

Pengaruh Jumlah Fitur pada Algoritma *Machine Learning* dalam Memprediksi Aktivitas Jatuh

Made Liandana¹, I Made Darma Susila²

Institut Teknologi dan Bisnis STIKOM Bali

e-mail: ¹liandana@stikom-bali.ac.id, ²darma_s@stikom-bali.ac.id

Diajukan: 31 Januari 2023; Direvisi: 4 Maret 2023; Diterima: 8 Maret 2023

Abstrak

Mendeteksi terjadinya jatuh sangat penting dilakukan karena jatuh dapat memberikan dampak yang serius bagi kesehatan. Salah satu perangkat sensor yang dapat digunakan untuk menyediakan data aktivitas jatuh adalah sensor accelerometer. Data sensor tersebut perlu diekstraksi menjadi fitur dan diklasifikasi menggunakan algoritma machine learning. Selain itu, untuk memilah dan memilih fitur, dan mengetahui kombinasi fitur yang relevan diperlukan algoritma seleksi fitur. Pada penelitian ini, jatuh dideteksi berdasarkan data sensor accelerometer tiga sumbu (x , y , dan z), data yang digunakan merupakan data publik. Data diekstraksi menggunakan fungsi statistik yang terdiri dari: minimum, maksimum, rata-rata, nilai tengah, dan standar deviasi. Terdapat 15 fitur yang akan dievaluasi oleh algoritma machine learning. Algoritma machine learning yang digunakan adalah: k -Nearest Neighbors (KNN), Decision Tree (D-Tree), Random Forest (RF), Support Vector Machine (SVM), AdaBoost, dan Gradient Boosting. Untuk mengevaluasi jumlah fitur yang paling optimal pada algoritma machine learning, seleksi fitur yang digunakan adalah Analysis of Variance (ANOVA). Penggunaan fitur sebanyak 7, 8, 9, 10, dan 11 fitur menghasilkan performa machine learning yang paling optimal yang dicapai oleh machine learning: Decision Tree (D-Tree), Random Forest (RF), dan Gradient Boosting. Secara berturut-turut, ketiga classifier ini memiliki nilai accuracy, $F-1$, precision, recall, dan specificity adalah 1.000, 1.000, 1.000, 1.000, dan 1.000.

Kata kunci: Accelerometer, Deteksi jatuh, ANOVA, Machine learning.

Abstract

Detecting a fall is very important because falls can have serious impacts on health. One sensor device that can be used to provide fall activity data is the accelerometer sensor. The sensor data needs to be extracted into features and classified using machine learning algorithms. In addition, to sort and select relevant features, a feature selection algorithm is needed. In this study, falls were detected based on three-axis accelerometer sensor data (x , y , and z), the data used was public. Data is extracted using statistical functions consisting of minimum, maximum, average, median value, and standard deviation. 15 features will be evaluated by the machine learning algorithm. The machine learning algorithms used are k -nearest Neighbors (KNN), Decision Tree (D-Tree), Random Forest (RF), Support Vector Machine (SVM), AdaBoost, and Gradient Boosting. To evaluate the most optimal number of features that influence machine learning algorithms, the feature selection algorithm used is Analysis of Variance (ANOVA). The use of 7, 8, 9, 10, and 11 features produces the most optimal machine learning performance achieved by machine learning: Decision Tree (D-Tree), Random Forest (RF), and Adaboost. Respectively, these three classifiers have accuracy, $F-1$, precision, recall, and specificity values of 1,000, 1,000, 1,000, 1,000, and 1,000.

Keywords: Accelerometer, Fall detection, ANOVA, Machine learning.

1. Pendahuluan

Data statistik menunjukkan kematian karena cedera pasca jatuh sering terjadi pada lanjut usia berusia 80 tahun ke atas [1]. Dalam bidang kesehatan, deteksi jatuh pada lanjut usia merupakan hal yang sangat penting karena dapat digunakan untuk membantu mengurangi dampak serius yang diakibatkan jatuh [2][3][4]. Deteksi jatuh dapat dilakukan menggunakan perangkat kamera[5] atau perangkat sensor [2], [3], [6], [7]. Isu privasi menjadi hal yang menjadi perdebatan dalam hal penggunaan kamera [8][9][10], sehingga penggunaan perangkat sensor menjadi salah satu pilihan.

Sensor-sensor yang dapat digunakan untuk melakukan deteksi jatuh adalah: accelerometer, gyroscope [2], [6], [7], [11]. Sensor accelerometer mampu memberikan data yang akurat tentang gerakan tubuh yang mengandung pola gerakan tertentu, studi menunjukkan bahwa sensor ini dapat memenuhi kebutuhan pemantauan pergerakan tubuh manusia sehari-hari, dan dapat bekerja secara normal dalam berbagai kondisi [12]. Oleh karena, itu sensor *accelerometer* sangat populer digunakan sebagai perangkat untuk mendeteksi aktivitas, salah satunya aktivitas jatuh.

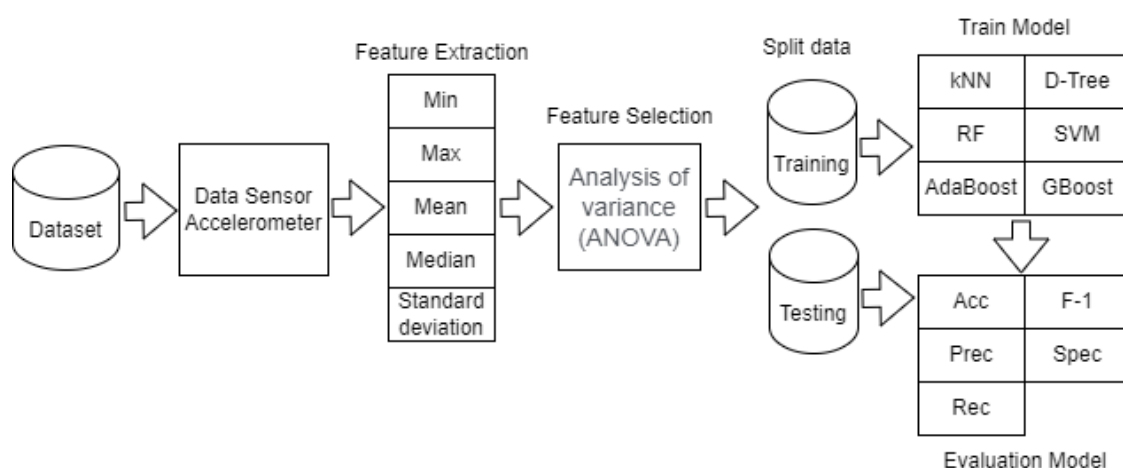
Penggunaan *machine learning* juga menjadi komponen penting dalam pengembangan sistem deteksi jatuh yang efektif [4][13][14]. Teknik ini memungkinkan sistem untuk belajar dari data yang diperoleh dari sensor accelerometer dan mengenali pola-pola yang mencirikan kejadian jatuh [15]. Dengan menggunakan algoritma *machine learning* yang tepat, sistem dapat meningkatkan akurasi deteksi dan mengurangi jumlah false positives [16]. *Machine learning* juga dapat membantu dalam mengekstraksi fitur-fitur penting dari data accelerometer yang kompleks untuk meningkatkan kinerja deteksi [2], [15], [16].

Selain penggunaan sensor accelerometer dan *machine learning*, teknik seleksi fitur juga dapat meningkatkan kinerja sistem deteksi jatuh [17][18]. Salah satu teknik seleksi fitur yang umum digunakan adalah analisis varians (ANOVA), teknik ini digunakan untuk memilih fitur-fitur yang paling signifikan dalam membedakan antara kejadian jatuh dan aktivitas normal [19]. Dengan menggunakan teknik seleksi fitur ANOVA, sistem dapat meningkatkan akurasi deteksi dengan fokus pada fitur-fitur yang paling informatif dan relevan [19].

Kombinasi antara sensor accelerometer, *machine learning*, dan teknik seleksi fitur ANOVA menawarkan pendekatan yang lebih lengkap dalam deteksi jatuh [19]. Penelitian sebelumnya telah dilakukan berkaitan dengan kombinasi tiga hal tersebut [20][21]. Namun, jumlah optimal dari jumlah fitur yang digunakan untuk melakukan deteksi jatuh perlu diinvestigasi lebih lanjut. Sehingga pada ini, pengaruh jumlah fitur yang diseleksi oleh seleksi fitur ANOVA (Analysis of Variance) terhadap performa *machine learning* menjadi fokus utama. Pada studi ini algoritma *machine learning* yang digunakan untuk mengevaluasi fitur yang dipilih adalah: k-Nearest Neighbors (KNN), Decision Tree (D-Tree), Random Forest (RF), Support Vector *Machine* (SVM), AdaBoost, dan Gradient Boosting. Pertimbangan menggunakan lebih dari satu algoritma *machine learning* adalah untuk mengetahui algoritma *machine learning* mana yang paling cocok untuk dataset yang digunakan. Sedangkan data sensor akselerometer akan diklasifikasi menjadi dua *class* atau label, yaitu: data jatuh dan data tidak jatuh.

2. Metode Penelitian

Dataset yang digunakan adalah data publik, data terdiri dari dua jenis aktivitas manusia, yaitu aktivitas jatuh dan aktivitas tidak jatuh. Data dari dataset dipilah dan dipilih data yang hanya berasal dari sensor accelerometer tiga sumbu (x, y, z). Nilai data sensor accelerometer diekstraksi menjadi 5 kategori fitur, yaitu: minimum (*Min*), maksimum (*Max*), rata-rata (*Mean*), nilai tengah (Median), dan standar deviasi (*Standard Deviation*).



Gambar 1. Model yang diusulkan.

Dengan jumlah sumbu sensor sebanyak 3 sumbu (x, y, z) sehingga jumlah fitur yang dihasilkan menjadi 15 fitur. Untuk mengetahui fitur-fitur yang memberikan pengaruh paling optimal pada performa algoritma *machine learning*, pada penelitian ini fitur diseleksi menggunakan Analysis of Variance (ANOVA). Selanjutnya data dibagi menjadi data latih dan data uji untuk digunakan untuk melatih dan

mengevaluasi algoritma *machine learning*. Algoritma *machine learning* yang digunakan untuk mengklasifikasi aktivitas jatuh dan tidak jatuh yaitu: k-Nearest Neighbors (KNN), Decision Tree (D-Tree), Random Forest (RF), Support Vector *Machine* (SVM), AdaBoost, dan Gradient Boosting. Model yang sudah dilatih dievaluasi menggunakan sejumlah parameter yaitu: *accuracy*, *precision*, *recall*, *recall*, *specificity*, dan *F-1 score*.

2.1. Ekstraksi Fitur

Ekstraksi fitur yang digunakan untuk mendapatkan fitur dari data sensor accelerometer adalah dengan menggunakan pendekatan statistik. Persamaan yang digunakan untuk menghitung nilai minimum (*min*), maksimum (*max*), rata-rata (*Mean*), nilai tengah (*Median*), dan standar deviasi (*Std*), secara berturut-turut adalah Persamaan (1), Persamaan (2), Persamaan (3), Persamaan (4), dan Persamaan 5. *A* adalah nilai akselerasi, baik untuk sumbu x, y, dan z. \bar{A} adalah nilai rata-rata, *Med* adalah nilai tengah, *Std* adalah nilai standar deviasi. Sedangkan *i* adalah data ke-*i* dari sejumlah data serial yang akan dievaluasi.

$$min=min(A_i) \tag{1}$$

$$max=max(A_i) \tag{2}$$

$$\bar{A} = \frac{1}{n} \sum_{i=1}^n (A_i) \tag{3}$$

$$Med = \begin{cases} A_{\lfloor \frac{n+1}{2} \rfloor}, & \text{if } n \text{ odd} \\ \frac{1}{2} \left(A_{\lfloor \frac{n}{2} \rfloor} + x_{\lfloor \frac{n+1}{2} \rfloor} \right), & \text{if } n \text{ even} \end{cases} \tag{4}$$

$$std = \sqrt{\frac{1}{n} \sum_{i=1}^n (A_i - \bar{A})^2} \tag{5}$$

2.2. Seleksi Fitur ANOVA (Analysis of Variance)

ANOVA (Analysis of Variance) digunakan untuk melakukan seleksi fitur berdasarkan nilai dari skor ANOVA. Untuk menghitung skor ANOVA dapat menggunakan Persamaan (6). Variabel *i* bernilai 1 sampai *k*, *k* adalah banyaknya kelompok yang dikelompokkan dari semua sampel data. Sedangkan \bar{A}_i adalah rata-rata dari setiap kelompok, g_{ij} adalah sample data dari setiap kelompok, dengan panjang setiap kelompok adalah n_i . \bar{A} adalah rata-rata dari semua sample dan G_m merupakan semua sample data dengan jumlah sebanyak *K*. df_w adalah *K* - *k*, sedangkan df_b adalah *k* - 1.

$$skor\ anova = \frac{df_w}{df_b} \times \frac{\sum_{i=1}^k n_i (\bar{A}_i - \bar{A})^2}{\sum_{i=1}^k \sum_{j=1}^{n_i} (g_{ij} - \bar{A}_i)^2} \tag{6}$$

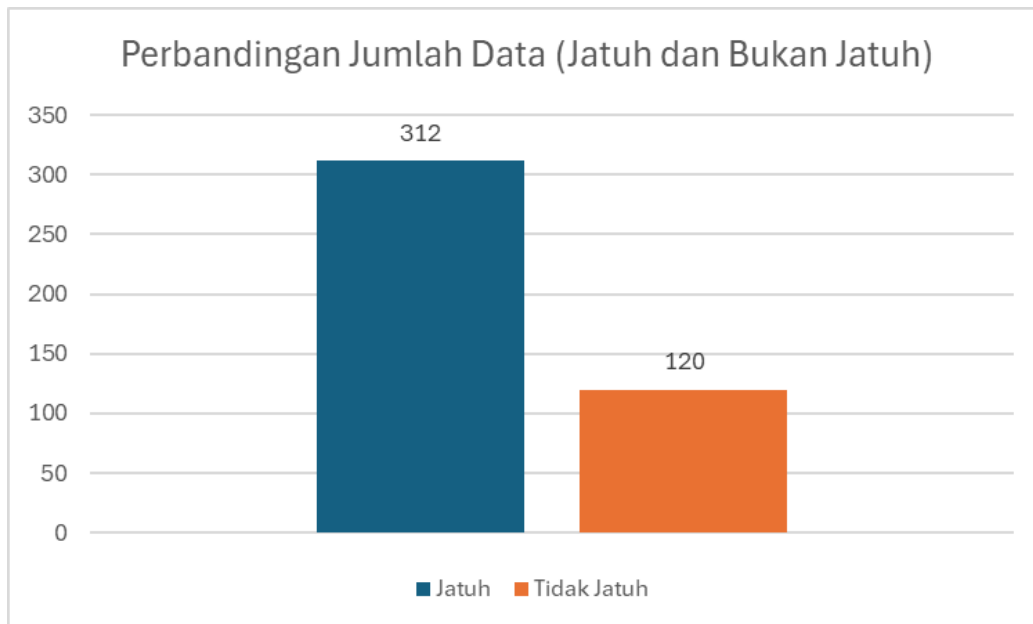
$$\bar{A}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} g_{ij} \tag{7}$$

$$\bar{A} = \frac{1}{K} \sum_{m=1}^K G_m \tag{8}$$

3. Hasil dan Pembahasan

3.1. Deskripsi Dataset

Penelitian ini menggunakan dataset publik [22], data yang digunakan bersumber dari sensor magnetometer (HMC5883L, Honeywell, USA), accelerometer (ADXL345, Analog Devices, USA) dan *gyroscope* (ITG-3200, InvenSense Inc., USA). Dalam proses pengambilan data, sensor dikenakan pada pinggang. Subjek pengambilan data terdiri dari 2 orang perempuan dan 6 orang laki-laki, masing-masing subjek melakukan 3 kali simulasi gerakan atau aktivitas. Walaupun dataset yang digunakan terdiri dari 3 sensor, namun pada penelitian ini hanya berfokus pada penggunaan single sensor, yaitu sensor accelerometer, hal ini bertujuan untuk mengurangi dimensi data. Dari dataset yang digunakan, data dikelompokkan menjadi dua aktivitas, yaitu: aktivitas jatuh dan aktivitas tidak jatuh. Data bukan jatuh merupakan data sensor dari aktivitas sehari-hari. Total data yang digunakan adalah sebanyak 432, yang terdiri dari 312 data aktivitas jatuh dan 120 aktivitas tidak jatuh, perbandingan data tersebut ditunjukkan pada Gambar 2.



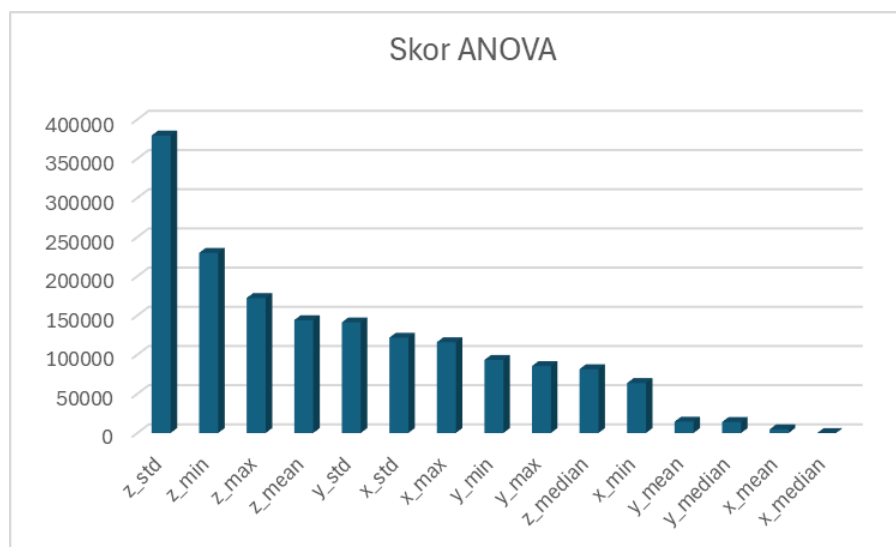
Gambar 2. Perbandingan Jumlah Data Jatuh dan Bukan Jatuh.

3.2. Penentuan Jumlah Fitur Berdasarkan Nilai Skor ANOVA (Analysis of Variance)

Untuk mengetahui pengaruh jumlah fitur terhadap performa *machine learning*, tahap lanjutan setelah proses ekstraksi fitur dilakukan adalah menghitung nilai skor dari setiap fitur menggunakan ANOVA (Analysis of Variance). Tabel 1 menunjukkan nilai skor dari ANOVA, nilai-nilai tersebut digunakan sebagai acuan mengevaluasi fitur yang digunakan dengan menyusun kombinasi fitur. Kombinasi fitur mana yang memberikan performa *machine learning* terbaik tentunya diperlukan proses evaluasi pada setiap kombinasi fitur. Gambar 3 menunjukkan fitur dengan skor tertinggi adalah *z_std* atau standar deviasi dari sumbu z sensor *accelerometer*. Sedangkan skor terkecil adalah *x_median* atau nilai tengah dari sumbu x sensor *accelerometer*.

Tabel 1. Skor ANOVA Setiap Fitur.

No	Nama fitur	Skor Anova	Keterangan
1	z_std	379.508	standar deviasi untuk sumbu z
2	z_min	229.802	nilai minimum untuk sumbu z
3	z_max	172.271	nilai maksimum untuk sumbu z
4	z_mean	143.968	nilai rata-rata untuk sumbu z
5	y_std	141.124	standar deviasi untuk sumbu y
6	x_std	121.609	standar deviasi untuk sumbu x
7	x_max	115.821	nilai maksimum untuk sumbu x
8	y_min	93.151	nilai minimum untuk sumbu y
9	y_max	85.356	nilai maksimum untuk sumbu y
10	z_median	81.453	nilai median untuk sumbu z
11	x_min	63.704	nilai minimum untuk sumbu x
12	y_mean	14.456	nilai rata-rata untuk sumbu y
13	y_median	14.105	nilai median untuk sumbu y
14	x_mean	4.667	nilai rata-rata untuk sumbu x
15	x_median	0.549	nilai median untuk sumbu x



Gambar 3. Perbandingan Skor ANOVA Setiap Fitur.

Total fitur yang dihasilkan dari proses ekstraksi fitur menggunakan Persamaan (1) sampai dengan (5) adalah sebanyak 15 fitur. Dari 15 fitur tersebut ditentukan kombinasi fitur berdasarkan skor ANOVA, dimulai dari skor ANOVA tertinggi hingga terendah, sehingga dihasilkan 15 kombinasi fitur seperti ditunjukkan pada Tabel 2. Kombinasi fitur tersebut selanjutnya dievaluasi menggunakan 6 algoritma *machine learning*, seperti ditunjukkan pada Gambar 1. Evaluasi pertama dilakukan untuk jumlah fitur dengan skor tertinggi, selanjutnya evaluasi kedua dilanjutkan dengan mengombinasikan fitur yang menempati urutan pertama dengan fitur yang menempati urutan kedua, begitu seterusnya sampai semua kombinasi fitur dievaluasi.

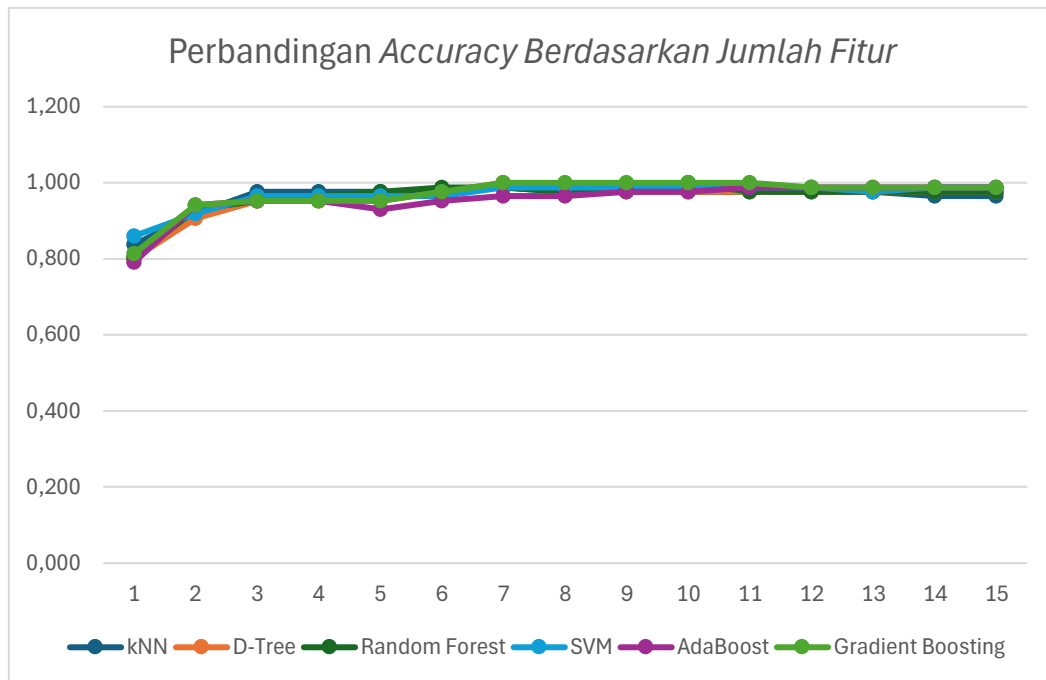
Tabel 2. Kombinasi Fitur.

No	Kombinasi fitur	Jumlah Fitur
1	z_std	1
2	z_std, z_min	2
3	z_std, z_min, z_max	3
4	z_std, z_min, z_max, z_mean	4
5	z_std, z_min, z_max, z_mean, y_std	5
6	z_std, z_min, z_max, z_mean, y_std, x_std	6
7	z_std, z_min, z_max, z_mean, y_std, x_std, x_max	7
8	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_min	8
9	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max	9
10	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median	10
11	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median, x_min	11
12	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median, x_min, y_mean	12
13	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median, x_min, y_mean, y_median	13
14	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median, x_min, y_mean, y_median, x_mean	14
15	z_std, z_min, z_max, z_mean, y_std, x_std, x_max, y_max, z_median, x_min, y_mean, y_median, x_mean, x_median	15

3.3. Performa *Machine Learning* Berdasarkan Jumlah Fitur

Performa dari *machine learning* berdasarkan jumlah fitur ditunjukkan pada Tabel 3 sampai dengan Tabel 17. Penggunaan hanya 1 fitur performa *machine learning* Support Vector *Machine* (SVM) menghasilkan nilai tertinggi dibandingkan dengan *machine learning* yang lain sedangkan performa terendah dihasilkan oleh *machine learning* Adaboost.

Pada kombinasi fitur sebanyak 15 atau semua fitur digunakan untuk memprediksi aktivitas jatuh atau tidak jatuh, *machine learning* Support Vector *Machine* (SVM), AdaBoost, dan Gradient Boosting memiliki performa paling tinggi dibandingkan dengan algoritma *machine learning* lain yang digunakan pada penelitian ini. Seperti ditunjukkan pada Tabel 17, secara kronologis, nilai *accuracy*, *F-1*, *precision*, *recall*, dan *specificity* adalah 0.988, 0.988, 0.989, 0.988, dan 0.995 untuk ketiga *clasiffier* tersebut.



Gambar 4. Perbandingan Accuracy Berdasarkan Jumlah Fitur.

Tabel 3. Performa Dengan Kombinasi 1 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.837	0.835	0.834	0.837	0.733
2	D-Tree	0.802	0.798	0.796	0.802	0.668
3	Random Forest	0.802	0.798	0.796	0.802	0.668
4	SVM	0.860	0.860	0.860	0.860	0.793
5	AdaBoost	0.791	0.784	0.782	0.791	0.638
6	Gradient Boosting	0.814	0.808	0.807	0.814	0.678

Tabel 4. Performa Dengan Kombinasi 2 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.919	0.908	0.937	0.919	0.968
2	D-Tree	0.907	0.908	0.910	0.907	0.887
3	Random Forest	0.930	0.931	0.933	0.930	0.922
4	SVM	0.919	0.921	0.930	0.919	0.943
5	AdaBoost	0.942	0.943	0.946	0.942	0.952
6	Gradient Boosting	0.942	0.943	0.946	0.942	0.952

Tabel 5. Performa Dengan Kombinasi 3 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.953	0.955	0.960	0.953	0.982
3	Random Forest	0.953	0.955	0.960	0.953	0.982
4	SVM	0.965	0.966	0.969	0.965	0.986
5	AdaBoost	0.953	0.955	0.960	0.953	0.982
6	Gradient Boosting	0.953	0.955	0.960	0.953	0.982

Tabel 6. Performa Dengan Kombinasi 4 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.953	0.955	0.960	0.953	0.982
3	Random Forest	0.965	0.966	0.969	0.965	0.986
4	SVM	0.965	0.966	0.969	0.965	0.986
5	AdaBoost	0.953	0.955	0.960	0.953	0.982
6	Gradient Boosting	0.953	0.955	0.960	0.953	0.982

Tabel 7. Performa Dengan Kombinasi 5 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.953	0.955	0.960	0.953	0.982
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.965	0.966	0.969	0.965	0.986
5	AdaBoost	0.930	0.932	0.944	0.930	0.973
6	Gradient Boosting	0.953	0.955	0.960	0.953	0.982

Tabel 8. Performa Dengan Kombinasi 6 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.988	0.988	0.989	0.988	0.995
4	SVM	0.965	0.966	0.969	0.965	0.982
5	AdaBoost	0.953	0.955	0.960	0.953	0.982
6	Gradient Boosting	0.977	0.977	0.979	0.977	0.991

Tabel 9. Performa Dengan Kombinasi 7 Fitur.

No	Machine Learning	Accuracy	F-I	Precision	Recall	Specificity
1	kNN	0.988	0.988	0.989	0.988	0.995
2	D-Tree	1.000	1.000	1.000	1.000	1.000
3	Random Forest	0.988	0.988	0.989	0.988	0.995
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.965	0.966	0.969	0.965	0.986
6	Gradient Boosting	1.000	1.000	1.000	1.000	1.000

Tabel 10. Performa Dengan Kombinasi 8 Fitur.

No	Machine Learning	Accuracy	F-I	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.965	0.966	0.969	0.965	0.986
6	Gradient Boosting	1.000	1.000	1.000	1.000	1.000

Tabel 11. Performa Dengan Kombinasi 9 Fitur.

No	Machine Learning	Accuracy	F-I	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.988	0.988	0.989	0.988	0.995
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.977	0.977	0.979	0.977	0.991
6	Gradient Boosting	1.000	1.000	1.000	1.000	1.000

Tabel 12. Performa Dengan Kombinasi 10 Fitur.

No	Machine Learning	Accuracy	F-I	Precision	Recall	Specificity
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	1.000	1.000	1.000	1.000	1.000
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.977	0.977	0.979	0.977	0.991
6	Gradient Boosting	1.000	1.000	1.000	1.000	1.000

Tabel 13. Performa Dengan Kombinasi 11 Fitur.

<i>No</i>	<i>Machine Learning</i>	<i>Accuracy</i>	<i>F-I</i>	<i>Precision</i>	<i>Recall</i>	<i>Specificity</i>
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.988	0.988	0.989	0.988	0.995
6	Gradient Boosting	1.000	1.000	1.000	1.000	1.000

Tabel 14. Performa Dengan Kombinasi 12 Fitur.

<i>No</i>	<i>Machine Learning</i>	<i>Accuracy</i>	<i>F-I</i>	<i>Precision</i>	<i>Recall</i>	<i>Specificity</i>
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.988	0.988	0.989	0.988	0.995
6	Gradient Boosting	0.988	0.988	0.989	0.988	0.995

Tabel 15. Performa Dengan Kombinasi 13 Fitur.

<i>No</i>	<i>Machine Learning</i>	<i>Accuracy</i>	<i>F-I</i>	<i>Precision</i>	<i>Recall</i>	<i>Specificity</i>
1	kNN	0.977	0.977	0.979	0.977	0.991
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.977	0.977	0.979	0.977	0.991
5	AdaBoost	0.988	0.988	0.989	0.988	0.995
6	Gradient Boosting	0.988	0.988	0.989	0.988	0.995

Tabel 16. Performa Dengan Kombinasi 14 Fitur.

<i>No</i>	<i>Machine Learning</i>	<i>Accuracy</i>	<i>F-I</i>	<i>Precision</i>	<i>Recall</i>	<i>Specificity</i>
1	kNN	0.965	0.966	0.969	0.965	0.986
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.988	0.988	0.989	0.988	0.995
6	Gradient Boosting	0.988	0.988	0.989	0.988	0.995

Tabel 17. Performa Dengan Kombinasi 15 Fitur.

No	Machine Learning	Accuracy	F-1	Precision	Recall	Specificity
1	kNN	0.965	0.966	0.969	0.965	0.986
2	D-Tree	0.977	0.977	0.979	0.977	0.991
3	Random Forest	0.977	0.977	0.979	0.977	0.991
4	SVM	0.988	0.988	0.989	0.988	0.995
5	AdaBoost	0.988	0.988	0.989	0.988	0.995
6	Gradient Boosting	0.988	0.988	0.989	0.988	0.995

Dari 15 kombinasi yang telah dievaluasi, penggunaan fitur sebanyak 7, 8, 9, 10 dan 11 fitur menghasilkan *performa machine learning* yang paling optimal yang dicapai oleh *machine learning*: Decision Tree (D-Tree), Random Forest (RF), dan Gradient Boosting. Secara berturut-turut, ketiga *classifier* ini memiliki nilai *accuracy*, *F-1*, *precision*, *recall*, dan *specificity* adalah 1.000, 1.000, 1.000, 1.000, dan 1.000, seperti ditunjukkan pada Tabel 9 sampai dengan Tabel 13. Gambar 4 menunjukkan perbandingan *accuracy machine learning* pada setiap kombinasi fitur.

4. Kesimpulan

Dalam eksperimen yang telah dilakukan, pengaruh jumlah fitur dalam memprediksi jatuh dan tidak jatuh memiliki pengaruh yang signifikan, hal ini dapat dilihat dari *performa machine learning* yang dihasilkan berdasarkan kombinasi fitur yang digunakan. Kombinasi fitur ditentukan dengan seleksi fitur ANOVA (Analysis Variance), yaitu pembentukan kombinasi fitur diawali dari fitur yang memiliki nilai skor tertinggi, kemudian dilanjutkan dengan membuat kombinasi fitur dengan fitur lain yang memiliki nilai skor lebih rendah. Terdapat 3 kombinasi fitur yang memberikan pengaruh signifikan terhadap *performa algoritma machine learning* khususnya *machine learning* Decision Tree (D-Tree), Random Forest (RF), dan Gradient Boosting, yaitu: kombinasi 7 fitur, 8 fitur, 10 fitur, 11 fitur, dan 12 fitur. Secara berturut-turut, nilai *accuracy*, *F-1*, *precision*, *recall*, dan *specificity* adalah 1.000, 1.000, 1.000, 1.000, dan 1.000.

Limitasi dari penelitian ini adalah perlu diuji lebih lanjut menggunakan dataset yang lain sehingga diketahui apakah model yang diusulkan memberikan dampak yang sama dengan karakteristik data yang berbeda. Di samping itu, algoritma seleksi fitur dan teknik ekstraksi fitur yang lain, seperti Information Gain dan penggunaan fitur berbasis frekuensi dapat digunakan sebagai perbandingan.

Daftar Pustaka

- [1] J. Santiago, E. Cotto, L. G. Jaimes, and I. Vergara-Laurens, "Fall detection system for the elderly," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*, Mar. 2017.
- [2] R. Jain and V. B. Semwal, "A Novel Feature Extraction Method for Preimpact Fall Detection System Using Deep Learning and Wearable Sensors," *IEEE Sens J*, vol. 22, no. 23, pp. 22943–22951, Dec. 2022.
- [3] Y. H. Nho, J. G. Lim, and D. S. Kwon, "Cluster-Analysis-Based User-Adaptive Fall Detection Using Fusion of Heart Rate Sensor and Accelerometer in a Wearable Device," *IEEE Access*, vol. 8, pp. 40389–40401, 2020.
- [4] R. Jayakarhik, A. Srinivasan, S. Goswami, Shivaranjini, and R. Mahaveerakannan, "Fall Detection Scheme based on Deep Learning Model for High-Quality Life," *3rd International Conference on Electronics and Sustainable Communication Systems, ICESC 2022 - Proceedings*, pp. 1582–1588, 2022.
- [5] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, "Fall Detection and Activity Recognition Using Human Skeleton Features," *IEEE Access*, vol. 9, pp. 33532–33542, 2021.
- [6] T. Althobaiti, S. Katsigiannis, and N. Ramzan, "Triaxial Accelerometer-Based Falls and Activities of Daily Life Detection Using Machine Learning," *Sensors 2020, Vol. 20, Page 3777*, vol. 20, no. 13, p. 3777, Jul. 2020.
- [7] L. M. Martins, N. F. Ribeiro, F. Soares, and C. P. Santos, "Inertial Data-Based AI Approaches for ADL and Fall Recognition," *Sensors 2022, Vol. 22, Page 4028*, vol. 22, no. 11, p. 4028, May 2022.

-
- [8] H. Wang, J. Zhao, J. Li, L. Tian, P. Tu, T. Cao, Y. An, K. Wang, and S. Li, "Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques," *Security and Communication Networks*, vol. 2020, 2020.
- [9] Z. Hussain, M. Sheng, and W. E. Zhang, "Different Approaches for Human Activity Recognition: A Survey," *Journal of Network and Computer Applications*, vol. 167, Jun. 2019.
- [10] F. Serpush, M. B. Menhaj, B. Masoumi, and B. Karasfi, "Wearable Sensor-Based Human Activity Recognition in the Smart Healthcare System," *Comput Intell Neurosci*, vol. 2022, 2022.
- [11] C. Vong, T. Theptit, V. Watcharakonpipat, P. Chanchotisatien, and S. Laitrakun, "Comparison of Feature Selection and Classification for Human Activity and Fall Recognition using Smartphone Sensors," *2021 Joint 6th International Conference on Digital Arts, Media and Technology with 4th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, ECTI DAMT and NCON 2021*, pp. 170–173, Mar. 2021.
- [12] Z. He, K. Wang, Z. Zhao, T. Zhang, Y. Li, and L. Wang, "A Wearable Flexible Acceleration Sensor for Monitoring Human Motion," *Biosensors 2022, Vol. 12, Page 620*, vol. 12, no. 8, p. 620, Aug. 2022.
- [13] S. Rastogi and J. Singh, "A systematic review on machine learning for fall detection system," *Comput Intell*, vol. 37, no. 2, pp. 951–974, May 2021.
- [14] A. K. Alhazmi, M. A. Alanazi, C. Liu, and V. P. Chodavarapu, "Machine Learning Enabled Fall Detection with Compact Millimeter Wave System," *Proceedings of the IEEE National Aerospace Electronics Conference, NAECON*, vol. 2021-August, pp. 217–222, 2021.
- [15] A. Choi, T. H. Kim, O. Yuhai, S. Jeong, K. Kim, H. Kim, and J. H. Mun, "Deep Learning-Based Near-Fall Detection Algorithm for Fall Risk Monitoring System Using a Single Inertial Measurement Unit," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2385–2394, 2022.
- [16] R. K. Hatkeposhti, M. Y. Tabari, and M. GolsorkhtabariAmiri, "Fall Detection using Deep Learning Algorithms and Analysis of Wearable Sensor Data by Presenting a New Sampling Method," *International Journal of Engineering*, vol. 35, no. 10, pp. 1941–1958, Oct. 2022.
- [17] T. M. Le, V. T. Ly, and S. V. T. Dao, "A Feature Selection Approach for Fall Detection Using Various Machine Learning Classifiers," *IEEE Access*, vol. 9, pp. 115895–115908, 2021.
- [18] C. A. Silva, R. García-Bermúdez, and E. Casilari, "Features Selection for Fall Detection Systems Based on Machine Learning and Accelerometer Signals," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12862 LNCS, pp. 380–391, 2021.
- [19] D. Kelly, J. Condell, J. Gillespie, K. Munoz Esquivel, J. Barton, S. Tedesco, A. Nordstrom, M. Åkerlund Larsson, and A. Alamäki, "Improved screening of fall risk using free-living based accelerometer data," *J Biomed Inform*, vol. 131, p. 104116, Jul. 2022.
- [20] J. A. Santoyo-Ramón, E. Casilari-Pérez, and J. M. Cano-García, "Study of the Detection of Falls Using the SVM Algorithm, Different Datasets of Movements and ANOVA," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11465 LNBI, pp. 415–428, 2019.
- [21] J. Silva, D. Gomes, I. Sousa, and J. S. Cardoso, "Automated Development of Custom Fall Detectors: Position, Model and Rate Impact in Performance," *IEEE Sens J*, vol. 20, no. 10, pp. 5465–5472, May 2020.
- [22] V. Cotechini, A. Belli, L. Palma, M. Morettini, L. Burattini, and P. Pierleoni, "A dataset for the development and optimization of fall detection algorithms based on wearable sensors," *Data Brief*, vol. 23, p. 103839, Apr. 2019.