KLASIFIKASI DATA DETEKSI JATUH MENGGUNAKAN MACHINE LEARNING DENGAN ALGORITMA ADAPTIVE BOOSTING (ADABOOST)

(FALL DETECTION DATA CLASSIFICATION USING MACHINE LEARNING WITH ADAPTIVE BOOSTING (ADABOOST) ALGORITHM)

Reza Rabbani[1], Ida Wahidah[2], Iman Hedi Santoso[3]
[1,2,3] Universitas Telkom, Bandung
[1]rezarabbaanii@student.telkomuniversity.ac.id, **[2]wahidah@telkomuniversity.ac.id**, **[3]imanhedis@telkomuniversity.ac.id**

Abstrak

Jatuh merupakan kecelakaan utama yang sering terjadi pada lanjut usia. Faktor penyebab terjadinya jatuh terdiri dari faktor intrinsik yang berhubungan dengan kondisi kesehatan dan faktor ekstrinsik yang berhubungan dengan kondisi lingkungan. Pada penelitian ini akan dilakukan klasifikasi yang terdiri dari 8 indikasi yaitu berdiri normal, duduk normal, tidur normal, naik tangga, turun tangga, jatuh kedepan, jatuh kebelakang dan jatuh kesamping. Mikrokontroler berupa Arduino serta sensor MPU-6050 sebagai accelerometer. Data yang didapat diklasifikasikan menggunakan Orange Data Mining. Metode klasifikasi yang digunakan yaitu Algoritma Adaptive Boosting (AdaBoost). AdaBoost merupakan metode boosting yang mampu menyeimbangkan kelas dengan memberikan bobot pada tingkat error klasifikasi. Data diklasifikasi pada 5 kondisi perbandingan rasio antara data training dan data testing yaitu 10%:90%, 20%:80%, 30%:70%, 40%:60% dan 50%:50%. Hasil klasifikasi berupa analisis performansi yang akan dibandingkan dengan 2 algoritma ensemble method berbasis metode tree lainnya yaitu Random Forest (RF) dan Gradient Boosting (GB). Hasil analisis perbandingan menggunakan AdaBoost didapatkan akurasi terbaik sebesar 100% pada rasio 50%:50%. Hasil perbandingan 3 klasifikasi antara AdaBoost, RF dan GB didapatkan hasil Adaboost sebagai model yang terbaik dengan nilai akurasi tertinggi di 4 rasio yaitu 97,5% pada rasio 20%:80%, 98,7% pada rasio 30%:70%, 99,3% pada rasio 40%:60% dan 100% pada rasio 50%:50%.

Kata kunci : Deteksi jatuh, MPU-6050, Klasifikasi, Machine Learning, Orange Data Mining, AdaBoost.

Abstract

Falls are the main accidents that often occur in the elderly. Factors causing falls consist of intrinsic factors related to health conditions and extrinsic factors related to environmental conditions. In this study, a classification will be carried out consisting of 8 indications, namely normal standing, normal sitting, normal sleep, climbing stairs, descending stairs, falling forward, falling backwards and falling sideways. The microcontroller is an Arduino and the MPU-6050 sensor as an accelerometer. The data obtained are classified using Orange Data Mining. The classification method used is the Adaptive Boosting Algorithm (AdaBoost). AdaBoost is a boosting method that is able to balance the class by giving weight to the classification error level. The data is classified in 5 conditions, the ratio between training data and testing data is 10%: 90%, 20%: 80%, 30%: 70%, 40%: 60% and 50%: 50%. The results of the classification are in the form of performance analysis which will be compared with 2 other tree-based ensemble method algorithms, namely Random Forest (RF) and Gradient Boosting (GB). The results of the comparative analysis using AdaBoost obtained the best accuracy of 100% at a ratio of 50%:50%. The results of the comparison of 3 classifications between AdaBoost, RF and GB showed that Adaboost was the best model with the highest accuracy value in 4 ratios, 97.5% at a ratio of 20%:80%, 98.7% at a ratio of 30%:70%, 99 ,3% at a ratio of 40%:60% and 100% at a ratio of 50%:50%.

Keywords: Fall detection, MPU-6050, Classification, Machine Learning, Orange Data Mining, AdaBoost.

1. Pendahuluan

Jatuh merupakan resiko kesehatan lanjut usia utama diseluruh dunia. Jatuh tanpa adanya bantuan bisa menyebabkan gangguan fungsional pada lanjut usia dan penurunan yang signifikan dalam mobilitas, kemandirian, dan kualitas hidup[1]. Menurut Direktur *Institute on Aging, University of Florida*, Marco Pahor dalam tajuk rencana menyatakan bahwa satu dari tiga orang usia 65 tahun atau lebih jatuh setiap tahun[2].

Lanjut usia dapat jatuh disebabkan oleh dua faktor yaitu faktor intrinsik dan ekstrinsik. Faktor intrinsik adalah faktor dari dalam yang berhubungan dengan kondisi kesehatan seseorang seperti gangguan gaya berjalan, kekakuan sendi, kelemahan ekstremitas bawah dan lain-lain. Faktor ekstrinsik adalah faktor dari luar yang berhubungan dengan lingkungan sekitar seperti cahaya ruangan yang kurang terang, lantai licin, tempat berpegangan yang kurang kuat dan lain-lain[3]. Dampak jatuh oleh lansia diantaranya bisa menyebabkan gangguan kondisi kesehatan yang serius seperti dehidrasi, pneumonia, dan hiportemia yang dapat menyebabkan kematian dalam 6 bulan setelah jatuh. Dalam kondisi jatuh yang dialami korban akan memiliki dampak yang berbeda berdasarkan posisi jatuh yang dialami, apakah itu jatuh kedepan, kebelakang atau kesamping. Maka dari itu, diperlukan untuk mengetahui indikasi jatuh yang akan dialami oleh korban.

Pada penelitian ini akan dilakukan klasifikasi deteksi jatuh dan tidak jatuh. Untuk deteksi jatuh terdiri dari 3 indikasi yaitu jatuh ke depan, ke belakang dan ke samping. Untuk deteksi tidak jatuh terdiri dari 5 indikasi yaitu berdiri normal, tidur normal, duduk normal, naik tangga dan turun tangga. Untuk merekam data kejadian menggunakan Arduino sebagai pusat kontrol dan MPU-6050 sebagai sensor *accelerometer* dan *gyroscope*. Kemudian akan dilakukan klasifikasi dengan menggunakan *software* dengan jenis *Machine Learning*. *Machine Learning* terdiri dari perancangan algoritma prediksi yang akurat dan efisien dimana telah digunakan untuk beberapa standar tugas yang telah dipelajari secara luas, salah satu diantaranya yaitu klasifikasi[4]. Salah satu *software* yang akan digunakan sebagai alat pengklasifikasian yaitu *Orange Data Mining*. *Orange Data Mining* merupakan sebuah *software* pembelajaran mesin terarah yang dapat digunakan untuk eksplorasi analisis data dan visualisasi yang diberikan untuk seleksi, prediksi dan sistem rekomendasi dari suatu percobaan[5]. Pada tahun 2017, Singh Rawat [6] melakukan perbandingan dari beberapa *software* data *mining tool* dimana *Orange Data Mining* memiliki 3 karakteristik yang dibutuhkan dalam penelitian ini yaitu *machine learning*, *data mining* dan visualisasi data yang independen dibandingkan dengan beberapa *software mining tool* lainnya.

Pada klasifikasi kali ini menggunakan algoritma *Adaptive Boosting* (*AdaBoost*) sebagai metode klasifikasi yang digunakan. *AdaBoost* merupakan *ensemble learning* dengan metode *boosting* yang mampu menyeimbangkan kelas dengan memberikan bobot pada tingkat eror klasifikasi yang dapat merubah distribusi data[7]. Pada penelitian ini bertujuan untuk mengklasifikasi jatuh dan tidak jatuh dengan 8 indikasi yang ditentukan dengan melakukan analisis performansi akurasi pada klasifikasi. Pada tahun 2020, Arman Zharmagambetov [8] melakukan penelitian perbandingan dengan 3 algoritma klasifikasi yaitu AdaBoost, *Random Forest* (RF) dan *Gradient Boosting* (GB). Algoritma tersebut digunakan sebagai perbandingan karena memiliki persamaan yaitu algoritma klasifikasi *ensemble method* berbasis metode *tree*. Berdasarkan referensi, penelitian ini akan menggunakan algoritma klasifikasi tersebut sebagai metode perbandingan dengan *AdaBoost*.

2. Konsep Dasar

Pada bagian ini, akan dijelaskan konsep dasar dan tinjauan pustaka mengenai Arduino, MPU-6050, *Machine Learning*, *Orange Data Mining*, *Adaptive Boosting* (*AdaBoost*), *Random Forest* (RF), *Gradient Boosting* (GB) dan *Confusion Matrix*.

2.1. Arduino

Arduino adalah sebuah *platform open-source* yang digunakan untuk membangun dan pemograman elektronik. Arduino dapat menerima dan mengirimkan informasi ke banyak perangkat bahkan melalui internet dapat memerintah perangkat elektronik tertentu. Ini menggunakan perangkat keras yang disebut Arduino Uno berbentuk *circuit board* dan perangkat lunak berupa pemograman *Simplified C++*. Pada zaman modern ini, Arduino banyak digunakan sebagai mikrokontroler karena penggunaan yang mudah dari pada mikrokontroler lainnya. Arduino memiliki *chip* yang dapat diprogram untuk melakukan sejumlah tugas yang dimana informasi akan dikirim dari program komputer ke mikrokontroler dengan beberapa sirkuit untuk melaksanakan perintah yang spesifik[9] .

Gambar 1. Arduino

2.2. MPU-6050

MPU-6050 adalah modul sensor yang terdapat dua fungsi di dalamnya yaitu, *Accelerometer* dengan *Micro-Electromechanical System* (MEMS) dan *Gyroscope* dengan *Micro-Electromechanical System* (MEMS) dalam sebuah *chip*. Terdapat 16 pin analog yang dilakukan pengkonversian terlebih dahulu untuk menentukan sumbu, sehingga sensor ini dapat bekerja dengan maksimal. Nilai dari sumbu X, Y, dan Z pada sensor ini dapat diambil secara bersamaan dalam satu waktu. Sensor ini menggunakan *Inter Integrated Circuit* (*interface 12C-bus*) sebagai koneksi antara sensor dan arduino[10].

Gambar 2. Sensor MPU-6050

2.3. *Machine Learning*

Machine Learning secara luas dapat didefinisikan sebagai metode perhitungan menggunakan pengalaman untuk meningkatkan kinerja atau membuat prediksi yang akurat. Pengalaman disini merujuk pada informasi masa lalu yang biasanya bentuk data elektronik yang dikumpulkan dan disediakan untuk dianalisis. *Machine Learning* telah digunakan untuk beberapa standar tugas yang telah dipelajari secara luas, diantaranya *classification* (klasifikasi), *regression* (regresi), *ranking* (peringkat), *clustering* (pengelompokan), dan *dimensionality reduction* atau *manifold learning*. Klasifikasi merukapan permasalahan penentuan kategori untuk suatu data. Misalnya klasifikasi dokumen yang terdiri dari dokumen politik, bisnis, olahraga, dan lain-lain. Klasifikasi dalam jumlah sub yang besar tentu akan menyulitkan pembagian klasifikasi tersebut, maka dibutuhkan *Machine Learning* untuk mempermudah pengklasifikasian tersebut[11].

2.4. Orange Data Mining

Orange Data Mining merupakan alat pembelajaran mesin terarah dan perangkat lunak *data mining* yang ditulis dengan bahasa *Python*. *Orange Data Mining* berbasis komponen perangkat lunak pemrograman visual yang terdiri dari visualisasi data, *machine learning*, *data mining* dan analisis data. Program ini dibuat dan dikembangkan oleh Laboratorium Bioinformatika, Fakultas Komputer dan Informasi Sains di Universitas Lljubjana[12].

Gambar 3. Tampilan Awal *Orange Data Mining*

2.5. *Adaptive Boosting* (*AdaBoost*)

Algoritma AdaBoost menggunakan metode boosting untuk membangun klasifikasi gabungan. Padaalgoritma ini, pentingnya dasar pengklasifikasian tergantung dari *error rate*. Pada dasarnya, Algoritma *AdaBoost* memiliki 3 tahap. Tahap pertama memulai pendistribusian bobot data pelatihan. Jika dimiliki nilai sampel N, pada awalnya setiap sampel pelatihan akan diberikan bobot yang sama. Tahap kedua yaitu melatih pengelompokan dasar. Dalam proses yang spesifik, jika sebuah sampel telah diklasifikasikan dengan akurat, maka bobot akan dikurangi pada set yang akan dibangun berikutnya. Jika sampel diklasifikasikan belum akurat, maka bobot akan ditingkatkan pada set yang akan dibangun berikutnya. Set berikutnya digunakan untuk melatih pengelompokan berikutnya. Pada tahap terakhir yaitu menggabungkan *weak classifiers* yang diperoleh dari setiap pelatihan menjadi *strong classifier*. Setelah melatih tiap golongan yang lemah, meningkatkan bobot pada

weak classifier dengan klasifikasi *error rate* yang kecil akan berperan lebih baik pada klasifikasi akhir. Dengan kata lain, *weak classifier* dengan nilai *error rate* yang kecil akan memiliki bobot yang besar pada pengklasifikasian akhir, begitu pula sebaliknya[13].

2.6. *Random Forest* (RF)

RF merupakan metode *ensemble learning* yang diusulkan pertama kali oleh Breiman pada tahun 2001 yang terdiri dari kombinasi dari pohon klasifikasi sehingga pada setiap pohon bergantung pada nilai acak vektor sampel secara mandiri dengan distribusi yang sama untuk semua pohon[14]. Langkah-langkah pada RF terdiri dari 3 tahap. Tahap pertama yaitu menghasilkan sampel acak, setiap sub pohon keputusan harus memasukkan set pelatihan acak yang akan diambil sampelnya dari kumpulan data asli. Tahap kedua yaitu membangun sub pohon keputusan, RF terdiri dari beberapa pohon keputusan yang dapat tumbuh sempurna tanpa pemangkasan karena batas atas teoritis dalam kesalahan generalisasi. Tahap terakhir yaitu mensintesis hasil klasifikasi, setiap sub pohon akan menghasilkan hasil yang berbeda-beda. Oleh karena itu mekanisme *voting* diperlukan untuk membuat keputusan akhir[15].

2.7 *Gradient Boosting* (GB)

GB menghasilkan model prediksi dalam bentuk *ensemble* dari model prediksi yang lemah. Ini melibatkan 3 elemen seperti *loss function*, *weak learner* dan *additive model*. Rangka pada GB merupakan kerangka kerja umum dimana setiap *loss function* yang berbeda-beda dapat digunakan. Pada GB *weak learner* sama dengan pohon keputusan. Untuk meminimalkan biaya, pohon dirakit dengan cara metode *spitting*. Tujuan dari peningkatan prosedur gradien adalah untuk meminimalkan kerugian saat menambahkan pohon. Di dalam prosedur, kerugian dihitung di setiap langkah dan setelah menghitung kerugian, pohon yang mengurangi kerugian harus ditambahkan dan bobot diperbarui[16].

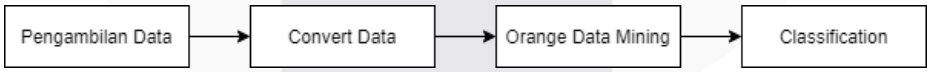
2.8 *Confusion Matrix*

Confusion Matrix merupakan sebuah matriks yang menampilkan visualisasi kinerja dari algoritma klasifikasi menggunakan data dalam matriks. Hal tersebut membandingkan klasifikasi prediksi terhadap klasifikasi *actual* dalam bentuk *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN). TP adalah jumlah poin data yang benar diklasifikasikan dari kelas positif. FP adalah jumlah poin data yang diperkirakan berada di kelas positif tetapi sebenarnya milik kelas negatif. TN adalah jumlah poin data yang benar diklasifikasikan dari kelas negatif. FN adalah jumlah poin yang diperkirakan berada di kelas negatif tetapi sebenarnya milik kelas positif[4].

Gambar 4. *Confusion Matrix*

3. Desain Sistem

Pada penelitian ini akan dilakukan klasifikasi data deteksi jatuh pada sensor MPU-6050 menggunakan *Orange Data Mining* dengan algoritma *Adaptive Boosting* (*AdaBoost*). Tahapan yang perlu dilakukan yaitu pengambilan data, *convert* data, mengatur *workflow* para *Orange Data Mining* dan klasifikasi dengan metode *AdaBoost.* Desain sistem dapat dilihat pada Gambar 5.



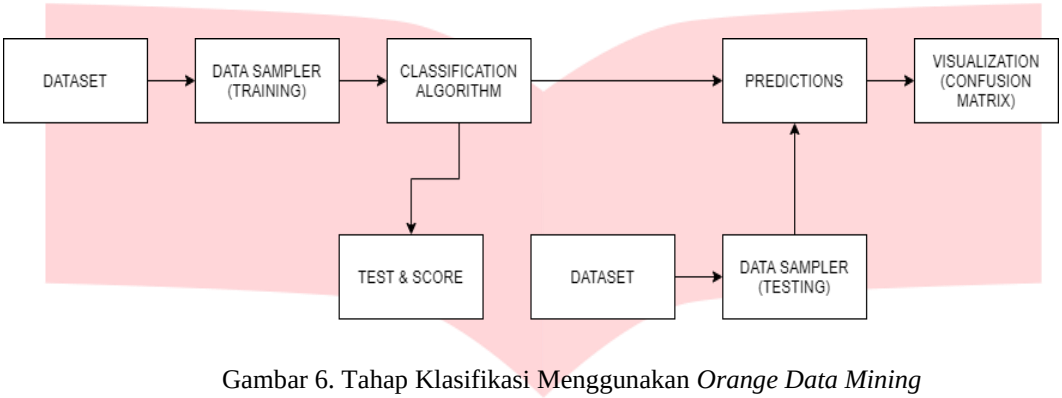
Gambar 5. Desain Sistem

Pada pengambilan data, sensor MPU-6050 akan dimasukkan ke dalam tas yang dimana akan ditempelkan di dalam sebuah boks kecil. Boks kecil tersebut akan ditempelkan di dalam tas agar sensor berada diposisi stabil. Kemudian dilakukan pengambilan data dengan 8 indikasi yaitu berdiri normal, duduk normal, tidur normal, naik tangga, turun tangga, jatuh ke depan, jatuh ke belakang dan jatuh ke samping dengan 3 subjek yang berbeda.

Setelah pengambilan data, data akan diubah ke dalam *Microsoft Excel* menggunakan PLX-DAQ tanpa dilakukannya *pre-processing*. *Pre-Processing* tidak dilakukan karena hasil performansi yang didapat nantinya sudah cukup baik

3.1 Klasifikasi Menggunakan *Orange Data Mining*

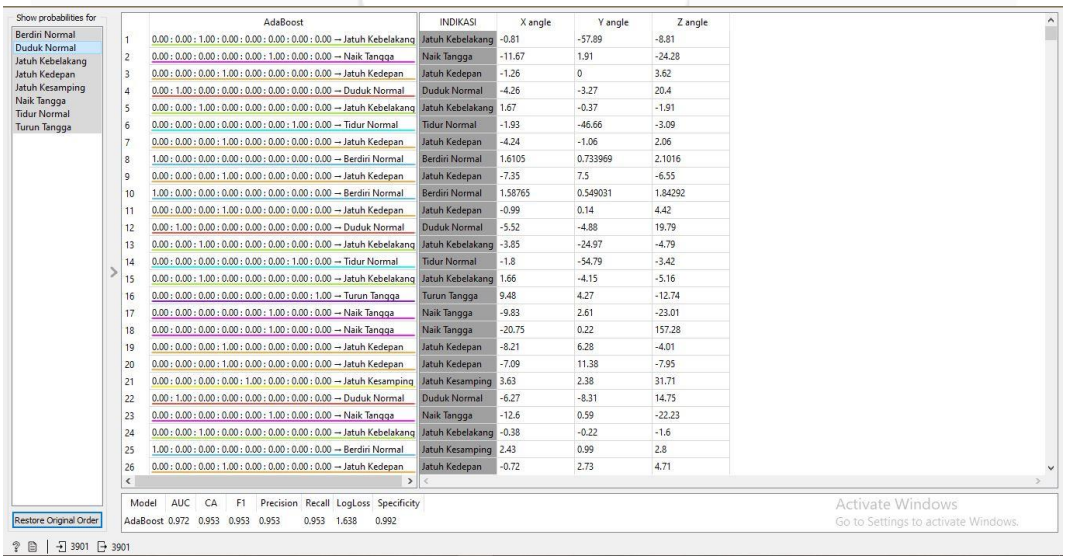
Perancangan *software* pada penelitian ini menggunakan *Orange Data Mining* sebagai *Machine Learning* untuk klasifikasi data untuk melihat hasil performansi akurasi yang didapat menggunakan algoritma *AdaBoost*.



Gambar 6. Tahap Klasifikasi Menggunakan *Orange Data Mining*

Gambar 6. Tahap Klasifikasi Menggunakan *Orange Data Mining*Gambar 6 menjelaskan tentang tahap klasifikasi menggunakan *Orange Data Mining*. Untuk melakukan tahap klasifikasi, akan dilakukan pengaturan setiap *widget* pada *workflow* yang telah disediakan. Dataset terdiri nilai *angle* pada *accelerometer* yaitu sumbu X, Y dan Z. Dataset kemudian akan disampel untuk membagi data *training* dan data *testing* untuk prediksi. Salah satu metode sampel yang digunakan yaitu *fixed proportion*. Penggunaan *fixed proportion* didasari dengan tujuan penelitian ini yaitu melakukan perbandingan berdasarkan rasio antara data *training* dan data *testing*. Terdapat 5 perbandingan rasio yang digunakan dengan porsi data *training* : porsi data *testing* yaitu 10%:90%, 20%:80%, 30%:70%, 40%:60% dan 50%:50%. Penggunaan *fixed proportion* menjadi metode yang sesuai dimana 3 metode lainnya *fixed sample*, *cross validation* dan *bootstrap* tidak melakukan *sampling* berdasarkan rasio. Terdapat penelitian yang dilakukan oleh Yoga Pristyanto pada tahun 2019 [17] dimana proses pengujian menggunakan metode *sampling* yang serupa dengan membagi proporsi antara data *training* dan data *testing*. Kemudian dihubungkan ke algoritma klasifikasi yang digunakan yaitu *AdaBoost*. Terdapat beberapa parameter yang bisa ditentukan. Berdasarkan percobaan dan referensi, nilai setiap parameter yang digunakan yaitu *number of estimators* = 50, *learning rate* = 1 dan *boosting method* menggunakan SAMME.R (*Stagewise Additive Modeling using a Multi-class Exponential loss function)*[18].

Untuk melihat performansi pada data *training* menggunakan *widget Test & Score*. Setelah dilakukan penentuan rasio antara data *training* dan data *testing*, *widget predictions* akan digunakan sebagai hasil klasifikasi yang didapat dan akan divisualisasikan menggunakan *confusion matrix*. Tahap serupa akan dilakukan pada 2 metode klasifikasi lainnya yaitu *Random Forest* dan *Gradient Boosting* sebagai analisis metode perbandingan



Show probabilities for
Berdiri Normal
Duduk Normal
Jatuh Kebelakang
Jatuh Kedepan
Jatuh Kesamping
Naik Tangga
Tidur Normal
Turun Tangga

	AdaBoost	INDIKASI	X angle	Y angle	Z angle
1	0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kebelakang	Jatuh Kebelakang	-0.81	-57.89	-8.81
2	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 → Naik Tangga	Naik Tangga	-11.67	1.91	-24.28
3	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-1.26	0	3.62
4	0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Duduk Normal	Duduk Normal	-4.26	-3.27	20.4
5	0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kebelakang	Jatuh Kebelakang	1.67	-0.37	-1.91
6	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → Tidur Normal	Tidur Normal	-1.93	-46.66	-3.09
7	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-4.24	-1.06	2.06
8	1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Berdiri Normal	Berdiri Normal	1.6105	0.733969	2.1016
9	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-7.35	7.5	-6.55
10	1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Berdiri Normal	Berdiri Normal	1.58765	0.549031	1.84292
11	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-0.99	0.14	4.42
12	0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Duduk Normal	Duduk Normal	-5.52	-4.88	19.79
13	0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kebelakang	Jatuh Kebelakang	-3.85	-24.97	-4.79
14	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 → Tidur Normal	Tidur Normal	-1.8	-54.79	-3.42
15	0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kebelakang	Jatuh Kebelakang	1.66	-4.15	-5.16
16	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 → Turun Tangga	Turun Tangga	9.48	4.27	-12.74
17	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 → Naik Tangga	Naik Tangga	-9.83	2.61	-23.01
18	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 → Naik Tangga	Naik Tangga	-20.75	0.22	157.28
19	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-8.21	6.28	-4.01
20	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-7.09	11.38	-7.95
21	0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 → Jatuh Kesamping	Jatuh Kesamping	3.63	2.38	31.71
22	0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Duduk Normal	Duduk Normal	-6.27	-8.31	14.75
23	0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 → Naik Tangga	Naik Tangga	-12.6	0.59	-22.23
24	0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kebelakang	Jatuh Kebelakang	-0.38	-0.22	-1.6
25	1.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 : 0.00 → Berdiri Normal	Jatuh Kesamping	2.43	0.99	2.8
26	0.00 : 0.00 : 0.00 : 1.00 : 0.00 : 0.00 : 0.00 : 0.00 → Jatuh Kedepan	Jatuh Kedepan	-0.72	2.73	4.71

Model	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
AdaBoost	0.972	0.953	0.953	0.953	0.953	1.638	0.992

Restore Original Order

Activate Windows
Go to Settings to activate Windows.

Gambar 7. *Predictions*

3.2 Analisis Performansi

Kinerja algoritma klasifikasi pada *Orange Data Mining* terdiri dari 7 parameter yaitu *Area Under* ROC (*Receiver Operating Characteristic*) *Curve* (AUC), *Classification Acuracy* (CA), F1, *Precision*, *Recall*, *LogLoss* dan *Specificity*.

$$CA = \frac{TP + FN}{TP + TN + FN + FP}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Dimana TP = *True Positives*, TN = *True Negatives*, FP = *False Positives*, FN = *False Negatives*, N = Jumlah sampel, p_i = Probabilitas i^{th} sampel milik *class* dan y_i = Label sebenarnya dari i^{th}.

4. Pengujian dan Analisa Sistem

Bagian ini berisi hasil analisa sistem saat melakukan pengujian pada klasifikasi menggunakan *Orange Data Mining* dengan *AdaBoost* sebagai algoritma pengklasifikasian.

4.1. Tahapan Pengujian

Adapun tahapan pengujian sistem sebagai berikut:

1. Pengambilan dataset
   Pengambilan data dilakukan di Indekos Nasta, Bandung. Percobaan dilakukan selama kurang lebih 10 detik setiap 8 indikasi yaitu berdiri normal, duduk normal, tidur normal, naik tangga, turun tangga, jatuh ke depan, jatuh ke belakang dan jatuh ke samping. Terdiri dari 3 subjek laki-laki dengan tinggi badan yang berbeda. Subjek 1 dengan tinggi badan 178 cm, subjek 2 dengan tinggi badan 165 cm dan subjek 3 dengan tinggi badan 170 cm.
2. Proses pengaturan pada *Orange Data Mining*
   Data yang telah didapat akan digunakan sebagai dataset dengan mengatur setiap *widget* yang dibutuhkan. Pengaturan tersebut terdiri dari input dataset, pembagian sampel data sebagai data *training* dan data *testing*, input algoritma yang ingin digunakan dan divisualisasikan menggunakan *confusion matrix*.
3. Perbandingan
   Pada penelitian ini akan dilakukan analisis perbandingan hasil analisis performansi akurasi yang didapat pada *AdaBoost* dengan dua model klasifikasi lainnya yaitu *Random Forest* dan *Gradient Boosting*.

Gambar 8. Proses Pengambilan Data

5. Data Hasil Pengujian Sistem

Pada bagian ini akan membahas hasil klasifikasi menggunakan algoritma *AdaBoost* dengan visualisasi *confusion matrix* setiap 5 kondisi perbandingan rasio antara data *training* dan data *testing* yaitu 10%:90%, 20%:80%, 30%:70%, 40%:60% dan 50%:50%. Untuk keterangan pada tabel dimana I = Indikasi, BN = Berdiri Normal, DN = Duduk Normal, JKeb = Jatuh Ke belakang, JKed = Jatuh Ke depan, JKes = Jatuh Ke samping, NT = Naik Tangga, TN = Tidur Normal dan TT = Turun Tangga. Terdapat 2 warna pada tabel yang membedakan dimana kolom berwarna biru memiliki arti klasifikasi yang benar (*classified)* sedangkan kolom berwarna merah memiliki arti salah klasifikasi (*misclassified*).

Tabel 1. *Confusion Matrix* Dengan Rasio 10%:90%

PREDICTED

ACTUAL	I	BN	DN	JKeb	JKed	JKes	NT	TN	TT	Σ
	BN	**339**	0	**8**	**1**	**6**	0	0	0	**354**
	DN	0	**480**	0	0	**1**	0	0	0	**481**
	JKeb	0	0	**457**	**22**	**7**	0	**12**	**6**	**504**
	JKed	0	0	**20**	**922**	0	0	0	**2**	**944**
	JKes	**38**	0	**10**	0	**437**	0	0	**5**	**490**
	NT	0	**5**	**4**	**9**	0	**361**	0	**5**	**384**
	TN	0	0	**10**	0	0	0	**344**	0	**354**
	TT	0	0	**7**	**2**	**2**	**3**	0	**376**	**390**
	Σ	**377**	**485**	**516**	**956**	**453**	**364**	**356**	**394**	**3901**

Pada Tabel 1 dapat dilihat bahwa dalam data prediksi yang berjumlah 3901 data, banyak data *classified* berjumlah 3716 data dan banyak data *misclassified* berjumlah 185 data.

\

Tabel 2. *Confusion Matrix* Dengan Rasio 20%:80%

ACTUAL \ PREDICTED	BN	DN	JKeb	JKed	JKes	NT	TN	TT	Σ
BN	307	0	0	1	7	0	0	0	315
DN	0	426	0	0	1	0	0	0	427
JKeb	0	0	421	8	0	0	11	8	448
JKed	2	0	1	834	0	2	0	0	839
JKes	26	0	0	0	410	0	0	0	436
NT	0	4	0	3	0	332	0	3	342
TN	0	0	2	0	0	0	312	0	314
TT	0	0	5	2	0	2	0	338	347
Σ	**335**	**430**	**429**	**848**	**418**	**336**	**323**	**349**	**3468**

Pada Tabel 2 dapat dilihat bahwa dalam data prediksi yang berjumlah 3468 data, banyak data *classified* berjumlah 3380 data dan banyak data *misclassified* berjumlah 88 data.

Tabel 3. *Confusion Matrix* Dengan Rasio 30%:70%

ACTUAL \ PREDICTED	BN	DN	JKeb	JKed	JKes	NT	TN	TT	Σ
BN	273	0	0	2	0	0	0	0	275
DN	0	370	3	0	1	0	0	0	374
JKeb	0	0	388	1	0	0	0	3	392
JKed	0	0	1	730	0	3	0	0	734
JKes	15	0	0	0	367	0	0	0	382
NT	0	1	0	1	0	295	0	2	299
TN	0	0	0	0	0	0	275	0	275
TT	0	0	4	2	0	1	0	296	303
Σ	**288**	**371**	**396**	**736**	**368**	**299**	**275**	**301**	**3034**

Pada Tabel 3 dapat dilihat bahwa dalam data prediksi yang berjumlah 3034 data, banyak data *classified* berjumlah 2994 data dan banyak data *misclassified* berjumlah 40 data.

Tabel 4. *Confusion Matrix* Dengan Rasio 40%:60%

ACTUAL \ PREDICTED	BN	DN	JKeb	JKed	JKes	NT	TN	TT	Σ
BN	235	0	0	1	0	0	0	0	236
DN	0	319	0	0	1	0	0	0	320
JKeb	0	0	333	0	0	0	0	3	336
JKed	0	0	0	630	0	0	0	0	630
JKes	7	0	0	0	320	0	0	0	327
NT	0	2	0	1	0	251	0	2	256
TN	0	0	0	0	0	0	236	0	236
TT	0	0	2	0	0	0	0	258	260
Σ	**242**	**321**	**335**	**632**	**321**	**251**	**236**	**263**	**2601**

Pada Tabel 4 dapat dilihat bahwa dalam data prediksi yang berjumlah 2601 data, banyak data *classified* berjumlah 2581 data dan banyak data *misclassified* berjumlah 20 data.

Tabel 5. *Confusion Matrix* Dengan Rasio 50%:50%

ACTUAL \ PREDICTED	BN	DN	JKeb	JKed	JKes	NT	TN	TT	Σ
BN	**196**	0	0	0	0	0	0	0	**196**
DN	0	**267**	0	0	0	0	0	0	**267**
JKeb	0	0	**280**	0	0	0	0	0	**280**
JKed	0	0	0	**524**	0	0	0	0	**524**
JKes	0	0	0	0	**273**	0	0	0	**273**
NT	0	0	0	0	0	**214**	0	0	**214**
TN	0	0	0	0	0	0	**197**	0	**197**
TT	0	0	0	0	0	0	0	**216**	**216**
Σ	**196**	**267**	**280**	**524**	**273**	**214**	**197**	**216**	**2167**

Pada Tabel 5 dapat dilihat bahwa dalam data prediksi yang berjumlah 2167 data, banyak data *classified* berjumlah 2167 data dan banyak data *misclassified* berjumlah 0 data. Berdasarkan 5 tabel diatas dapat dilihat bahwa terbentuknya pola dimana jika rasio pada data *training* semakin besar, maka data *misclassified* akan semakin menurun, begitu juga sebaliknya. Dari hasil uji klasifikasi setiap 5 kondisi rasio, terdapat 7 parameter performansi yang dihasilkan yaitu *Area Under* ROC (*Receiver Operating Characteristic*) *Curve* (AUC), *Classification Acuracy* (CA), F1, *Precision*, *Recall*, *LogLoss* dan *Specificity*. Terdapat keterangan pada rasio dimana TR = Data *Training* dan TS = Data *Testing*. Hasil performansi dapat dilihat di Gambar 6.

Rasio	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
10%TR:90%TS	0.972	0.953	0.953	0.953	0.953	1.638	0.992
20%TR:80%TS	0.985	0.975	0.975	0.975	0.975	0.876	0.996
30%TR:70%TS	0.992	0.987	0.987	0.987	0.987	0.455	0.998
40%TR:60%TS	0.996	0.993	0.993	0.993	0.993	0.252	0.999
50%TR:50%TS	1.000	1.000	1.000	1.000	1.000	0.000	1.000

Gambar 9. Hasil Performansi menggunakan *AdaBoost*

Dari Gambar 6 dapat setiap kenaikan rasio pada data *training*, maka hasil performansi setiap 7 parameter akan meningkat. Dimana nilai AUC, CA, F1, *Precision*, *Recall* dan *Specificity* mengalami peningkatan nilai berdasarkan kenaikan rasio pada data *training*. Sedangkan untuk nilai *LogLoss* mengalami penurunan yang dimana semakin rendah nilai *LogLoss* maka performansi akan semakin baik.

Rasio	Model	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
	AdaBoost	0.972	0.953	0.953	0.953	0.953	1.638	0.992
10%TR:90%TS	RF	0.993	0.961	0.951	0.949	0.952	0.524	0.992
	GB	0.988	0.935	0.935	0.938	0.935	0.400	0.989
Rasio	**Model**	**AUC**	**CA**	**F1**	**Precision**	**Recall**	**LogLoss**	**Specificity**
	AdaBoost	0.985	0.975	0.975	0.975	0.975	0.876	0.996
20%TR:80%TS	RF	0.997	0.971	0.971	0.971	0.971	0.259	0.996
	GB	0.997	0.963	0.963	0.964	0.963	0.159	0.994
Rasio	**Model**	**AUC**	**CA**	**F1**	**Precision**	**Recall**	**LogLoss**	**Specificity**
	AdaBoost	0.992	0.987	0.987	0.987	0.987	0.455	0.998
30%TR:70%TS	RF	0.998	0.981	0.981	0.981	0.981	0.188	0.997
	GB	0.994	0.967	0.967	0.967	0.967	0.229	0.996
Rasio	**Model**	**AUC**	**CA**	**F1**	**Precision**	**Recall**	**LogLoss**	**Specificity**
	AdaBoost	0.996	0.993	0.993	0.993	0.993	0.252	0.999
40%TR:60%TS	RF	0.999	0.991	0.991	0.991	0.991	0.115	0.999
	GB	0.894	0.885	0.883	0.889	0.885	3.555	0.982
Rasio	**Model**	**AUC**	**CA**	**F1**	**Precision**	**Recall**	**LogLoss**	**Specificity**
	AdaBoost	1.000	1.000	1.000	1.000	1.000	0.000	1.000
50%TR:50%TS	RF	1.000	0.996	0.996	0.996	0.996	0.040	1.000
	GB	0.886	0.880	0.861	0.885	0.880	3.637	0.984

Gambar 10. Hasil Perbandingan

Penentuan performansi dilihat berdasarkan nilai dari 4 parameter utama yaitu CA, *Precision*, *Recall* dan *LogLoss*. Pada model *AdaBoost* didapatkan hasil performansi tertinggi di 4 rasio yaitu pada rasio 20%TR:80%TS, 30%TR:70%TS, 40%:60% dan 50%:50%. Pada rasio 20%:80%, didapatkan hasil CA = 0,971 , *Precision* = 0,975, *Recall* = 0,975 dan *LogLoss* = 0.876. Pada rasio 30%:70%, didapatkan hasil CA = 0,987 , *Precision* = 0,987, *Recall* = 0,987 dan *LogLoss* = 0,455. Pada rasio 40%:60%, didapatkan hasil CA = 0,993 , *Precision* = 0,993, *Recall* = 0,993 dan *LogLoss* = 0,252. Pada rasio 50%:50%, didapatkan hasil CA = 1,000 , *Precision* = 1,000, *Recall* = 1,000 dan *LogLoss* = 0,000. Walaupun *AdaBoost* selalu memiliki nilai performansi lebih baik di 3 paramter yaitu CA, *Precision* dan *Recall*, pada parameter *LogLoss*, *AdaBoost* memiliki nilai performansi yang tidak lebih baik diantara RF dan GB dikarenakan nilai *LogLoss* yang tinggi. Pada rasio 10%TR:90%TS didapatkan hasil nilai performansi RF lebih baik diantara 2 model karena 2 alasan. Alasan yang pertama yaitu konsep dasar model RF bekerja. RF merupakan kumpulan algoritma pada *decision tree* yang dimana setiap *decision tree* pada RF menginputkan set pelatihan acak dari kumpulan data original[15]. Input set pelatihan secara acak ini yang mendasari setiap percobaan, nilai performansi pada RF selalu menghasilkan nilai yang berbeda-beda. Alasan yang kedua yaitu model RF selalu menghasilkan nilai performansi yang berbeda-beda disetiap 5 kali percobaan yang telah ditentukan. Berbeda dengan *AdaBoost* dan GB yang selalu menghasilkan nilai performansi yang sama disetiap percobaan.

6. Kesimpulan

Dari analisis dan percobaan yang telah dilakukan pada sensor MPU-6050 sebagai pendeteksi jatuh dan *Orange Data Mining* sebagai perangkat lunak klasifikasi menggunakan Algoritma *Adaptive Boosting* (*AdaBoost*) didapatkan hasil Dari 5 rasio yang telah ditentukan, perbandingan 50% data *training* (TR) dan 50% data *testing* (TS) merupakan rasio dengan nilai performansi akurasi yang paling tinggi yaitu 100% dan *AdaBoost* merupakan model klasifikasi terbaik diantara dua model klasifikasi lainnya yaitu *Random Forest* dan *Gradient Boosting* dengan nilai performansi Akurasi tertinggi di 4 rasio yaitu 97,5% pada rasio 20%TR:80%TS, 98,7% pada rasio 30%TR:70%TS, 99,3% pada rasio 40%TR:60%TS dan 100% pada rasio 50%TR:50%TS.

7. Referensi

[1] D. Yacchirema, J. S. De Puga, C. Palau, and M. Esteve, “Fall detection system for elderly people using IoT and Big Data,” *Procedia Comput. Sci.*, vol. 130, pp. 603–610, 2018, doi: 10.1016/j.procs.2018.04.110.

[2] Geriatri, “Jatuh Merupakan ‘Epidemi Utama’ Lansia,” *5 September*, 2019. https://www.geriatri.id/artikel/139/jatuh-merupakan-epidemi-utama-lansia (accessed Oct. 11, 2020).

[3] K. B. Sapata *et al.*, “Kejadian Jatuh Pada Lansia,” *Rev. Bras. Med. do Esporte*, vol. 12, no. 1, pp. 1–13, [Online]. Available: https://doi.org/10.1080/17461391.2020.1733672%0Ahttp://www.efdeportes.com/efd48/trein2.htm%0Ahttp://seer.uftm.edu.br/revistaeletronica/index.php/aces%0Ahttp://www.tandfonline.com/doi/abs/10.1080/02701367.2004.10609174%5Cnhttp://insights.ovid.com/crossref?

[4] M. Awad and R. Khanna, *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, no. April. 2015.

[5] S. Kodati and R. Vivekanandam, “Analysis of Heart Disease using in Data Mining Tools Orange and Weka Sri Satya Sai University Analysis of Heart Disease using in Data Mining Tools Orange and Weka,” *Glob. J. Comput. Sci. Technol.*, vol. 18, no. 1, 2018.

[6] K. Singh Rawat and C. Author, “Comparative Analysis of Data Mining Techniques, Tools and Machine Learning Algorithms for Efficient Data Analytics,” vol. 19, no. 4, pp. 56–61, 2017, doi: 10.9790/0661-1904035661.

[7] Y. H. Agustin, . K., and E. T. Luthfi, “Klasifikasi Penerimaan Mahasiswa Baru Menggunakan Algortima C4.5 Dan Adaboost (Studi Kasus : STMIK XYZ),” *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 9, no. 1, p. 1, 2017, doi: 10.22303/csrid.9.1.2017.1-11.

[8] M. A. Carreira-Perpiñán and A. Zharmagambetov, “Ensembles of Bagged TAO Trees Consistently Improve over Random Forests, AdaBoost and Gradient Boosting,” *FODS 2020 - Proc. 2020 ACM-IMS Found. Data Sci. Conf.*, no. section 3, pp. 35–46, 2020, doi: 10.1145/3412815.3416882.

[9] Y. A. Badamasi, “The working principle of an Arduino,” *Proc. 11th Int. Conf. Electron. Comput. Comput. ICECCO 2014*, 2014, doi: 10.1109/ICECCO.2014.6997578.

[10] A. Suprayogi and H. Fitriyah, “Sistem Pendeteksi Kecelakaan Pada Sepeda Motor Berdasarkan Kemiringan Menggunakan Sensor Gyroscope Berbasis Arduino,” *Sist. Pendeteksi Kecelakaan Pada Sepeda Mot. Berdasarkan Kemiringan Menggunakan Sens. Gyroscope Berbas. Arduino*, vol. 3, no. 3, pp. 3079–3085, 2019.

[11] A. T. Mehryar Mohri, Afshin Rostamizadeh, *Foundations of Machine Learning*. 2018.

[12] A. Naik and L. Samant, “Correlation Review of Classification Algorithm Using Data Mining Tool: WEKA, Rapidminer, Tanagra, Orange and Knime,” *Procedia Comput. Sci.*, vol. 85, no. Cms, pp. 662–

668, 2016, doi: 10.1016/j.procs.2016.05.251.

[13] R. Yao, J. Li, M. Hui, L. Bai, and Q. Wu, "Pattern recognition for partial discharge using multi-feature combination adaptive boost classification model," *IEEE Access*, vol. 9, pp. 48873–48883, 2021, doi: 10.1109/ACCESS.2021.3067009.

[14] A. Syukron and A. Subekti, "Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk Klasifikasi Penilaian Kredit," *J. Inform.*, vol. 5, no. 2, pp. 175–185, 2018, doi: 10.31311/ji.v5i2.4158.

[15] S. Xu, Z. Zhang, D. Wang, J. Hu, X. Duan, and T. Zhu, "Cardiovascular risk prediction method based on CFS subset evaluation and random forest classification framework," *2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA 2017*, pp. 228–232, 2017, doi: 10.1109/ICBDA.2017.8078813.

[16] A. Iqbal and K. Barua, "A Real-time Emotion Recognition from Speech using Gradient Boosting," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, pp. 1–5, 2019, doi: 10.1109/ECACE.2019.8679271.

[17] Y. Pristyanto, "Penerapan Metode Ensemble Untuk Meningkatkan Kinerja Algoritme Klasifikasi Pada Imbalanced Dataset," *J. Teknoinfo*, vol. 13, no. 1, p. 11, 2019, doi: 10.33365/jti.v13i1.184.

[18] A. Ferrario and R. Hämmerli, "On Boosting: Theory and Applications," *SSRN Electron. J.*, pp. 1–39, 2019, doi: 10.2139/ssrn.3402687.