

Rancang Bangun Sistem Deteksi Jatuh (*Fall Detection System*) pada Penghuni Rumah berbasis *Convolutional Neural Network*

¹Alejandro Junito, ²Sularso Budilaksono
¹Informatika, Universitas Persada Indonesia YAI, Jakarta
¹Informatika, Universitas Persada Indonesia YAI, Jakarta

E-mail: ¹alejandrojuni@gmail.com, ²sularso@upi-yai.ac.id

ABSTRAK

Insiden jatuh pada penghuni rumah merupakan masalah serius yang berdampak pada kesehatan fisik, psikologis, dan ekonomi. Proposal ini mengusulkan pengembangan sistem deteksi jatuh berbasis Convolutional Neural Network (CNN) yang diintegrasikan ke dalam aplikasi web menggunakan Flask. Sistem ini bertujuan untuk mendeteksi insiden jatuh secara akurat dan real-time tanpa memerlukan perangkat tambahan seperti IoT atau wearable devices. Model CNN akan dilatih menggunakan dataset publik seperti Le2i Fall Detection dan UP-Fall Detection untuk meningkatkan sensitivitas dan spesifisitas deteksi. Dengan mengadopsi algoritma YOLO, sistem mampu memproses data visual secara cepat dan efisien. Implementasi berbasis web memungkinkan penghuni rumah, pengasuh, atau keluarga untuk memantau kondisi penghuni rumah dengan mudah melalui antarmuka yang sederhana. Hasil dari penelitian ini diharapkan dapat memberikan solusi praktis untuk meningkatkan keamanan penghuni rumah, mendukung respons cepat terhadap insiden, dan meminimalkan dampak serius akibat keterlambatan penanganan.

Kata kunci : *deteksi jatuh, penghuni rumah, convolutional neural network, YOLO, Flask*

ABSTRACT

Falls among household members are a serious issue with significant impacts on physical, psychological, and economic well-being. This proposal outlines the development of a fall detection system based on a Convolutional Neural Network (CNN) integrated into a web application using Flask. The system aims to detect fall incidents accurately and in real-time without requiring additional devices such as IoT sensors or wearable gadgets. The CNN model will be trained using public datasets like the *Le2i Fall Detection Dataset* and *UP-Fall Detection Dataset* to enhance detection sensitivity and specificity. By adopting the YOLO (You Only Look Once) algorithm, the system can process visual data quickly and efficiently. The web-based implementation allows household members, caregivers, or family to monitor residents easily through a user-friendly interface. The outcomes of this research are expected to provide a practical solution to improve household safety, enable rapid response to incidents, and minimize severe consequences caused by delayed intervention.

Keyword : *fall detection, household members, Convolutional Neural Network, You Only Look Once, Flask*

1. PENDAHULUAN

Insiden jatuh pada penghuni rumah merupakan masalah serius dalam bidang kesehatan masyarakat global. Setiap orang, tanpa memandang usia, dapat berisiko mengalami jatuh akibat berbagai faktor, seperti penurunan kondisi fisik, gangguan keseimbangan, atau kecelakaan yang tidak terduga. Jatuh dapat menyebabkan cedera fisik seperti patah tulang, trauma kepala, serta dampak psikologis seperti rasa takut untuk bergerak, yang mengarah pada penurunan mobilitas dan kemandirian (Alanazi et al., 2022).

Jatuh merupakan salah satu faktor utama penyebab keterbatasan fisik dan meningkatkan risiko kematian pada lansia. Jika tidak segera ditangani, insiden ini dapat memperburuk kualitas hidup dan meningkatkan kebutuhan akan layanan kesehatan jangka panjang. Kondisi ini juga memberikan beban tambahan pada sistem kesehatan, keluarga, dan pengasuh, baik secara emosional maupun finansial. Oleh karena itu, pencegahan dan deteksi dini insiden jatuh sangat penting untuk meminimalkan dampak buruk dan memastikan respons cepat untuk memberikan pertolongan.

Dalam beberapa tahun terakhir, berbagai solusi teknologi telah dikembangkan untuk mendeteksi jatuh. Sistem deteksi tradisional, seperti akselerometer dan giroskop yang dipasang pada perangkat wearable, telah digunakan untuk mengidentifikasi pola gerakan abnormal. Namun, sistem ini memiliki keterbatasan, seperti banyaknya false positives (kesalahan deteksi), di mana aktivitas normal seperti duduk atau berbaring terdeteksi sebagai jatuh. Selain itu, perangkat wearable sering kali tidak nyaman digunakan, yang mengurangi efektivitasnya dalam jangka panjang (Casilari et al., 2020).

Melalui pendekatan ini, diharapkan sistem deteksi jatuh dapat memberikan solusi praktis dan efisien untuk

meningkatkan keamanan penghuni rumah, baik di rumah maupun di fasilitas kesehatan. Dengan adanya akses realtime melalui aplikasi web, pengasuh atau anggota keluarga dapat segera merespons jika terjadi insiden, sehingga mencegah risiko yang lebih besar akibat keterlambatan penanganan (Alanazi et al., 2022; Casilari et al., 2020).

2. LANDASAN TEORI

Computer Vision

Computer Vision adalah bidang yang dominan untuk digunakan pada penelitian dan serbaguna di era saat ini. *Computer Vision* memberikan intruksi pada komputer untuk dapat memahami dan menganalisis dengan pemahaman tingkat tinggi mengenai konten visual. Sub bidang dari *computer vision* meliputi pengenalan pada adegan atau objek, deteksi objek, pelacakan video, segmentasi objek, pose dan estimasi gerakan, pemodelan pemandangan, dan restorasi gambar (Morris dalam Diwan et al., 2023). *Computer vision* memiliki kemampuan untuk dapat mengembangkan sistem secara otomatis dengan melihat, mengenali, dan menganalisis gambar atau video. Tujuan utama yang dimiliki oleh *computer vision* adalah agar komputer dapat melihat dan memahami dunia nyata yang berupa visual layaknya manusia, dengan menggunakan algoritma dan teknik dalam melakukan proses gambar agar dapat dianalisis dan mendapatkan informasi yang berguna (Sutisna, 2024).

Deep Learning

Teknik dan algoritma dalam *deep learning* dapat digunakan dalam memenuhi kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*), dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. *Deep Learning* juga disebut sebagai *Deep* karena struktur dan jumlah

jaringan saraf yang ada pada algoritmanya sangat banyak hingga dapat mencapai hingga ratusan lapisan (Lorentius et al., 2019). Pendeteksian objek dapat dilakukan salah satunya adalah dengan menggunakan metode *deep learning*. Penggunaan dari *deep learning* didasarkan oleh beberapa faktor, contohnya adalah karena penggunaan data *training* dengan jumlah yang cukup banyak untuk dapat dipelajari dan dianalisis karakteristik dari data *training* tersebut, sehingga komputer memerlukan performa yang cukup tinggi agar data *training* dapat berkembang dengan signifikan (Zhao et al., 2019).

Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan salah satu bagian dari metode *deep learning* yang biasanya digunakan untuk melakukan pengolahan data yang berupa gambar atau video. CNN memiliki beberapa *layer*, seperti *input layer*, *output layer*, dan *layer* tersembunyi yang terdiri dari *convolution layer* yang berfungsi untuk mengekstraksi fitur dengan mengatur ukuran dari kernel, pergeseran setiap kernel, dan menambahkan *padding* pada *input* supaya ukuran dari *output* sama dengan ukuran *input*. *Pooling layer* yang berfungsi untuk dapat mengurangi dimensi agar *layer* berikutnya tidak memerlukan komputasi yang besar. *fully connected layer* yang berfungsi untuk melakukan klasifikasi berdasarkan fitur yang telah diekstrak (Narayan & Muthalagu, 2021).

CNN dibuat dengan menggunakan cara *Multi Layer Perceptron* dan banyak digunakan dalam pembuatan aplikasi *computer vision* seperti aplikasi untuk dapat melakukan klasifikasi pada citra, mendeteksi objek, dan melakukan pengenalan pada wajah manusia (Radikto, 2022). Metode CNN dianggap menghasilkan data yang paling signifikan pada pengenalan citra digital, karena CNN memang diimplementasikan berdasarkan sistem pengenalan pada citra

pada *visual cortex* manusia (Bili et al., 2022). Dengan memanfaatkan jaringan dari syaraf tiruan, CNN dapat mempelajari data-data yang telah tersedia dengan bantuan *deep learning*.

Referensi

Gunakan sistem *author-date* untuk menulis referensi, seperti '(Kosko, 1999)' atau 'menurut pendapat Kosko (1999)'. Gunakan nama belakang, sekali lagi, nama belakang pengarang diikuti tahun terbit. Jika Anda ingin menghemat energi, akan lebih baik jika Anda menginstall aplikasi EndNote pada komputer Anda. :-)

Persamaan

Ini juga masih menggunakan style. Untuk menulis persamaan matematika, pastikan untuk menggunakan Microsoft Equation dengan perintah **Insert | Object...** Jika Anda tidak bisa menemukan, sebaiknya Anda install dulu. Jika tidak memungkinkan, untuk persamaan yang sederhana tidak menjadi masalah diketik secara manual, klik  pada toolbar TRANSIT.

$$z_{in_j} = b1_j + \sum_{i=1}^n x_i v_{ij} \quad (1)$$

Gunakan style Persamaan untuk menulis persamaan seperti di atas. Pastikan Anda tidak lupa menuliskan nomor persamaan terurut menurut penampakan :-). Nomor persamaan ini dituliskan dalam kurung rapat kanan. Rujuk persamaan dengan dengan cara seperti ini: dalam Persamaan (1) dan seterusnya. Dalam menuliskan persamaan, gunakan font size 10 untuk variabel, sedangkan untuk pangkat dan indeks gunakan font size 8. Font dalam persamaan dengan style miring, kecuali untuk angka.

3. METODOLOGI

Pada penelitian kali ini menggunakan perancangan *waterfall* yang meliputi model dari SDLC (*System Development Life Cycle*), metode *waterfall* merupakan suatu perkembangan yang mengikuti Langkah-langkah secara berurutan dan linier, pendekatan tersebut meliputi berbagai fase yang harus diselesaikan secara berurutan dan harus selesai pada tahap berikutnya untuk dapat melanjutkan fase.

Tahapan dari metode SDLC dapat dijelaskan seperti berikut ini:

a) *Requirement Analysis*

Fase ini adalah tahap pengumpulan untuk kebutuhan dari perancangan aplikasi, mengidentifikasi secara merinci, dan melibatkan interaksi untuk memahami persyaratan sistem.

b) *Design*

Perancangan arsitektur dari sistem agar dapat memberikan gambaran dari rancangan aplikasi yang akan dibuat.

c) *Development*

Pembuatan kode program berdasarkan perancangan yang dibuat sebelumnya meliputi pengujian unit serta integrasi komponen

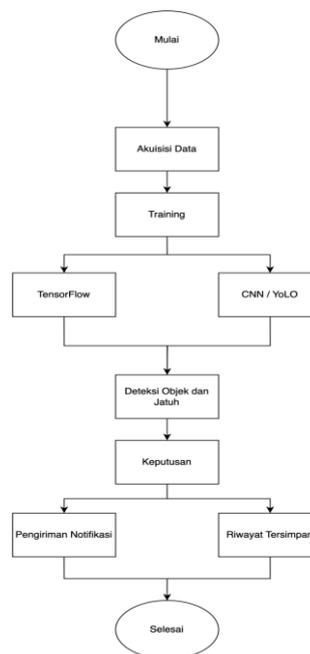
d) *Testing*

Tahap pengujian dalam memastikan sistem yang dirancang memenuhi syarat. Pengujian dilakukan meliputi pengujian fungsional, performa, dan yang lainnya.

e) *Maintenance*

Tahap pemeliharaan apabila diperlukan seperti perbaikan *bug* pada sistem, peningkatan dan pemeliharaan sistem lainnya.

Pada gambar di bawah adalah menjelaskan alur data dikumpulkan hingga aplikasi sudah menghasilkan *output* yang diharapkan.



Gambar 1. Flowchart Algoritma

4. HASIL DAN PEMBAHASAN

Implementasi Sistem

Implementasi adalah tahap dalam proses pengembangan sistem di mana desain yang telah dirancang sebelumnya diubah menjadi bentuk nyata atau operasional. Tujuan dari implementasi sistem ini adalah untuk menghasilkan sistem yang berfungsi sesuai dengan keinginan dan tujuan yang telah ditetapkan. Ini melibatkan penerapan semua elemen yang telah dirancang, termasuk perangkat lunak, perangkat keras, proses, dan sumber daya lain yang diperlukan untuk operasi normal sistem.

Analisa Hasil

Analisa hasil yang dilakukan dalam penelitian ini, bertujuan untuk melihat apakah proses dari pendeteksian jatuh melalui pengujian pada sistem berjalan dengan baik atau tidak, serta untuk mengetahui kekurangan apa saja yang terdapat pada sistem pada saat melakukan pendeteksian objek jatuh. Hal pertama yang dilakukan adalah dengan pengujian pada sistem aplikasi baik dari tampilan

antarmuka, logika, dan model pemrograman.

Tabel 1. Pengujian Hasil

No	Pengujian	Hasil
1	Halaman Daftar	Baik
2	Halaman Masuk	Baik
3	Halaman Dasbor	Baik
4	Halaman Riwayat	Baik
5	Halaman Deteksi	Baik

Tabel 2. Hasil Akurasi Model

No	Label	Akurasi
1	Jatuh	98.9%
2	Normal	98.9%

Hasil dari dilakukan beberapa pengujian terhadap objek jatuh, diperoleh data hasil pengujian beserta nilai kepercayaan (*confidence*) dari setiap deteksi yang telah dilakukan. Data yang dihasilkan mencakup berbagai parameter penting yang dianalisis secara mendetil untuk memastikan akurasi dan efektivitas dari metode deteksi yang digunakan. Setiap kali pengujian, metode dievaluasi berdasarkan kinerja dalam mendeteksi jatuh, dengan mempertimbangkan faktor-faktor seperti tingkat kesalahan (*error rate*), sensitivitas, dan spesifitas.

Algoritma Program Pendeteksian

```
1. result = model(frame)
2. for result in results:
3. if hasattr(result, "boxes"): for box
   in result.boxes:
4. confidence = box.conf[0].item()
5. class_id = int(box.cls[0].item())
6. label = LABEL_MAP.get(class_id,
   "Unknown")
7. if confidence >=
   CONFIDENCE_THRESHOLD:
8. x1,y1,x2,y2 = map(int,
   box.xyxy[0].tolist())
9. color = (0, 255, 0)
10. if label == "Jatuh":
11. cv2.putText(frame, f"(label)
   ({confidence:.2f})",
```

Gambar 2. Algoritma Program Pendeteksian

5. KESIMPULAN

1) Insiden jatuh pada lansia merupakan masalah serius yang membutuhkan solusi teknologi untuk deteksi dini. Penelitian ini menunjukkan bahwa algoritma YOLO, berbasis Convolutional Neural Network (CNN), mampu mendeteksi objek jatuh dalam bentuk citra atau gambar dengan tingkat akurasi tinggi, yakni sebesar 98% secara keseluruhan. Hal ini mendukung upaya dalam meminimalkan dampak insiden jatuh yang dapat membahayakan kualitas hidup lansia.

2) Implementasi YOLO dalam aplikasi berbasis web menggunakan framework Flask memberikan solusi yang fleksibel dan mudah diakses oleh pengasuh atau keluarga melalui browser. Hal ini menunjukkan efisiensi dalam pendeteksian real-time tanpa memerlukan perangkat tambahan seperti IoT atau perangkat wearable, sesuai dengan tujuan penelitian.

6. UCAPAN TERIMA KASIH

Terima kasih kepada :

1. Kedua Orang Tua dan keluarga yang telah memberikan doa dan dukungan kepada penulis dalam segala hal baik moril maupun material.
2. Ibu Prof. Ir. Sri Astuti Indriyati, MS, Ph selaku Rektor Universitas Persada Indonesia Y.A.I
3. Ibu Henni, ST. MT selaku Dekan Fakultas Teknik Universitas Persada Indonesia Y.A.I
4. Bapak M. Anno Suwarno., S.kom., M.Kom selaku Ketua Program Studi Teknik Informatika Strata 1 (satu) Universitas Persada Indonesia Y.A.I

DAFTAR PUSTAKA

Alanazi, T., & Muhammad, G. (2022). Human Fall Detection Using 3D Multi-Stream Convolutional Neural Networks with Fusion. *Diagnostics*, 12(12), 3060. Diperoleh dari MDPI.

- Altman, D.G., & Bland, J.M. (1994). Diagnostic Tests 2: Predictive Values. *BMJ*.
- Ambrose, A. F., Paul, G., & Hausdorff, J. M. (2013). Risk factors for falls among older adults: A review of the literature. *Maturitas*, 75(1), 51-61.
- Casilari, E., & Álvarez-Marco, M. (2020). A Study of the Use of Gyroscope Measurements in Wearable Fall Detection Systems. *Symmetry*, 12(4), 649.
- Dong, W., et al. (2021). A Real-Time Face Recognition System Based on Flask and CNN. *Journal of Intelligent & Fuzzy Systems*, 40(3), 5647-5654.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
- Muro-De-La-Herran, A., Garcia-Zapirain, B., & Mendez-Zorrilla, A. (2020). Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications. *Sensors*, 14(2), 3362-3394.
- Kwolek, B., & Kepski, M. (2015). Human Fall Detection Using Depth Maps and Wireless Accelerometer. *Sensors*.
- Kong, Y., Huang, J., & Wang, S. (2019). Learning Spatiotemporal Representations for Human Fall Detection in Surveillance Video. *Journal of Visual Communication and Image Representation*.