

Peningkatan Akurasi Deteksi Jatuh Menggunakan Sensor Akselerometer dan Giroskop pada *Smartphone*

Muhammad Luthfi Arya Widagdo^{*1}, Muhammad Idham Ananta Timur²

Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia

e-mail: ^{*1}muhammadluthfiaw@mail.ugm.ac.id, ²idham@ugm.ac.id

Abstrak

Peningkatan akurasi deteksi jatuh pada smartphone menggunakan sensor akselerometer dan giroskop sangat penting untuk mendukung penerapan tindakan preventif. Penelitian ini mencoba mendekati masalah jatuh dengan mendesain tujuh model arsitektur berdasarkan pendekatan baru menggunakan metode varian recurrent neural network (RNN), yaitu long short-term memory (LSTM), gated recurrent unit (GRU), dan simple recurrent neural network (SimpleRNN) dan variasi dengan convolutional neural network (CNN), yaitu 1D Convolutional Neural Network (1D CNN).

Hasil validasi klasifikasi menunjukkan bahwa metode yang digunakan dalam eksperimen untuk kelas duduk, berdiri dan jatuh mendapatkan nilai yang sempurna, sedangkan kelas jatuh mendapatkan nilai yang berbeda tiap arsitektur model yang dirancang. Untuk keseluruhan kelas, performa terendah dimiliki oleh kombinasi model arsitektur 1D CNN dan SimpleRNN dengan nilai akurasi sebesar 95,6%, sedangkan performa tertinggi dimiliki model arsitektur SimpleRNN dan kombinasi model arsitektur CNN dan GRU dengan nilai akurasi mencapai 99,0%.

Kata kunci— *Peningkatan akurasi deteksi jatuh pada smartphone menggunakan sensor akselerometer dan giroskop sangat penting untuk mendukung penerapan tindakan preventif. Penelitian ini mencoba mendekati masalah jatuh dengan mendesain tujuh model arsitektur berdasarkan pendekatan baru menggunakan metode varian recurrent neural network (RNN), yaitu long short-term memory (LSTM), gated recurrent unit (GRU), dan simple recurrent neural network (SimpleRNN) dan variasi dengan convolutional neural network (CNN), yaitu 1D Convolutional Neural Network (1D CNN).*

Abstract

The aging population is a global concern, partly because as the body ages, physical conditions weaken, increasing the likelihood of falls. Falls are particularly dangerous for the elderly as they can lead to serious problems and even death. Detecting falls quickly and accurately is crucial to implement preventive measures and timely intervention when a fall occurs.

This research focuses on designing a human physical activity classification system, primarily used for fall detection. Seven model architectures are proposed using a novel approach involving the variant of recurrent neural network (RNN) methods, including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Simple Recurrent Neural Network (SimpleRNN). Additionally, variations with Convolutional Neural Network (CNN) are explored, specifically 1D Convolutional Neural Network (1D CNN).

Validation results of the classification show that the experimented methods for the classes of sitting, standing, and falling achieved perfect scores, while the falling class showed varying scores for each designed model architecture. For the overall classes, the lowest

performance is observed in the combination of 1D CNN and SimpleRNN architecture with an accuracy of 95.6%, whereas the highest performance is attributed to the SimpleRNN architecture and the combined CNN and GRU architecture with an accuracy reaching 99.0%.

Keywords— *Population aging, falling conditions, Human physical activity classification system, Recurrent Neural Network (RNN), Convolutional Neural Network (CNN)*

1. PENDAHULUAN

Penuaan populasi merupakan perhatian global dengan proyeksi jumlah lansia yang terus meningkat. Indonesia sendiri, saat ini mengalami peningkatan lansia dari 7,56 persen pada tahun 2010 menjadi 9,7 persen pada tahun 2019 dan diperkirakan pada tahun 2035 sebesar 15,77 persen. [1]. Kondisi ini memperlihatkan risiko tinggi terjadinya kejadian jatuh pada lansia, yang dapat mengakibatkan cedera serius atau bahkan kematian. Oleh karena itu, deteksi jatuh menjadi krusial sebagai langkah preventif untuk meminimalkan penanganan yang lama saat kejadian jatuh terjadi.

Pendekatan untuk deteksi jatuh dapat dibagi menjadi metode berbasis vision dan berbasis sensor. Metode berbasis vision menggunakan perangkat pencitraan seperti kamera video untuk melacak perilaku pengguna, tetapi memiliki masalah mengalami kesulitan dalam mengenali objek-objek kecil yang berdekatan dan objek-objek baru dari gambar yang memiliki konfigurasi maupun aspek rasio yang berbeda [2]. Di sisi lain, metode berbasis sensor menggunakan akselerometer dan giroskop, yang memberikan informasi akurat tentang gerakan tubuh dan orientasi [3]. Sensor dapat dikenakan sehari-hari, termasuk pada smartphone, dan dapat bekerja di berbagai lingkungan.

Pengembangan perangkat lunak menggunakan pembelajaran mesin, khususnya dengan algoritma seperti Long Short Term Memory (LSTM) dan Convolutional Neural Network (CNN), telah meningkatkan akurasi deteksi jatuh pada lansia [3][4]. Meski demikian, tantangan seperti kompleksitas algoritma dan waktu pemrosesan data masih perlu diatasi untuk memastikan respons sistem yang cepat dan akurat [5]. Penambahan data dapat menjadi solusi untuk mengatasi *overfitting* tanpa perlu menambah jumlah neuron pada model (Lisha et al., 2022).

2. METODE PENELITIAN

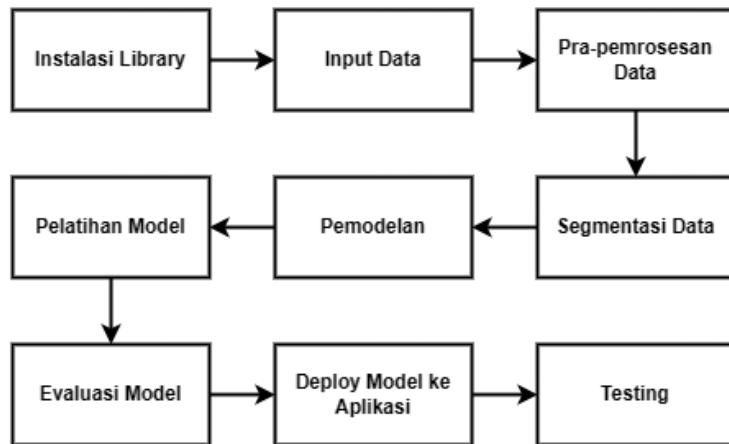
2.1 Analisis Sistem

Penelitian ini dirancang untuk mengembangkan sistem deteksi kejadian jatuh pada lansia menggunakan sensor akselerometer dan giroskop pada *smartphone*. Data aktifitas fisik manusia dilakukan pra-pemrosesan sebelum dilakukan proses pelatihan dan evaluasi. Pada proses pelatihan dilakukan dengan tujuh rancangan arsitektur model yang dibuat, yaitu menggunakan LSTM, GRU, SimpleRNN, 1D CNN, dan kombinasi 1D CNN dengan LSTM, GRU, serta SimpleRNN.

2. 1.1 Rancangan Sistem Perangkat Lunak

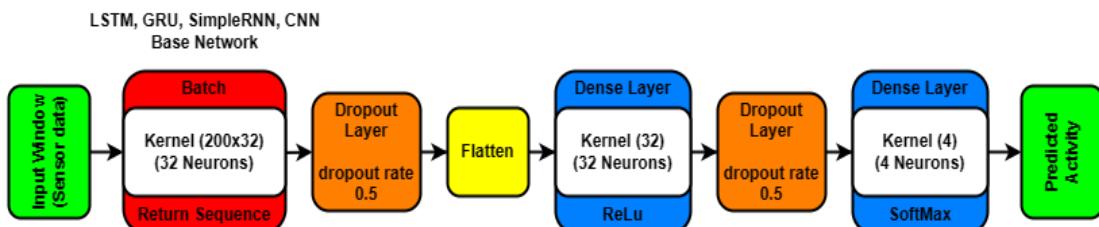
Rancangan sistem perangkat lunak deteksi jatuh menggunakan sensor *smartphone* melewati serangkaian tahapan. Instalasi *library* seperti Tensorflow dan Numpy diperlukan sebelum perangkat lunak menerima data sensor, termasuk akselerasi, rotasi, dan percepatan akselerasi. Proses pra-pemrosesan melibatkan pembersihan dan penggabungan data, sementara data sensor dipecah menjadi segmen waktu kecil untuk pemrosesan bertahap. Berbagai arsitektur model *deep learning* seperti LSTM, RNN-GRU, SimpleRNN, dan 1D CNN digunakan untuk klasifikasi aktifitas fisik manusia, dengan evaluasi menggunakan metrik

seperti akurasi dan *F1-score*. Dalam implementasi *real-case scenario*, sistem dapat mendeteksi jatuh dan memberikan notifikasi getaran selama 5 detik. Dengan langkah-langkah ini, sistem menyediakan solusi deteksi jatuh yang efisien menggunakan sensor *smartphone* dan model *deep learning*. Rancangan sistem perangkat lunak dapat dilihat pada Gambar 1.



Gambar 1 Rancangan Sistem Perangkat Lunak

2. 1.2 Topologi Model



Gambar 2 Topologi Model

Topologi model pada Gambar 2 dirancang untuk membuat rancangan arsitektur model klasifikasi aktivitas fisik manusia, khususnya kejadian jatuh, menggunakan data sensor dari *smartphone*. Dengan menerima jendela masukan yang berisi nilai akselerometer, giroskop, dan percepatan akselerasi dalam interval waktu tertentu, model ini menggunakan lapisan arsitektur seperti LSTM, GRU, SimpleRNN, dan CNN untuk memahami informasi urutan waktu dan karakteristik data. Lapisan *dropout* ditempatkan untuk mencegah *overfitting*, dan lapisan *flatten* meratakan *output* sehingga dapat dimasukkan ke dalam lapisan *dense*. Lapisan *dense* pertama bertindak sebagai penghubung antara lapisan rekurent dan *output*, menggunakan fungsi aktivasi ReLU, sedangkan lapisan *dense* terakhir memiliki fungsi softmax untuk mengklasifikasikan aktivitas fisik manusia ke dalam empat kategori: "duduk", "berdiri", "berjalan", atau "jatuh". Penggunaan regularisasi L2 membantu mengontrol kompleksitas model dan mencegah *overfitting*, sementara keluaran dari model ini adalah aktivitas yang diklasifikasi berdasarkan probabilitas tertinggi.

Model ini menghadirkan pendekatan yang komprehensif dalam pemrosesan data sensor dan pemodelan aktivitas manusia, dengan fokus pada deteksi jatuh. Dengan menggabungkan elemen-elemen seperti lapisan rekurent, *dropout*, dan *dense*, topologi ini dirancang untuk mencapai keseimbangan antara akurasi dan generalisasi model, memastikan respons yang andal dalam mengidentifikasi aktivitas fisik manusia.

2. 2 Akusisi Data

Penelitian ini menggunakan data aktifitas fisik pada manusia. Data yang digunakan merupakan data sekunder yang diperoleh dari platform penyedia dataset dan data primer melalui uji coba mandiri. Data yang diperlukan yaitu data akselerasi, rotasi dan total akselerasi yang harus direkam dalam tiga sumbu (x, y, dan z) untuk mendapatkan informasi spasial yang lengkap.

2. 3 Pra-pemrosesan

Raw data hasil akusisi data akan dilakukan pra-pemrosesan data sebelum dilakukan pelatihan menggunakan model yang dirancang. Pra-pemrosesan data dilakukan dengan memuat data ke *jupyter notebook*, membagi data menjadi data latih dan data uji, menggabungkan data menjadi satu data frame, dan segmentasi data.

2. 4 Pengujian Sistem

2. 4.1 Evaluasi

Proses evaluasi sistem akan dilakukan dengan menguji apakah sistem mampu untuk mengklasifikasikan data akselerometer, giroskop, dan percepatan akselerasi ke dalam empat kelas aktifitas fisik manusia. Pengukuran akan dilakukan dengan memanfaatkan teori *Confusion Matrix* yang akan membandingkan keluaran dari sistem dengan label sebenarnya dari data dan selanjutnya akan menghasilkan angka akurasi, presisi, *Recall*, dan *f1-score*.

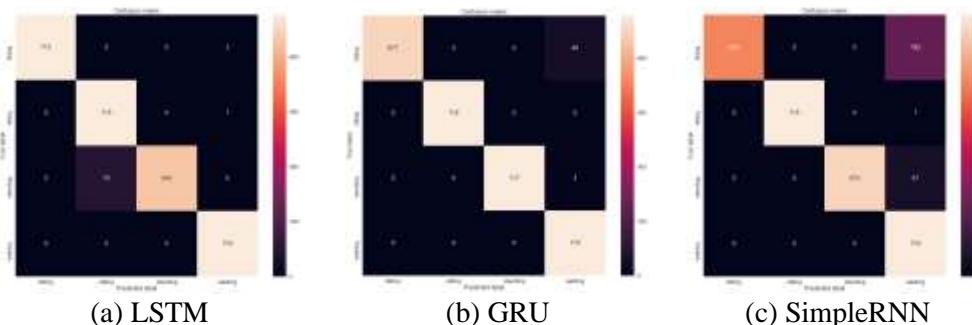
2. 4.2 Validasi

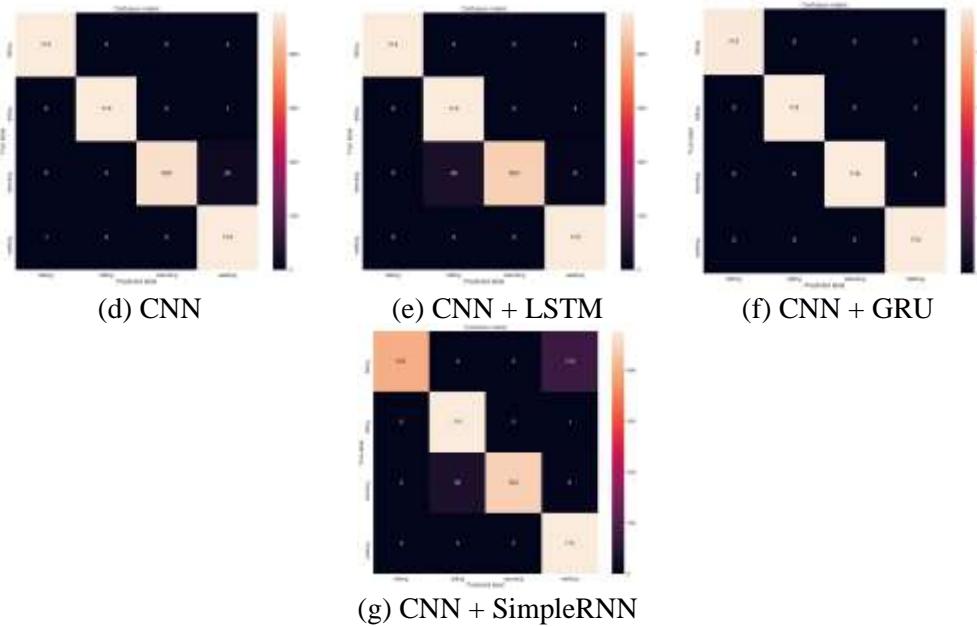
Proses validasi sistem akan dilakukan dengan menguji apakah sistem mampu untuk mengklasifikasikan data ke dalam kelas aktifitas fisik manusia dengan menggunakan data validasi yang belum pernah dites maupun dilatih sebelumnya. Proses validasi data dilakukan dengan menguji model dengan skenario *real-case* dari empat aktifitas fisik manusia yaitu duduk, berdiri, berjalan, dan jatuh lalu data akan dibandingkan untuk mengetahui mana hasil validasi model yang terbaik.

3. HASIL DAN PEMBAHASAN

3.3 Hasil Evaluasi Model Arsitektur

Model arsitektur yang telah dibuat pada setiap eksperimen akan dievaluasi untuk mengetahui dan menilai performa arsitektur model yang telah dibuat dalam mengklasifikasikan data yang diberikan. Pada evaluasi sistem, dilakukan evaluasi pada setiap arsitektur model dengan memanfaatkan teori *Confusion*. Dalam proses evaluasi sistem klasifikasi, digunakan data tes yang berjumlah 20% dari setiap eksperimen yang terdiri dari seluruh kelas aktifitas fisik manusia yakni duduk, berdiri, berjalan, dan jatuh. Hasil *confusion matrix* dari setiap arsitektur model dapat dilihat pada Gambar 3.





Gambar 3 *Heatmap Hasil Confusion Matrix* Pada Model: (a) LSTM, (b) GRU, (c) SimpleRNN, (d) CNN, (e) CNN + LSTM, (f) CNN + GRU, (g) CNN + SimpleRNN

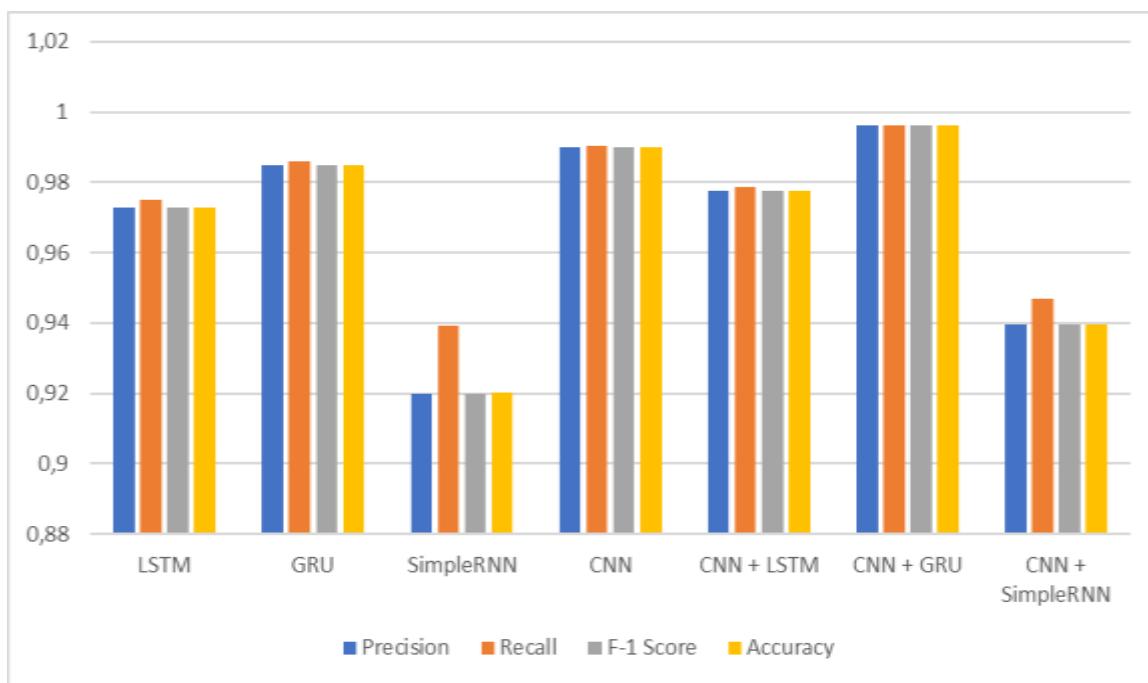
Gambar 3 menunjukkan komponen-komponen utama dari *Confusion Matrix*, yaitu *true positive* (prediksi jatuh yang benar), *false positive* (prediksi tidak jatuh padahal jatuh), *false negative* (prediksi jatuh padahal tidak jatuh), dan *true negative* (prediksi tidak jatuh yang benar). Nilai-nilai ini terperinci dalam Tabel 1 dan digunakan untuk mengevaluasi hasil dari berbagai percobaan klasifikasi yang telah dilakukan.

Tabel 1 Nilai *Confusion Matrix* Pada Arsitektur Model

No.	Arsitektur Model	TP	FP	FN	FP
1.	LSTM	2790	8	79	5723
2.	GRU	2825	43	43	5689
3.	SimpleRNN	2638	230	230	5502
4.	CNN	2839	29	28	5703
5.	CNN + LSTM	2804	6	64	5726
6.	CNN + GRU	2857	11	11	5721
7.	CNN + SimpleRNN	2695	121	173	5611

Tabel 2 Hasil Evaluasi dari *Confusion Matrix* Pada Evaluasi Model

No.	Arsitektur Model	Accuracy	Precision	Recall	F-1 Score
1.	LSTM	0.972698	0.972803	0.974941	0.972912
2.	GRU	0.985006	0.985007	0.985818	0.985011
3.	SimpleRNN	0.920410	0.919805	0.939153	0.919873
4.	CNN	0.989909	0.989888	0.990217	0.989923
5.	CNN + LSTM	0.977593	0.977685	0.978787	0.977768
6.	CNN + GRU	0.996171	0.996165	0.996212	0.996169
7.	CNN + SimpleRNN	0.939479	0.939679	0.946867	0.939763



Gambar 4 Hasil Evaluasi Pada Setiap Arsitektur Model

Dari grafik batang hasil evaluasi pada setiap arsitektur model pada Gambar 6.16, terlihat bahwa ketujuh model arsitektur yang diusulkan memiliki performa yang baik. Ketujuh arsitektur model yang dirancang memiliki akurasi yang cukup tinggi dengan rata-rata 97.7% dan nilai F-1 Score dengan rata-rata 0.98. Kemudian untuk model arsitektur yang memiliki performa paling baik dimiliki oleh model arsitektur CNN + GRU dan untuk performa terendah dari ketujuh model arsitektur yang diusulkan dimiliki oleh model arsitektur SimpleRNN.

3.4 Hasil Pengujian dan Validasi Model

Pada tahap ini, dilakukan validasi secara *real-case scenario* terhadap arsitektur model yang telah dibuat untuk mengklasifikasikan atau mendeteksi data validasi di setiap data dan kelasnya. Proses validasi ini menggunakan smartphone Xiaomi Poco M4 PRO yang sudah dipasang aplikasi testing terintegrasi dengan arsitektur model yang dibuat. Validasi dilakukan dengan melakukan 20x pengujian di setiap kelas dalam setiap arsitektur yang dibuat. Hasil *confusion matrix* validasi model dari setiap arsitektur model yang dibuat dapat dilihat pada Tabel 3.

Tabel 3 Hasil *Confusion Matrix* Validasi Setiap Arsitektur Model

		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	17	2	1	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	13	5	2	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	19	0	1	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	19	0	1	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	15	5	0	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	18	0	1	1
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20
		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	19	0	1	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20

(a) LSTM

(b) GRU

(c) SimpleRNN

(d) CNN

(e) CNN + LSTM

(f) CNN + GRU

		Kelas Prediksi			
		Jatuh	Duduk	Berdiri	Berjalan
Kelas Benar	Jatuh	13	1	6	0
	Duduk	0	20	0	0
	Berdiri	0	0	20	0
	Berjalan	0	0	0	20

(g) CNN + SimpleRNN

Tabel 3 menunjukkan *confusion matrix* untuk kelas duduk, berdiri, dan jatuh yang memiliki nilai konstan, menunjukkan bahwa ketujuh arsitektur model yang dibuat mampu mengklasifikasikan aktivitas fisik manusia secara efektif dalam skenario real. *Nilai true positive, false positive, false negative, dan true negative* terperinci dalam Tabel 4 untuk kelas jatuh, serta Tabel 5 untuk keempat kelas. Informasi dari *Confusion Matrix* ini digunakan untuk mengevaluasi hasil dari berbagai pengujian yang telah dilakukan.

Tabel 4 Nilai *Confusion Matrix* Pada Validasi Model Kelas Jatuh

No.	Arsitektur Model	Kelas	TP	FP	FN	TN
1.	LSTM	Jatuh	17	3	1	59
2.	GRU	Jatuh	13	7	5	55
3.	SimpleRNN	Jatuh	19	1	1	59
4.	CNN	Jatuh	15	5	3	57
5.	CNN + LSTM	Jatuh	18	2	2	58
6.	CNN + GRU	Jatuh	19	1	1	59
7.	CNN + SimpleRNN	Jatuh	13	7	7	53

Tabel 5 Nilai *Confusion Matrix* Pada Validasi Model Keempat Kelas

No.	Arsitektur Model	TP	FP	FN	TN
1.	LSTM	77	3	1	239
2.	GRU	73	7	5	237
3.	SimpleRNN	79	1	1	239
4.	CNN	75	5	3	237
5.	CNN + LSTM	78	2	2	238
6.	CNN + GRU	79	1	1	239
7.	CNN + SimpleRNN	73	7	7	233

Dari nilai *Confusion Matrix* pada Tabel 4 dan 5, kemudian dihitung nilai dari evaluasi *Precision, Recall, F-1 score, dan Accuracy* dari setiap model arsitektur yang dirancang dengan rumus pada persamaan (3.5), (3.6), (3.7), dan (3.8) serta hasilnya ditunjukkan pada Tabel 6 untuk khusus kelas jatuh dan Tabel 7 untuk nilai *confusion matrix* keempat kelas.

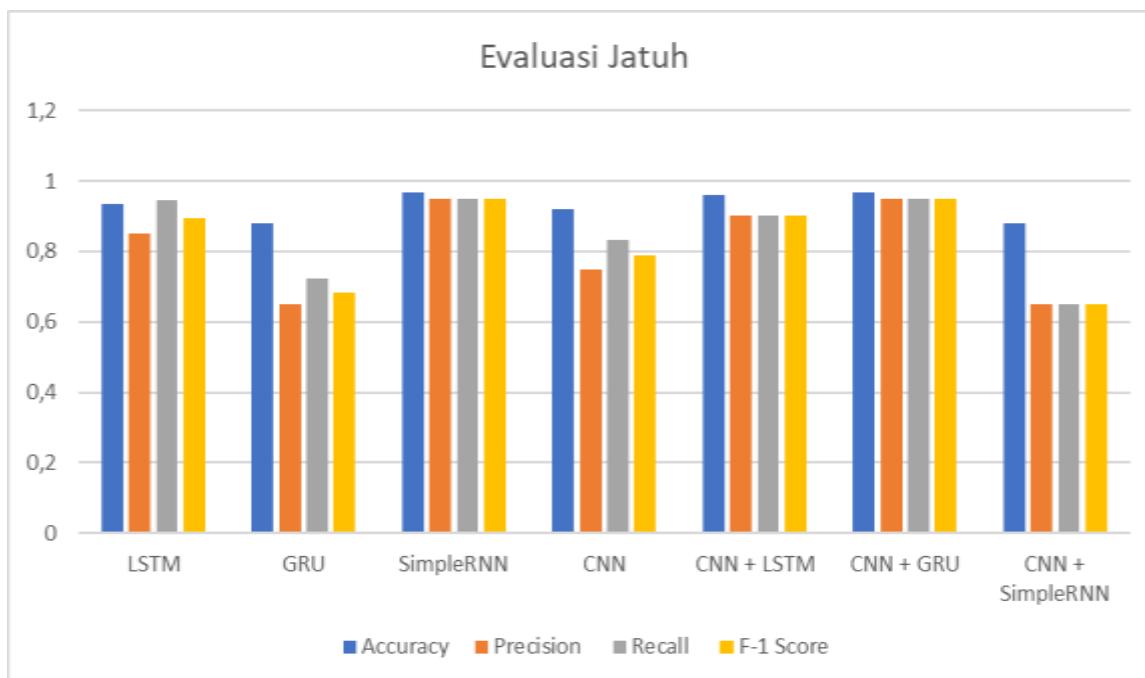
Tabel 6 Hasil Evaluasi dari *Confusion Matrix* Pada Validasi Model Kelas Jatuh

No.	Arsitektur Model	Kelas	Accuracy	Precision	Recall	F-1 Score
1.	LSTM	Jatuh	0.933	0.850	0.944	0.895
2.	GRU	Jatuh	0.880	0.650	0.722	0.684
3.	SimpleRNN	Jatuh	0.966	0.950	0.950	0.950
4.	CNN	Jatuh	0.920	0.750	0.833	0.789
5.	CNN + LSTM	Jatuh	0.960	0.900	0.900	0.900
6.	CNN + GRU	Jatuh	0.966	0.950	0.950	0.950
7.	CNN + SimpleRNN	Jatuh	0.880	0.650	0.650	0.650

Tabel 7 Hasil Evaluasi dari *Confusion Matrix* Pada Validasi Model Keempat Kelas

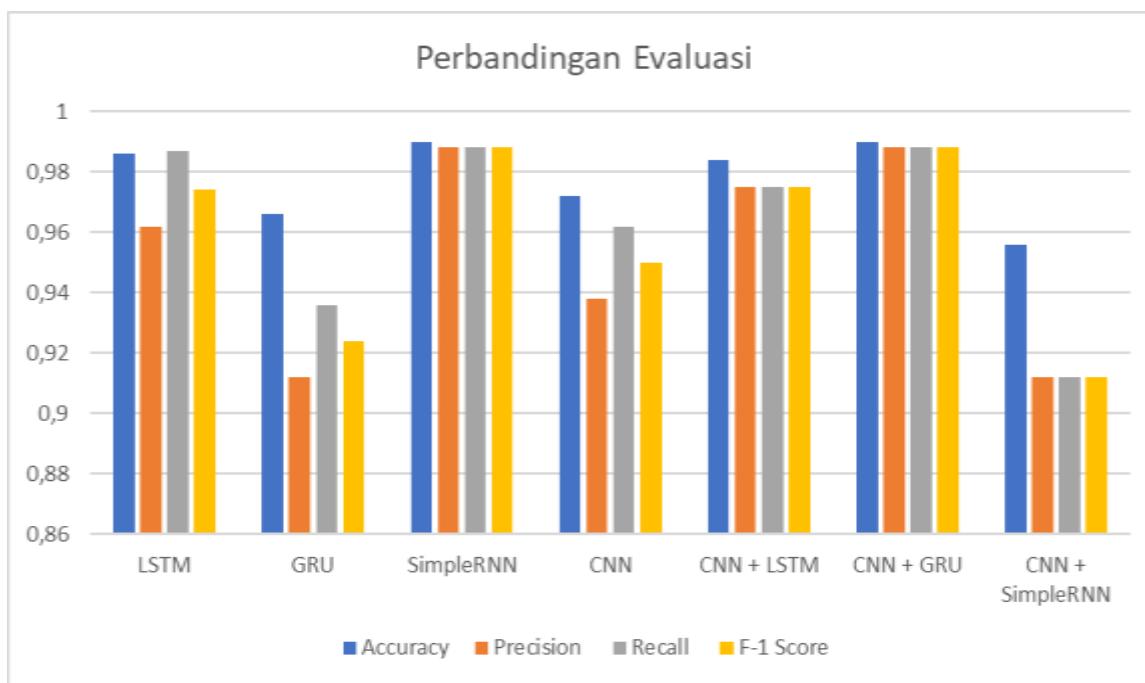
No.	Arsitektur Model	Accuracy	Precision	Recall	F-1 Score
1.	LSTM	0.986	0.962	0.987	0.974

2.	GRU	0.966	0.912	0.936	0.924
3.	SimpleRNN	0.990	0.988	0.988	0.988
4.	CNN	0.972	0.938	0.962	0.950
5.	CNN + LSTM	0.984	0.975	0.975	0.975
6.	CNN + GRU	0.990	0.988	0.988	0.988
7.	CNN + SimpleRNN	0.956	0.912	0.912	0.912



Gambar 5 Hasil Evaluasi Validasi Kelas Jatuh Pada Setiap Arsitektur Model

Dari grafik batang hasil evaluasi validasi kelas jatuh pada setiap arsitektur model pada Gambar 5 terlihat bahwa ketujuh model arsitektur yang diusulkan memiliki performa yang baik. Ketujuh arsitektur model yang dirancang memiliki akurasi yang cukup tinggi dengan rata-rata 93,3% dan nilai *F-1 Score* dengan rata-rata 0.83. Kemudian untuk model arsitektur yang memiliki performa paling baik dimiliki oleh model arsitektur SimpleRNN dan CNN + GRU, serta untuk performa terendah dari ketujuh model arsitektur yang diusulkan dimiliki oleh model arsitektur CNN + SimpleRNN.



Gambar 6 Hasil Evaluasi Validasi Pada Setiap Arsitektur Model

Dari grafik batang hasil evaluasi validasi keempat kelas pada setiap arsitektur model pada Gambar 6 terlihat bahwa ketujuh model arsitektur yang diusulkan memiliki performa yang baik. Ketujuh arsitektur model yang dirancang memiliki akurasi yang cukup tinggi dengan rata-rata 97,6% dan nilai *F-1 Score* dengan rata-rata 0.96. Kemudian untuk model arsitektur yang memiliki performa paling baik dimiliki oleh model arsitektur SimpleRNN dan CNN + GRU serta untuk performa terendah dari ketujuh model arsitektur yang diusulkan dimiliki oleh model arsitektur CNN + SimpleRNN.

4. KESIMPULAN

Hasil penelitian pengolahan data dengan variasi klasifikasi menegaskan beberapa poin kunci. Meskipun model arsitektur dengan tingkat akurasi tinggi tidak selalu menjamin kinerja optimal dalam kasus pengujian real, ditemukan bahwa rancangan model dengan performa terbaik adalah *convolutional neural network* (CNN) + *gated recurrent unit* (GRU), mencapai akurasi, presisi, *recall*, dan *F1-Score* di atas 99%. Di sisi lain, model arsitektur dengan performa terendah adalah *simple recurrent neural network* (SimpleRNN). Meskipun demikian, hasil uji dan validasi untuk kelas duduk, berdiri, dan berjalan menunjukkan performa yang sangat baik dan konsisten, seperti dibuktikan oleh nilai sempurna pada *confusion matrix* evaluasi. Namun, jika fokus hanya pada kelas jatuh, model dengan hasil validasi tertinggi adalah SimpleRNN dan CNN + GRU, sementara model dengan hasil terendah adalah CNN + SimpleRNN, dengan perbedaan metrik evaluasi yang signifikan antara keduanya.

5. SARAN

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran yang dapat meningkatkan kualitas hasil penelitian. Pertama, perlu ditambahkan *data set* untuk meningkatkan performa sistem agar lebih optimal. Kedua, pengembangan variasi subjek pengambilan data set sangat diperlukan dengan mempertimbangkan variasi kelas aktivitas fisik manusia, posisi *smartphone* saat pengambilan *data set*, serta penambahan sensor yang

digunakan. Dengan memperluas dan memperkaya data serta variasi subjek, penelitian ini memiliki potensi untuk memberikan hasil yang lebih komprehensif dan dapat diandalkan.

DAFTAR PUSTAKA

- [1] Sumandar, (2019). "Pengantar keperawatan gerontik dengan pendekatan asuhan keperawatan," *Deepublish*. ISBN: 978-623-209-087-3
- [2] Qurotul Aini et al., (2021). Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model Yolo. *CESS Vol. 6 No. 2 Juli 2021*. p-ISSN :2502-7131, e-ISSN :2502-714x
- [3] Bin K.S., Park J-H., Kwon C. et al, (2019) An Energy-Efcient Algorithm for Classification of Fall Types Using a Wearable Sensor. *IEEE Access* 7:31321–31329. DOI: 10.1109/ACCESS.2019.2902718
- [4] Kulwarun Warunsin, Thongchai Phairoh, (2022). Wristband Fall Detection System Using Deep Learning. *IEEE*. DOI: 10.1109/ICCCS55155.2022.9846023
- [5] Ge Zhang et al, (2019) Using Machine Learning Techniques to Optimize Fall Detection Algorithms in Smart Wristband. *IEEE*. DOI: 10.23919/IConAC.2019.8895065. Available: <https://doi.org/10.23919/IConAC.2019.8895065>
- [6] Lisha Chen, Songtao Lu, Tianyi Chen, (2022). Understanding Benign Overfitting in Gradient-Based Meta Learning. Arxiv. <https://doi.org/10.48550/arXiv.2206.13482>
- [7] Emad Ul Haq Qazi, Abdulrazaq Almorjan, Tanveer Zia, (2022). A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection. *MDPI*. DOI: 10.3390/app12167986
- [8] Na Lu, Yidan Wu, Li Feng, Jinbo Song, (2018). Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data. *IEEE*. DOI: 10.1109/JBHI.2018.2808281
- [9] Sheng-Ta Hsieh, Chun-Ling Lin, (2020). Fall Detection Algorithm Based on MPU6050 and Long-Term Short-Term Memory network. *IEEE*. DOI: 10.1109/CACS50047.2020.9289769
- [10] Seunghee Lee, Bummo Koo, Sumin Yang, Jongman Kim, Yejin Nam, Youngho Kim, (2022). Fall-from-Height Detection Using Deep Learning Based on IMU Sensor Data for Accident Prevention at Construction Sites. *MDPI*. DOI: 10.3390/s22166107