan 1.

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

2. Recall

Ketika kelas aktualnya positif, seberapa sering model memprediksi positif. Persamaan recall dapat ditulis menggunakan persamaan 2.

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

3. F1 – Score

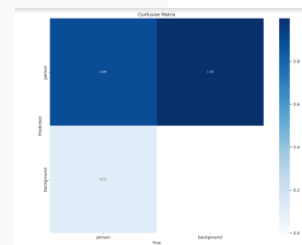
Merupakan rata-rata dari Precision dan Recall. Persamaan f1-score dapat ditulis menggunakan persamaan 3.

$$F1 - Score = 2 \times \frac{presisi \times recall}{presisi+recall} \quad (3)$$

4. Akurasi

Total keseluruhan seberapa sering model benar mengklasifikasi. Persamaan akurasi dapat ditulis menggunakan persamaan 4.

$$Akurasi = \frac{TP+TN}{Total} \quad (4)$$

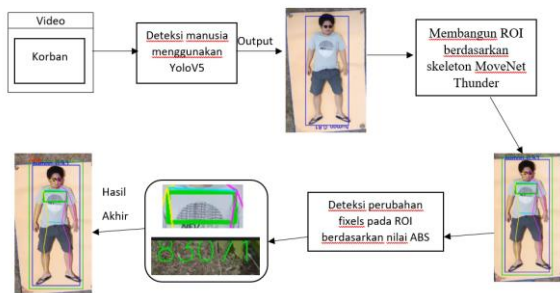


Gambar 10  
Contoh hasil confusion matrix YOLOv5

III. METODE

A. Gambaran Umum Sistem

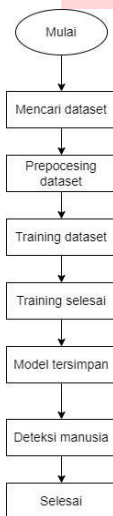
Pada penelitian ini menggunakan YOLOv5 untuk human detection, setelah korban dalam penelitian ini adalah manusia terdeteksi selanjutnya akan mendeteksi skeleton korban menggunakan Movenet Thunder yang bertujuan untuk membangun ROI (Region of Interest) yang dipusatkan pada bagian dada atau punggung korban. Berikut adalah gambaran sistem yang akan dirancang :



Gambar 11  
Gambaran Umum Sistem

B. Alur Pembuatan Model

Alur pembuatan model YoloV5 dapat dilihat pada gambar dibawah ini:



Gambar 12  
Flowchart Pembuatan Model

Langkah-langkah sistem dalam pembuatan model deteksi manusia adalah sebagai berikut :

1. Mulai melakukan tahapan untuk membuat model YoloV5.
2. Mencari dataset diambil dari google yang berjumlah awal 200 lebih.
3. *Pre-processing dataset* dimulai dari pelabelan manual pada roboflow hingga mendapatkan keysource untuk training dataset.
4. *Training dataset* proses untuk mendapatkan model dengan cara training pada Google Colab.
5. Setelah melakukan *training dataset* selanjutnya tinggal menunggu proses training berjalan.
6. Model yang sudah selesai di *training* akan tersimpan berupa "*best.pt*" dan "*last.pt*".
7. Setelah model tersimpan dalam bentuk "*best.pt*" dan "*last.pt*" maka model siap digunakan untuk mendeteksi manusia.

C. Deteksi Gerakan Dada atau Punggung Objek

Pada *flowchart* kali ini bertujuan untuk mengetahui apakah korban yang dideteksi hidup atau tidak hidup berdasarkan penjelasan *flowchart* berikut :



Gambar 13  
Flowchart Deteksi Kehidupan

Langkah-langkah sistem pendeteksian gerakan dada atau punggung korban bencana alam sebagai berikut:

1. Tahapan awal adalah sistem mendeteksi manusia.
2. Setelah sistem dapat mendeteksi manusia kemudian dilanjutkan untuk membentuk titik skeleton objek manusia menggunakan *movenet thunder*.
3. Setelah skeleton dapat dibangun pada objek manusia kemudian sistem membentuk ROI pada daerah dada ataupun punggung objek.
4. Setelah ROI dapat dibangun pada area dada ataupun punggung korban kemudian mendeteksi pergerakan daerah ROI tersebut menggunakan *motion detection*.
5. Setelah dideteksi ada pergerakan atau tidak sistem akan mengeluarkan hasil "*Died*" atau "*Alive*" berdasarkan pergerakan dada atau punggung objek

IV. HASIL DAN PEMBAHASAN

A. Skenario Training Model Deteksi Manusia

*Training* untuk *dataset* deteksi menggunakan 2 rasio berbeda yaitu rasio 90%:10%, dan rasio 80%:20% dengan keterangan data *train* : data *validation*. *Dataset* diambil dari *Google* yang telah disediakan kemudian diolah atau diberikan label manual menggunakan roboflow dan kemudian di *training* menggunakan *Google Colab*. *Dataset* tersebut terdapat satu *class* yaitu *person*.

Tabel 1  
Persentass Partisi Data

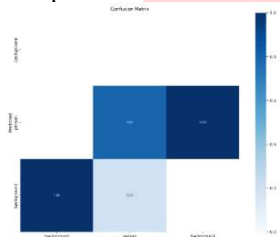
No	Data Train : Data Validation	Bat ch	Epo ch	Precis ion	Rec all	F1 - Sco re	Accur acy (%)
1	90% : 10%	2	5	0,679	0,611	0.643	63

2	80% : 20%	2	5	0.459	0.53	0.4 9	42
3	90% : 10%	10	200	0.923	0.75 9	0.8 33	80

Pada proses training menggunakan Google Colab nilai precision dan recall untuk YoloV5 langsung diberikan saat proses training selesai, kemudian dari precision dan recall dapat dicari nilai F1 – Score dengan persamaan 5.

$$F1 - Score = 2 \times \frac{presisi \times recall}{presisi+recall} \quad (5)$$

Dapat dilihat pada tabel 1 model terbaik saat training adalah dengan partisi dataset 90:10 dengan nilai batch 10 dan epoch 200, yang kemudian digunakan sebagai model dalam sistem yang akan diimplementasikan secara langsung.



Gambar 14  
Nilai akurasi training


B. Hasil Pengujian Sistem dan Analisis

Pada pembahasan ini akan menunjukkan hasil kinerja sistem setelah diimplementasikan dengan model yang sudah didapatkan melalui proses training sebelumnya serta analisis yang menjelaskan selama pengujian berlangsung.

1. Hasil Deteksi Manusia (YoloV5)

a. Manekin

Tabel 2  
Objek Manekin

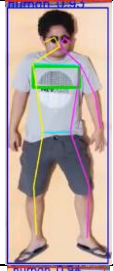



No	Postur	Hasil	Akurasi (%)
1	Terlentang (180 derajat)		93

Pada tabel 2 pengujian objek manekin untuk mengetahui akurasi deteksi manusia dengan 1 pose yaitu 93%.

b. Objek 1

Tabel 3  
Objek Manusia 1

No	Postur	Hasil	Akurasi (%)

1	Terlentang (180 derajat)		93
2	Telungkup (180 derajat)		94
3	Miring Kanan (180 derajat)		93
4	Miring Kiri (180 derajat)		94

Pada tabel 3 pengujian objek manusia 1 untuk mengetahui akurasi deteksi manusia dengan 4 pose berbeda ditunjukkan dengan persamaan 6 berikut :

$$Akurasi = \frac{ak1+ak2+ak3+ak4}{Total percobaan} \quad (6)$$

$$Akurasi = \frac{93+94+93+94}{4} = 93\%$$

dimana:

akurasi = Nilai akurasi deteksi manusia

ak1 = akurasi deteksi pada pose 1

ak2 = akurasi deteksi pada pose 2

ak3 = akurasi deteksi pada pose 3

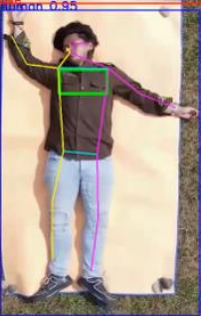

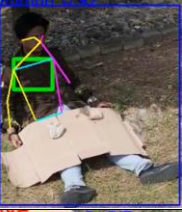
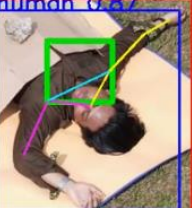
ak4 = akurasi deteksi pada pose 4

Total Percobaan = Total percobaan dari semua pose dalam pengujian

c. Objek 2

Tabel 4  
Objek Manusia 2

No	Postur	Hasil	Akurasi (%)

1	Terlentang (180 derajat)		95
2	Telungkup (180 derajat)		93
3	Duduk nyender pohon (90 derajat)		92
4	Terlentang dada tertutup (90 derajat)		87

Pada tabel 4 pengujian objek manusia 1 untuk mengetahui akurasi deteksi manusia dengan 4 pose berbeda ditunjukkan dengan persamaan 7 berikut :

$$Akurasi = \frac{ak1+ak2+ak3+ak4}{Total percobaan} \quad (7)$$

$$Akurasi = \frac{95+93+92+87}{4} = 91\%$$





dimana:

- akurasi = Nilai akurasi deteksi manusia
- ak1 = akurasi deteksi pada pose 1
- ak2 = akurasi deteksi pada pose 2
- ak3 = akurasi deteksi pada pose 3
- ak4 = akurasi deteksi pada pose 4
- Total Percobaan = Total percobaan dari semua pose dalam pengujian

d. Objek 3

Tabel 5  
Objek Manusia 3

No	Postur	Hasil	Akurasi (%)
----	--------	-------	-------------

1	Terlentang (180 derajat)		87
2	Telungkup (180 derajat)		86
3	Duduk nyender pohon (90 derajat)		90
4	Terlentang dada tertutup (90 derajat)		82

Pada tabel 5 pengujian objek manusia 1 untuk mengetahui akurasi deteksi manusia dengan 4 pose berbeda ditunjukkan dengan persamaan 8 berikut :

$$Akurasi = \frac{ak1+ak2+ak3+ak4}{Total percobaan} \quad (8)$$

$$Akurasi = \frac{87+86+90+82}{4} = 86\%$$

dimana:

- akurasi = Nilai akurasi deteksi manusia
- ak1 = akurasi deteksi pada pose 1
- ak2 = akurasi deteksi pada pose 2
- ak3 = akurasi deteksi pada pose 3
- ak4 = akurasi deteksi pada pose 4
- Total Percobaan = Total percobaan dari semua pose dalam pengujian.

2. Hasil Akhir Pengujian Deteksi Manusia

Setelah semua objek memiliki nilai akurasi dengan berbagai pose yang terdapat pada pengujian, selanjutnya didapatkan nilai akurasi akhir dari deteksi manusia pada korban bencana dengan persamaan 9 berikut :

$$Akurasi = \frac{M+Obj1+Obj2+Obj3}{Total percobaan} \quad (9)$$

$$Akurasi = \frac{93+93+91+86}{4} = 90,75\%$$

Dimana :

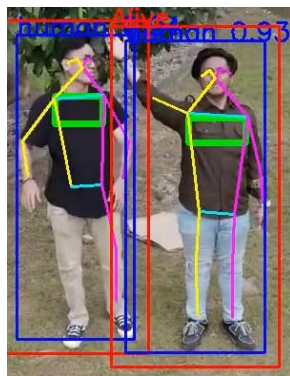
- akurasi = Nilai akurasi deteksi manusia
- M = akurasi deteksi objek manekin
- Obj1 = akurasi deteksi pada objek manusia 1
- Obj2 = akurasi deteksi pada objek manusia 2
- Obj3 = akurasi deteksi pada objek manusia 3
- Total Percobaan = Total percobaan dari semua pose dalam pengujian

3. Hasil Deteksi Tanda Kehidupan ROI dan Tangan

Berikut adalah hasil yang didapatkan dari sistem dengan 3 objek manusia dan 1 objek manekin.

Tabel 6  
Pengujian deteksi gerakan ROI

Subjek	Pose	Deceased	Alive	True/False
Manekin	Terlentang	√		True
Objek Manusia 1	Terlentang		√	True
	Telungkup		√	True
	Miring kanan		√	True
	Miring kiri		√	True
Objek Manusia 2	Terlentang		√	True
	Telungkup		√	True
	Duduk nyender		√	True
	Terlentang dada tertutup		√	True
Objek Manusia 3	Terlentang		√	True
	Telungkup		√	True
	Duduk nyender		√	True
	Terlentang dada tertutup		√	True
Objek Manusia tambah	Objek tangan bergerak		√	True



Gambar 15  
Deteksi gerakan tangan

#### 4. Hasil Akhir Deteksi Kehidupan

Dari semua pengujian yang telah dilakukan, maka dapat diambil nilai akurasi akhir untuk sistem dapat mendeteksi tanda-tanda kehidupan pada korban yang ditunjukkan pada persamaan 10 berikut ini:

$$Akurasi = \frac{M+Obj1+Obj2+Obj3+Dtangan}{Total Percobaan} \quad (10)$$

$$Akurasi = \frac{100+100+100+100+100}{5}$$

$$Akurasi = 100\%$$

dimana:

M = Akurasi yang didapatkan pada pengujian manekin  
 Obj1 = Akurasi yang didapatkan pada pengujian objek manusia 1  
 Obj2 = Akurasi yang didapatkan pada pengujian objek manusia 2  
 Obj3 = Akurasi yang didapatkan pada pengujian objek manusia 3  
 Dtangan = Akurasi yang didapatkan pada pengujian deteksi tangan  
 Total Percobaan = Total percobaan skenario pengujian

## V. KESIMPULAN

Berdasarkan implementasi sistem dan pengujian yang telah dijalankan, berikut adalah kesimpulan dari penelitian yang telah dilakukan sebagai berikut :

- Pembuatan mode yang sudah di *training* dengan berbagai parameter memiliki model terbaik dengan partisi dataset data *training*: data *validation* (90:10) dengan *Confusion Matrix* person (80:20) atau 80% akurasi pendeteksian manusia setelah diimplementasikan terhadap sistem mendapatkan akurasi akhir 90,75%.
- Hasil deteksi tanda kehidupan menggunakan *MoveNet Thunder* pada 4 objek yang dideteksi terdiri dari 13 objek manusia dan 1 objek manekin diperoleh akurasi 100%.

## REFERENSI

- Suprayitno dan Aryani Soemitro, "Pemikiran Awal tentang Prinsip Tindakan Mitigasi Preventif Resiko Bencana Alam bagi Manajemen Aset Infrastruktur & Fasilitas" *Jurnal Manajemen Aset Infrastruktur & Fasilitas* – Vol. 3, Edisi Khusus 1, Maret 2019.
- Rizki Wahyudi dan Tri Astuti, "Sistem Informasi Geografis (SIG) Pemetaan Bencana Alam Kabupaten Banyumas Berbasis Web" *Jurnal Teknologi dan Informasi* – Vol. 9, Edisi Maret 2019.
- Juju Jumadi, Yupianti dan Devi Sartika, "Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Hierarchical Agglomerative Clustering" *Jurnal Sains dan Teknologi* – Vol. 10, Tahun 2021.
- L. Mi-young, L. Joo-hyun, K. Jin-kyu, K. Byung-jo, dan K. Ju-yeob "The Sparsity and Activation Analysis of Compressed CNN Networks in a HW CNN Accelerator Model" *Jurnal IEEE International SoC Design Conference (ISODC)* tahun 2019.
- S. Moechammad, M. Nainul "Implementasi Algoritma You Only Look Once (YOLO) Untuk Deteksi Korban Bencana Alam" *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)* – Vol. 8 no 4, Agustus 2021.
- A. Qurotul, L. Ninda, K. Hendra, Z. Muhammad Suzaki "Deteksi dan Pengenaln Objek dengan Model Machine Learning : Model YOLO" *Journal of Computer Engineering System and Science* – Vol.6 No. 2, Juli 2021.
- K. Nur, S. Judi Prajetno "Deteksi Jalur Yang Terputus Pada Rangkaian Listrik dalam PCB Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Sistem Cerdas dan Rekayasa (JSCR)*, Volume 3, Nomor 1, Tahun 2021.
- N. Upesh, E. Hossein "Comparing Yolov3, Yolov4, Yolov5 for Autonomous Landing Spot Detection in Faulty UAVs" *Article Diakses* 4 Februari 2023.
- J. Beomjun, K. Seongki "Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices" *Journal of IIETA* – Vol. 39, no 1, Februari 2022.
- P. Awang Hendrianto, K. Wilis, M. Siti "Implementasi Algoritma Region of Interest (ROI) untuk Meningkatkan Performa Algoritma Deteksi dan Klasifikasi Kendaraan"

Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),  
Vol. 7 No. 1, Februari 2020.

- [11]I. W. Saputro, B. W. Sari “Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa,” Citec Journal, Vol. 6, No. 1, Januari 2019 ISSN: 2460-4259.

