Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

Sistem Deteksi Jatuh Lansia *Real-Time* Berbasis YOLOv8 dengan Notifikasi Telegram dan *Dashboard* Web

Claudio Syanu Mareta Dinata*, Rina Firliana, Arie Nugroho

Fakultas Teknik dan Ilmu Komputer, Sistem Informasi, Universitas Nusantara PGRI Kediri, Kediri, Indonesia
Email: claudiosyanu@gmail.com[1*], rina@unpkediri.ac.id[2], arienugroho@unpkediri.ac.id[3]
Email Penulis Korespondensi: claudiosyanu@gmail.com

Abstrak- Kejadian jatuh pada lansia merupakan permasalahan kesehatan global yang serius. Menurut WHO, satu dari tiga lansia berusia di atas 65 tahun mengalami kejadian jatuh setiap tahunnya. Di lingkungan Panti Werda Kediri, pemantauan aktivitas lansia masih dilakukan secara manual oleh petugas sehingga kejadian jatuh tidak selalu terdeteksi dengan cepat. Penelitian terdahulu mengenai deteksi jatuh berbasis YOLO umumnya hanya menghasilkan model tanpa mengintegrasikannya ke dalam platform monitoring yang dapat digunakan langsung oleh pengguna non-teknis dan tanpa notifikasi otomatis. Penelitian ini bertujuan untuk mengetahui efektivitas sistem monitoring dalam mendeteksi aktivitas normal dan kejadian jatuh pada lansia secara *real-time* di Panti Werda Kediri menggunakan YOLOv8. Sistem dikembangkan menggunakan metode *Waterfall* melalui tahapan analisis kebutuhan, desain sistem, implementasi, dan pengujian. Komponen deteksi menggunakan model YOLOv8n yang dilatih ulang pada dataset UR Fall dengan 2.600 gambar untuk mengenali dua kelas, yaitu kondisi normal dan kondisi jatuh. *Backend* dibangun menggunakan FastAPI dan PostgreSQL, dilengkapi *dashboard* monitoring berbasis web dan notifikasi otomatis melalui Telegram Bot. Mekanisme konfirmasi jatuh diterapkan berdasarkan 3 *frame* berturut-turut dengan *cooldown* 1,5 detik untuk menekan *false positive*. Evaluasi model menghasilkan *precision* 93,9%, *recall* 97,9%, mAP@0.5 99,1%, dan mAP@0.5:0.95 81,8%. Hasil pengujian *blackbox* yang dilaksanakan di Panti Werda Kediri menunjukkan bahwa seluruh 10 skenario pengujian dinyatakan valid. Sistem berhasil mengirimkan notifikasi Telegram secara *real-time* dalam waktu kurang dari 2 detik beserta bukti visual setiap kali kejadian jatuh terkonfirmasi, menyajikan *live streaming* kamera, dan menampilkan seluruh riwayat deteksi melalui *dashboard* web yang dapat diakses petugas tanpa latar belakang teknis.

Kata Kunci: *fall detection*; YOLOv8; *real-time*; *dashboard* monitoring; notifikasi Telegram; *computer vision*

Abstract- Falls in the elderly represent a serious global health problem. According to the WHO, one in three elderly people aged over 65 years experiences a fall each year. In Panti Werda Kediri, monitoring of elderly activities is still performed manually by staff, causing fall incidents to go undetected quickly. Previous YOLO-based fall detection studies generally produce models without integrating them into monitoring platforms usable by non-technical end users and without automatic notification. This research aims to determine the effectiveness of a monitoring system in detecting normal activities and fall incidents in elderly residents in real time at Panti Werda Kediri using YOLOv8. The system was developed using the Waterfall method through stages of requirements analysis, system design, implementation, and testing. The detection component uses a retrained YOLOv8n model on the UR Fall dataset with 2,600 images to recognize two classes: normal and fall. The backend is built with FastAPI and PostgreSQL, equipped with a web-based monitoring dashboard and automatic notifications via Telegram Bot. A fall confirmation mechanism based on 3 consecutive frames with a 1.5-second cooldown suppresses false positives. Model evaluation yielded precision of 93.9%, recall of 97.9%, mAP@0.5 of 99.1%, and mAP@0.5:0.95 of 81.8%. Blackbox testing conducted at Panti Werda Kediri shows all 10 test scenarios passed. The system successfully sends real-time Telegram notifications in under 2 seconds with visual evidence each time a fall is confirmed, provides live camera streaming, and displays complete detection history through a web dashboard accessible to non-technical staff.

Keywords: fall detection; YOLOv8; real-time; dashboard monitoring; Telegram notification; computer vision

1. PENDAHULUAN

Proses penuaan yang tidak dapat dihindari membawa konsekuensi berupa penurunan fungsi fisiologis secara bertahap pada kelompok lanjut usia (lansia), mencakup melemahnya kekuatan otot, berkurangnya kemampuan keseimbangan, serta melambatnya respons refleks terhadap perubahan lingkungan sekitar. Kombinasi dari ketiga faktor tersebut menempatkan lansia pada posisi yang sangat rentan terhadap risiko kejadian jatuh dalam rutinitas sehari-hari. Dampak yang ditimbulkan tidak hanya bersifat fisik patah tulang pinggul dan cedera kepala merupakan komplikasi yang paling sering terjadi tetapi juga menyentuh dimensi psikologis berupa tumbuhnya rasa takut untuk bergerak bebas yang berakibat pada penurunan kualitas hidup secara menyeluruh. WHO mencatat bahwa insiden jatuh dialami oleh sekitar 30% populasi lansia berusia 65 tahun ke atas setiap tahunnya, menjadikannya ancaman kesehatan yang tidak dapat diabaikan pada kelompok usia ini [1]. Tanpa penanganan yang cepat dan tepat, konsekuensi dari kejadian jatuh dapat berujung fatal, terutama pada lansia dengan kondisi kesehatan yang sudah menurun.

Fenomena penuaan populasi turut terjadi di Indonesia dalam skala yang semakin besar. Data Badan Pusat Statistik memperlihatkan tren pertumbuhan proporsi penduduk lansia yang konsisten dari tahun ke tahun dan diproyeksikan akan terus berlanjut pada dekade mendatang [2]. Merespons realitas tersebut, Kementerian Kesehatan Republik Indonesia menetapkan pemantauan kesehatan proaktif pada kelompok lansia sebagai salah satu prioritas dalam agenda peningkatan kualitas layanan kesehatan nasional [3]. Di lingkungan Panti Werda Kediri, pemantauan terhadap aktivitas lansia umumnya masih dilakukan secara manual oleh petugas. Keterbatasan jumlah tenaga pengawas serta tidak adanya

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

sistem pemantauan yang berjalan secara terus-menerus menyebabkan kejadian jatuh tidak selalu dapat terdeteksi dengan cepat. Kondisi ini berpotensi menimbulkan keterlambatan dalam penanganan darurat, terutama apabila kejadian terjadi di luar jam pengawasan aktif petugas. Selain itu, sistem pemantauan yang ada juga belum dilengkapi dengan *dashboard* monitoring yang informatif maupun mekanisme notifikasi otomatis, sehingga petugas tidak dapat menerima peringatan secara langsung ketika terjadi kondisi darurat. Oleh karena itu, diperlukan solusi berbasis teknologi yang mampu mendeteksi kejadian jatuh secara otomatis dan memberikan notifikasi segera kepada petugas tanpa bergantung pada pengawasan manual.

Kemajuan teknologi di bidang kecerdasan buatan, khususnya *computer vision*, telah membuka peluang besar untuk mengembangkan sistem deteksi aktivitas manusia berbasis video secara otomatis [4]. Teknologi ini memungkinkan sistem untuk mengenali pola gerakan manusia tanpa memerlukan interaksi langsung, sehingga aktivitas seperti berjalan, duduk, maupun jatuh dapat dianalisis secara *real-time* melalui rekaman kamera. Salah satu algoritma yang banyak digunakan dalam sistem berbasis video adalah YOLO (*You Only Look Once*). Kemampuan YOLO dalam mendeteksi objek secara *real-time* dengan latensi rendah menjadikannya komponen yang sesuai untuk diintegrasikan ke dalam sistem monitoring. YOLOv8 yang dikembangkan oleh Jocher et al. menunjukkan peningkatan performa yang signifikan dibandingkan versi-versi sebelumnya sehingga dapat diandalkan sebagai komponen deteksi utama [5].

Sejumlah penelitian telah mengeksplorasi penerapan YOLO sebagai algoritma inti deteksi jatuh. Hwang et al. merancang FD-YOLO, varian YOLOv5 yang direkayasa ulang untuk mencapai keseimbangan antara kecepatan inferensi dan akurasi deteksi; akan tetapi, penelitian tersebut tidak mencakup pengembangan antarmuka pengguna maupun lapisan monitoring [6]. Pada pendekatan berbeda, Tîrziu et al. menggabungkan YOLOv7-W6-Pose dengan analisis *pose estimation* dan melaporkan tingkat akurasi mencapai 95%, namun sistem yang dihasilkan tidak dilengkapi mekanisme notifikasi maupun *dashboard* yang dapat dioperasikan pengguna akhir [7]. Dalam konteks Indonesia, Syarif et al. membangun sistem pemantauan berbasis YOLOv7 dengan antarmuka yang relevan secara lokal, meskipun integrasi *dashboard* interaktif dan notifikasi otomatis belum tersedia [8]. Luo mengevaluasi berbagai arsitektur YOLO untuk skenario *smart home* dan memvalidasi efektivitas pendekatan berbasis deteksi objek, namun tanpa mekanisme pemberitahuan kepada penghuni atau pengasuh [9]. Wang et al. mengusulkan LFD-YOLO sebagai model yang efisien secara komputasi untuk perangkat dengan sumber daya terbatas, tetapi aspek integrasi dengan platform monitoring tidak dibahas [10]. Junito dan Budilaksono merancang sistem deteksi jatuh berbasis CNN khusus untuk lingkungan hunian di Indonesia, namun belum mencakup komponen *dashboard* berbasis web maupun notifikasi terhubung [11]. Ye turut membuktikan potensi kombinasi YOLO dan *pose estimation* dalam mendeteksi jatuh pada lansia, meskipun cakupan evaluasi masih dibatasi pada performa model tanpa validasi sistem secara menyeluruh [13].

Dari kajian tersebut, teridentifikasi tiga kesenjangan utama. Pertama, sebagian besar penelitian berfokus pada pengembangan model deteksi tanpa mengintegrasikannya dengan platform monitoring yang dapat digunakan langsung oleh pengguna non-teknis. Kedua, belum adanya mekanisme notifikasi otomatis yang terintegrasi langsung dengan sistem deteksi sehingga respons petugas terhadap kondisi darurat masih bergantung pada pemantauan manual. Ketiga, minimnya penelitian yang mengembangkan dan menguji sistem deteksi jatuh secara khusus di lingkungan panti werda Indonesia. Perbandingan posisi penelitian ini terhadap penelitian terdahulu disajikan pada Tabel 1.

Tabel 1. Perbandingan dengan Penelitian Terdahulu

Penelitian	Algoritma	Dashboard Web	Notifikasi Otomatis	Evaluasi Model
Hwang et al. (2025) [6]	FD-YOLO (YOLOv5)	-	-	✓
Tîrziu et al. (2024) [7]	YOLOv7-W6-Pose	-	-	✓ (95%)
Syarif et al. (2024) [8]	YOLOv7	✓	-	-
Luo (2023) [9]	Multi-YOLO	-	-	✓
Wang et al. (2025) [10]	LFD-YOLO	-	-	✓
Junito & Budilaksono (2025) [11]	CNN	-	-	✓
Ye (2024) [13]	YOLO + Pose	-	-	✓
Penelitian ini	YOLOv8	✓	✓ (Telegram)	✓

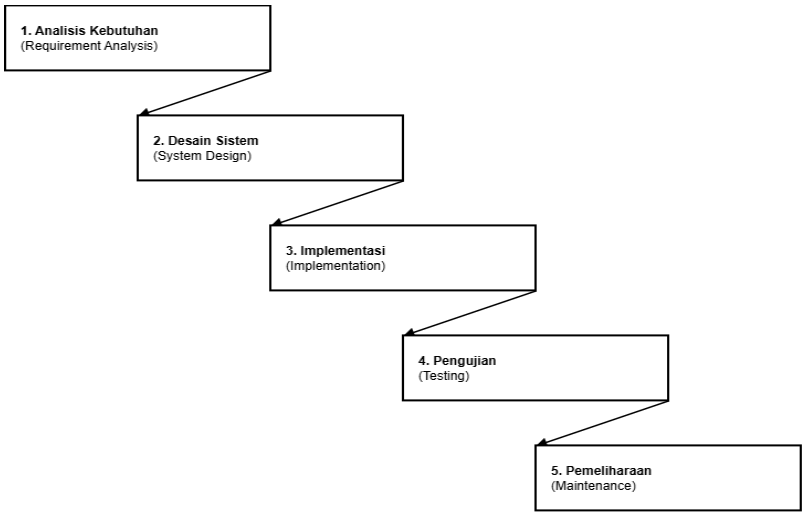
Berdasarkan perbandingan pada Tabel 1, terlihat bahwa penelitian ini merupakan satu-satunya yang mengintegrasikan ketiga komponen evaluasi model, *dashboard* monitoring berbasis web, dan notifikasi otomatis secara bersamaan dalam satu platform yang divalidasi di fasilitas perawatan lansia nyata. Berdasarkan kesenjangan tersebut, penelitian ini

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

membangun sistem deteksi jatuh lansia *real-time* secara menyeluruh menggunakan YOLOv8 yang diintegrasikan dengan *dashboard* web, *live streaming* kamera, dan notifikasi Telegram, serta diuji langsung di Panti Werda Kediri dengan tujuan untuk mengetahui efektivitasnya dalam mendeteksi aktivitas normal dan kejadian jatuh pada lansia secara *real-time*.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode pengembangan sistem *Waterfall* yang bersifat sekuensial, di mana setiap tahap harus diselesaikan secara tuntas sebelum tahap berikutnya dimulai. Metode ini dipilih karena kebutuhan sistem dapat didefinisikan secara jelas sejak awal berdasarkan hasil observasi di Panti Werda Kediri, sehingga pengembangan dapat berjalan terarah dan terdokumentasi dengan baik. Adapun alur penelitian yang diterapkan tersaji pada Gambar 1



Gambar 1. Tahapan Pengembangan Sistem *Waterfall*

2.1 Studi Literatur

Tahap studi literatur dilakukan untuk memperoleh landasan teoritis yang diperlukan dalam merancang dan mengimplementasikan sistem. Kajian mencakup algoritma deteksi objek berbasis *deep learning*, khususnya keluarga YOLO (*You* Only *Look Once*), serta metode pelatihan dan evaluasi model kecerdasan buatan. YOLOv8 yang dikembangkan oleh Jocher et al. [5] dipilih sebagai komponen deteksi utama karena arsitekturnya yang lebih ringan dan akurat dibandingkan versi pendahulunya, serta dukungan ekosistem pelatihan yang komprehensif melalui pustaka Ultralytics. Selain kajian algoritma deteksi, studi literatur juga mencakup pengembangan sistem *backend* berbasis FastAPI [15], teknologi *Server-Sent Events* (SSE) untuk pembaruan status *real-time*, serta integrasi notifikasi melalui Telegram Bot [14]. Kajian terhadap penelitian terdahulu seperti Hwang et al. [6], Tîrziu et al. [7], dan Syarif et al. [8] dilakukan untuk mengidentifikasi kesenjangan yang akan dijawab oleh penelitian ini, sebagaimana telah diuraikan pada bagian pendahuluan.

2.2 Analisis Kebutuhan

Pengumpulan data dilakukan melalui observasi langsung di Panti Werda Kediri dan studi literatur. Selain observasi, wawancara dengan kepala panti dilakukan untuk memperoleh gambaran alur kerja petugas sehari-hari dan mengidentifikasi titik-titik lemah dalam sistem pemantauan yang berjalan. Dari hasil tersebut ditemukan bahwa petugas mengalami keterbatasan dalam memantau seluruh area panti secara bersamaan, terutama pada malam hari dan saat jumlah staf terbatas. Kondisi ini meningkatkan risiko keterlambatan penanganan apabila terjadi insiden jatuh. Berdasarkan analisis tersebut, ditetapkan tujuh kebutuhan fungsional sistem sebagaimana ditampilkan pada Tabel 2.

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

Tabel 2. Kebutuhan Fungsional Sistem

No	Kebutuhan Fungsional
KF-01	Autentikasi pengguna dengan dua peran (admin dan petugas)
KF-02	Manajemen konfigurasi kamera (tambah, edit, hapus)
KF-03	Deteksi otomatis menggunakan YOLOv8 pada video maupun kamera *live*
KF-04	Pencatatan kondisi normal dan kondisi jatuh
KF-05	Notifikasi Telegram otomatis saat kejadian jatuh terkonfirmasi
KF-06	*Dashboard* web dengan riwayat deteksi dan status pekerjaan
KF-07	*Live streaming* MJPEG dengan pembaruan status *real-time* via SSE

2.3 Desain Sistem

Tahap desain menghasilkan cetak biru teknis berupa *Data Flow Diagram* (DFD) untuk menggambarkan aliran data antar komponen, dan *Entity Relationship Diagram* (ERD) untuk menggambarkan struktur basis data. DFD Level 1 dirancang dengan menguraikan lima proses utama yang saling berkaitan: Autentikasi Pengguna, Manajemen Kamera, Deteksi Aktivitas, Pencatatan dan Notifikasi, serta Pemantauan dan Pelaporan. Tiga entitas eksternal yang berinteraksi dengan sistem adalah Petugas sebagai pengguna utama, Kamera sebagai sumber data visual, dan Telegram sebagai penerima notifikasi otomatis. ERD dirancang untuk menggambarkan relasi antar empat entitas utama, yaitu *users*, *cameras*, *detection_jobs*, dan *detection_logs*, yang dihubungkan melalui *foreign key* untuk menjaga integritas data. Arsitektur sistem dirancang secara berlapis mengikuti pola *router* → *service* → *worker* → *model* agar tanggung jawab setiap komponen terpisah secara jelas dan pengujian dapat dilakukan per lapisan. Desain *background job* berbasis *daemon thread* dipilih agar *endpoint* HTTP tetap responsif saat proses deteksi berlangsung tanpa memblokir antarmuka pengguna. Hasil rancangan DFD dan ERD ditampilkan lebih lanjut pada bagian Hasil dan Pembahasan.

2.4 Dataset dan Pelatihan Model

Dataset yang digunakan untuk melatih model adalah UR Fall (*University of Rzeszow Fall Detection Dataset*) yang diperoleh dari Roboflow Universe [19]. Dataset ini terdiri dari 2.600 gambar yang mencakup dua kelas aktivitas *fall* (kondisi jatuh) dan *normal* (kondisi berdiri, duduk, dan berjalan) dengan pembagian 2.000 gambar data latih (*training*), 400 gambar data validasi (*validation*), dan 200 gambar data uji (*testing*). Pemilihan dataset ini didasarkan pada keseimbangan jumlah kelas dan ketersediaan anotasi kotak pembatas (*bounding box*) yang sesuai dengan format YOLOv8.

Pelatihan dilakukan menggunakan arsitektur YOLOv8n (*nano*) dengan bobot pra-latih (*pretrained*) yang kemudian dilatih ulang (*fine-tuned*) selama 40 *epoch* dengan ukuran *batch* 8 dan ukuran gambar 640×640 piksel. Teknik augmentasi data RandAugment [20] diterapkan selama pelatihan untuk meningkatkan variasi data latih dan mengurangi risiko *overfitting*. Evaluasi kinerja model dilakukan menggunakan metrik standar deteksi objek, yaitu *precision*, *recall*, mAP@0.5 (*mean Average Precision* pada IoU 0,5), dan mAP@0.5:0.95 (rata-rata mAP pada rentang IoU 0,5–0,95). *Epoch* terbaik ditetapkan berdasarkan nilai mAP@0.5 tertinggi pada data validasi.

2.5 Spesifikasi Sistem

Sistem dibangun menggunakan Python 3.10 dengan FastAPI [15] sebagai *framework backend* dan PostgreSQL sebagai basis data relasional. Model deteksi menggunakan YOLOv8 dari pustaka Ultralytics [5] yang dilatih ulang untuk mengenali dua kelas aktivitas. Setiap *frame* video di-*resize* ke resolusi 640×360 piksel sebelum inferensi. Keputusan label menggunakan *threshold* per kelas 0,35 untuk *fall* dan 0,30 untuk *normal*. Kejadian jatuh dikonfirmasi setelah 3 *frame* berturut-turut (*fall confirmation*) dengan *cooldown* 1,5 detik untuk mencegah pencatatan log ganda. Notifikasi dikirim secara otomatis melalui Telegram Bot [14] setiap kali kejadian jatuh terkonfirmasi. Spesifikasi pustaka utama yang digunakan disajikan pada Tabel 3.

Tabel 3. Pustaka Utama yang Digunakan

Pustaka	Fungsi
FastAPI	*Framework* HTTP API dan *rendering* template web
Ultralytics	Pustaka YOLOv8 untuk *inferensi* model deteksi
OpenCV	Pemrosesan *frame* video (*resize*, *encode* JPEG)
SQLAlchemy	Pencatatan kondisi normal dan kondisi jatuh
python-telegram-bot	Pengiriman notifikasi ke Telegram Bot

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

2.6 Pengujian

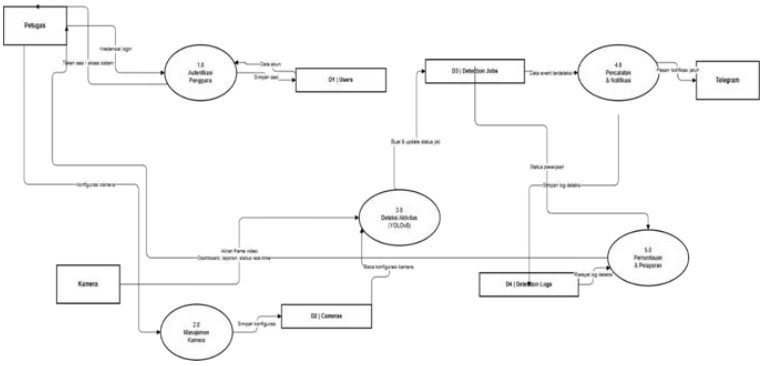
Pengujian menggunakan metode *blackbox* untuk memverifikasi fungsionalitas sistem dari sudut pandang pengguna tanpa memperhatikan implementasi internal. Pengujian dilaksanakan langsung di Panti Werda Kediri menggunakan perangkat peneliti pada kondisi operasional nyata. Perangkat yang digunakan adalah laptop peneliti yang terhubung dengan kamera USB sebagai sumber video *live*, serta beberapa *file* video rekaman uji sebagai masukan untuk skenario deteksi video. Sepuluh skenario dirancang mencakup seluruh fungsi utama sistem, mulai dari autentikasi, manajemen kamera, deteksi video, deteksi kamera *live*, notifikasi, log, penghentian *job*, hingga *logout*. Setiap skenario dijalankan minimal dua kali untuk memverifikasi konsistensi *output* sistem. Kriteria keberhasilan adalah seluruh skenario menghasilkan *output* sesuai ekspektasi tanpa *error* pada antarmuka, serta notifikasi Telegram diterima dalam waktu yang wajar setelah kejadian jatuh terkonfirmasi.

3. HASIL DAN PEMBAHASAN

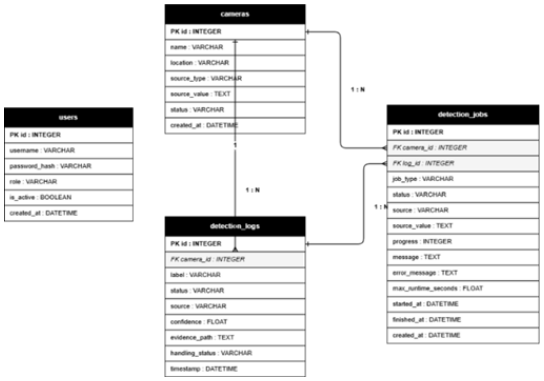
3.1 Perancangan Sistem

Perancangan sistem menggunakan DFD dan ERD sebagaimana ditampilkan pada Gambar 2 dan Gambar 3. DFD Level 1 menguraikan lima proses utama yang saling berkaitan: Autentikasi Pengguna, Manajemen Kamera, Deteksi Aktivitas, Pencatatan dan Notifikasi, serta Pemantauan dan Pelaporan. Tiga entitas eksternal yang berinteraksi dengan sistem adalah Petugas sebagai pengguna utama, Kamera sebagai sumber data visual, dan Telegram sebagai penerima notifikasi otomatis. Setiap proses utama terhubung dengan penyimpanan data yang terdiri dari empat tabel, yaitu *users*, *cameras*, *detection_jobs*, dan *detection_logs*.

ERD menggambarkan relasi antar keempat tabel tersebut melalui *foreign key*. Tabel *users* berfungsi sebagai titik kontrol akses seluruh pengguna wajib terautentikasi sebelum menggunakan fitur sistem. Tabel *detection_jobs* menyimpan antrean dan status pekerjaan dengan siklus *queued* → *running* → *done* / *failed* / *cancelled*, sedangkan tabel *detection_logs* menyimpan setiap *event* deteksi beserta *path* bukti visual yang direkam. Relasi antara *detection_jobs* dan *detection_logs* dirancang satu-ke-banyak sehingga satu pekerjaan deteksi dapat menghasilkan banyak *event* log sepanjang durasi pekerjaan berlangsung. Selain itu, perancangan mencakup alur *background job* sebagai komponen inti sistem yang bekerja secara asinkron melalui *daemon thread* tunggal. Status *job* dipantau melalui *Server-Sent Events* (SSE) agar halaman web diperbarui secara otomatis tanpa teknik *polling* yang boros sumber daya



Gambar 2. DFD Level 1 Sistem Monitoring Deteksi Jatuh Lansia



Gambar 3. Entity Relationship Diagram Basis Data Sistem

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

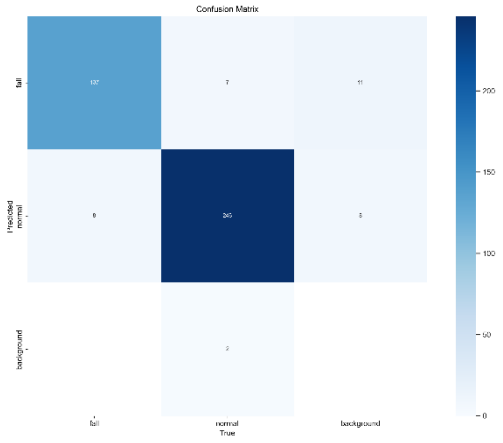
3.2 Evaluasi Model

Model YOLOv8n yang dilatih pada dataset UR Fall [19] mencapai kinerja terbaik pada *epoch* ke-34 dari 40 *epoch* yang dijalankan. Hasil evaluasi model pada data uji disajikan pada Tabel 4.

Tabel 4. Hasil Evaluasi Kinerja Model YOLOv8n

Metrik	Nilai
Precision	93,9%
Recall	97,9%
mAP@0.5	99,1%
mAP@0.5:0.95	81,8%

Nilai *precision* 93,9% menunjukkan bahwa dari seluruh prediksi *fall* yang dihasilkan model, 93,9% di antaranya benar-benar merupakan kondisi jatuh. Nilai *recall* 97,9% menunjukkan bahwa model berhasil mendeteksi 97,9% dari seluruh kejadian jatuh yang sebenarnya ada pada data uji, sehingga tingkat kejadian jatuh yang terlewat (*false negative*) sangat rendah. Nilai mAP@0.5 sebesar 99,1% menunjukkan bahwa pada ambang IoU 0,5, model mencapai rata-rata presisi yang sangat tinggi untuk kedua kelas secara bersamaan. Tingginya nilai *recall* pada kelas *fall* menjadi prioritas utama dalam konteks deteksi jatuh lansia, karena kegagalan mendeteksi kejadian jatuh (*false negative*) memiliki dampak yang lebih serius dibandingkan *false positive* yang dapat ditekan oleh mekanisme *fall confirmation* pada level sistem. *Confusion matrix* hasil evaluasi sebagaimana ditunjukkan pada Gambar 4 memperlihatkan bahwa model mampu memisahkan kelas *fall* dan *normal* dengan baik, dengan jumlah *false negative* yang minimal

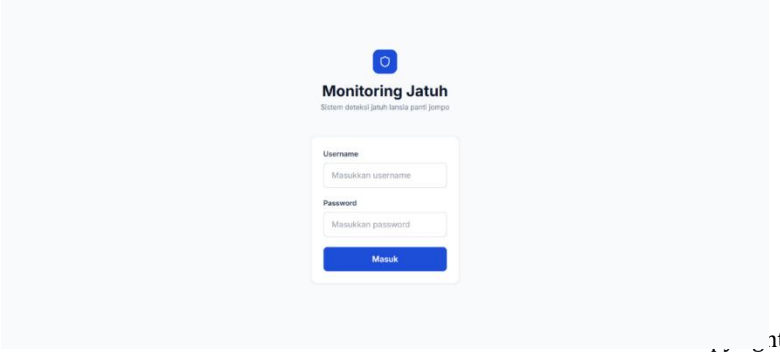


Gambar 2. Confusion Matrix Evaluasi Model YOLOv8n pada Data Uji

3.3 Implementasi

Sistem berhasil diimplementasikan sebagai aplikasi web berbasis FastAPI yang dapat diakses melalui *browser* tanpa instalasi tambahan di sisi pengguna. Seluruh tujuh kebutuhan fungsional yang ditetapkan pada tahap analisis berhasil diimplementasikan dan berjalan sesuai rancangan.

1. Halaman Login

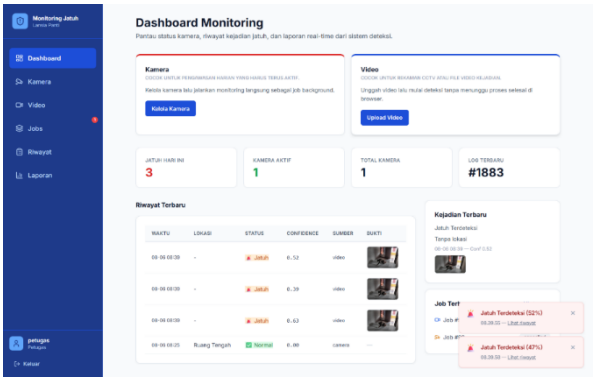


Gambar 3. Tampilan Halaman *Login* Sistem

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

Halaman *login* merupakan pintu masuk utama sistem yang hanya dapat diakses oleh pengguna terdaftar. Sistem menyediakan dua peran pengguna, yaitu admin dan petugas, yang masing-masing memiliki hak akses berbeda terhadap fitur manajemen sistem. Autentikasi menggunakan mekanisme *session cookie* sehingga pengguna tetap login selama sesi aktif tanpa perlu memasukkan kredensial berulang kali. Kata sandi disimpan dalam bentuk hash sehingga tidak tersimpan dalam teks biasa di basis data, menjaga keamanan akun pengguna dari potensi kebocoran data. Seluruh halaman web dilindungi dari akses tanpa autentikasi percobaan mengakses halaman manapun tanpa sesi aktif akan dialihkan secara otomatis ke halaman login.

2. Halaman *Dashboard*

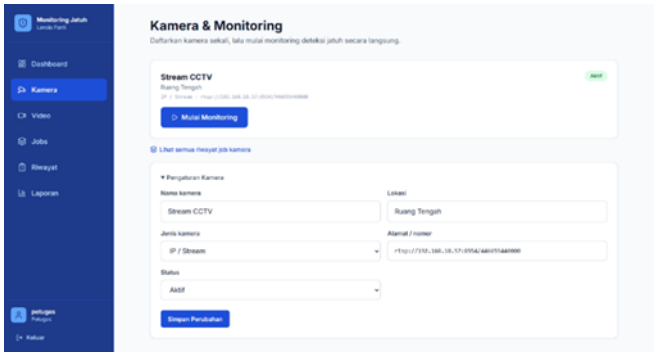


Gambar 4. Halaman *Dashboard* Sistem

Halaman *dashboard* menampilkan ringkasan kondisi sistem secara menyeluruh, mencakup kartu statistik total kejadian jatuh, jumlah log hari ini, dan status kamera aktif. *Sidebar* menampilkan lima pekerjaan deteksi terbaru beserta statusnya untuk memudahkan petugas memantau aktivitas sistem tanpa harus berpindah halaman. Kartu statistik diperbarui setiap kali petugas memuat ulang halaman sehingga angka yang ditampilkan selalu mencerminkan kondisi terkini sistem. Notifikasi *real-time* berupa *toast* berwarna merah muncul di pojok kanan bawah layar selama 10 detik setiap kali kejadian jatuh terkonfirmasi melalui mekanisme *Server-Sent Events* (SSE). Mekanisme ini memungkinkan notifikasi dikirim dari *background worker* ke seluruh tab *browser* yang aktif secara bersamaan tanpa teknik *polling* yang boros sumber daya.

3. Halaman Manajemen Kamera

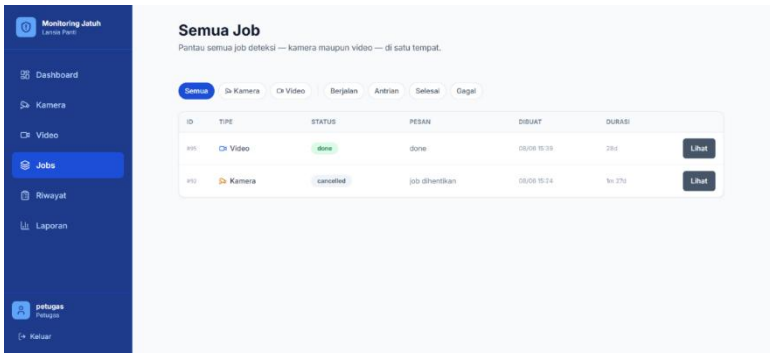
Halaman manajemen kamera memungkinkan petugas menambahkan konfigurasi kamera berupa nama, lokasi, tipe sumber (*stream* URL atau USB), dan nilai sumber perangkat. Sistem mendukung dua tipe kamera, yaitu kamera USB yang menggunakan indeks perangkat *integer* dan kamera *network stream* yang menggunakan URL RTSP. Sistem membatasi hanya satu kamera aktif sesuai batasan desain yang ditetapkan apabila petugas mencoba menambahkan kamera kedua, sistem mengembalikan HTTP 400 dengan pesan kesalahan secara otomatis. Pembatasan ini diterapkan pada lapisan *service* sehingga penolakan terjadi sebelum data disimpan ke basis data, menjamin konsistensi konfigurasi sistem. Halaman ini juga menyediakan tautan langsung untuk memulai pekerjaan deteksi dari kamera yang terdaftar.



Gambar 5. Tampilan Halaman Manajemen Kamera

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

4. Halaman Daftar Pekerjaan

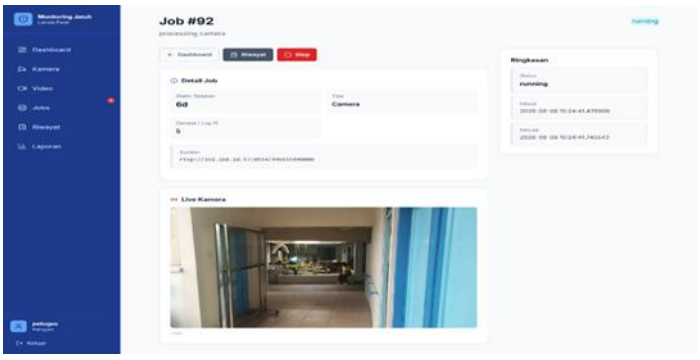


Gambar 6. Tampilan Halaman Daftar Pekerjaan Deteksi

Halaman daftar pekerjaan menampilkan seluruh pekerjaan deteksi dari kamera maupun *file* video dalam satu tampilan terpusat yang dapat disaring berdasarkan tipe dan status. Sistem membedakan dua tipe pekerjaan: pekerjaan bertipe *video* yang memiliki total *frame* dan persentase kemajuan, serta pekerjaan bertipe *camera* yang berjalan tanpa batas waktu tetap dan menampilkan *elapsed time* sebagai gantinya. *Badge* berwarna merah di *navbar* menampilkan jumlah pekerjaan aktif secara *real-time* dengan pembaruan setiap 10 detik sehingga petugas selalu mengetahui kondisi sistem tanpa harus membuka halaman ini terlebih dahulu. Halaman ini juga menampilkan informasi durasi pekerjaan yang telah selesai sehingga petugas dapat mengevaluasi lama proses deteksi pada setiap sesi pemantauan.

5. Halaman Detail Pekerjaan

Halaman detail pekerjaan menampilkan informasi lengkap pekerjaan yang sedang berjalan, termasuk *live stream* MJPEG dari kamera dan *elapsed time* yang diperbarui setiap detik. Untuk pekerjaan bertipe video, halaman ini menampilkan persentase *progress* pemrosesan *frame* secara *real-time* sehingga petugas dapat memperkirakan kapan pekerjaan akan selesai. Petugas dapat menghentikan pekerjaan yang sedang berjalan melalui tombol *stop* yang mengubah status menjadi *cancel_requested* dan *worker* akan melakukan penghentian secara *graceful* sebelum mengubah status menjadi *cancelled*. Saat pekerjaan berstatus *done*, *failed*, atau *cancelled*, *live stream* berhenti secara otomatis melalui sinyal SSE sehingga tidak terjadi konsumsi sumber daya yang tidak perlu.

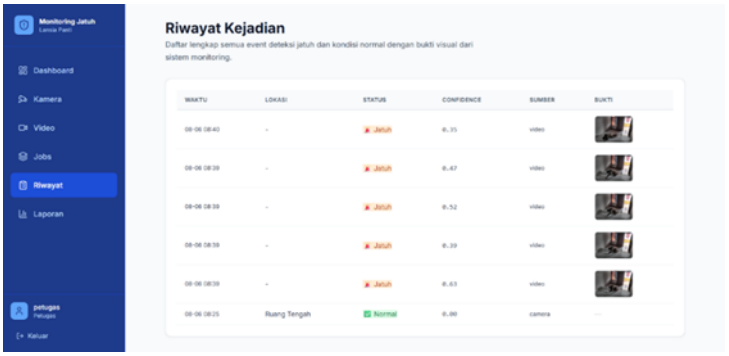


Gambar 7. Tampilan Halaman Detail Pekerjaan Deteksi

6. Halaman Riwayat Log

Sistem mengirimkan notifikasi otomatis ke Telegram Bot beserta gambar bukti visual *frame* yang direkam setiap kali kejadian jatuh terkonfirmasi [14]. Pesan notifikasi menyertakan informasi waktu kejadian sehingga petugas dapat menilai situasi secara langsung tanpa harus membuka *dashboard* terlebih dahulu. Notifikasi dikirim dalam waktu kurang dari 2 detik melalui mekanisme *event callback* yang dipanggil langsung oleh *worker* tanpa menunggu keseluruhan pekerjaan selesai [16].

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS



Gambar 10. Tampilan Halaman Riwayat Log Deteksi

7. Notifikasi Telegram

Sistem mengirimkan notifikasi otomatis ke Telegram Bot beserta gambar bukti visual *frame* yang direkam setiap kali kejadian jatuh terkonfirmasi [14]. Pesan notifikasi menyertakan informasi waktu kejadian dan gambar *frame* yang direkam sehingga petugas dapat menilai situasi secara langsung tanpa harus membuka *dashboard* terlebih dahulu. Notifikasi dikirim secara *real-time* melalui mekanisme *event callback* yang dipanggil langsung oleh *worker* tanpa menunggu keseluruhan pekerjaan selesai, sehingga petugas mendapatkan peringatan segera setelah kejadian terdeteksi. Penggunaan Telegram sebagai kanal notifikasi dipilih karena platform tersebut sudah sangat familiar bagi petugas panti dan dapat diakses melalui perangkat *mobile* kapan saja dan di mana saja tanpa memerlukan instalasi aplikasi tambahan [16].Pengujian

Pengujian *blackbox* dilakukan langsung di Panti Werda Kediri pada kondisi operasional nyata menggunakan perangkat peneliti. Pengujian dilaksanakan dalam dua sesi pada hari yang berbeda sesi pertama berfokus pada fungsi autentikasi, manajemen kamera, dan antarmuka *dashboard*, sedangkan sesi kedua menguji skenario deteksi video, deteksi kamera *live*, notifikasi, serta log dan penghentian pekerjaan. Sepuluh skenario dijalankan secara berurutan dan setiap skenario dinilai lulus (✓) apabila *output* sesuai ekspektasi. Hasil pengujian disajikan pada Tabel 5.

Gambar 11. Notifikasi Telegram Saat Kejadian Jatuh Terkonfirmasi

Tabel 5. Hasil Pengujian Blackbox

No	Fungsi yang Diuji	Skenario	Ekspektasi	Status
1	*Login*	Kredensial valid	*Redirect* ke *dashboard*, sesi aktif	✓
2	*Login*	Kredensial tidak valid	Pesan kesalahan, tetap di halaman	✓
3	Manajemen Kamera	Tambah kamera baru	Kamera tersimpan, muncul di daftar	✓
4	Manajemen Kamera	Tambah kamera kedua	HTTP 400, pesan kamera sudah ada	✓
5	Deteksi Video	*Upload* video .mp4	*Job* terbuat, *queued* → *running* → *done*	✓
6	Deteksi Kamera	Mulai deteksi *live*	*Live stream* MJPEG tampil, *elapsed time* berjalan	✓
7	Notifikasi Jatuh	Video berisi aktivitas jatuh	Notifikasi Telegram terkirim dengan bukti visual	✓
8	Log Deteksi	Lihat riwayat log	Daftar log tampil dengan label, *confidence*, waktu	✓

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

9	*Stop Job*	Klik tombol *stop*	Status berubah menjadi *cancelled*	✓
10	*Logout*	Klik *logout*	Sesi dihapus, *redirect* ke halaman *login*	✓

Seluruh 10 skenario dinyatakan lulus dengan status valid (✓). Pada skenario 7, notifikasi Telegram diterima dalam waktu kurang dari 2 detik setelah kejadian jatuh terkonfirmasi oleh sistem, membuktikan bahwa mekanisme *event callback* berjalan secara *real-time* sesuai kebutuhan. Catatan tambahan ditemukan pada skenario 6 dan 7, yaitu terdapat *false positive* pada beberapa kasus di mana orang berdiri terdeteksi sebagai kondisi jatuh, khususnya ketika *bounding box* subjek memiliki rasio aspek horizontal yang menyerupai posisi berbaring. Meskipun demikian, mekanisme *fall confirmation* 3 *frame* berturut-turut berhasil menekan sebagian besar *false positive* sesaat yang tidak konsisten antar *frame*.

3.4 Pembahasan

Hasil pengujian *blackbox* menunjukkan bahwa seluruh kebutuhan fungsional sistem berjalan sesuai yang ditetapkan. Mekanisme *fall confirmation* berbasis 3 *frame* berturut-turut terbukti efektif dalam menekan *false positive* dari deteksi sesaat. Cooldown 1,5 detik juga berfungsi mencegah pencatatan log ganda pada satu kejadian yang sama. Kondisi normal yang dicatat secara periodik setiap 60 detik membuktikan bahwa sistem tidak hanya reaktif terhadap kejadian jatuh, tetapi juga aktif memantau secara berkelanjutan sehingga memberikan jaminan operasional kepada petugas.

Dari sisi evaluasi model, YOLOv8n yang dilatih pada dataset UR Fall mencapai *recall* 97,9% pada kelas *fall*, yang merupakan metrik prioritas dalam konteks ini karena kegagalan mendeteksi kejadian jatuh (*false negative*) memiliki konsekuensi keselamatan yang lebih serius dibandingkan *false positive*. Nilai mAP@0.5 sebesar 99,1% menunjukkan kemampuan model yang sangat baik dalam membedakan kedua kelas secara keseluruhan. Dibandingkan dengan Tîrziu et al. yang melaporkan akurasi 95% menggunakan YOLOv7-W6-Pose [7], model YOLOv8n pada penelitian ini mencapai *recall* dan mAP yang lebih tinggi dengan arsitektur yang lebih ringan (*nano*), menjadikannya pilihan yang efisien untuk sistem monitoring berbasis perangkat keras terbatas.

Dari sisi arsitektur, penggunaan *background job* berbasis *daemon thread* memberikan keunggulan responsivitas yang signifikan. Sistem mengembalikan *job_id* segera setelah permintaan diterima dan memproses deteksi di *background*, sehingga antarmuka pengguna tidak pernah mengalami *timeout* meskipun video yang diproses berdurasi panjang. Pola ini relevan untuk sistem monitoring berbasis IoT yang memerlukan pengolahan data berat secara berkelanjutan tanpa mengorbankan pengalaman pengguna [17].

Dibandingkan dengan penelitian terdahulu, sistem ini memiliki keunggulan dalam hal integrasi komponen. Hwang et al. dan Wang et al. berhasil membangun model deteksi jatuh berperforma tinggi, namun keduanya tidak menyediakan antarmuka pengguna maupun notifikasi terintegrasi [6], [10]. Tîrziu et al. mencapai akurasi 95% menggunakan *pose estimation*, namun tanpa *dashboard* interaktif yang dapat digunakan petugas non-teknis [7]. Syarif et al. menyediakan antarmuka sederhana yang relevan dengan konteks Indonesia, namun tanpa *background job system* dan notifikasi otomatis [8]. Sistem pada penelitian ini mengintegrasikan seluruh komponen evaluasi model YOLOv8 dengan *recall* 97,9%, *background job* asinkron, *dashboard* web, *live streaming* MJPEG, dan notifikasi Telegram dalam satu platform yang dapat dioperasikan langsung oleh petugas non-teknis di lingkungan panti werda Indonesia [12].

Dari perspektif kontribusi lokal, penelitian ini merupakan salah satu implementasi *end-to-end* sistem deteksi jatuh yang divalidasi langsung di fasilitas perawatan lansia di Indonesia. Sebagian besar penelitian terdahulu melakukan evaluasi hanya pada *dataset benchmark* tanpa validasi lapangan di fasilitas perawatan nyata [18]. Pengujian di Panti Werda Kediri memberikan gambaran kondisi operasional nyata, termasuk variasi pencahayaan, ukuran subjek yang beragam, dan kebutuhan respons petugas yang sesungguhnya. Pencatatan kondisi normal secara periodik setiap 60 detik juga menjadi pembeda signifikan, memberikan bukti audit bahwa sistem sedang aktif beroperasi. Keterbatasan sistem mencakup adanya *false positive* pada kondisi posisi tubuh tertentu yang menyerupai posisi berbaring, *threshold* deteksi yang bersifat global untuk semua kamera, serta sistem yang hanya mendukung satu kamera aktif sesuai batasan desain yang ditetapkan.

4. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang dilaksanakan di Panti Werda Kediri, sistem deteksi jatuh lansia *real-time* berbasis YOLOv8 dinyatakan efektif sebagai alat bantu pemantauan aktivitas lansia. Model YOLOv8n yang dilatih pada dataset UR Fall dengan 2.600 gambar mencapai *precision* 93,9%, *recall* 97,9%, mAP@0.5 99,1%, dan

Management of Information System Journal
Vol 4, No 3, Juli 2026, Hal. 203 - 213
ISSN: 2964-9455 (media online)
DOI: 10.47065/mis.v2i3.2885
https://journal.fkpt.org/index.php/MIS

mAP@0.5:0.95 81,8% pada *epoch* terbaik ke-34. Hasil pengujian *blackbox* terhadap 10 skenario di Panti Werda Kediri menunjukkan seluruh fungsi sistem berjalan valid sesuai kebutuhan fungsional yang ditetapkan. Sistem berhasil mendeteksi kejadian jatuh secara otomatis menggunakan mekanisme *fall confirmation* 3 *frame* berturut-turut dengan *cooldown* 1,5 detik, serta mengirimkan notifikasi Telegram beserta bukti visual dalam waktu kurang dari 2 detik setiap kali kejadian terkonfirmasi. *Dashboard* monitoring berbasis web berhasil dioperasikan petugas tanpa latar belakang teknis, menjadikan pemantauan kondisi lansia lebih efisien dibandingkan metode manual sebelumnya. Keterbatasan yang teridentifikasi meliputi *false positive* pada posisi tubuh horizontal yang menyerupai berbaring dan dukungan satu kamera aktif. Untuk pengembangan selanjutnya, disarankan pelatihan ulang model menggunakan dataset lokal Panti Werda Kediri, pengaktifan filter gerak berbasis *pose estimation* yang telah tersedia pada sistem, serta penambahan dukungan multi-kamera agar seluruh area panti dapat dipantau secara komprehensif.

REFERENCES

[1] World Health Organization, "Falls," WHO Fact Sheet, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls
[2] Badan Pusat Statistik, Statistik Penduduk Lanjut Usia 2023. Jakarta: BPS Republik Indonesia, 2023.
[3] Kementerian Kesehatan Republik Indonesia, Profil Kesehatan Indonesia 2024. Jakarta: Kemenkes RI, 2024.
[4] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th ed. Hoboken: Pearson Education, 2021.
[5] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO (Version 8.0.0)," Computer Software, Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[6] H. Hwang, D. Kim, and H. Kim, "FD-YOLO: A YOLO network optimized for fall detection," Applied Sciences, vol. 15, no. 1, p. 453, 2025. https://doi.org/10.3390/app15010453
[7] E. Tirziu, A. M. Vasilevschi, A. Alexandru, and E. Tudora, "Enhanced fall detection using YOLOv7-W6-Pose for real-time elderly monitoring," Future Internet, vol. 16, no. 12, p. 472, 2024. https://doi.org/10.3390/fi16120472
[8] M. Syarif, C. Setianingsih, and A. Novianty, "Pengembangan sistem monitoring perawatan lanjut usia menggunakan webcam dan algoritma YOLOv7," e-Proceeding of Engineering, Telkom University, 2024.
[9] B. Luo, "Human fall detection for smart home caring using YOLO networks," International Journal of Advanced Computer Science and Applications, vol. 14, no. 4, 2023. http://dx.doi.org/10.14569/IJACSA.2023.0140409
[10] H. Wang, S. Xu, Y. Chen, and C. Su, "LFD-YOLO: A lightweight fall detection network with enhanced feature extraction and fusion," Scientific Reports, vol. 15, no. 1, p. 5069, 2025. https://doi.org/10.1038/s41598-025-89214-7
[11] A. Junito and S. Budilaksono, "Rancang bangun sistem deteksi jatuh pada penghuni rumah berbasis convolutional neural network," Tekinfo, vol. 26, no. 1, 2025. https://doi.org/10.37817/Tekinfo.v26i1
[12] M. Y. Efendi, R. Wulanningrum, and A. B. Setiawan, "Rancang bangun sistem deteksi manusia dengan YOLO pada video CCTV," Prosiding Seminar Nasional, vol. 8, 2024.
[13] Z. Ye, "Elderly fall detection based on YOLO and pose estimation," in Proc. International Conference on Pattern Recognition and Machine Learning, INSTICC, 2024, pp. 10-17.
[14] M. I. M. Abu.Zaid, R. B. Abdullah, S. Izawana, and N. N. S. Nik Dzulkefli, "IoT-based emergency alert system integrated with Telegram Bot," in 2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2023. https://doi.org/10.1109/I2CACIS57635.2023.10193550
[15] S. Ramirez, "FastAPI (Version 0.95.0)," Computer Software, 2021. [Online]. Available: https://fastapi.tiangolo.com
[16] R. Firliana, A. S. Wardani, M. N. Muzzaki, H. Stiawan, and A. W. M. Gamas, "Implementasi manajemen proyek pada pengembangan website pemetaan biodiversitas tanaman obat di Kabupaten Kediri," Bulletin of Information Technology (BIT), vol. 3, no. 4, pp. 289-293, 2022.
[17] I. Zamarli, H. Kurniawan, and W. Darwin, "Implementasi algoritma YOLOv8 untuk mendeteksi objek manusia secara real-time menggunakan IP camera," Jurnal Penelitian Nusantara, vol. 1, pp. 432-441, 2025.
[18] R. N. Zakaria, R. Wulanningrum, and A. B. Setiawan, "Penerapan segmentasi wajah menggunakan YOLOv8 untuk presensi mata kuliah," Prosiding SEMNAS INOTEK, vol. 8, 2024.
[19] "UR Fall Detection Dataset," Roboflow Universe, ver. 1, Apr. 2024. [Online]. Available: https://universe.roboflow.com/fall-detection-w7nxl/ur-fall
[20] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2020, pp. 702–703.