

Pemilihan Fitur Statistik serta Implementasi Model Decision Tree Machine Learning Pada Arduino Nano 33 BLE Untuk Pendeteksian dan Klasifikasi Gerak Jatuh dan Kecenderungan Jatuh Lansia Berbasis Nilai Akselerasi

1st Azzahra Nadya Kahpiasa
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

kahpiasakanya@student.telkomuniversity.
@.id

2nd Istiqomah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

istiqomah@telkomuniversity.ac.id

3rd Husneni Mukhtar
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

husnenimukhtar@telkomuniversity.ac.id

Abstrak — Orang yang berusia di atas 60 tahun dianggap sudah lanjut usia. Penurunan fungsi fisiologis terkait usia, termasuk fungsi tulang dan otot, berkontribusi terhadap peningkatan risiko jatuh pada lansia. Salah satu keadaan yang bisa berakibat fatal adalah terjatuh. Kemunduran berbagai proses organ yang terlibat dalam menjaga keseimbangan tubuh dapat dipengaruhi oleh dampak ini, yang mungkin berdampak pada kematian. Orang tua banyak jatuh sekarang, namun keluarga sering tidak menyadari keberadaan mereka. Secara tradisional, tetangga di dekat rumah lansia adalah sumber utama informasi mengenai kondisi mereka. Studi sebelumnya telah menggunakan sifat statistik dan sensor inersia untuk mengenali aktivitas manusia pada orang tua. Dalam studi ini, kami akan mengevaluasi metode ekstraksi fitur dan pembelajaran mesin terbaik. Variabel maximum, minimum, mean, median, kurtosis, skewness, dan variance yang dikumpulkan dari data sensor akselerometer menggunakan sensor IMU akan diperiksa untuk ekstraksi fitur menggunakan metode Fast Fourier Transform. Digunakan Cross-validation untuk mengetahui performa mode Decision Tree. Dengan nilai akurasi 99,8% dan nilai ekstraksi ciri terbaik pada Maksimum *accelX*, *median accelZ*, *variance accelZ*, *variance Magnitude*, dan maksimum *Magnitude*.

Kata kunci—Elderly, Fall, Machine Learning, Accelerometer, Micromlgen, Kodular

I. PENDAHULUAN

Lansia merupakan tahap akhir dari siklus hidup manusia. Berdasarkan UU No. 13 tahun 1998, seseorang yang berusia lebih dari 60 tahun digolongkan sebagai lansia. Lansia dapat dibagi menjadi dua kelompok menurut mata pencaharian: lansia potensial, yaitu orang lanjut usia yang masih dapat bekerja atau berpartisipasi dalam produksi barang atau jasa, dan orang lanjut usia yang lemah yang tidak dapat lagi memenuhi kebutuhan sehari-hari, dan mereka bergantung pada orang lain untuk kebutuhan mereka.

Menurut sebuah studi UNICEF, Indonesia memiliki jumlah lansia yang tumbuh paling cepat di dunia antara tahun 1990 dan 2025 [16]. Sebagai hasil dari keberhasilan pembangunan negara, harapan hidup juga meningkat. Namun saat ini, lansia semakin kurang mendapat perhatian dari

masyarakat, terutama dalam hal kecukupan gizi yang mereka butuhkan. Selain itu, menurut Badan Pusat Statistik (BPS), akan ada 29,3 juta lansia di Indonesia pada tahun 2021. Jumlah tersebut setara dengan 10,82 persen dari total penduduk Indonesia (BPS, 2022). Jumlah ini diprediksi mencapai 33,69 juta pada tahun 2025, meningkat menjadi 40,95 juta di tahun 2030 dan 48,19 juta pada 2035 (Segita et al., 2021). Peningkatan jumlah lansia memiliki konsekuensi yang kompleks. Jatuh adalah masalah umum pada tahap ini (Mahendra, 2016). Penelitian ini ditujukan agar dapat melakukan pencegahan dini pada lansia jika terjadi insiden jatuh dan kecenderungan jatuh karena hal tersebut sangat berbahaya dan jika tidak segera dilakukan pertolongan dapat berakibat fatal hingga menyebabkan kematian.

Saat ini sudah banyak dilakukan penelitian mengenai Human Activity Recognition. Baik pada percobaan untuk variasi model machine learning, feature yang digunakan, tipe dataset yang digunakan hingga pada aplikasi penggunaan mikrokontroler yang digunakan. Salah satu contoh penelitian sebelumnya adalah pendeteksian aktivitas manusia pada lansia dengan menggunakan sensor inersia dan fitur statistik [17]. Nilai yang digunakan merupakan pembacaan nilai *akselerasi* dan *gyroscope* dari sensor IMU kemudian menggunakan metode XGBoost akan dipilih urutan model machine learning dan lima fitur ekstraksi terbaik untuk dilakukan pelatihan.

Tujuan akhir dari riset ini yakni untuk membangun suatu alat yang dapat melakukan klasifikasi gerak lansia pada mikrokontroler Arduino Nano 33 BLE berdasarkan nilai akselerasi yang diperoleh dari sensor IMU LSM9DS1 untuk lima kelas kategori gerakan meliputi jatuh, kecenderungan jatuh, duduk, berdiri dan jalan. Akan digunakan sebanyak 61,925 data akselerasi dalam proses pelatihan. Hasil dari klasifikasi juga akan ditampilkan pada antarmuka pengguna yakni aplikasi berbasis android.

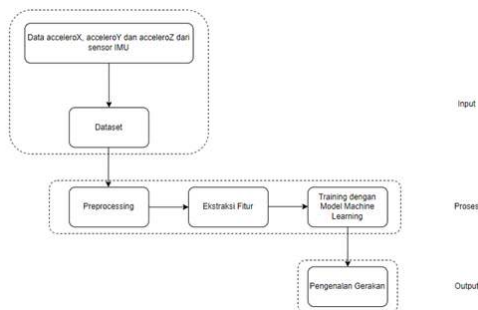
Akan dipilih fitur terbaik untuk dilakukan training sesungguhnya untuk melihat performa model Decision Tree yang kita bangun. Pemilihan fitur ekstraksi disesuaikan dengan keterbatasan memori komputasi dari mikrokontroler Arduino Nano 33 BLE yang kita gunakan. Selain itu juga

penggunaan model Decision Tree disesuaikan dengan library Micromlgen yang akan menjadi ekstensi hasil training model untuk di aplikasikan di Arduino Nano 33 BLE. Hasil percobaan fitur ekstraksi dengan model Decision Tree disajikan dalam tabel pengujian dan confusion matrix.

II. KAJIAN TEORI

A. Dataset

Dalam pengambilan dataset, digunakan sensor IMU (Inertial Measurement Unit) LSM9DS1 yang ada pada Arduino Nano BLE 33 yang ditempatkan di bagian dada untuk mengukur nilai percepatan (acceleroX, acceleroY, dan acceleroZ) setiap 0,01 detik setelah tombol *push button* ditekan. Dataset ini diperoleh dari 15 sampel subjek yang melakukan setiap gerakan. Terdapat 5 kelas gerakan yang diidentifikasi, yaitu berjalan, duduk (perubahan gerakan dari berdiri menjadi duduk), berdiri (perubahan gerakan dari duduk menjadi berdiri), oleng, dan jatuh. Setiap subjek melakukan masing-masing aktivitas gerakan sebanyak 15 kali.



GAMBAR 1
Alur Preprocessing

Metode yang digunakan dalam pengambilan dataset adalah dengan menghubungkan mikrokontroler yang telah terintegrasi sensor IMU LSM9DS1 dengan tombol *push button* yang dikenakan oleh peserta. Ketika peserta menekan tombol, itu menunjukkan bahwa data percepatan diambil. Posisi Alat terdapat pada Gambar. 2.



GAMBAR 1
Posisi Penempatan Arduino Nano 33 BLE

Untuk skema gerakan yang digunakan sebagai berikut:

1. Aktivitas Jatuh:

Dataset gerakan jatuh hanya mencakup gerakan saat jatuh ke depan. Posisi Jatuh dapat lurus ke depan, sedikit serong kanan dan serong kiri. Pengujian aktivitas Jatuh dilakukan selama 10 kali. Skenario gerakan yakni posisi seseorang dari kondisi berdiri kemudian jatuh ke depan, akan dihitung berapa kali alat dapat mendeteksi gerakan jatuh pada masing-masing partisipan.

2. Aktivitas Kecenderungan Jatuh:

Dataset aktivitas gerakan Kecenderungan Jatuh meliputi gerakan saat seseorang "Akan" Jatuh saat dalam posisi ketika

berjalan ke depan. Pengujian Gerakan kecenderungan jatuh akan dilakukan selama 10 detik dengan posisi partisipan ketika sedang berjalan namun kondisi cenderung akan jatuh.

3. Aktivitas Duduk:

Dataset gerakan duduk meliputi perubahan gerakan dari posisi berdiri menuju posisi duduk (Standing to Sitting). Pengujian gerakan Duduk akan dilakukan duduk selama 10 kali.

4. Aktivitas Berdiri:

Dataset gerakan berdiri meliputi perubahan gerakan dari posisi Duduk menuju posisi berdiri (Sitting to Standing). Pengujian gerakan Berdiri akan dilakukan duduk selama 10 kali.

5. Aktviitas Jalan:

Dataset gerakan jalan akan dilakukan dengan skenario satu kali pengambilan gerakan meliputi lima langkah kedepan. Langkah tersebut akan di sesuaikan dengan timer selama 15 detik. Pengujian gerakan jatuh dilakukan dengan cara partisipan melakukan gerakan jalan seperti biasa.

B. Preprocessing dan Ekstraksi Fitur

Setelah pengambilan dataset, data harus dibersihkan terlebih dahulu dan masuk ke proses preprocessing. Metode preprocessing yang digunakan yakni metode statistik. Untuk perhitungan statistik yang digunakan meliputi nilai Maksimum, minimum, mean, median, skewness, kurtosis dan variance.

1. Maksimum (Max) adalah adalah nilai terbesar dalam kumpulan data, mewakili nilai tertinggi dalam dataset.
2. Minimum (Min) adalah nilai terkecil dalam kumpulan data, mewakili nilai terendah dalam dataset.
3. Median adalah nilai tengah dalam dataset ketika data disusun secara berurutan, dan merupakan ukuran sentral yang tidak dipengaruhi oleh nilai ekstrim, sehingga bermanfaat untuk data dengan distribusi condong atau adanya nilai ekstrim.

$$Median = \frac{x_{\frac{n}{2}} + x_{(\frac{n}{2}+1)}}{2}$$

(1)

- a. n adalah jumlah data dalam kumpulan data
 - b. X adalah Kumpulan data yang diurutkan dari yang terkecil hingga terbesar
 - c. $\frac{n+1}{2}$ adalah posisi data tengah jika jumlah data ganjil.
4. Rata-rata (Mean) adalah hasil membagi jumlah semua nilai dalam kumpulan data dengan jumlah nilai tersebut. Ini adalah ukuran sentral yang memberikan nilai khas dalam dataset.

$$\bar{x} = \frac{\sum x}{N} \quad (2)$$

Dimana:

- a. N jumlah data dalam sampel atau populasi
- b. x nilai dari data

5. Skewness mengukur ketidaksimetrisan distribusi data. Skewness positif menunjukkan ekor panjang di sebelah kanan distribusi, sedangkan skewness negatif menunjukkan ekor panjang di sebelah kiri.

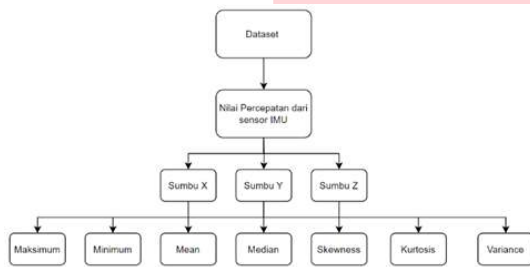
$$Skewness (\tilde{\mu}_3) = \frac{\sum_i^N (x_i - \bar{x})^2}{(N-1)\sigma^3} \quad (3)$$

- 6. Varians mengukur sebaran atau dispersi dalam dataset. Varians tinggi menunjukkan variasi atau sebaran yang besar dalam data.

$$S^2 = \frac{\sum (x_i - \bar{x})^2}{n-1} \quad (4)$$

- 7. Kurtosis mengukur bentuk ekor dan puncak distribusi dibandingkan dengan distribusi normal. Kurtosis positif menunjukkan ekor yang lebih berat dan puncak yang lebih tajam, sedangkan kurtosis negatif menunjukkan ekor yang lebih ringan dan puncak yang lebih datar.

$$Kurtosis = \frac{\mu_4}{\sigma^4} \quad (5)$$



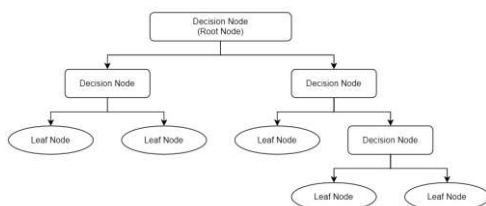
GAMBAR 3 Diagram Ekstraksi Fitur

C. Pemilihan Fitur

Dari seluruh fitur yang telah diekstraksi, akan dilakukan seleksi fitur untuk mengurangi jumlah fitur ekstraksi yang digunakan pada data pelatihan. Dalam proses seleksi ini, akan dipilih 5 fitur terbaik dari model Decision Tree yang kita gunakan untuk dilatih pada model yang kita gunakan. Alasan penggunaan fitur penting adalah untuk memilih fitur yang memiliki nilai atau kinerja yang baik dalam mendukung kerja mesin [15]. Selain itu, keterbatasan memori dalam mikrokontroler arduino nano 33 ble juga menjadi alasan mengapa digunakan pemilihan fitur terbaik saja yang akan dijalankan pada proses klasifikasi.

D. Classifier

Pada riset sebelumnya, model machine learning Random Forest telah digunakan untuk mendeteksi Pengenalan Aktivitas Manusia. Oleh karena itu, pada riset kali ini, kami akan menggunakan model non-ensemble baru yaitu Decision Tree. Akan dilakukan juga pengujian apakah important feature dapat menghasilkan mdoel yang lebih efektif dengan metode klasifikasi yang digunakan.



GAMBAR 2 DecisionTree

Pohon Keputusan (*Decision Tree Classifier*) adalah salah satu metode klasifikasi dalam pembelajaran mesin yang digunakan untuk memodelkan dan memecahkan masalah klasifikasi. Metode ini menghasilkan model dalam bentuk struktur pohon, di mana setiap simpul mewakili keputusan berdasarkan fitur-fitur tertentu, dan setiap cabang mewakili kemungkinan hasil atau kelas yang berbeda [12].

Pada gambar 4 digambarkan proses pembentukan pohon dimulai dari simpul akar yang merupakan fitur paling penting untuk memisahkan data ke dalam kelompok yang berbeda. Selanjutnya, setiap cabang menggambarkan pemisahan berdasarkan fitur lain yang relevan, dan proses ini berlanjut hingga terbentuknya simpul daun yang mewakili hasil akhir atau klasifikasi.

Pada Decision Tree Classifier, terdapat beberapa rumus dan metrik yang digunakan untuk membangun dan mengukur performa model. Berikut beberapa rumus penting yang terkait dengan Decision Tree Classifier.

- 1. Entropy (Entropi): Entropy digunakan untuk mengukur tingkat ketidakteraturan atau ketidakpastian dalam data. Persamaan entropi sebagai berikut:

$$H(Y) = - \sum (p_i \log_2 p_i)$$

Di mana:

- a. $H(Y)$ adalah entropi dari variabel acak Y.
- b. $\sum(p_i \log_2 p_i)$ adalah jumlah dari seluruh kemungkinan nilai pi (probabilitas) dari semua kemungkinan nilai variabel acak Y, dikalikan dengan logaritma basis 2 dari pi.

- 2. Information Gain (Gain Informasi): Information Gain digunakan untuk menilai seberapa banyak informasi baru yang diperoleh dari memisahkan dataset berdasarkan fitur tertentu. Rumus Information Gain untuk kelas Y terhadap fitur X dalam dataset adalah:

$$Gain\ Information\ (Y\ |X) = H(Y) - H(Y|X)$$

di mana:

- a. $Gain\ Information\ (Y\ |X)$ adalah Information Gain dari target variabel Y terhadap fitur X.
- b. $H(Y)$ adalah Entropi dari target variabel Y sebelum pemisahan (entropi node induk).
- c. $H(Y|X)$ adalah Entropi dari target variabel Y setelah pemisahan berdasarkan fitur X (entropi children node).

- 3. Gini Impurity: Gini Impurity adalah ukuran ketidakhomogenan (impuritas) dari sebuah node dalam Decision Tree. Rumus Gini Impurity untuk node kelas Y dalam dataset adalah:

$$Gini\ (Y) = 1 - \sum p_i^2$$

di mana:

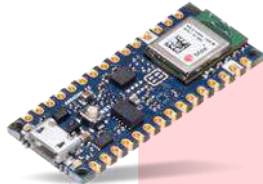
- a. pi adalah probabilitas dari setiap kelas Y dalam dataset.

E. Arduino Nano BLE 33

Arduino Nano 33 BLE adalah salah satu jenis mikrokontroler yang dikembangkan oleh Arduino. Papan ini memiliki bentuk yang kecil dan compact, sehingga cocok untuk aplikasi yang membutuhkan ukuran fisik yang

terbatas. Arduino Nano 33 BLE menggunakan sistem chip Nordic nRF52840 yang mengintegrasikan CPU ARM Cortex-M4 berkecepatan 64MHz, serta memori Flash 1MB dan RAM 256KB. Fitur yang dimiliki Arduino Nano 33 BLE ini memberikan kemampuan yang cukup untuk menjalankan berbagai macam aplikasi, baik dari implementasi Internet of Things hingga Machine Learning.

Salah satu fitur utama Arduino Nano 33 BLE adalah kemampuannya untuk berkomunikasi melalui BLE atau Bluetooth 5.0. komunikasi tersebut memungkinkan untuk dapat berkomunikasi dengan perangkat lainnya secara nirkabel. Dengan koneksi nirkabel ini, pengguna bahkan dapat mengatur pemantauan jarak jauh melalui aplikasi.



GAMBAR 5
Arduino Nano 33 BLE

Selain chip komunikasi yang sudah BLE, Arduino Nano 33 BLE juga memiliki sensor LSM9DS1 yang dapat mengukur nilai akselerometer, giroskop, dan magnetometer. Kombinasi BLE dan fungsi komunikasi sensor ini membuat Arduino Nano 33 BLE sangat cocok untuk berbagai aplikasi seperti perangkat pelacak gerak.

Konsumsi energi penting dalam pengembangan perangkat nirkabel. Arduino Nano 33 BLE memiliki pengisi daya baterai Li-Po yang memungkinkan Anda menggunakan baterai sebagai sumber daya. Selain itu, kartu ini dapat ditenagai oleh USB atau catu daya eksternal (7-12V), yang menawarkan fleksibilitas dalam penggunaan daya. Arduino Nano 33 BLE kompatibel dengan ekosistem Arduino yang luas. Pengguna dapat memprogramnya dengan Arduino IDE, yang menyediakan lingkungan pemrograman yang mudah digunakan dan beragam kode sampel dan pustaka yang tersedia untuk memfasilitasi proyek.

Dalam hal konektivitas dan ekspansi, Arduino Nano 33 BLE menawarkan pin I/O digital, input analog, PWM (Pulse Width Modulation), UART (komunikasi serial), SPI dan I2C. Ini memungkinkan pengguna untuk menghubungkan berbagai perangkat eksternal dan memperluas fungsi sesuai dengan kebutuhan proyek.

Arduino Nano 33 BLE adalah pilihan yang efisien dan serbaguna untuk mengembangkan berbagai aplikasi nirkabel dan IoT. Untuk pemula dan pemrogram berpengalaman, Arduino Nano 33 BLE menawarkan platform yang andal dan kuat dalam faktor bentuk yang kecil dan ringkas.

F. MicroMLgen Eloquent

Library Eloquent Micromlgen merupakan sebuah pustaka (library) yang digunakan untuk menghasilkan kode bahasa pemrograman C/C++ dari model Machine Learning yang telah dilatih sebelumnya. Library ini dikembangkan untuk mendukung implementasi model Machine Learning pada perangkat berdaya terbatas, seperti mikrokontroler atau perangkat IoT (Internet of Things).

Library Eloquent Micromlgen memungkinkan pengguna untuk mengkonversi model Machine Learning

yang telah dibuat dengan menggunakan framework atau pustaka Machine Learning populer, seperti Scikit-learn, Keras, atau TensorFlow, menjadi kode bahasa C/C++. Sehingga, model Machine Learning tersebut dapat dijalankan di mikrokontroler tanpa memerlukan komputer dengan daya komputasi tinggi.

Proses penggunaan Eloquent Micromlgen cukup sederhana. Pengguna hanya perlu menyediakan model Machine Learning yang telah dilatih dan menyebutkan target platform (misalnya Arduino, ESP8266, atau ARM) saat memanggil pustaka ini. Selanjutnya, Eloquent Micromlgen akan menghasilkan kode C/C++ yang dioptimalkan untuk platform target tersebut [7].

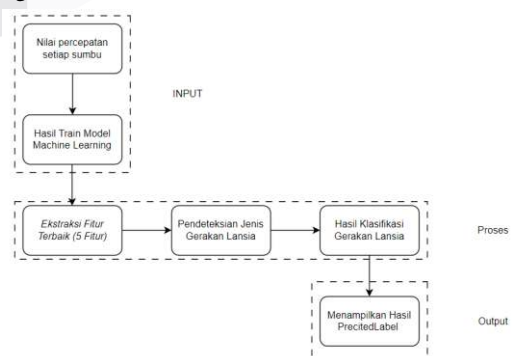
Keuntungan menggunakan Eloquent Micromlgen adalah sebagai berikut:

1. Efisien pada Perangkat Terbatas: Kode C/C++ yang dihasilkan sangat efisien dalam penggunaan sumber daya, sehingga dapat dijalankan pada perangkat dengan keterbatasan daya komputasi, memori, dan energi.
2. Dukungan untuk Berbagai Model: Eloquent Micromlgen mendukung berbagai jenis model Machine Learning, seperti DecisionTree, RandomForest, GaussianNB, SVC, OneClassSVM, XGBoost, Relevant Vector Machines dan SEFR.
3. Kompatibilitas dengan Berbagai Framework: Pustaka ini kompatibel dengan berbagai framework Machine Learning yang populer, sehingga memudahkan pengguna untuk mengimplementasikan model yang telah dibuat sebelumnya.
4. Tidak Memerlukan Koneksi Internet: Pengguna dapat menghasilkan kode C/C++ dari model Machine Learning tanpa memerlukan koneksi internet, sehingga memudahkan implementasi di lingkungan yang tidak terhubung internet.

Dengan menggunakan pustaka Eloquent Micromlgen, para pengembang dan peneliti memiliki kesempatan untuk menghadirkan kecerdasan buatan ke dalam berbagai perangkat terbatas, membuka potensi baru untuk menerapkan teknologi Machine Learning di berbagai bidang seperti IoT, automasi, robotika, dan aplikasi yang memerlukan analisis data di perangkat yang terbatas sumber dayanya.

III. METODE

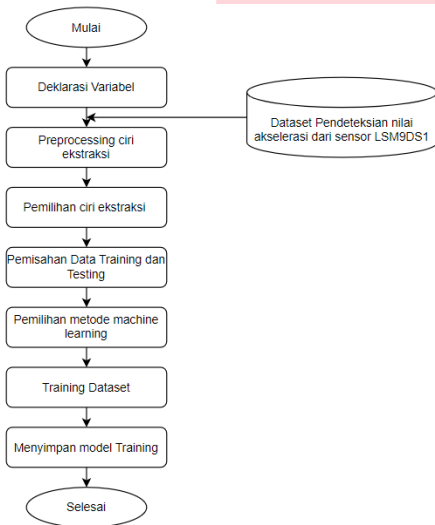
A. Diagram Blok Sistem



GAMBAR 3
Diagram Sistem

Gambar di atas merupakan diagram sistem dari alat yang dibuat. Alat akan diletakkan di tas kecil dan digunakan dibagian depan dada. Alat akan memproses seluruh nilai percepatan (akselerasi) sumbu X, Y dan Z dari sensor IMU LSM9DS1 yang sudah ada pada mikrokontroler Arduino Nano 33 BLE. Nilai tersebut akan menjadi nilai input yang akan menghasilkan suatu klasifikasi gerak. Data hasil pendeteksian klasifikasi gerak tersebut akan dikirimkan ke *Dongle Komunikasi* melalui *Bluetooth Low Energy* sebelum nantinya akan dikirimkan bersamaan dengan nilai *Latitude dan Longitude* menggunakan *Wifi Ke IoT Platform Firebase*. Selanjutnya Aplikasi akan menjadi antarmuka yang digunakan oleh pihak pemantau (Keluarga lansia). Aplikasi akan menampilkan Klasifikasi gerakan dan Titik Lokasi. Jika terjadi kondisi berbahaya Jatuh dan Kecenderungan Jatuh akan ditampilkan *warning pop-up message* disertai *vibrate sound* selama 2000 ms.

B. Flowchart



GAMBAR 4
Flowchart Machine Learning

Penelitian ini memanfaatkan dataset yang merekam nilai percepatan yang diperoleh dari sensor inersia LSM9DS1 yang ada pada Arduino Nano BLE 33 sebagai masukan. Titik pengukuran sensor inersia ada di dada pengguna karena alat digunakan di bagian depan dada. Tahapan preprocessing dataset akan dilakukan dengan metode statistika meliputi mean, median, maksimum, minimum, skewness, kurtosis, dan varians. Kemudian, dilakukan cross validation untuk mengetahui akurasi yang dihasilkan oleh model Decision Tree. Akan dilakukan pengujian untuk menentukan performa mesin menggunakan 3,5 dan 7 terbaik untuk di train menggunakan model Decision Tree. Hasil akhir dari sistem ini, dihasilkan suatu algoritma untuk mengenali gerakan jatuh, kecenderungan jatuh, duduk, berdiri dan jalan.

C. Model Machine Learning untuk bahasa C

Setelah kita melakukan train data dengan model `DecisionTreeClassifier()` dan menentukan variabel `X_train`, `X_validation`, `Y_train` dan `Y_validation`. Selanjutnya, kita perlu menyimpan model hasil train kita dalam bentuk header file agar nantinya dapat di implementasikan di arduino yang menggunakan bahasa C. Kali ini digunakan bantuan library *“Micromlgen by Eloquent”* sebagai jembatan komunikasi antara data yang sudah di train dengan arduino.

```

c_code = port(clf)
with open('ara bisa.h', 'w') as file:
    file.write(c_code)
  
```

GAMBAR 8
Ekstensi port untuk menyimpan file .h

Library *Micromlgen* dari *Eloquent* menggunakan metode fungsi port dalam menyimpan hasil train model. Nantinya bentuk yang akan di tampilkan pada bagian serial monitor yakni sesuai dengan yang sudah kita inisialisasi pada proses preprocessing yakni hanya dalam bentuk label kegiatan tanpa penjelasan deskriptif.

D. Implementasi Machine Learning pada bahasa C

Setelah berhasil menyimpan hasil predict label ke file header selanjutnya kita perlu membuat implementasi code pada bahasa C. Hal tersebut dikarenakan bahasa pemrograman yang digunakan pada bagian machine learning yakni bahasa Python sementara arduino menggunakan bahasa C. Kita akan menggunakan bantuan library *“Statistical.h”* yang akan menghitung preprocessing statistik di arduino. Nantinya hasil dari preprocessing akan dijadikan input untuk predict label.

```

Serial.println(" ");
Serial.println("Data Array : ");
Serial.println("Data Array Size : ");
Serial.println("Data Array Min : ");
Serial.println("Data Array Max : ");
Serial.println("Data Array Sq Sum : ");
Serial.println("Data Array Sum : ");
Serial.println("Data Array Avg : ");
Serial.println("Data Array Geometric Avg : ");
Serial.println("Data Array RMS Avg : ");
Serial.println("Data Array Extended RMS Avg : ");
Serial.println("Data Array Q1 : ");
Serial.println("Data Array Q2 (Median) : ");
Serial.println("Data Array Q3 : ");
Serial.println("Data Array Std Deviation : ");
Serial.println("Data Array Standard Deviation Error : ");
Serial.println("Data Array Coefficient Factor : ");
Serial.println("Data Array Variance : ");
Serial.println("Data Array Sorted : ");
Serial.println("Sorted Data Array : ");
  
```

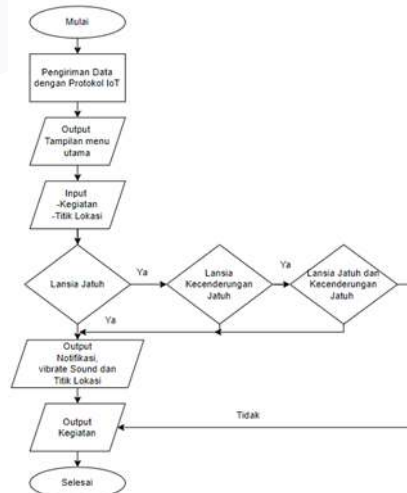
GAMBAR 5
Perhitungan statistik yang ada pada library *Statistic.h*

E. Aplikasi

Alat ini dilengkapi dengan aplikasi yang berfungsi sebagai interface antara alat observasi yang digunakan oleh lansia dengan pengawas (keluarga lansia). Aplikasi dibangun di atas platform Kodular.

Tentunya dalam membangun sebuah aplikasi dibutuhkan suatu platform tertentu yang berperan sebagai jembatan antara lalu lintas data. Salah satu platform dalam menciptakan suatu antarmuka (aplikasi) yakni kodular. Kodular termasuk salah satu website yang menyediakan fungsi mirip MIT App Inventor untuk dalam membangun suatu aplikasi berbasis Android menggunakan sistem drag-and-drop block program. Selain menggunakan sistem drag-and-drop, Kodular memiliki fitur tambahan yaitu Kodular Store dan Kodular Extension IDE sehingga pengembang dapat mengunggah aplikasi yang telah di buat ke Kodular Store dan membuat blok ekstensi IDE.

Nantinya data hasil pendeteksian akan dikirimkan ke *platform iot firebase* yang kemudian aplikasi akan memvisualisasikan data tersebut. Bagian yang akan ditampilkan pada aplikasi adalah jenis gerakan lansia dan maps. Untuk jenis gerakan berbahaya seperti *Jatuh “0”* dan *Kecenderungan jatuh (Oleng) “1”* akan ditampilkan *pop up message* beserta *vibrate sound* selama 2000 ms.



Gambar 10
Alur Tampilan Aplikasi



GAMBAR 11
Desain User Interface Aplikasi dengan Figma

IV. HASIL DAN PEMBAHASAN

A. Preprocessing

Tujuan dari tahap preprocessing adalah untuk mengubah format data masukan agar dapat diproses dengan tepat. Untuk mencapai hal ini, digunakan preprocessing metode statistika. Kemudian dilakukan ekstraksi beberapa fitur seperti rata-rata, median, nilai maksimum, nilai minimum, skewness, kurtosis, dan varians. Terdapat lima fitur ekstraksi yang dianggap paling berpengaruh dalam kasus ini meliputi varians Magnitude, maksimum accelX, maksimum Magnitude, median accelZ, varians accelZ, berdasarkan hasil seleksi fitur ekstraksi yang paling signifikan. Hasil dari ekstraksi fitur yang memiliki dampak terbesar terhadap pengenalan gerakan serta skor yang dihasilkan akan ditampilkan dalam Tabel 1.

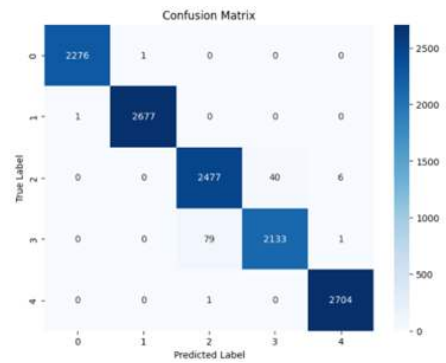
TABLE 1
Sepuluh Fitur Terbaik dengan model Decision Tree

NO	Ekstraksi Fitur	Skor
1	A_Magnitude variance	0.250
2	accelX Maksimum	0.227
3	A_Magnitude maksimum	0.216
4	accelZ median	0.150
5	accelZ Variance	0.034
6	AccelY Variance	0.028
7	A_Magnitude minimum	0.021
8	accelZ minimum	0.0177
9	acceY median	0.015
10	A_Magnitude skewness	0.092

B. Machine Learning

Dari model machine learning Decision Tree yang digunakan, akan digunakan percobaan penggunaan 3, 5 dan 7 fitur ekstraksi sebagai perbandingan untuk melihat akurasi dari tiap penggunaan fitur ekstraksi.

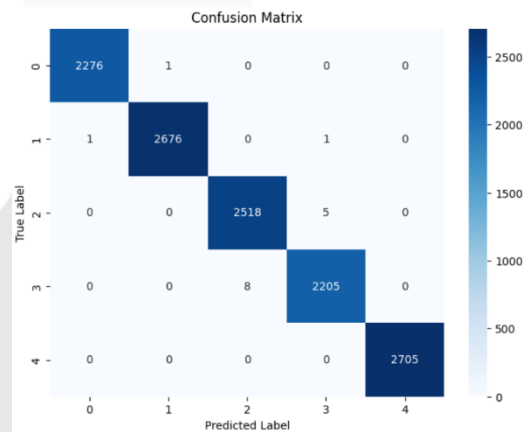
1. Untuk 3 fitur digunakan fitur Akselerasi Magntiude Varians, akselerasi X maksimum dan akselersi Magnitude Maksimum.



GAMBAR 6
Confusion Matrix 3 Fitur

Label	Precision	Recall	F1-score	support
0 Jatuh	1.00	1.00	1.00	1.00
1 Kecenderungan Jatuh	1.00	1.00	1.00	1.00
2 Duduk	1.00	1.00	0.99	1.00
3 Berdiri	1.00	0.99	1.00	1.00
4 Jalan	1.00	1.00	1.00	1.00

2. 5 Fitur Ekstraksi meliputi Akselerasi Magntiude Varians, akselerasi X maksimum, akselersi Magnitude Maksimum, akselerasi Z median dan akselerasi Z Varians. Berikut tabel hasil pengujian.

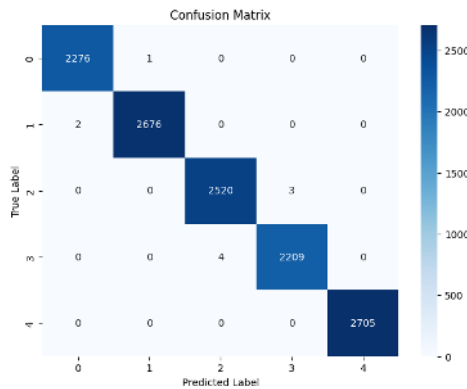


GAMBAR 7
Confusion Matrix 5 Fitur

TABLE 2
Precision, Recall, F-1 Score 5 Fitur

Label	Precision	Recall	F1-score	support
0 Jatuh	1.00	1.00	1.00	1.00
1 Kecenderungan Jatuh	1.00	1.00	1.00	1.00
2 Duduk	1.00	1.00	1.00	1.00
3 Berdiri	1.00	1.00	1.00	1.00
4 Jalan	1.00	1.00	1.00	1.00

3. Dengan 7 Fitur ekstraksi meliputi Akselerasi Magnitude Varians, akselerasi X maksimum, akselerasi Magnitude Maksimum, akselerasi Z median, akselerasi Z Varians, akselerasi Y varians dan Akselerasi Magnitude minimum



GAMBAR 8
Confusion Matrix 7 Fitur

TABLE 3
Precision, Recall, F-1 Score 7 Fitur

Label	Precision	Recall	F1-score	support
0 Jatuh	1.00	1.00	1.00	1.00
1 Kecenderungan Jatuh	1.00	1.00	1.00	1.00
2 Duduk	1.00	1.00	1.00	1.00
3 Berdiri	1.00	1.00	1.00	1.00
4 Jalan	1.00	1.00	1.00	1.00

Berdasarkan pengujian hasil train dataset menggunakan model machine learning decision tree, diperoleh hasil akurasi yang cukup beragam dari penggunaan 3, 5 dan 7 fitur ekstraksi. Untuk 3 Fitur ekstraksi diperoleh akurasi sebesar 45.52%. Untuk Penggunaan 5 Fitur Ekstraksi diperoleh akurasi sebesar 60.1% dan Untuk penggunaan 7 Fitur Ekstraksi diperoleh akurasi sebesar 34.06%. Percobaan dilakukan masing-masing oleh 3 partisipan.

Dari percobaan dengan 3, 5 dan 7 Fitur Ekstraksi di atas terlihat bahwa penggunaan 5 fitur memiliki nilai akurasi yang lebih baik dibandingkan penggunaan 3 dan 7 fitur ekstraksi. Hal ini disebabkan karena ketika penggunaan fitur ekstraksi terlalu sedikit dapat

mengakibatkan “Underfitting” atau fitur penting pada data kita hilang tidak ikut dalam pelatihan. Sementara itu, untuk kasus 7 Fitur Ekstraksi dapat mengalami “Overfitting” atau Model melatih data yang terlalu kompleks, hal tersebut juga dapat mengakibatkan Bias antara fitur penting pada gerakan dengan fitur lainnya. Selain untuk diperoleh hasil bahwa model kesulitan untuk melakukan pendeteksian gerakan Duduk dan Berdiri. Hal tersebut dikarenakan Aktivitas duduk dan berdiri memiliki pola gerakan yang relatif mirip, terutama dalam hal variasi akselerasi. Sehingga model sulit untuk mengenali aktivitas tersebut. Selain itu, hal tersebut juga bisa dikarenakan kurangnya variasi dalam pengambilan dataset. Sehingga model memiliki variasi pola yang kurang dalam melakukan pelatihan.

C. Akurasi klasifikasi

Sebelumnya alat sudah di latih dalam proses machine learning untuk dapat melakukan klasifikasi data untuk pengidentifikasian gerakan. Gerakan yang diidentifikasi sesuai dengan kegiatan yang kita label dan latih pada proses machine learning yakni gerakan jatuh, kecenderungan jatuh, duduk, berdiri dan jalan. Data percepatan masing-masing sumbu yang diperoleh dari sensor IMU yang ada pada arduino nano 33 BLE akan diklasifikasikan oleh Machine Learning. Akan dilakukan pengujian untuk mengidentifikasi ketepatan hasil train model machine learning yang telah kita lakukan dalam melakukan klasifikasi gerakan. Dilakukan percobaan pada satu partisipan untuk model machine learning dengan 3 Fitur ekstraksi, 5 Fitur Ekstraksi dan 7 Fitur Ekstraksi. Fitur ekstraksi yang digunakan sebagai berikut:

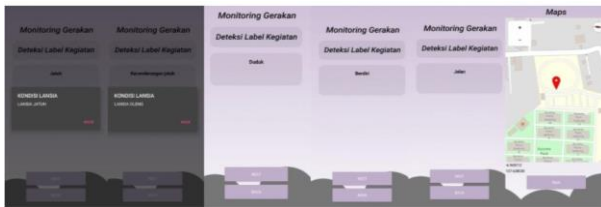
1. Untuk 3 fitur digunakan fitur Akselerasi Magnitude Varians, akselerasi X maksimum dan akselerasi Magnitude Maksimum.
2. 5 Fitur Ekstraksi meliputi Akselerasi Magnitude Varians, akselerasi X maksimum, akselerasi Magnitude Maksimum, akselerasi Z median dan akselerasi Z Varians.
3. Dengan 7 Fitur ekstraksi meliputi Akselerasi Magnitude Varians, akselerasi X maksimum, akselerasi Magnitude Maksimum, akselerasi Z median, akselerasi Z Varians, akselerasi Y varians dan Akselerasi Magnitude minimum.

Kegiatan	Akurasi (%)		
	3 Fitur Ekstraksi	5 Fitur Ekstraksi	7 Fitur Ekstraksi
Jatuh	80	100	100
	60	90	90
	50	90	70
Kecenderungan Jatuh	50	80	50
	30	10	30
	40	0	40
Duduk	60	80	0
	40	60	0
	30	60	0
Berdiri	40	60	0
	40	40	0
	30	40	0
Jalan	50	90	50
	50	50	40
	30	50	40
Total Akurasi	45,52	60,1	34,06

GAMBAR 9
Hasil Akurasi Klasifikasi Gerakan

D. Aplikasi

Berdasarkan pengujian yang dilakukan. Aplikasi saat ini sudah dapat menampilkan data hasil pendeteksian dan titik lokasi maps. Selanjutnya untuk gerakan berbahaya “Jatuh” dan “Kecenderungan Jatuh” sudah dapat ditampilkan pop-up message disertai dengan vibrate sound selama 2000 ms. Peringatan pop-up message akan ditutup ketika tombol “Back” ditekan. Selain itu, Pada bagian aplikasi maps akan diperbarui titik lokasinya jika terdeteksi kondisi lansia Jatuh atau Kecenderungan Jatuh saja. Diharapkan pihak pemantau (keluarga lansia) dapat memberikan pertolongan pertama pada lansia saat terdeteksi kondisi berbahaya tersebut.



GAMBAR 10

Tampilan Pop-up message dan Klasifikasi gerak biasa

V. KESIMPULAN

Menurut hasil penelitian ini, model machine learning dapat di implementasikan di arduino menggunakan library “*Statistical.h*”. Klasifikasi gerakan mencakup kegiatan berjalan, jatuh, berdiri ke duduk, dan duduk ke berdiri. Dikarenakan keterbatasan memory Arduino nano 33 BLE sebesar 1 MB, hanya akan dipilih fitur terbaik untuk dilakukan pelatihan. Setelah melakukan percobaan terhadap 3 fitur, 5 fitur dan 7 fitur terbaik diperoleh penggunaan 5 fitur ekstraksi menghasilkan akurasi yang lebih baik. Lima fitur ekstraksi tersebut meliputi *Aksekerasi Magnitude Varians*, *akselerasi X maximum*, *akselerasi Magnitude Maksimum*, *akselerasi Z Median* dan *akselerasi Z Varians*. Mean Akurasi yang diperoleh hasil training data tersebut yakni 99,8%. Hasil pelatihan menunjukkan nilai *Recall*, *Precision* dan *F1-Score* yang tinggi dengan nilai *Recall* 1.00, *Precision* 1.00 dan *F1-Score* sebesar 1.00 yang menunjukkan bahwa sistem mungkin untuk mengidentifikasi lima label kegiatan.

Selain itu aplikasi sebagai antarmuka alat yang digunakan lansia dengan pihak keluarga sudah dapat menampilkan hasil klasifikasi gerakan. Jika terdeteksi kondisi bahaya “Jatuh” dan “Kecenderungan Jatuh” maka akan disertai *pop-up message* dan *vibrate sound* selama 2000 ms.

REFERENSI

- [1] V. S. Thomas, S. Darvesh, C. MacKnight, and K. Rockwood, “Estimating the prevalence of dementia in elderly people: a comparison of the Canadian Study of Health and Aging and National Population Health Survey approaches,” *Int Psychogeriatr*, vol. 13 Supp 1, no. SUPPL. 1, pp. 169–175, 2001, doi: 10.1017/S1041610202008116.
- [2] J. Liang, Z. Qin, L. Xue, X. Lin and X. Shen, "Efficient and Privacy-Preserving Decision Tree Classification for Health Monitoring Systems," in *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12528-12539, 15 Aug.15, 2021, doi: 10.1109/IJOT.2021.3066307.
- [3] M. M. Alam, H. Malik, M. I. Khan, T. Pardy, A. Kuusik, and Y. le Moullec, “A survey on the roles of communication technologies in IoT-Based personalized healthcare applications,” *IEEE Access*, vol. 6, pp. 36611–36631, Jul. 2018, doi: 10.1109/ACCESS.2018.2853148.
- [4] N. G. Nia, E. Kaplanoglu, A. Nasab and H. Qin, "Human Activity Recognition Using Machine Learning Algorithms Based on IMU Data," 2023 5th International Conference on Bioengineering for Smart Technologies (BioSMART), Paris, France, 2023, pp. 1-8, doi: 10.1109/BioSMART58455.2023.10162095.
- [5] Viswanatha, et al. (2022). "Implementation of Tiny Machine Learning Models on Arduino 33-BLE for Gesture and Speech Recognition." *Journal of Xi'an University of Architecture & Technology*.
- [6] Mutlu, O, “Processing data where it makes sense in modern computing systems: Enabling in-memory computation”, *IEEE* 2018.
- [7] Eloquent Arduino. [online] 2020. micromlgen. available on September 19, 2022: <https://github.com/eloquentarduino/micromlgen>.
- [8] L. Fan, Z. Wang and H. Wang, "Human Activity Recognition Model Based on Decision Tree," 2013 International Conference on Advanced Cloud and Big Data, Nanjing, China, 2013, pp. 64-68, doi: 10.1109/CBD.2013.19.
- [9] R. Kumar and R. Verma, “Classification algorithms for data mining: A survey,” *International Journal of Innovations in Engineering and Technology (IJJET)*, vol. 1, no. 2, pp. 7–14, 2012.
- [10] S. S. Nikam, “A comparative study of classification techniques in data mining algorithms,” *Oriental journal of computer science & technology*, vol. 8, no. 1, pp. 13–19, 2015.
- [11] Ismail, Istiqomah, and Husneni Mukhtar “Development Human Activity Recognition for the Elderly Using Inertial Sensor and Statistical Feature “ *Proceeding of the 3rd International Conference on Electronics, Biomedical Engineering, and Health Informatics, Lecture Notes in Electrical Engineering 1008*, https://doi.org/10.1007/978-981-99-0248-4_21.
- [12] riyanka and D. Kumar, “Decision tree classifier: a detailed survey,” *International Journal of Information and Decision Sciences*, vol. 12, no. 3, pp. 246–269, 2020.
- [13] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, “Decision tree classifier for network intrusion detection with GA-based feature selection,” in *Proceedings of the 43rd annual Southeast regional conference Volume 2*, 2005, pp. 136–141.
- [14] I. S. Damanik, A. P. Windarto, A. Wanto, S. R. Andani, and W. Saputra, “Decision Tree Optimization in C4. 5 Algorithm Using Genetic Algorithm,” in *Journal of Physics: Conference Series*, 2019, vol. 1255, no. 1, p. 012012.
- [15] Y. Liu, L. Hu, F. Yan, and B. Zhang, “Information Gain with Weight Based Decision Tree for the Employment Forecasting of Undergraduates,” in 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, Aug. 2013, pp. 2210–2213, doi: 10.1109/GreenCom-iThingsCPSCom.2013.417
- [16] UNICEF (2021). Laporan Tahunan 2021 UNICEF INDONESIA. [On-line].