

# Sistem Deteksi Jatuh Untuk Lansia Menggunakan *Openpose dan Convolutional Neural Network*

## *Fall Detection System For Elderly Using Openpose dan Convolutional Neural Network*

1<sup>st</sup> Ridho Adha Hardiyanto

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

dhohardiyanto@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> Tito Waluyo Purboyo

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

titowaluyo@telkomuniversity.a.cid

**Abstrak**—Manusia Ketika memasuki umur senja (lansia) membuat sistem kerja tubuh berkurang dari psikologi maupun jasmani mengalami penurunan fungsionalitas. sehingga rentan sekali terjadinya bahaya kepada lansia tersebut sehingga membuat keluarga lansia menginginkan sebuah solusi untuk membantu mereka apabila terjadinya sebuah marabahaya ke lansia Cara kerja sistem ini adalah dengan menggunakan webcam, yang mana akan mendeteksi pergerakan dan posisi lansia di dalam rumah dengan menggunakan algoritma You Only Look Once (YOLO) dan openpose. Cara kerja sistem ini , kamera perekam akan diposisikan pada suatu ruangan dengan area yang dideteksi seluas 3 x 3 meter lalu kamera di taruh di pojok atas ruangan, sistem akan melakukan inferensi input gambar dan di tugas akhir ini akan dibandingkan performansi yang terbaik antara sistem jatuh dengan openpose atau tidak dengan openpose. Hasil yang di dapat adalah model terbaik dengan performansi terbaik terdapat pada model openpose dengan mendapatkan hasil Presisi 100%, Recall 100%, F1 Score 100 mAP 100% serta akurasi yang dihasilkan mencapai 100% dengan parameter model yang digunakan adalah Rasio 70% : 30%, Batchsize 64, Learning rate 0.01.

**Kata Kunci**— *fall detection, lansia, yolo, object detection.*

**Abstract**—*Elderly Humans (seniors) When humans enter old age (elderly) make the body's work system, both psychological and physical, experience a decrease in functionality. Because it is very vulnerable to danger to the elderly, so that elderly families want a solution to help them in the event of a distress to the elderly. The way this system works is to use a webcam, which will detect the movement and position of the elderly in the house using the You Only algorithm. Look Once (YOLO). The YOLO algorithm applies a single neural network to the entire image. This network will divide the image into regions and then*

*predict bounding boxes and probabilities, for each bounding region box the probability is weighed so that it can classify an object or not. At the end, the bounding box with the highest value will be selected to be used as a separator of an object from another. How this system works is that first the recording camera will be positioned in a room with a detected area of 3 meters x 3 meters then the camera is placed in the upper corner of the room, the system will perform image input inference and in this final project will be compared which performance is the best between falling system with openpose or not with openpose. The results that have been obtained are the best model with the best performance in the openpose model by getting 100% precision, 100% recall, F1 score 100 mAP 100% and the resulting accuracy reaches 100% with the model parameters used is the ratio 70%: 30%, Batchsize 64, Learning rate 0.01.*

**Keywords**— *elderly, fall detection, yolo, object detection*

### I. PENDAHULUAN

Ketika manusia sudah memasuki masa usia senja, atau bisa disebut manusia lanjut usia (lansia) membuat sistem tubuh lansia, baik dari sisi psikologi maupun dari sisi fisiknya menurun dan tidak bekerja secara optimal[1].Apabila seorang lansia terjatuh dapat menyebabkan dampak kerusakan tubuh lansia tersebut ataupun penyakit lain akibat pertambahan usia lansia[2], yang bisa berakibat fatal ataupun hal yang tidak diinginkan. Penurunan fungsi tubuh dari lansia yang dikhawatirkan ini memerlukan sebuah solusi untuk membantu keluarga lansia agar dapat mengawasi maupun melakukan Tindakan apabila suatu hal terjadi kepada lansia tersebut.

Sistem yang bekerja secara realtime dan cerdas menjadi salah satu solusi[3] dimana salah satu sistemnya yaitu kecerdasan buatan menjadi jawaban untuk permasalahan yang disebut.Sistem secara realtime juga dibutuhkan agar lansia dapat terpantau

sekaligus dapat melihat apa saja yang terjadi terhadap lansia tersebut [4]. Salah satu kecerdasan buatan yang akan digunakan di tugas akhir ini adalah Openpose sebagai perangkat kerja yang akan melakukan deteksi objek dan Artificial Neural network, dan variasi Artificial neural network yang akan digunakan adalah YOLO (You Only Look Once) untuk melakukan deteksi apakah lansia jatuh atau tidak.

## II. KAJIAN TEORI

### A. Artificial Neural Network (ANN)

Artificial Neural Network atau jaringan saraf tiruan adalah sebuah teknik machine learning untuk membuat sebuah model AI (artificial intelligence) dengan mencontoh sistem jaringan saraf pada tubuh manusia. Proses pembuatan jaringan saraf tiruan adalah dengan menentukan titik – titik (node) yang teroganisir dengan setiap titik saling terhubung membentuk titik-titik saraf, lapisan – lapisan saraf tersebut biasa di kategorikan sebagai ,input , output , dan hidden node layer. Setelah jaringan saraf terbentuk kemudian untuk mengaktifkan setiap node berlaku dengan setiap fungsi aktivasi (activation function), yang mana fungsi aktivasi tersebut bisa berbentuk fungsi sigmoid, fungsi tanh, ataupun fungsi rectified linear unit (ReLU)[3]. Beberapa implementasi dari Artificial neural network antara lain, sparse encoder, convolutional neural network (CNN), restricted boltzmann machine (RBMs), dan Long short-term memory (LSTM).

### B. YOLO (You Only Look Once)

You Only Look Once adalah sebuah algoritma yang terbentuk dari model AI yang jaringan syaraf nya dibentuk dari Convolutional Neural Network (CNN) DarkNet[5]. YOLO Versi 1 diciptakan oleh Joseph Redmon pada tahun 2015, selanjutnya pada tahun 2017 Joseph Redmon dan Ali Farhandi merilis YOLO Versi 2 atau disebut YOLOv2 perbedaan dengan versi ke-1, YOLO Versi 2 meningkatkan performansi dan akurasinya dibandingkan dengan YOLOv1, Pada tahun 2018 Joseph Redmon dan Ali Farhandi kemudian merilis kembali Versi terbaru dari YOLO yaitu Yolo Versi 3 atau YOLOv3[6]. Dan yang terbaru pada tahun 2020 YOLOv5 rilis yang di develop oleh Glenn Jocher perbedaan YOLOv5 dibanding dengan YOLO versi sebelumnya YOLOv5 hadir dengan berbagai alat-alat bantu tambahan untuk melakukan deteksi dan pelatihan

untuk AI seperti teintegrasi dengan pytorch HUB, wandb AI, roboflow dan lain-lain. YOLO sesuai dengan kepanjangannya akan bekerja dengan cara dengan hanya 1 kali melihat untuk mendeteksi dengan 1 fungsi aktivasinya.

Paramater performansi matriks evaluasi pada YOLO CNN yang dapat dihasilkan saat model melakukan suatu pendeteksian adalah

#### 1. Intersection over Union (IoU)

Metric IoU adalah hasil ukur ununtuk mengevaluasi tingkat tumpang tindih antara kebenaran dasar (gt) dan prediksi (pd). Kebenaran dasar (gt) dan prediksi bisa berbentuk apa saja seperti (kotak, persegi panjang, lingkaran, atau bahkan bentuk tidak beraturan) dalam hal ini di yolo kita menggunakan bounding box (kotak)[7].

#### 2. Presisi

Presisi adalah pengukur kemampuan model untuk mengidentifikasi hanya objek yang relevan[7].

#### 3. Recall

Recall adalah pengukur kemampuan model untuk mendeteksi semua kebenaran objek berdasarkan proposisi di antara semua kebenaran objek[7].

#### 4. average precision (AP)

Average precision adalah hasil pengukuran model mewakili area di bawah kurva (AUC) Precision Recall Curve. Semakin tinggi kurva di sudut kanan atas, semakin besar areanya, sehingga semakin tinggi AP, dan semakin baik model pembelajaran mesin[7].

#### 5. mean average precision (mAP)

mean average precision adalah hasil rata-rata pengukuran dari rata-rata presisi (average precision) jadi semakin tinggi nilai mAP suatu model maka performansi nya semakin baik[7].

#### 6. Mean average precision .95(mAP@.95)

Metrik ini memperluas metrik AP@.5 dan AP@.75 dengan menghitung AP@ dengan 10 perbedaan IoU yang berbeda ( $t = [0,5, 0,55, \dots, 0,95]$ ) dan mengambil rata-rata di antara semua hasil yang dihitung[8].

### C. Pose Estimation

Pose estimation adalah sebuah teknik Machine learning untuk mengkategorikan tubuh manusia dengan menghubungkan titik-titik persendian dan anggota tubuh manusia[9]. Dengan bantuan Deep learning untuk melakukan klasifikasi pembentukan pose tubuh, pose estimation membuat komputer dapat membaca gerakan-gerakan tubuh, baik dari bentuk tangan, kepala, kaki, dll[10]. Saat ini banyak sekali bentuk-bentuk dari Pose estimation seperti, Openpose, alphapose, TF Pose estimation, openPifPaf, dll.



GAMBAR 1  
CONTOH POSE ESTIMATION [9]

#### D. Openpose

OpenPose adalah sistem deteksi multi real-time pertama yang dimana Openpose akan mendeteksi titik tubuh, tangan, wajah, dan kaki manusia (total 135 titik tubuh) pada satu gambar[11]. Yang diusulkan oleh para peneliti di Carnegie Mellon University pada tahun 2017. Mereka telah merilis dalam berbagai bentuk kode Python, implementasi C++ dan Plugin Unity[12]. Berikut ini adalah bentuk dari openpose:



GAMBAR 2  
CONTOH OPENPOSE

#### E. Inferensi pada machine learning

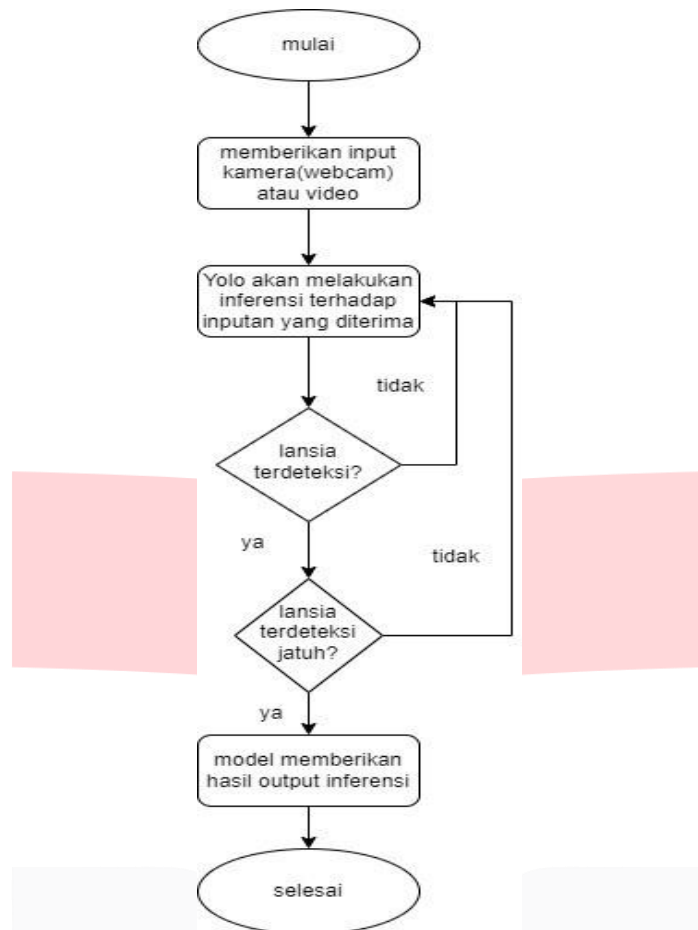
Inferensi pada machine learning adalah proses menalarakan sebuah prediksi dan pengeluaran hasil prediksi kearah hasil yang sebenarnya dari data ke dalam model pembelajaran mesin dan menghitung output Skor numerik tunggal. Proses ini juga disebut "pengaturan model pembelajaran mesin"[13]. Sementara menurut [14] inferensi pada machine

learning adalah penentuan tingkat pencocokan input saat ini mengenai setiap aturan dan pemutusan sesuai dengan input. Selanjutnya, aturan yang diabaikan digabungkan untuk membentuk tindakan kontrol pada machine learning. Inferensi menurut [15] adalah komponen dari sistem yang menerapkan aturan logis ke basis pengetahuan untuk menyimpulkan informasi baru. Mesin inferensi pertama adalah komponen sistem pakar. Sistem pakar tipikal terdiri dari basis pengetahuan dan mesin inferensi. Basis pengetahuan menyimpan fakta tentang dunia. Mesin inferensi menerapkan aturan logis ke basis pengetahuan dan menyimpulkan pengetahuan baru. Proses ini akan berulang karena setiap fakta baru di basis pengetahuan dapat memicu aturan tambahan di mesin inferensi. Mesin inferensi bekerja terutama dalam salah satu dari dua mode baik aturan khusus atau fakta: rantai maju dan rantai mundur. Forward chaining dimulai dengan fakta yang diketahui dan menegaskan fakta baru. Backward chaining dimulai dengan tujuan,

dan bekerja ke belakang untuk menentukan fakta apa yang harus ditegaskan agar tujuan dapat tercapai.

### III. METODE

#### A. Gambaran Umum Sistem

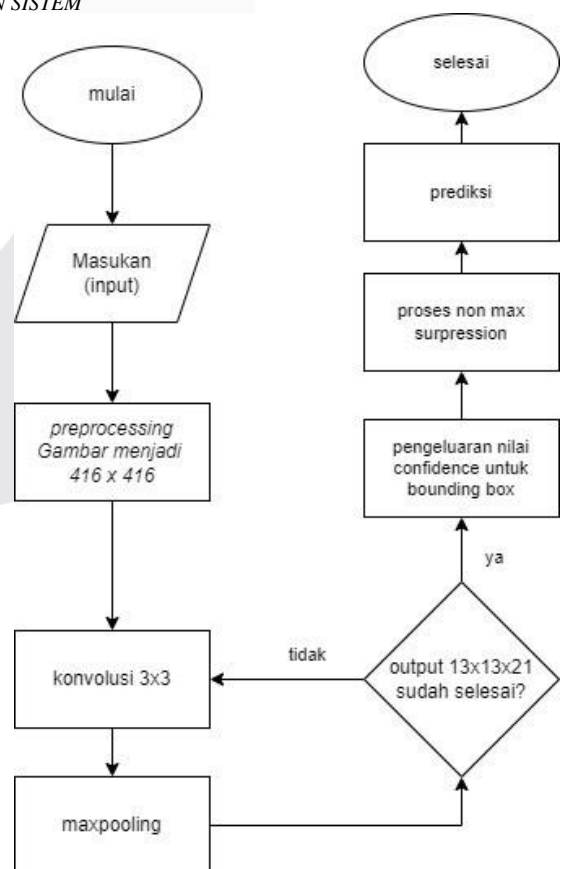


GAMBAR 3  
FLOWCHART KESELURUHAN SISTEM

Proses pertama yang akan dilakukan di sistem ini adalah user akan memberikan inputan kepada sistem baik dari sumber webcam atau video inputan, kemudian inputan tersebut akan dibaca dan dilakukan inferensi oleh sistem deteksi jatuh ini(Yolo inference), di saat proses inferensi ini sistem akan menentukan apakah elderly yang terdeteksi akan dideteksi jatuh atau tidak , apabila tidak maka akan terus di deteksi, dan apabila iya maka akan di outputkan hasil inferensi nya bahwa lansia terdeteksi jatuh.

B. Flowchart algoritma Yolo

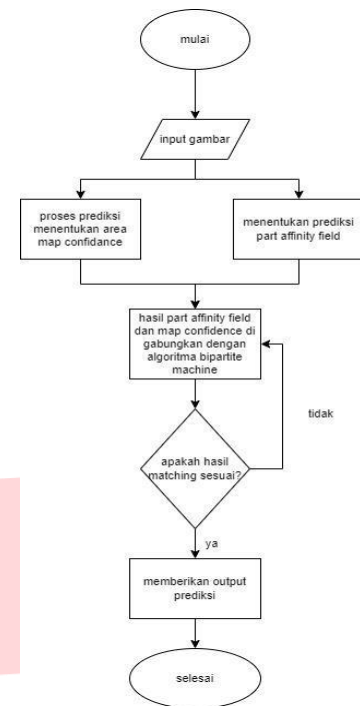
Sistem yang dibuat akan menggunakan algoritma You Only Look Once (YOLO). Berikut flowchart dari algoritma yolo pada sistem deteksi jatuh



GAMBAR 4  
FLOWCHART ALGORITMA YOLO

Proses pertama yang dilakukan algoritma YOLO adalah menginputkan dataset berupa image objek yang diinginkan seperti lansia dan terdeteksi jatuh untuk melakukan training. Kemudian dataset akan dilakukan sebuah perubahan dan membuat ukuran gambar dataset menjadi 416 x 416 pixels. Setelah itu Langkah selanjutnya adalah melakukan dua proses utama dalam algoritma ini, yaitu Convolution dan Max Pooling. Convolution atau konvolusi adalah suatu proses yang dimana hasil olahan gambar akan dimanipulasi dengan menggunakan eksternal mask / subwindows untuk menghasilkan sebuah gambar yang baru menggunakan kernel 3x3. Sementara Max Pooling merupakan proses mengecilkan input secara spasial (mengurangi jumlah parameter) dengan operasi down-sampling yang akan mengambil nilai terbesar dari bagian tersebut menggunakan kernel 2x2 dan stride 2 (Setiap matriks akan selalu terbagi menjadi setengahnya (416x416 menjadi 208x208 dst)). Kedua proses ini akan terus terulang sampai menghasilkan sebuah output grid cell yang bernilai 13x13x21 dan berakhir dengan 21 channels yang berisi data berguna untuk membentuk bounding boxes dan class prediction. 21 channels ini merupakan tiap sel grid yang memprediksi beberapa kotak pembatas dan tiap satu kotak pembatas akan dideskripsikan oleh beberapa elemen data. YOLO akan membagi gambar menjadi grid senilai 13x13 sel dan setiap bounding box, sel juga akan memprediksi kelas yang berfungsi sebagai klasifikasi. Skor kepercayaan untuk bounding box dan prediksi kelas akan digabungkan menjadi satu hasil akhir yang menghasilkan kemungkinan bahwa bounding box ini berisi seorang lansia atau terdeteksi jatuh. Terdapat 169 sel kotak (13x13) dan masing-masing selnya memprediksi beberapa bounding box. Setelah itu, ada sebagian kotak yang akan memiliki nilai confident rendah, sehingga algoritma YOLO hanya akan menyimpan dan menampilkan box yang memiliki skor prediksi diatas 30% lalu dilabelkan sesuai kelasnya. Proses menghilangkan bounding box yang mempunyai nilai confident yang rendah itu disebut dengan non- max Suppression.

### C. Flowchart openpose



GAMBAR 5  
FLOWCHART ALGORITMA YOLO

Proses pertama kali yang dilakukan dalam openpose adalah openpose akan menerima inputan suatu gambar, selanjutnya gambar akan melewati arsitektur CNN disini menggunakan CNN VGG-19, lalu hasil ekstraksi fitur akan melewati 2 tahap selanjutnya, yang pertama adalah openpose akan memberikan prediksi map confidence, yaitu titik-titik interest dari setiap pixel pada gambar yang diyakini sebagai bagian tubuh manusia, dan setelah itu bersamaan dengan prediksi map confidence, terjadi proses prediksi part affinity field[11] yang dimana disini ditentukannya letak titik lokasi tubuh manusia dalam gambarnya. Setelah 2 prediksi hasil tadi selesai tahap selanjutnya adalah menggabungkan hasil 2 prediksi tersebut dengan algoritma bipartite machine learning, kemudian Openpose akan menentukan apakah hasil penyamaan sesuai dengan dataset, apabila sesuai maka Openpose akan melakukan render gambar hasil prediksi.

### D. Kebutuhan dataset

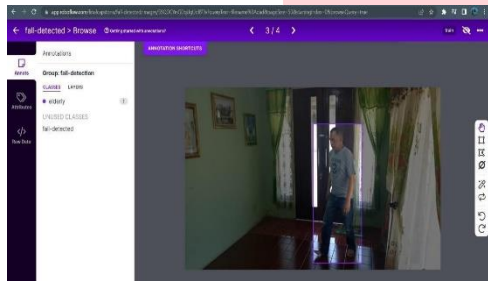
Dataset yang digunakan adalah dataset yang diambil secara manual. Pada perancangan tugas akhir ini, telah didapatkan 190 gambar tanpa openpose dan 240 gambar dengan openpose yang akan digunakan untuk training model klasifikasi. Gambar-gambar yang diambil adalah gambar lansia dan orang terdeteksi jatuh.



GAMBAR 6  
CONTOH GAMBAR DATASET

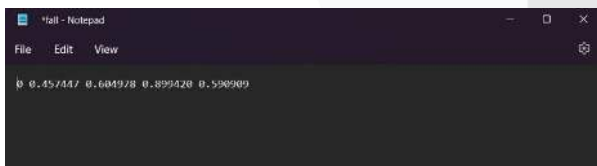
E. Anotasi pelabelan gambar

Setelah dataset diatur, dilakukanlah pelabelan. Anotasi merupakan sebuah proses membuat label dengan cara memberikan bounding box atau kotak batas beserta nama kelas pada objek gambar yang akan di training. Anotasi gambar ini dilakukan menggunakan web Roboflow.



GAMBAR 7  
PELABELAN GAMBAR

Setelah melakukan proses labeling pada dataset maka hasil dari labeling tersebut berupa titik koordinat yang disimpan dalam bentuk .txt dengan nilai 0 berupa elderly dan 1 berupa fall detected. Berikut tampilan koordinat setelah melakukan labeling dataset.



GAMBAR 8  
CONTOH HASIL ANOTASI GAMBAR

IV. HASIL DAN PEMBAHASAN

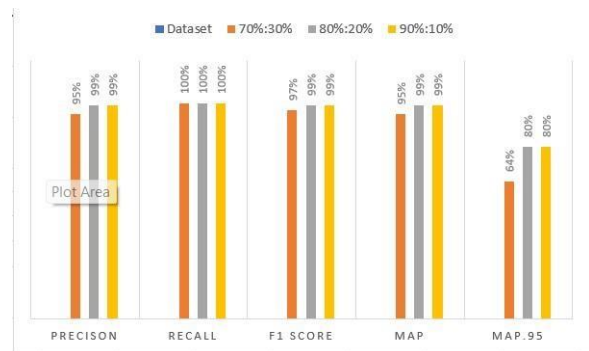
A. Pengujian training

Untuk skenario pengujian yang akan dilakukan dalam tugas akhir ini adalah pengujian training dengan dataset yang telah di labelkan secara manual dengan pengujiannya adalah pengujian rasio, learning rate, batchsize. Untuk pengujian rasio, dataset akan dibagi menjadi tiga rasio, yaitu 70%:30%, 80%:20%, dan 90%:10%. Untuk

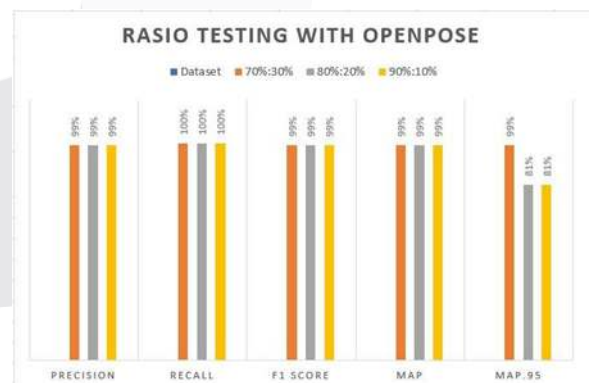
pengujian learning rate dibagi menjadi 0.01, 0.03, dan 0.05. Untuk pengujian batchsize dibagi menjadi 16, 32, dan 64. Pada proses training tersebut dilakukan menggunakan Google Colab Notebook.

1. Hasil pengujian rasio

Setelah dilakukannya training dengan pembagian dataset mulai dari 70% data training : 30% data test, 80% data training : 20% data test, dan 90% data training : 10% data test, terlihat bahwa rasio 90% data training : 10% data test menghasilkan performansi yang paling baik sebesar 99% yaitu model dengan openpose dengan konfigurasi rasio 70:30 %, learning rate 0.01, dan batchsize 64. Hal ini dapat terjadi dikarenakan semakin banyak dan semakin unik data yang dilatih, maka akan didapatkan model yang semakin baik dalam pembelajaran mesin dengan pemilihan data yang diambil secara acak. Berikut grafik perbandingannya.



GAMBAR 9  
TES RASIO TANPA OPENPOSE

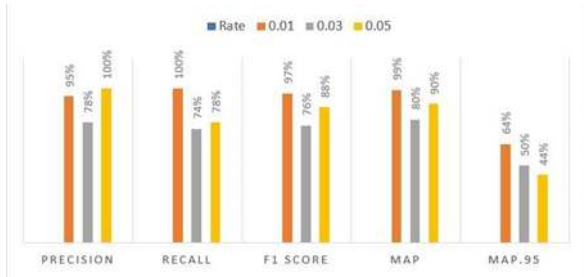


GAMBAR 10  
TES RASIO DENGAN OPENPOSE

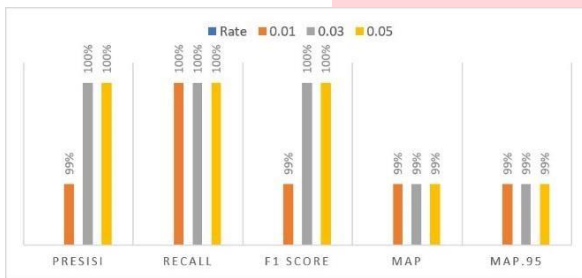
2. Hasil pengujian learning rate

Pada proses training dengan learning rate yang berbeda mulai dari 0.01, 0.03, dan 0.05 didapatkan hasil tertinggi sebesar 100% yang terdapat di model dengan learning rate 0.03 dan 0.05 dengan Openpose. Hal ini dapat terjadi dikarenakan semakin besar nilai learning

rately , maka akan didapatkan nilai akurasi yang lebih baik. Berikut hasil grafik perbandingannya.



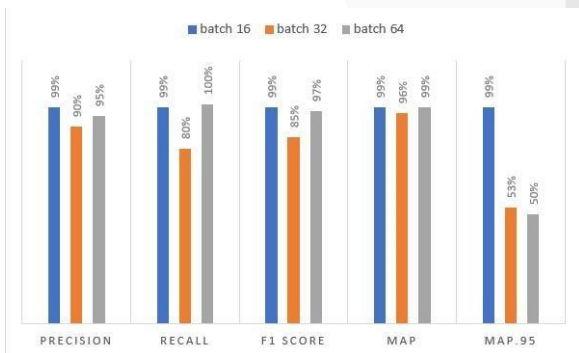
GAMBAR 11  
TES LEARNING RATE TANPA OPENPOSE



GAMBAR 12  
TES LEARNING RATE DENGAN OPENPOSE

3. Pengujian *batchsize*

Pada proses training dengan learning rate yang berbeda mulai dari 16, 32, dan 64 didapatkan hasil tertinggi sebesar 100% yang terdapat di model dengan batch size 32 dan 64 dengan Openpose. Hal ini dapat terjadi dikarenakan semakin batchsizenya , maka akan didapatkan nilai performansi yang lebih baik.



GAMBAR 13  
TES BATCH SIZE TANPA OPENPOSE



GAMBAR 14  
TES BATCH SIZE DENGAN OPENPOSE

B. Pengujian arah deteksi

Pengujian arah deteksi yang dilakukan pada tugas akhir ini harus menghasilkan deteksi fall-detected adalah sebagai berikut.



GAMBAR 15  
TES ARAH JATUH KIRI



GAMBAR 16  
TES ARAH JATUH KANAN



GAMBAR 17  
TES ARAH JATUH DEPAN



GAMBAR 18  
TES ARAH JATUH BELAKANG

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Berdasarkan hasil dari pengujian dan analisis yang telah dilakukan pada jurnal ini, maka dapat disimpulkan sebagai berikut:

1. Sistem deteksi jatuh lansia berhasil 100% mendeteksi lansia dan menentukan jatuh atau tidak dengan menggunakan algoritma YOLO (You Only Look Once).
2. Dengan melakukan pengujian hyperparameter pada algoritma yolo didapatkan performansi model terbaik pada Rasio 70% : 30%, Batchsize 64, Learning rate 0.1 dengan hasil Presisi 100%, Recall 100%, F1 Score 100%, mAP 100% serta akurasi yang dihasilkan mencapai 100%.
3. Selama melakukan pengujian bentuk dan arah objek yang di deteksi haruslah benar karena apabila melakukan kesalahan dari video/perekam kamera maka model akan menyebabkan salah deteksi

### B. Saran

Berdasarkan hasil penelitian dan pengujian yang dilakukan pada jurnal ini, maka saran yang dapat diusulkan untuk penelitian selanjutnya yaitu:

1. Menggunakan Google Colab Pro+ untuk melakukan training agar lamanya waktu training menjadi lebih cepat dan tidak terkena limit.
2. Menggunakan perangkat keras dengan spesifikasi yang tinggi seperti GPU NVIDIA dengan cuda core yang banyak dapat menggunakan versi YOLO yang lebih baik sehingga proses pendeteksian menjadi lebih baik dan mendapatkan fps yang stabil.
3. Menambahkan dataset yang lebih beragam lagi.

## REFERENSI

- [1] Leslie. Mayhew, *Health and elderly care expenditure in an aging world*. International Institute for Applied Systems Analysis, 2000.
- [2] D. Park, H. Jo, C. H. Yoon, E. S. Lee, M. K. Oh, and C. H. Lee, "Fall risk assessment of rural elderly population in Korea," *Ann Rehabil Med*, vol. 43, no. 3, pp. 269–278, Jun. 2019, doi: 10.5535/arm.2019.43.3.269.
- [3] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 53040–53065, 2019. doi: 10.1109/ACCESS.2019.2912200.
- [4] Z. He, D. Lu, Y. Yang, and M. Gao, "An Elderly Care System Based on Multiple Information Fusion," *J Healthc Eng*, vol. 2018, 2018, doi: 10.1155/2018/4098237.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Jun. 2015, [Online]. Available: <http://arxiv.org/abs/1506.02640>
- [6] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [7] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms."
- [8] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A



- comparative analysis of object detection metrics with a companion open- source toolkit,” *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–28, Feb. 2021, doi: 10.3390/electronics10030279.
- [9] M. Hofmann and D. M. Gavrilă, “Multi-view 3D human pose estimation in complex environment,” *Int J Comput Vis*, vol. 96, no. 1, pp. 103–124, Jan. 2012, doi: 10.1007/s11263-011-0451-1.
- [10] A. Toshev and C. Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks,” Dec. 2013, doi: 10.1109/CVPR.2014.214.
- [11] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “RMPE: Regional Multi-person Pose Estimation,” Nov. 2016, [Online]. Available: <http://arxiv.org/abs/1612.00137>
- [12] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” Dec.2018, [Online]. Available: <http://arxiv.org/abs/1812.08008>
- [13] M. H. Kazeminezhad, A. Etemad-Shahidi, and S. J. Mousavi, “Application of fuzzy inference system in the prediction of wave parameters,” *Ocean Engineering*, vol. 32, no. 14–15, pp. 1709–1725, Oct. 2005, doi: 10.1016/j.oceaneng.2005.02.001.
- [14] I. H. Sarker, “AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems,” *SN Comput Sci*, vol. 3, no. 2, Mar. 2022, doi: 10.1007/s42979-022-01043-x.
- [15] Hayes-Roth, D. A. (Donald A. Waterman, and D. B. Lenat, *Building expertsystems*, 1st ed., vol. 1. Reading, Mass. : Addison-Wesley Pub. Co., 1983.