KLASIFIKASI GERAKAN NAIK-TURUN TANGGA DAN BERJALAN MENGGUNAKAN SENSOR INERSIA DAN KNN

TUGAS AKHIR

HAGI ABHIPRAYA SAPUTRA
311710011

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MA CHUNG
MALANG
2022

LEMBAR PENGESAHAN

TUGAS AKHIR

KLASIFIKASI GERAKAN NAIK-TURUN TANGGA MENGGUNAKAN SENSOR INERSIA

Oleh:

HAGI ABHIPRAYA SAPUTRA

NIM. 311710021

dari:

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS dan TEKNOLOGI

UNIVERSITAS MA CHUNG

Telah dinyatakan lulus dalam melaksanakan Tugas Akhir sebagai syarat kelulusan dan berhak mendapatkan gelar Sarjana Komputer

Dosen Pembimbing 1,	Dosen Pembimbing 2,
Dr.Eng. Romy Budhi, ST., MT.	**Ir. Oesman Hendra Kelana, M.Cs.**
NIP. 20070035	**NIP. 20110022**

Dekan Fakultas Sains dan Teknologi,

Dr. Kestrilia Rega Prilianti, M.Si.

NIP. 20120035

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi Tugas Akhir ini dengan judul KLASIFIKASI GERAKAN NAIK-TURUN TANGGA DAN BERJALAN MENGGUNAKAN SENSOR INERSIA DAN KNN adalah benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Malang, 28 Juli 2022

Hagi Abhipraya Saputra
NIM. 311710011

KLASIFIKASI GERAKAN NAIK-TURUN TANGGA DAN BERJALAN MENGGUNAKAN SENSOR INERSIA DAN KNN

Hagi A. Saputra, Romy Budhi Widodo, Oesman Hendra Kelana

Universitas Ma Chung

Abstrak

Penelitian terhadap gerak tubuh manusia diperlukan diberbagai bidang seperti kesehatan, keamanan, dan pengawasan. Salah satu bentuk penelitian yang dilakukan berupa klasifikasi gerakan manusia. Klasifikasi gerakan manusia dapat digunakan untuk mengawasi keadaan orang-orang yang membutuhkan perhatian lebih seperti lansia yang rentan jatuh atau digunakan sebagai alat pengukur kesehatan kaki bagi yang cedera. Penelitian ini bertujuan untuk mengetahui kemampuan metode KNN yang digunakan untuk mengklasifikasi gerakan berjalan dan naik-turun tangga terhadap kumpulan data yang didapatkan dari rekaman data sensor inersia IMU.

Penelitian ini mengklasifikasi gerakan naik-turun tangga dan berjalan menggunakan KNN. Data yang diklasifikasi adalah hasil *feature extraction* dari data pengukuran pitch berupa nilai *wave length* dan *mean absolute deviation* dari masing-masing fitur sensor yang dipakai. Proses klasifikasinya dilakukan dalam tiga percobaan. Percobaan pertama klasifikasi menggunakan fitur wave length saja, percobaan kedua diklasifikasi menggunakan hanya fitur *mean absolute deviation*, percobaan ketiga menggunakan kedua fitur.

Akurasi klasifikasi menggunakan *wave length* pada proses *training* mendapatkan nilai 88% dengan nilai k sebesar 3 sedangkan pada pengujian akurasi data tes yang didapatkan sebesar 31%. Akurasi klasifikasi menggunakan *mean absolute deviation* pada proses pelatihan sebesar 91% dengan nilai k = 4 sedangkan pada pengujian data tes sebesar 50%. Hasil akurasi menggunakan kedua fitur ketika pelatihan sebesar 95 % pada k = 3 sedangkan pengujian data tes mendapatkan akurasi 65 %.

Kata kunci : Human Activity Recognition, Inertial Measurement Unit, K-Nearest Neighbor, Machine Learning

HUMAN ACTIVITY RECOGNITION ON CLIMB UPSTAIRS-DOWNSTAIRS AND WALKING USING INERTIA SENSOR AND KNN

Hagi A. Saputra, Romy Budhi Widodo, Oesman Hendra Kelana

Universitas Ma Chung

Abstract

Research on human motion are needed in wide variety like health, security, and monitoring. One of the research is how to classify human activity motion. This type of classification can be used by the medics to oversee on some people who need it the most, like on older people who have high chance to fall or it can be used to measure recovery state of an injured athlete. The purpose of this study is try to prove how good KNN is to classify walking motion, climb upstairs, and climb downstairs.

This study classify walking motion, upward, and downward using KNN. The training data is collected through feature extraction from pitch data. Feature extraction calculate the wave length and mean absolute deviation. This study has three experiment. The first experiment is to classify the motion using only wave length feature. Second experiment is to classify every motion using only mean absolute deviation, and the last one is classify all the motion using both the wave length and mean absolute deviation

The result of this study it was found that when in the process of training the model, the accuracy rate on wave length feature reaches to 88% . This accuracy achieved by using k = 3. In testing process got only 31 % accuracy rate. Training with mean absolute deviation got 91% accuracy rate with k = 4, but only 50% accuracy rate when using test data. The last one is training with both of the feature got 95% accuracy, k = 3, and 65% accuracy when using the test data

Keywords : Human Activity Recognition, Inertial Measurement Unit, K-Nearest Neighbor, Machine Learning

Kata Pengantar

Puji syukur kepada Tuhan Yang Maha Esa karena atas rahmat dan penyertaan-Nya sehingga tugas akhir ini dapat diselesaikan. Laporan ini berisi hasil dari penelitian yang telah dilaksanakan di Laboratorium Human Machine Interaction, Universitas Ma Chung. Atas semua dukungan yang telah diberikan dalam pengerjaan penelitian ini, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Bapak Dr.Eng. Romy Budhi Widodo selaku dosen pembimbing.
2. Bapak Ir. Oesman Hendra Kelana, M.Cs.selaku dosen pembimbing
3. Ibu Dr.Kestrilia Rega Prilianti, M.Si selaku Dekan dari Fakultas Sains dan Teknologi Universitas Ma Chung,
4. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika,
5. Orang tua dan dosen yang telah mendukung pengerjaan Tugas Akhir hingga selesai,
6. Teman-teman yang sudah membantu dan mendukung selama pengerjaan penelitian ini

Laporan ini disusun berdasarkan hasil tugas akhir "Klasifikasi Gerakan Naik-Turun Tangga dan Berjalan Menggunakan Sensor Inersia dan KNN", sebagai salah satu prasyarat kelulusan.

Malang, 28 Juli 2022

Hagi Abhipraya Saputra

Daftar Isi

Daftar Gambar

Daftar Tabel

Bab I
Pendahuluan

1.1 Latar Belakang

Riset terhadap gerak tubuh menjadi topik yang menarik. Riset mengenai gerak tubuh memiliki banyak aplikasi terapan yang digunakan di berbagai bidang contohnya robot penunjang bagi pasien yang cedera kaki atau sulit berjalan, sistem pengawas keamanan, dan kesehatan lain seperti rehabilitasi (Li, et al., 2019). Aplikasi dari penelitian terhadap gerak tubuh di bidang kesehatan dapat ditemukan pada penelitian yang dilakukan oleh (Ranakoti, et al., 2019). Dasar dari dilakukannya penelitian oleh Ranakoti, et al (2019) adalah banyaknya kasus tergelincir dan jatuh pada lansia atau pasien-pasien lain, dengan begitu dibutuhkan suatu sistem pendeteksi otomatis yang dapat mengingatkan tenaga ahli atau orang-orang terdekat ketika dikenali adanya kejadian tersebut. Penelitian oleh Ranakoti, et al (2019) membahas mengenai klasifikasi menggunakan *Support Vector Machine* (SVM) terhadap kejadian jatuh dan tidak jatuh menggunakan data akselerasi dari *triaxial* sensor. Kegiatan sehari-hari akan dikelompokkan dalam kelas tidak jatuh sedangkan kegiatan berbaring dengan berbagai posisi dianggap sebagai jatuh.

Kebanyakan aplikasi terapan ini banyak melibatkan kaki bagian bawah. Beberapa kegiatan yang melibatkan kaki bagian bawah contohnya adalah berjalan, naik-turun tangga, berlari. Alasan tersebut yang mendasari penelitian ini dalam memilih berjalan, naik, dan turun tangga sebagai kegiatan yang akan diuji coba.

Di bidang kesehatan, pasien yang bermasalah dengan kakinya membutuhkan ahli medis untuk diagnosis, perawatan, atau rehabilitasi. Sayangnya untuk mendapatkan pelayanan dari ahli medis itu mengeluarkan banyak biaya atau ahli medis tidak selalu berada di tempat, sehingga pasien harus menyesuaikan waktunya dengan ketersediaan ahli medis.

Pasien atau pasar menuntut mendapatkan kualitas yang terbaik dari para penyedia jasa dalam bidang kesehatan, solusi yang terbaik yang bisa ditawarkan adalah dengan menggunakan kecerdasan buatan atau *autonomous robot*. Kecerdasan buatan dapat memberikan pelayanan yang sama ke setiap pasien, selain

itu kecerdasan buatan tidak terbatas tempat dan waktu (Kim, et al., 2016), sehingga pasien tidak perlu harus selalu bertemu dengan dokter. Ini berarti dapat memberikan waktu yang lebih nyaman bagi pasien.

Penelitian ini melakukan klasifikasi kegiatan manusia menggunakan *machine learning* secara *real time* menggunakan algoritma K *Nearest Neighbor*. *Machine learning* adalah pendekatan dalam kecerdasan buatan di mana sebuah sistem dibuat dengan melalui proses pelatihan dan pembelajaran sebagai bentuk tiruan pada proses manusia dalam belajar dan mengeneralisasi suatu informasi dengan tujuan untuk menggantikan atau menirukan perilaku manusia dan menggantinya dengan sebuah proses automasi (Ahmad, 2017). Algoritma *machine learning* yang digunakan di penelitian ini adalah K-Nearest Neighbor, klasifikasi kegiatan manusia dilakukan dengan menggunakan data yang diambil dari sensor inersia atau sensor Inertial Measurement Unit (IMU).

KNN digunakan dalam penelitian yang dilakukan oleh Soleha et al., (2019) untuk melakukan klasifikasi pergerakan kepala sebagai kontrol untuk menggerakkan kursi roda, dalam penelitian ini akurasi terbaik 100% dengan menggunakan k sebanyak 5.

Penelitian yang dilakukan oleh Kim et al., (2016) melakukan kategorisasi kegiatan manusia dengan cara menghitung jarak 2 pusat gravitasi tubuh yang didapat dari tubuh bagian atas dan tubuh bagian bawah. Adapula penelitian yang dilakukan oleh Li et al., (2019) menggunakan Triplet Markov Model untuk melakukan klasifikasi dan akurasinya berhasil mencapai di angka 90%.

Tingkat akurasi mencapai 98% di penelitian yang dilakukan oleh Lewi, et al (2021). Penelitian tersebut melakukan proses penghilangan derau terhadap data sensor IMU dengan kegiatan tidur, berdiri, berlari, duduk, bungkuk ke depan, dan bungkuk ke belakang. Klasifikasi dilakukan dengan menggunakan metode KNN yang diujikan pada nilai k = {1, 3, 5, 7, 9}. Hasil akurasi terbaik yang didapat sebesar 98,4893%.

Calisan dan Talu (2020) membandingkan tiga metode dalam mengklasifikasi gerakan naik, turun, berjalan, berlari, duduk, dan berdiri diam yang datanya dikumpulkan menggunakan satu sensor IMU yang dipasang tepat pada

dada subjek. Pemasangan pada dada subjek bertujuan untuk mendapatkan pusat gravitasi tubuh secara akurat. Metode yang diujikan adalah KNN, SVM, ANN

Hasil klasifikasi yang dilakukan oleh Calisan dan Talu (2020) menggunakan KNN berhasil mendapatkan akurasi rata-rata sebesar 94,6%. Pada penelitian tersebut dijelaskan pula akurasi tertinggi pada klasifikasi KNN didapatkan ketika subjek sedang duduk atau sedang berdiri dibandingkan dengan kelas lainnya dan kesalahan klasifikasi paling banyak terjadi ketika subjek diminta untuk naik dan turun tangga.

Sensor IMU merupakan alat pengukur gerakan yang populer digunakan untuk mendeteksi gerak tubuh manusia. Sensor ini memiliki akurasi tinggi dan kemudahan dalam pemasangan dan pengoperasiannya. Sensor IMU merupakan gabungan dari 3 sensor yaitu *accelerometer, gyroscope*, dan *magnetometer*.

1.2 Identifikasi Masalah

Berdasarkan paparan latar belakang maka identifikasi masalahnya adalah diperlukan adanya klasifikasi terkait kegiatan keseharian manusia seperti berjalan, naik-turun tangga secara *realtime.*

1.3 Batasan Masalah

Batasan masalah dari pengerjaan penelitian ini sebagai berikut.

1. Kegiatan yang diklasifikasi adalah berjalan, naik tangga, turun tangga
2. Klasifikasi menggunakan algoritma KNN
3. Sensor yang digunakan adalah sensor IMU XSENS
4. Data diambil dari lima subjek dengan kondisi kaki yang sehat
5. Penelitian dilakukan di *operating system* Windows

1.4 Perumusan Masalah

Bagaimana kinerja algoritma KNN dalam melakukan klasifikasi gerakan berjalan, naik-turun tangga

1.5 Tujuan Penelitian

Tujuan dilakukannya penelitian ini untuk mengetahui apakah algoritma KNN mampu mengklasifikasikan gerakan jalan, naik-turun tangga.

1.6 Luaran

Luaran dari penelitian ini berupa laporan tugas akhir dan kekayaan intelektual perangkat lunak.

1.7 Manfaat

Manfaat yang didapat dengan mengerjakan penelitian ini adalah mengetahui apakah KNN dapat digunakan untuk mengklasifikasikan gerakan naik-turun tangga dan berjalan, serta menjadi tahap awal dalam penelitian selanjutnya terkait sistem deteksi gerakan tubuh manusia dan sistem deteksi jatuh.

1.8 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini ditulis dengan sistematika penulisan seperti berikut.

Bab 1. Pendahuluan

Bab ini menjelaskan mengenai latar belakang dilakukannya penelitian ini, batasan masalah, rumusan masalah, tujuan penelitian, dan hasil luaran dari penelitian yang dilakukan

Bab 2. Tinjauan Pustaka

Bab ini berisi penjelasan mendasar mengenai alat yang digunakan untuk menerima dan mengirim data, serta aplikasi yang digunakan untuk mengoperasikan alat tersebut. Penjelasan terhadap metode yang digunakan dalam klasifikasi di penelitian ini yaitu KNN.

Bab 3. Analisis dan Desain Sistem

Bab 3 berisi tata cara melakukan penelitian mulai dari proses persiapan sensor dan cara pemasangan, tata cara pengambilan data, persiapan data dan proses pelatihan.

Bab 4. Hasil dan Pembahasan

Hasil klasifikasi dan bagaimana kinerja dari masing-masing model dan *dataset*nya akan dijelaskan di bab ini.

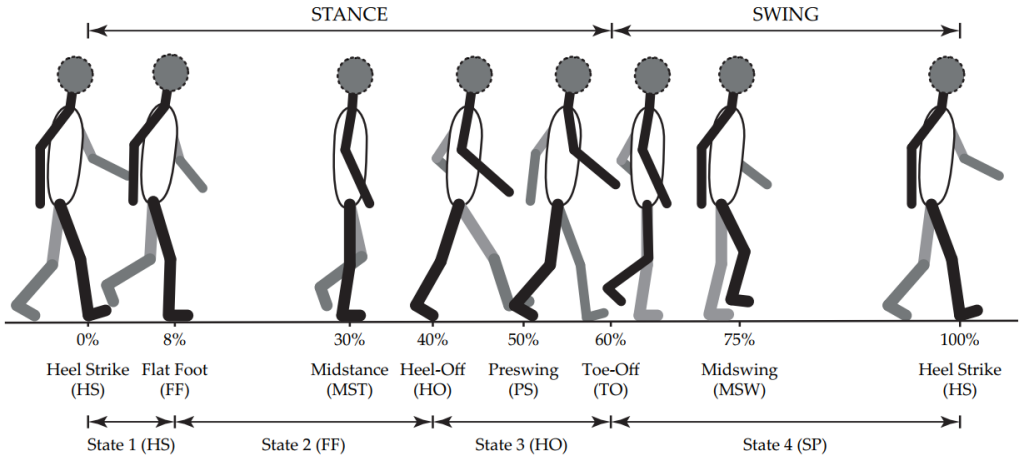
Bab 5. Simpulan dan Saran

Bab ini berisi kesimpulan dari hasil penelitian yang dilakukan beserta saran yang dapat dilakukan untuk penelitian selanjutnya.

Bab II
Tinjauan Pustaka

2.1 *Gait Phase*

Gait phase merupakan tahapan-tahapan gerak kaki ketika melakukan tindakan berjalan. Garis besar tahapan ini dapat dibagi menjadi dua model. Model pertama memiliki dua tahap berjalan yaitu *stance phase* dan *swing phase*.



Gambar 2.1 *Gait Phase*

(Sumber: *Gait Phase Detection for Lower-Limb Exoskeletons using Foot Motion Data from a Single Inertial Measurement Unit in Hemiparetic Individuals*, (Manchola, et al., 2019))

Model kedua terbagi menjadi empat tahapan, yaitu *heel strike* atau kontak awal kaki, *loading response phase* atau *flat foot*, heel off, dan swing atau proses ketika kaki terayun dan kembali lagi ke tahap awal *heel strike* (Manchola, et al., 2019).

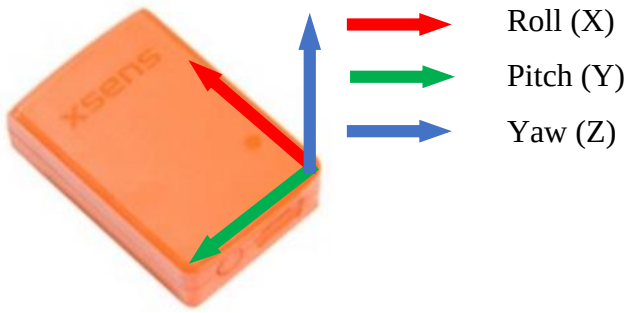
2.1.1 *Stride*

Penggerak manusia erat kaitannya dengan kaki. *Stride* merupakan siklus langkah kaki ketika melangkah. Satu *stride* merupakan langkah pertama dan langkah berikutnya pada sisi kaki yang sama, sehingga satu stride dapat berupa langkah pertama menggunakan kaki kiri dan berakhir di kaki kiri atau langkah pertama dari kaki kanan dan berakhir di kaki kanan juga.

Penelitian yang dilakukan oleh Widodo dan Wada (2016) menjelaskan dengan kecepatan berjalan 0,266 m/s yang termasuk ke dalam kecepatan berjalan yang lambat memiliki 21 *stride*/menit maka setiap detiknya memiliki 0,35 *stride*. Berdasarkan perhitungan tersebut durasi minimal untuk mendapatkan satu stride adalah 2,8 detik.

2.2 Sensor Inertial Measurement Unit (IMU)

IMU merupakan sensor yang digunakan sebagai *motion tracker* untuk aktivitas manusia. Sensor ini didalamnya memiliki 3D linear *accelerometer*, 3D *gyroscope*, 3D magnetometer. Sensor ini mudah dipasang dan nyaman karena tidak menggunakan kabel, dibagian belakangnya terdapat *velcro patch*, digunakan untuk merekatkan sensor ke *velcro strap* yang nanti akan dipasang di bagian tubuh objek yang akan diukur. Sensor yang dipakai di penelitian ini bernama MTw Awinda buatan XSENS.



Gambar 2.2 Sensor IMU

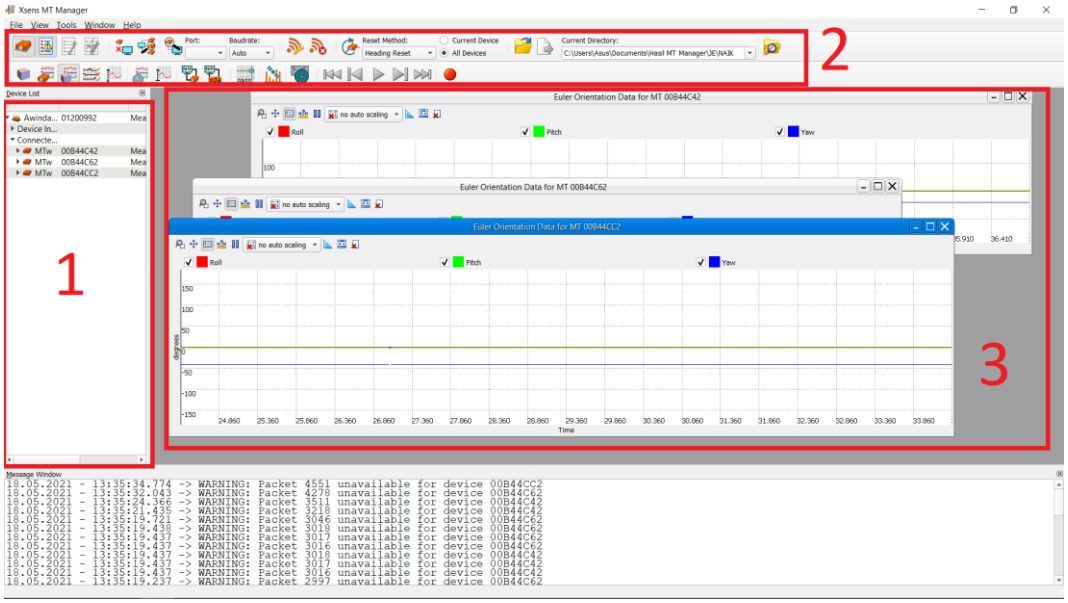
Sensor XSENS ini merupakan sensor IMU nirkabel berukuran 47 mm × 30 mm × 13 mm dengan berat 16 gram. Bagian depan sensor terdapat tulisan merk XSENS dan lampu led yang digunakan sebagai indikator sensor. Dibagian bawah terdapat micro-usb *port* untuk pengisian daya, disampingnya terdapat satu tombol power yang digunakan untuk menyalakan dan mematikan sensor. Bagian belakang sensor dipenuhi dengan *velcro patch* yang digunakan untuk merekatkan sensor ke *velcro strap* dan dibagian tengahnya terdapat *barcode* dan *serial number* untuk membedakan setiap sensor.

Tanda panah pada Gambar 2.2 menunjukkan gerakan sensor yang terekam. Gerakan memutar terhadap panah berwarna merah adalah *roll*, diputar terhadap

panah hijau menunjukkan gerakan *pitch*, dan *yaw* ditunjukkan dengan memutar sensor terhadap panah biru.

2.3 MT Manager

MT Manager merupakan perangkat lunak buatan XSENS yang digunakan untuk akuisisi data, visualisasi terhadap data sensor atau dapat juga memvisualisasikan data hasil rekaman, mengatur parameter sensor, dan ekspor data. MT Manager dapat memunculkan berbagai macam jenis grafik dari setiap sensor yang digunakan. Hasil rekaman data disimpan dengan format .mtb dan bisa ekspor menjadi file berformat .txt.



Gambar 2.3 Tampilan MT Manager 4.6

Gambar 2.3 merupakan tampilan MT Manager ketika memutar ulang hasil rekaman sensor maupun ketika melakukan rekaman, gambar ditunjukkan dengan nomor satu menjadi bagian di mana pengguna bisa melihat status sensor yang sedang digunakan, informasi yang terdapat didalamnya berupa jumlah sensor yang digunakan beserta dengan status baterainya.

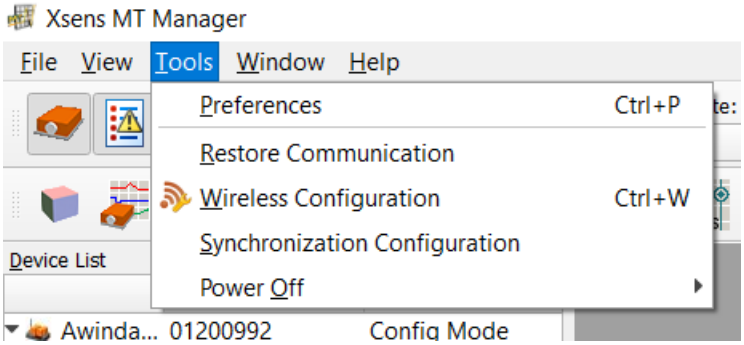
Bagian bertuliskan nomor dua menunjukkan berbagai pengaturan yang dapat diterapkan terhadap pembacaan sensor dan tampilan data pada saat

perekaman. Di bagian dua tersebut pengguna dapat menggunakan MT Manager untuk melakukan pencarian sensor yang sedang aktif.

Bagian yang bertuliskan nomor tiga menjadi tempat ditampilkannya visualisasi data. Tombol-tombol yang lain digunakan untuk mengatur bagaimana visualisasi data harus ditampilkan.

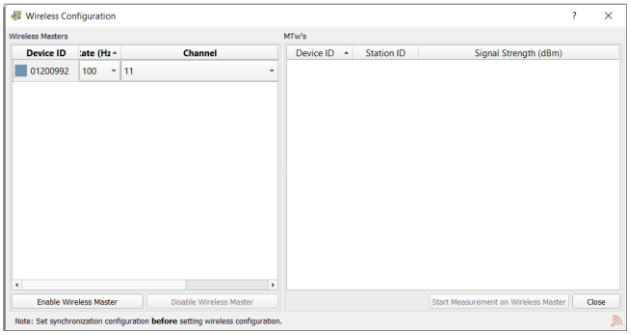
Berikut ini panduan pengoperasian MT Manager untuk melakukan perekaman data.

1. Pasang antena pada Awinda Station. Hubungkan Awinda Station dengan perangkat *desktop* yang digunakan melalui kabel usb yang sudah disediakan oleh Xsens.
2. Pada halaman utama, pilih tombol Tools pada bagian atas kiri. Pada menu Tools, pilih "Wireless Configuration". Seperti terlihat pada Gambar 2.4



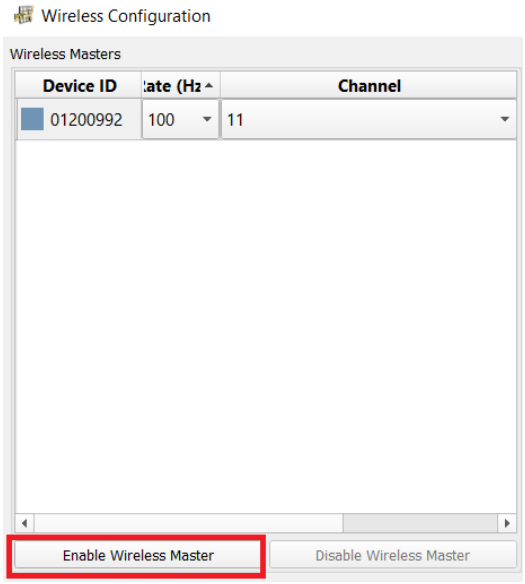
Gambar 2.4 Menu Tools

3. Gambar 2.5 berikut ini adalah halaman yang muncul ketika memilih "Wireless Configuration". Halaman ini digunakan untuk mengatur nilai *sampling rate* yang digunakan. Kotak putih di sebelah kanan digunakan untuk menampilkan setiap sensor IMU yang tersedia.



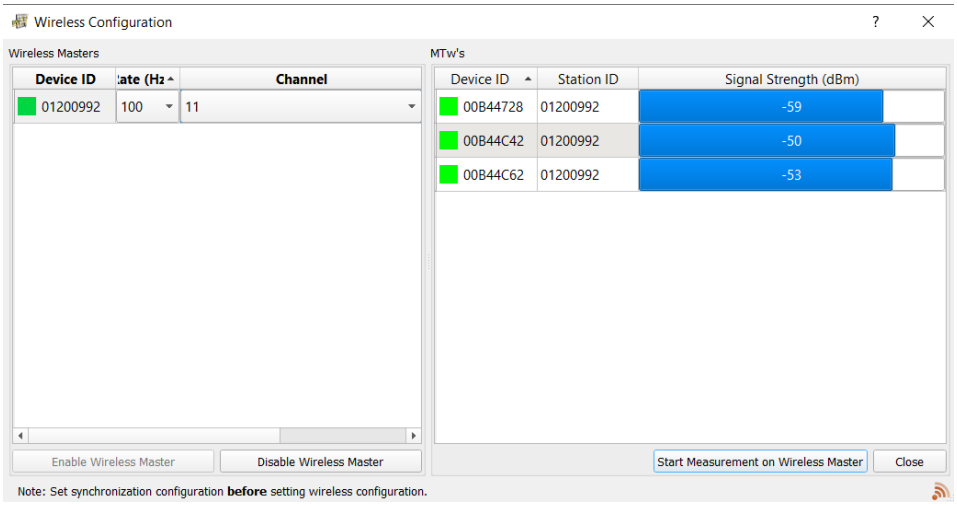
Gambar 2.5 Wireless configuration

4. Tekan tombol “Enable Wireless Master” yang tertanda pada Gambar 2.6. Nyalakan sensor IMU dengan menekan tombol *power* sampai lampu menyala



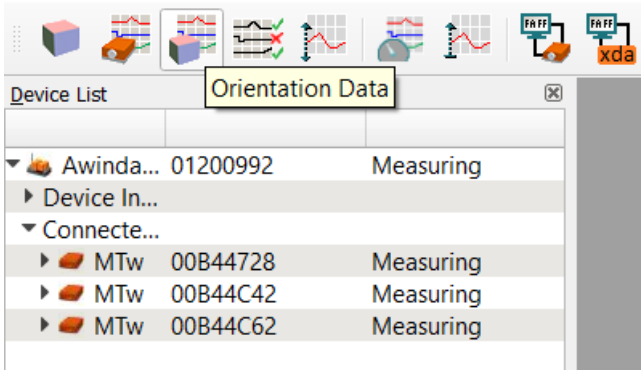
Gambar 2.6 Enable Wireless Master

5. Selanjutnya tekan tombol “Start Measurement on Wireless Master” untuk menghubungkan sensor dengan Awinda Station. Sensor yang berhasil terhubung akan muncul seperti Gambar 2.7. Lampu pada sensor yang berhasil terhubung akan berkelip-kelip selaras dengan kelip lampu pada Awinda Station. Tekan tombol “Close” untuk menutup halaman “Wireless Configuration”



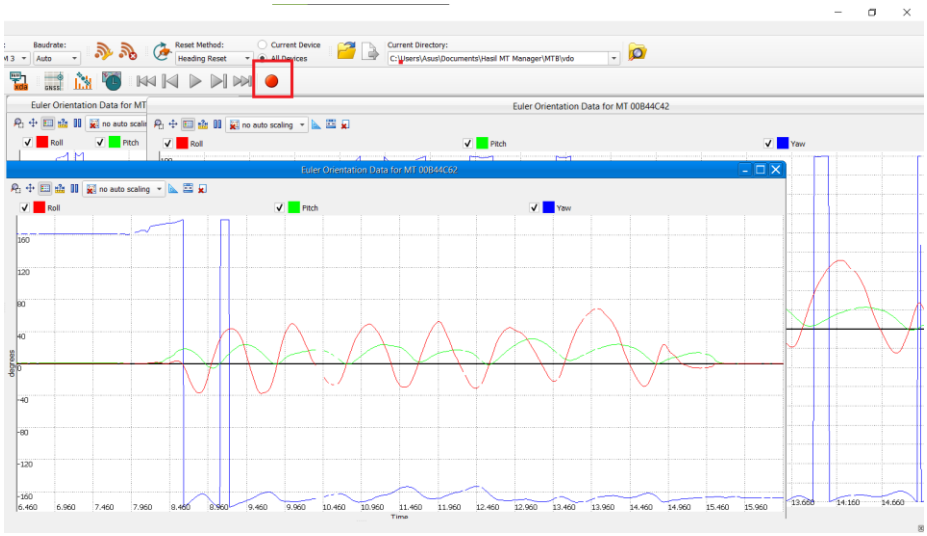
Gambar 2.7 Sensor yang terhubung dengan Awinda Station

6. Setelah konfigurasi selesai, akan kembali ke halaman awal. Selanjutnya pilih tombol “Orientation Data” untuk melihat visualisasi hasil data sensor seperti pada Gambar 2.8



Gambar 2.8 Visual data menggunakan Orientation Data

7. Hasilnya dapat dilihat pada Gambar 2.9. Untuk merekam dan menghentikan rekaman, gunakan tombol berbentuk lingkaran berwarna. Hasil rekaman akan otomatis tersimpan.



Gambar 2.9 Orientation Data

2.4 Awinda Station

Awinda Station adalah alat yang digunakan sebagai penerima hasil rekaman sensor secara nirkabel, alat ini mampu menerima data dan menyinkronkan dari dua puluh sensor secara bersamaan. Awinda Station memiliki enam slot *micro USB* yang digunakan untuk mengisi daya sensor IMU, untuk mengisi daya IMU, Awinda

Station membutuhkan sumber daya. Ada dua jenis sumber daya yang bisa digunakan, yaitu menggunakan arus DC dan melalui arus dari USB.

Gambar 2.10 Awinda Station

Angka satu pada Gambar 2.10 menunjukkan konektor DC dan konektor USB. Konektor DC sebesar 12v 1,5A digunakan sebagai *power supply* untuk pengisian daya sensor, konektor USB digunakan ketika ingin melihat antarmuka *awinda station*. Angka dua pada Gambar 2.10 merupakan konektor-konektor I/O yang dipakai ketika ingin menyinkronkan dengan alat-alat lain yang dipakai bersamaan.

2.5 Machine Learning

Machine learning merupakan bagian dari rumpun kecerdasan buatan, di mana machine learning dibuat untuk menggantikan atau menirukan proses manual yang dilakukan oleh manusia menjadi proses otomatis yang seolah-olah dilakukan oleh manusia melalui proses belajar terlebih dahulu (Ahmad, 2017). Ciri khas dari machine learning adalah adanya proses belajar atau *training*. Melalui proses ini diharapkan sebuah model dapat menirukan manusia dalam menentukan keputusan. Machine learning biasanya digunakan untuk klasifikasi atau prediksi.

2.5.1 Supervised Learning

Supervised learning didasarkan pada pelatihan data yang dengan kelas klasifikasi yang sudah diberikan diawal (Sathya & Abraham, 2013). Variabel input

dan output sudah diberikan diawal sehingga algoritma akan mencari fungsi untuk mendapatkan variabel output berdasarkan input yang diberikan.

2.5.2 Unsupervised Learning

Metode *unsupervised learning* mengidentifikasi pola tersembunyi dari data input yang tidak berlabel. Variabel input akan diberikan diawal tanpa memberikan variabel outputnya, algoritma akan diminta untuk melakukan proses pembelajaran dan mencari informasi terhadap variabel input dengan pengarahan yang kurang dari manusia atau penggunanya (Sathya & Abraham, 2013).

2.6 K-*Nearest Neighbor* (KNN)

Algoritma K-Nearest Neighbor adalah salah satu algoritma *supervised learning*. Algoritma ini digunakan untuk mengklasifikasikan suatu objek berdasarkan parameter yang dimiliki dengan melihat kemiripan dengan parameter yang dimiliki oleh objek lain (Ismail, 2018). Algoritma ini bekerja dengan cara melihat kemiripan suatu objek baru terhadap sejumlah objek yang sudah ditentukan jumlahnya atau biasa disebut tetangga. Jumlah tetangga yang digunakan tidak dibatasi sehingga perlu percobaan berulang-ulang untuk menentukan jumlah tetangga yang terbaik. Objek baru yang muncul akan dihitung jarak terdekatnya terhadap k tetangga menggunakan Euclidean.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (2\text{-}1)$$

Keterangan:

$d(x,y)$ = jarak dari dua obyek

x_i = data latih

y_i = data uji

n = jumlah data

2.7 Normalisasi Data

Data hasil rekaman sensor besarannya berbeda-beda antar variabel dan antar subjek sehingga perlu melakukan normalisasi data. Normalisasi adalah proses merubah data sehingga rentang nilainya berada di interval nol sampai satu, data terkecil mulai dari nol dan data terbesar bernilai satu. Metode ini membuat setiap

variabel dapat berkontribusi sama terhadap proses training atau praproses yang akan dilakukan di penelitian ini. Dalam penelitian ini, standardisasi dilakukan dengan menggunakan *Min Max Scaler*.

$$x_{std} = (X - Xmin) \div (Xmax - Xmin) \quad (2\text{-}2)$$

$$x_{scaled} = x_{std} * (max - min) + min \quad (2\text{-}3)$$

Keterangan :

X = Data yang ingin diubah

$Xmin$ = Data dengan nilai terkecil

$Xmax$ = Data dengan nilai terbesar

max = Nilai maksimal

min = Nilai terkecil

2.8 Ekstraksi Fitur

Ekstraksi fitur adalah proses mengurangi jumlah fitur dan data yang ada menjadi lebih sedikit. Ekstraksi fitur digunakan dalam penelitian ini untuk menyingkat ukuran dimensi data latih yang digunakan sehingga menjadi lebih kecil. Metode ekstraksi fitur yang digunakan adalah dengan menghitung nilai *mean absolute deviation* dan *wave length* dari masing-masing fitur yang digunakan.

2.8.1 Mean Absolute Deviation

Mean absolute deviation merupakan proses menghitung rata-rata jarak semua data terhadap nilai rata-rata dari data tersebut. Untuk menghitung *mean absolute deviation* menggunakan rumus sebagai berikut.

$$mad = \frac{1}{n}\sum_{i=1}^{n} |x_i - \bar{x}| \quad (2\text{-}4)$$

2.8.2 Wave Length

Wave length ini dilakukan untuk mendapatkan nilai panjang gelombang dari kumpulan data dalam satu interval waktu yang sudah ditentukan. Untuk menghitung wave length menggunakan rumus berikut.

$$wl = \frac{1}{n}\sum_{i=1}^{n-1} |x_{i+1} - x_i| \quad (2\text{-}5)$$

2.9 Confusion Matrix

Confusion matrix merupakan metode yang digunakan untuk melihat performa dari algoritma klasifikasi yang digunakan. Hal ini diperlukan karena kesalahan bisa saja terjadi karena jumlah data yang tidak sama di tiap kelasnya atau karena memiliki jumlah kelas yang banyak (Brownlee, 2020).

Tabel 2.1 Confusion Matrix

	Kelas 1 Prediksi	Kelas 2 Prediksi	Kelas 3 Prediksi
Kelas 1 Kenyataan	True Positive		
Kelas 2 Kenyataan		True Positive	
Kelas 3 Kenyataan			True Positive

Kelas 1 menjadi kelas positif sedangkan kelas 2 dan 3 adalah kelas negatif. Kategori *True Positive* (TP) terjadi ketika kenyataan datanya berupa kelas 1 dan hasil dari prediksi menunjukkan kelas 1 juga. Nilai *False Negative* (FN) didapat ketika data kenyataan merupakan kelas 1 tapi hasil dari prediksinya adalah kelas 2 atau kelas 3. *False Positive* (FP) adalah ketika data kelas 2 dan kelas 3 yang terprediksi sebagai kelas 1 atau kelas positif sedangkan True Negative (TN) adalah data kelas kelas negatif yang berhasil diidentifikasi sebagai kelas negatif juga. Nilai akurasi bisa dihitung berdasarkan nilai pada tabel *confusion matrix* dengan menggunakan rumus berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2\text{-}6)$$

$$Recall = \frac{TP}{TP + FN} \quad (2\text{-}7)$$

$$Precision = \frac{TP}{TP + FP} \quad (2\text{-}8)$$

$$Specificity = \frac{TN}{TN + FP} \quad (2\text{-}9)$$

Nilai *recall* membantu mengukur kesalahan prediksi yang terjadi di dalam kelas positif. Semakin besar nilainya semakin kecil kesalahan prediksi terjadi. *Specificity* adalah kebalikan dari *recall,* dimana nilai ini mengukur terjadinya kesalahan yang terjadi ketika model mengklasifikasi kelas negatif. Nilai *precision* menunjukkan kemampuan model dalam membedakan kelas negatif dengan kelas positif. Semakin besar nilai yang didapatkan maka semakin banyak kelas negatif yang terklasifikasi sebagai kelas positif.

2.10 Gradient Dua Titik

Gradient dihitung pada penelitian ini untuk melihat pergerakan datanya. Kumpulan baris data yang memiliki *gradient* yang tidak berbeda jauh menunjukkan subjek sedang pada keadaan yang sama. Nilai *gradient* yang berubah-ubah secara ekstrem menunjukkan subjek sedang berjalan sehingga data perekaman sensor yang didapatkan menjadi bervariasi. Teknik ini digunakan untuk membantu menghapus data yang tidak relevan dengan klasifikasi.

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{2-10}$$

2.11 Python

Python merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum (Muhardian, 2018), dirilis pada tahun 1991 dan terus dikembangkan oleh Python Software Foundation. Python dikembangkan oleh Guido van Rossum pertama kali dengan tujuan untuk menciptakan sebuah bahasa pemrograman yang mudah dibaca dan memiliki *syntax* yang lebih pendek bila dibandingkan dengan bahasa pemrograman lain seperti Java, C, dan C++ (Sohom, 2019).

Kelebihan yang dimiliki Python seperti berikut.

1. *Open source*. Kelebihan ini membuat Python memiliki banyak pengguna dengan begitu mudah pengguna lainnya mudah mencari dokumentasi dan panduan.
2. Memiliki banyak *library*. Python memiliki banyak library yang memudahkan *programmer* dalam membuat program.

3. Multifungsi, dapat diaplikasikan kedalam berbagai macam bidang ilmu komputer seperti *big data*, *web*, *mobile*, *machine learning*. Mudah dipahami. Sebagai bahasa pemrograman tingkat tinggi Python lebih mudah dibaca karena istilah-istilah yang dipakai seperti bahasa keseharian manusia.
4. Mudah dipahami. Sebagai bahasa pemrograman tingkat tinggi Python lebih mudah dibaca karena istilah-istilah yang dipakai seperti bahasa keseharian manusia.
5. Banyak dukungan fitur. Bahasa pemrograman Python cukup populer sehingga banyak dibuat produk-produk yang memudahkan programmer untuk mengeksekusi program seperti Colab, Jupyter.
6. *Virtual Environment*. Python memungkinkan untuk mengisolasi program yang dibuat secara lokal dengan berbagai macam versi *library* sehingga program-program yang lain tidak terganggu dengan perbedaan *library*.

2.11.1 Pandas

Pandas merupakan *library* yang digunakan sebagai alat manipulasi data yang sering digunakan untuk menganalisis. Pandas erat kaitannya dengan membuat objek *dataframe* dari suatu kumpulan data serta mempunyai kemampuan untuk menampilkan dan menghitung dengan kecepatan tinggi. Pada penelitian ini Pandas sering digunakan untuk *slicing* dan *merging* data, selain itu Pandas juga digunakan untuk membantu dalam praproses data seperti menghitung dan mencari nilai NaN. Nilai NaN ini bisa diganti dengan nilai yang lain atau bahkan menghapusnya, semua hal tersebut dapat dilakukan dengan bantuan Pandas (Pandas Team, 2014).

2. 11.2 NumPy

NumPy merupakan *open source library* yang dikembangkan sejak tahun 2005. NumPy digunakan untuk melakukan proses matematika di Python (NumPy, 2019). Tidak hanya untuk proses perhitungan numerik, NumPy memungkinkan untuk beroperasi dengan banyak dimensi matrik maupun *array*.

2. 11.3 Seaborn

Seaborn merupakan *library* visualisasi data yang dibangun berdasarkan matplotlib dan terintegrasi dengan Pandas sehingga proses membuat dan

menampilkan nilai-nilai statistik bisa dipermudah (Waskom, 2012). Walaupun dibangun atas *library* Matplotlib, dalam menggunakan Seaborn tidak perlu meng*import* Matplotlib.

2. 11.4 Scikit-learn (Sklearn)

Sklearn pertama kali dikembangkan oleh David Cornapeau sebagai bagian dari proyek Google Summer of Code di tahun 2007, kemudian pada tahun 2010 proyek ini dilanjutkan dengan dipimpin oleh Fabian Pedregosa bersama dengan tiga orang lainnya. Ada enam fungsi penting yang dimiliki seperti berikut. (Scikit Learn, 2007)

1. Klasifikasi, digunakan untuk menentukan kelas suatu data
2. Regresi, memprediksi nilai dengan variabel-variabelnya
3. *Clustering*, mengelompokkan data yang sejenis secara otomatis
4. *Dimensionality reduction*, mengurangi jumlah variabel untuk memperbaiki model
5. *Model selection*, melakukan validasi dan mengukur tingkat keberhasilan suatu model.
6. Praproses, melakukan berbagai macam manipulasi data agar keefektifan *feature* bisa ditingkatkan.

2.11.5 Matplotlib

Matplotlib merupakan *library* yang digunakan untuk membuat berbagai macam visualisasi di Python. Visualisasi yang disajikan oleh Matplotlib tidak terbatas pada gambar yang statis saja melainkan juga animasi dan interaktif. Banyak library pihak ketiga dibuat berdasarkan pada Matplotlib yang bisa digunakan sebagai alternatif seperti Seaborn (Matplotlib, 2002).

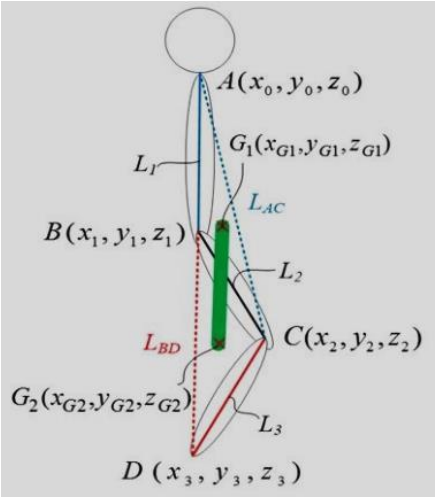
2.12 C#

C# merupakan bahasa pemrograman yang dikembangkan oleh Microsoft, digunakan untuk mengembangkan aplikasi *web, desktop, mobile, game* (W3School, n.d.). Versi 1.0 pertama kali diterbitkan pada tahun 2002 dengan tujuan untuk menjadi bahasa pemrograman berorientasi objek yang simple dan modern. Tahun 2003 diterbitkan versi 1.2 yang memberikan sedikit pengembangan dari versi sebelumnya. Versi 2.0 diterbitkan pada tahun 2005 dengan beberapa fitur baru

seperti *generics*, tipe data null, *anonymous methods* dan lainnya (Kulikov, et al., 2021). Pengembangan C# tetap dilanjutkan dan diterbitkan versi-versi yang lebih baru dengan berbagai fitur yang ditambahkan hingga versi terbaru saat ini yang stabil ada pada versi 9.0, dan versi 10.0 *preview*.

2.13 Penelitian Rujukan

Penelitian rujukan diambil dari penelitian Kim et al., (2016). Penelitian ini melakukan analisis pada hasil kategorisasi gerak manusia menggunakan kinect dan sensor IMU. Gerak manusia yang dikategorisasi adalah berjalan, berdiri, duduk, dan jatuh.

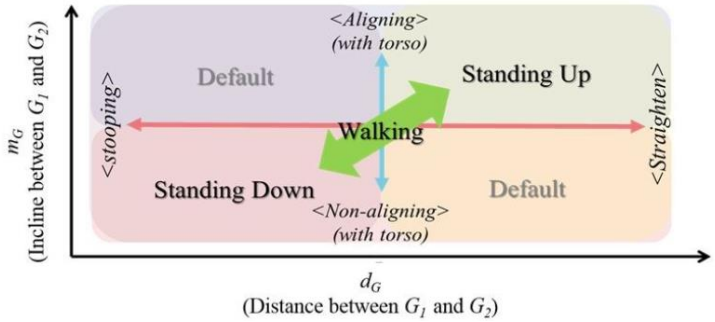


Gambar 2.11 Gambar COG

(Sumber: *An Approach to Categorization Analysis for Human Motion by Kinect and IMU,* Kim et al., (2016))

Kategorisasi gerak manusia dilakukan dengan cara membagi tubuh manusia menjadi dua *center of gravity* atau COG yaitu tubuh bagian atas dan tubuh bagian bawah seperti pada Gambar 2.11, segitiga berwarna biru menunjukkan tubuh bagian atas sedangkan segitiga berwarna merah menjadi tubuh bagian bawah. Setiap gerak tubuh akan dilihat perubahan COG-nya dan dihitung perubahan *incline* dan *decline* yang terjadi. Nilai *incline* dan *decline* ini yang digunakan sebagai kategorisasi pada gerak manusia. Gambar 2.12 mengilustrasikan semakin kecil

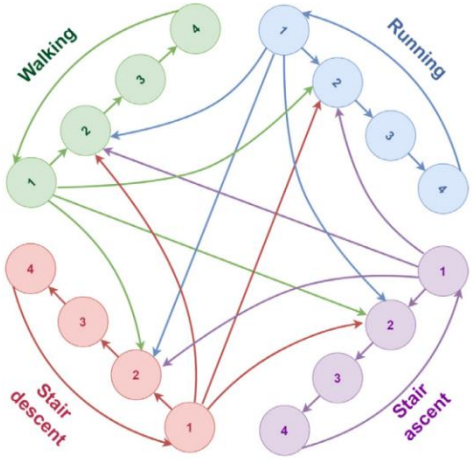
jarak antara COG pertama dan kedua menunjukkan badan sedang dalam keadaan membungkuk, sebaliknya semakin jauh menunjukkan badan sedang lurus.



Gambar 2.12 Klasifikasi dari COG

(Sumber: *An Approach to Categorization Analysis for Human Motion by Kinect and IMU,* Kim et al., (2016))

Penelitian lain oleh Li et al., (2019) melakukan klasifikasi aktivitas menggunakan triplet Markov Model. Klasifikasi yang dilakukan erat kaitannya dengan analisis terhadap gaya berjalan, sehingga dalam penelitian ini membagi fase berjalan manusia kedalam empat bagian yaitu *stance* (1), *push-up* (2), *swing* (3), dan *step down* (4) di mana setiap kegiatan selalu dimulai dengan posisi *stance.* Gambar 2.13 menjelaskan mengenai bagaimana alur gerakan kaki di setiap gerakan, yang mana setiap kegiatan ini selalu dimulai dengan posisi awal yang sama yaitu *stance.*

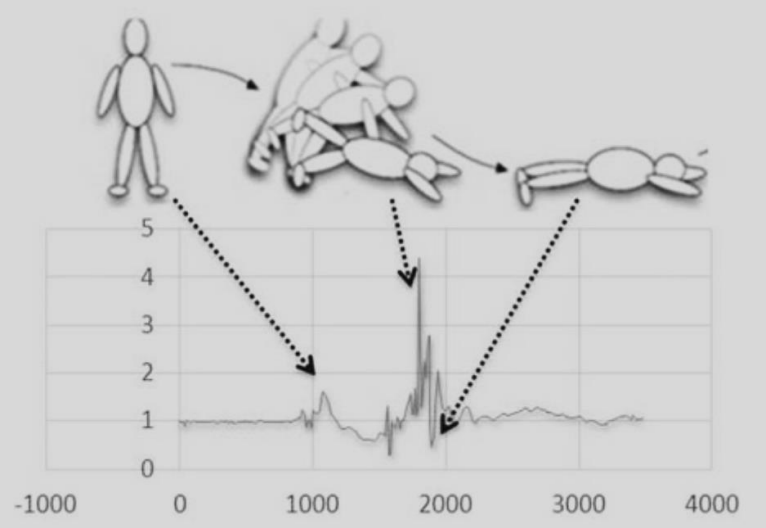


Gambar 2.13 Alur perubahan proses bergerak

(Sumber: *An Adaptive and On-line IMU-Based Locomotion Activity Classification Method Using A Triplet Markov Model,* Li et al., (2019))

Dalam penelitian ini klasifikasi dilakukan dengan cara meletakkan sensor IMU didalam sepatu. Ada dua hasil rekaman sensor yang tidak digunakan yaitu akselerasi dan data hasil magnetometer. Hasil data rekaman akselerasi tidak digunakan dengan pertimbangan bahwa kecepatan berjalan dan cara berjalan setiap orang berbeda-beda. Sedangkan data magnetometer yang berupa ukuran arah tujuan tidak cocok digunakan untuk klasifikasi jenis aktivitas atau kegiatan manusia.

Ranakoti et al (2019) melakukan penelitian terhadap klasifikasi gerak tubuh manusia sehari-hari dikombinasikan dengan prediksi gerakan jatuh. Teknik dasar yang digunakan dalam melakukan klasifikasi dengan melihat adanya lonjakan nilai data sensor akselerasi.



Gambar 2.14 Lonjakan nilai data ketika jatuh

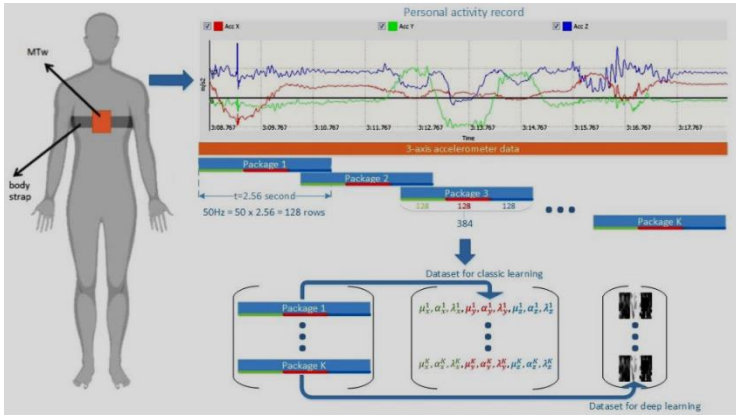
(Sumber: *Human Fall Detection System over IMU Sensors Using Triaxial Accelerometer* (Ranakoti, et al., 2019))

Gambar 2.14 menunjukkan munculnya data tidak normal yang ketika orang terjatuh. Ketika orang tepat akan membentur tanah, grafik yang melonjak tinggi menunjukkan ada percepatan pergerakan, sebaliknya ketika grafik turun kebawah dengan tajam menunjukkan orang tersebut sudah berada di bawah atau gerakan jatuh sudah berhenti.

Penelitian yang dilakukan oleh Ranakoti et al (2019) menggunakan dataset MobiFall dengan klasifikasi menggunakan Support Vector Machine (SVM) dengan hasil akurasi sebesar 78,04%. 147 data jatuh berhasil terprediksi dengan benar

sebesar 115 dan terprediksi salah sebanyak 32, sedangkan dari 149 data tidak jatuh terprediksi dengan benar sebanyak 116 data dan prediksi salah sebanyak 33.

Calisan dan Talu meneliti bagaimana performa KNN, SVM, ANN, dan CNN dalam mengklasifikasikan berjalan, berlari, duduk , berdiri di tempat, naik, dan turun tangga. Dalam penelitian ini hanya menggunakan satu sensor yang dipasang di dada subjek. Satu sensor akan menghasilkan tiga variabel yaitu *roll*, *pitch*, dan *yaw*. *Feature extraction* dilakukan terhadap masing-masing fitur yaitu menghitung nilai rata-rata, nilai deviasi, dan besar maksimal dari *eigenvalues* pada masing-masing *roll*, *pitch*, dan *yaw*. Fitur akhir yang didapatkan setelah *feature extraction* berupa sembilan kolom baru. Ketiga jenis *feature extraction* dihitung disetiap 128 baris data.



Gambar 2.15 Klasifikasi gerakan manusia menggunakan sensor di dada

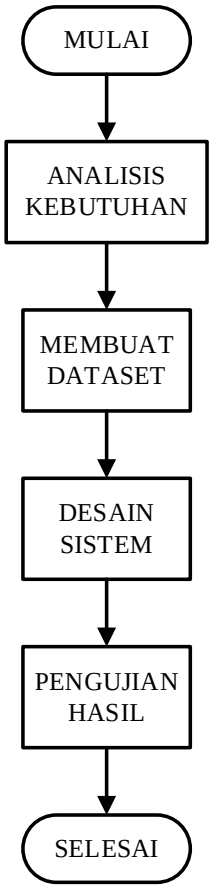
(Sumber : *Comparison of Methods for Determining Activity from Physical Movements* (Calisan & Talu, 2020))

Ketiga metode klasifikasi yang dibandingkan didapatkan nilai terbaik pada klasifikasi menggunakan ANN dengan nilai akurasi sebesar 99%. KNN tidak tertinggal jauh dengan akurasi rata-rata juga berada di nilai 90%. Penelitian tersebut menjelaskan gerakan manusia yang memiliki kemiripan akan cenderung salah diklasifikasi seperti gerakan naik dan turun tangga.

Bab III
Analisis dan Perancangan Sistem

3.1 Tahapan Penelitian

Tahapan penelitian dalam pembuatan aplikasi klasifikasi secara *realtime* terhadap gerakan naik-turun tangga dan berjalan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan pengerjaan

Tahap pertama yang dilakukan dalam pengerjaan penelitian ini adalah menentukan permasalahan yang ada dan dilanjutkan dengan analisis kebutuhan untuk mencapai solusi yang dibutuhkan dalam menyelesaikan masalah tersebut, langkah berikutnya adalah pengambilan *dataset.* Dataset ini diperlukan sebagai

acuan dalam melakukan klasifikasi gerakan. *Dataset* dikumpulkan dari lima orang dengan kondisi kaki yang sehat, data yang terkumpul merupakan hasil dari rekaman tiga sensor yang dipasang di tiga bagian kaki yang berbeda di kaki kanan. Tahap terakhir adalah pengujian terhadap hasil yang didapatkan untuk melihat bagaimana performa dari sistem atau aplikasi yang dibuat.

3.2 Analisis Kebutuhan

Kejadian tergelincir dan jatuh bisa dialami semua orang, efeknya tergantung terhadap keadaan tubuh masing-masing, semakin rentan seseorang maka efeknya akan semakin parah. Kasus seperti ini harus diperhatikan khususnya pada lansia, sehingga diperlukan suatu sistem yang dapat mengklasifikasikan gerakan manusia untuk mengatasi atau mencegah terjadinya jatuh atau tergelincir. Selain itu dibutuhkan juga suatu aplikasi atau sistem yang dapat dengan mudah dioperasikan dan dipasang oleh pasien-pasien lain yang membutuhkan perawatan kesehatan terhadap kakinya atau membutuhkan suatu pengawasan terhadap perkembangan kesehatan akibat cedera. Penelitian ini mengkhususkan dalam melakukan klasifikasi realtime terhadap kegiatan manusia berupa berjalan dan naik-turun tangga. Klasifikasi menggunakan algoritma KNN dengan eksperimen jumlah k={3,5,7}.

Perangkat utama yang digunakan untuk membantu dalam pengambilan data di penelitian ini berupa sensor IMU. Ada 3 sensor yang digunakan, sensor akan dipasang di kaki kanan, dibagian-bagian kaki yang memiliki sedikt jaringan lemak, hal ini dilakukan agar sensor lebih akurat dalam menerima data ketika anggota tubuh digerakkan. Sensor pertama dengan kode sensor CC2 dipasang dipasang pada paha lateral dekat dengan lutut seperti Gambar 3.2.

Gambar 3.2 Tempat pemasangan sensor di paha

Sensor kedua dengan kode C42 dipasang di tulang kering sekitar sepuluh cm diatas mata kaki sisi medial seperti ditunjukkan pada Gambar 3.3, sensor ketiga dengan kode C62 dipasang di atas *metatarsal* dimasukkan kedalam sepatu atau diikat di atas sepatu seperti ditunjukkan pada Gambar 3.3.

Gambar 3.3 Letak sensor kedua dan ketiga

Berikut ini daftar perangkat-perangkat yang digunakan dalam penelitian ini.

i. MTw XSENS

a. Tiga sensor IMU
b. Satu awinda station
c. Velcro strap
d. *Foot pad*

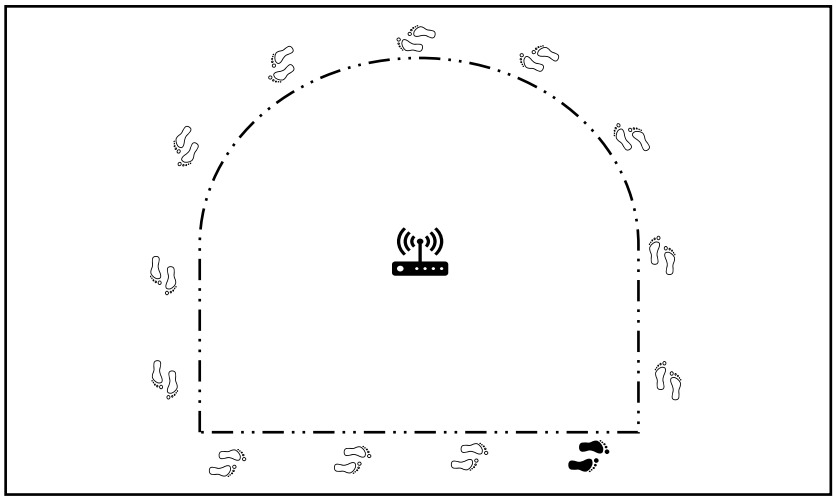
ii. Perangkat lunak

a. Python 3.x
b. C#
c. Jupyter Notebook
d. Microsoft Visual Studio

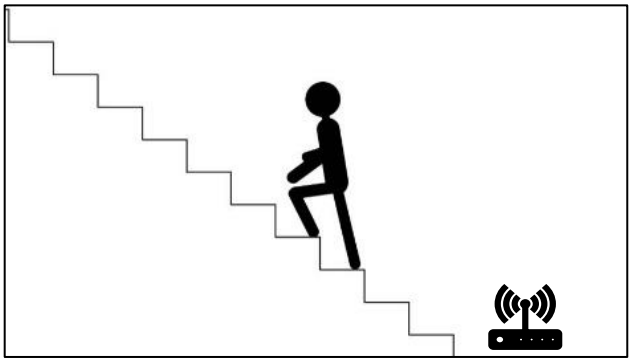
3.3 Prosedur Pengambilan Data

Sensor IMU yang digunakan memiliki beberapa aturan yang harus diperhatikan beberapa contohnya adalah peletakan sensor di anggota tubuh sebaiknya tidak di area yang memiliki banyak jaringan lemak, selain itu sensor ini perlu waktu untuk menyesuaikan *filter*nya ketika baru dinyalakan. Hal ini dilakukan karena agar pembacaan data oleh sensor tidak kacau karena pengaruh dari *filter* yang belum siap. Aturan lainnya berkaitan dengan arah kemana sensor menghadap ketika digunakan. Sensor yang sejalur atau searah sebaiknya menghadap ke arah yang sama, hal ini bisa dilakukan dengan melihat dari posisi tombol *power* atau tulisan XSENS yang ada di badan sensor. Bila salah satu sensor dipasang dengan posisi tombol *power* berada di bawah maka sebaiknya sensor lain yang sejalur juga dipasang dengan posisi tombol *power* di bawah juga.

Gerakan berjalan memutari lintasan dilakukan sebanyak sepuluh kali berlawanan dengan arah jarum jam. Dibagian tengah lintasan menjadi tempat diletakkannya Awinda Station beserta dengan perangkat komputer untuk mengawasi pergerakan graf hasil perekaman seperti yang diilustrasikan pada Gambar 3.4. Lambang kaki berwarna hitam menjadi titik awal dan titik akhir dalam proses berjalan. Gambar 3.5 menunjukkan proses naik-turun tangga yang dilakukan pada delapan anak tangga yang juga masing-masing diulang sepuluh kali. Awinda station diletakkan di bawah dekat dengan anak tangga terakhir.



Gambar 3.4 Lintasan berjalan



Gambar 3.5 Lintasan naik-turun tangga

3.4 Membuat Dataset

3.4.1 Pengambilan Dataset

Proses pengambilan data dilakukan pada subjek berumur 20-22 tahun dengan kondisi kaki sehat, diambil pada frekuensi *sampling* 100Hz. Data yang diambil berupa gerakan berjalan diulang sepuluh kali, turun tangga sejumlah delapan anak tangga diulang sepuluh kali, naik tangga sejumlah delapan anak tangga diulang sepuluh kali. Kecepatan berjalan, naik, dan turun tangga disesuaikan dengan kecepatan normal masing-masing subjek.

3.4.2 Membuat Data Latih

Hasil rekaman sensor akan disimpan dalam satu berkas dengan format mtb. Berkas mtb tersebut akan diubah menjadi tiga berkas berformat txt yang setiap berkasnya mewakili masing-masing sensor yang digunakan. Seperti pada Tabel 3.1,

setiap satu berkas mtb mewakili satu gerakan, dalam satu berkas mtb ini tersimpan tiga data sensor sekaligus, sehingga satu berkas mtb yang diubah akan menjadi tiga berkas txt yang setiap berkas tersebut akan dibedakan dengan nama kode serial sensor yaitu C42, C62, dan CC2 yang tertulis di akhir nama. Sepuluh hasil rekaman data per subjek, tujuh rekaman percobaan akan digunakan untuk sebagai data KNN dan sisa tiga rekaman data akan digunakan sebagai data uji.

Tabel 3.2 merupakan salah satu contoh data sensor yang terekam. Di setiap berkas, tertulis ukuran *sampling*, versi *firmware*, tipe filter dibagian awalnya, sedangkan rekaman sensornya sendiri awalnya terdapat kolom PacketCounter, SampleTimeFine, Roll, Pitch, dan Yaw.

Tabel 3.1 Konversi berkas berformat mtb ke format txt

mtb	txt
MT_01200992_000.mtb	Jehezkiel_JALAN_000-000_00B44C42.txt Jehezkiel_JALAN_000-000_00B44C62.txt Jehezkiel_JALAN_000-000_00B44CC2.txt

Tabel 3.2 Contoh data rekaman sensor

//Start Time : Unknown
// Update Rate: 100.0Hz
// Filter Profile: human (46.1)
// Firmware Version: 4.3.5				

PacketCounter	SampleTimeFine	Roll	Pitch	Yaw
11229		145,6145	-78,6957	55,78035
11230		147,3954	-79,002	54,03531
11231		146,0394	-78,7997	55,23932
11232		145,5686	-78,7863	55,61589
11233		145,1606	-78,8901	55,74023
11234		144,6299	-78,8844	56,12784

Ketiga berkas berformat txt akan digabungkan menjadi satu berkas berformat csv. Nama kolom dari masing-masing sensor akan dibedakan dengan angka, sensor pertama (CC2) mendapat imbuhan angka satu di akhir nama kolomnya, sensor kedua (C42) mendapat imbuhan angka dua, dan sensor ketiga (C62) mendapat imbuhan angka tiga seperti pada Tabel 3.3. *Dataset* masih memiliki data Yaw, untuk proses klasifikasi nanti tidak akan melibatkan data Yaw karena karakteristiknya yang mengacu pada kutub atau medan magnet dapat

menimbulkan bias terhadap informasi yang diberikan. Total seluruh data ada 250147 baris seperti terangkum dalam Tabel 3.4.

Tabel 3.3 Contoh data latih

Roll1	Pitch1	Yaw1	Roll2	Pitch2	Yaw2	Roll3	Pitch3	Yaw3	Kegiatan
-172,399	-66,9516	-124,169	117,0003	-80,3971	70,37973	-3,76979	-30,2265	146,9571	JALAN
-173,648	-67,1251	-122,48	117,4248	-80,5252	69,77613	-4,00526	-30,2611	147,3256	JALAN
-172,928	-67,0597	-123,507	117,2427	-80,4585	69,98698	-3,89902	-30,2539	147,1227	JALAN
-172,77	-67,0959	-123,751	117,2799	-80,4493	69,91381	-3,91717	-30,2684	147,1184	JALAN

Tabel 3.4 Total jumlah data mentah

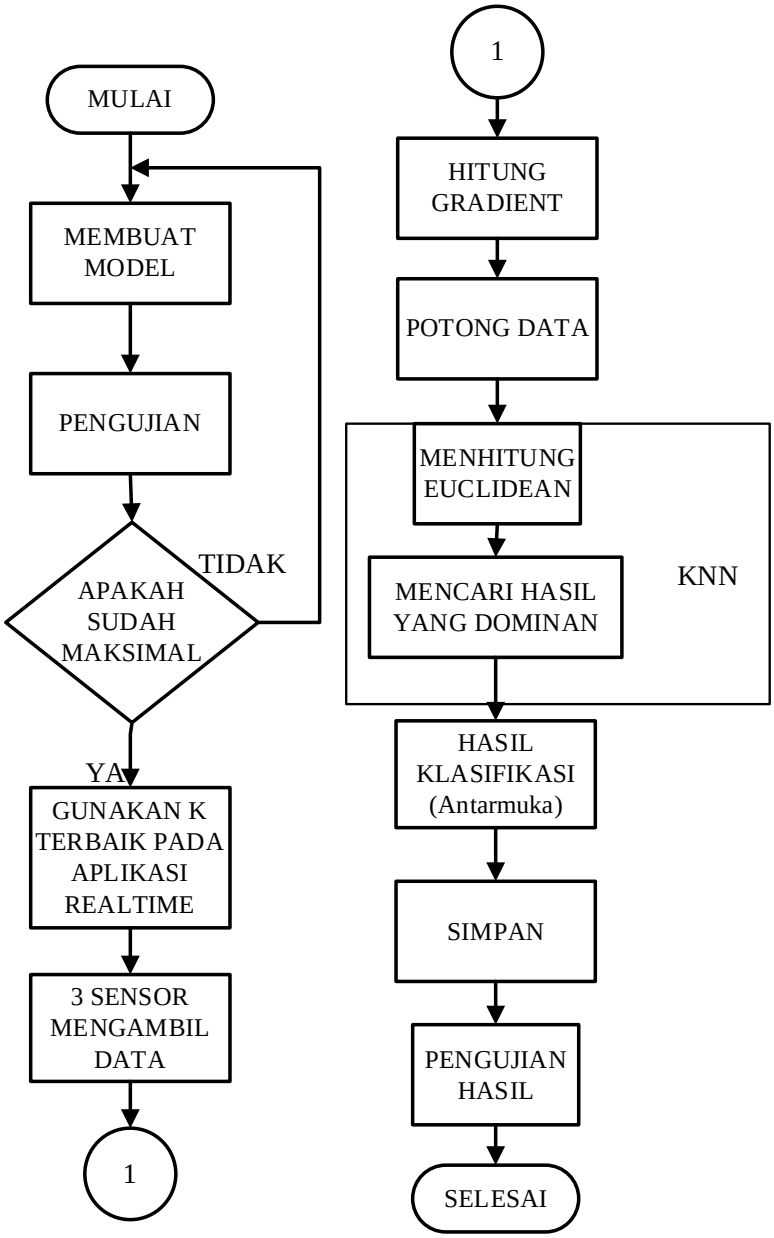
Kegiatan	Jumlah
Jalan	171685
Naik	39897
Turun	38565
Total	250147

Data latih berikutnya akan diseleksi. Fitur yang tidak relevan dengan proses klasifikasi gerakan akan dihapus, sehingga fitur yang tersisa hanya yang dipakai dan berkontribusi dengan baik terhadap proses klasifikasi.

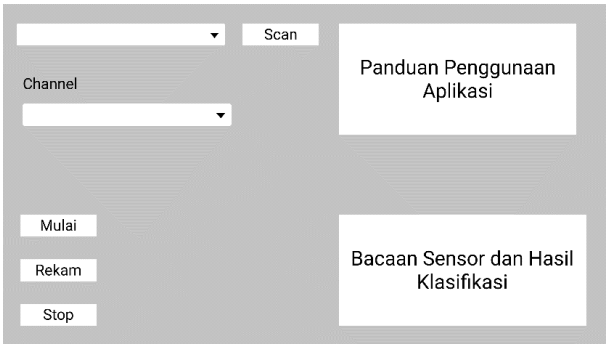
3.5 Desain Sistem

Klasifikasi yang dilakukan di penelitian ini menggunakan algoritma KNN, yang akan diujikan dengan jumlah k satu sampai dengan seratus. Pengujian dilakukan untuk menentukan nilai k yang optimal. Jumlah nilai k yang optimal akan digunakan sebagai alat klasifikasi final yang akan dilakukan secara *realtime*.

Gambar 3.6 menjelaskan alur berjalannya sistem klasifikasi. Sistem bekerja dengan menerima hasil bacaan sensor terhadap gerak kaki yang dilakukan ketika melakukan salah satu dari tiga kegiatan yang diujikan (berjalan, naik-turun tangga). Data sensor yang sudah terkumpul akan ditentukan jenis gerakannya berdasarkan nilai Euclidean, hasil dari klasifikasi ini akan ditampilkan di antar muka aplikasi.



Gambar 3.6 Alur sistem klasifikasi



Gambar 3.7 Rancangan desain tampilan aplikasi

Gambar 3.7 menjadi rancangan tampilan antarmuka aplikasi yang disarankan. Antarmuka diatas dibuat sepenuhnya dalam bahasa C# dengan *framework* Windows Form Application.

3.5.1 Desain Eksperimen

Pengujian dilakukan terhadap enam subjek. Keenam subjek tersebut dibagi menjadi dua kelompok. Kelompok pertama akan diuji gerakan berjalan terlebih dahulu, setelahnya baru diujikan pada gerakan naik-turun tangga. Kelompok kedua diuji pada gerakan naik-turun tangga terlebih dahulu baru dilanjutkan dengan gerakan berjalan. Setiap subjek diminta untuk mengulang setiap gerakan sebanyak tiga kali. Perulangan pada gerakan naik-turun tangga dilakukan secara bergantian.

3.5.2 Klasifikasi KNN

Klasifikasi gerak tubuh manusia secara *realtime* membutuhkan kecepatan yang tinggi, maka klasifikasi sebaiknya di hasilkan setiap satu *stride*. Jumlah data dalam satu *stride* ini dihitung dengan cara mengalikan frekuensi sensor yang digunakan yaitu 100 (hz) dengan lama waktu yang dibutuhkan untuk mendapatkan satu *stride* yaitu 2,8 (detik). Total jumlah data yang dibutuhkan untuk menghasilkan satu klasifikasi adalah 280 baris data sensor. Aplikasi klasifikasi akan mengumpulkan sebanyak 1120 baris, setara dengan empat *stride*. Klasifikasi akan dijalankan pada 560 baris data (setara dengan dua *stride*) yang diambil setelah melalui proses penghapusan data melalui nilai *gradient*.

Klasifikasi akan dilakukan dengan tiga metode. Metode pertama klasifikasi dilakukan melibatkan fitur *wave length* saja. Hasil dari klasifikasi ini akan diuji dengan *confusion matrix*. Metode kedua klasifikasi dilakukan dengan menggunakan fitur *mean absolute deviation* saja. Metode ketiga klasifikasi dilakukan dengan menggunakan kedua fitur tersebut.

3.5.3 Desain Pengujian

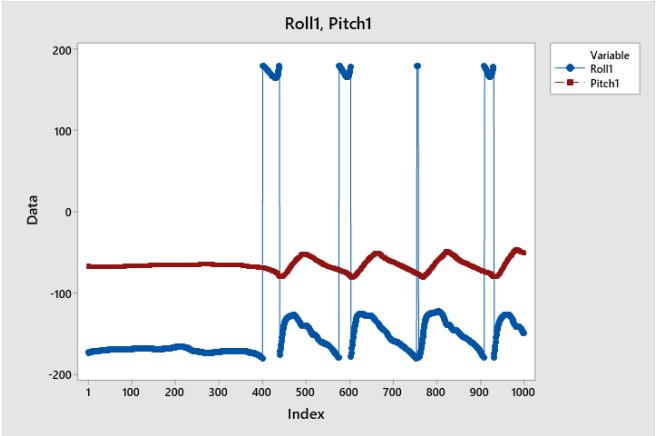
Confusion matrix akan menjadi instrumen pengujian yang digunakan dalam penelitian ini. Tahap pertama pengujian dilakukan untuk memilih nilai k yang memberikan nilai akurasi paling tinggi. Pengujian ini dilakukan terhadap KNN dengan menggunakan data uji yang sudah disiapkan sebelumnya. Nilai k terpilih akan diimplementasikan ke dalam sistem langsung untuk klasifikasi data uji. Tiga sensor akan mengambil data langsung dari subjek yang sedang bergerak. Data yang

terambil akan diklasifikasi dengan menggunakan KNN dengan nilai k terbaik yang didapatkan dari pengujian sebelumnya. Pengujian kedua akan dilakukan terhadap hasil klasifikasi dari aplikasi *desktop* untuk melihat akurasi final dari KNN dengan menggunakan *confusion matrix*. *Matrix* ini digunakan menghitung nilai *recall*, *precission*, dan *specificity* di setiap metode klasifikasi.

Bab IV
Hasil dan Pembahasan

4.1 Memilih Fitur

Proses ini dilakukan untuk mengurangi data agar memudahkan dalam proses klasifikasi. Di penelitian fitur roll dan yaw tidak digunakan. Karakteristik dari yaw tidak cocok digunakan didalam penelitian ini karena sifatnya yang dipengaruhi oleh medan magnetik. Roll diputuskan tidak digunakan setelah melihat dari grafik hasil sinyal roll seperti pada Gambar 4.1.



Gambar 4.1 Contoh grafik data roll

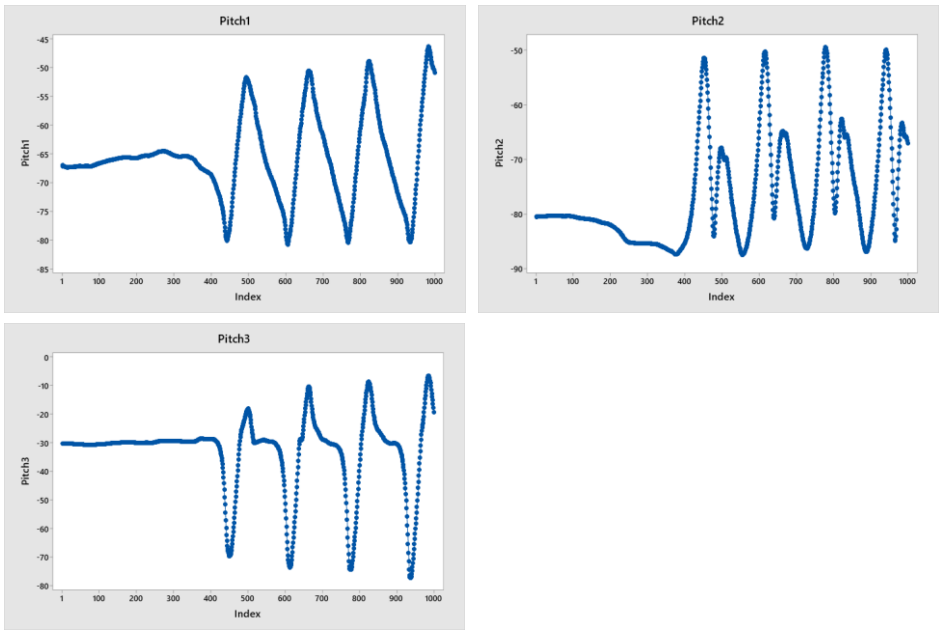
Gambar 4.1 membuktikan data fitur roll memiliki lonjakan-lonjakan yang ekstrem. Hal ini terjadi karena efek Gimbal Lock pada representasi sudut Euler yang digunakan dalam penelitian. Penelitian ini tidak menggunakan representasi Quaternion, maupun tidak dilakukan reset orientasi saat pengambilan data. Akibatnya data yang melebihi batas derajat akan terdeteksi sebagai data yang melonjak kebawah atau keatas Dengan melakukan reset orientasi, maka semua fitur (roll, pitch, dan yaw) akan diubah nilainya menjadi mendekati nol.

4.2 Hasil *Feature Extraction*

Feature extraction dilakukan untuk mengubah data latih yang digunakan untuk masukan dalam model KNN yang digunakan. Langkah awal yang dilakukan

sebelum melakukan *feature extraction* adalah memotong beberapa baris data yang dianggap tidak relevan. Pada subbab ini akan dijelaskan data apa yang dihapus dan hasil dari *feature extraction* yang akan dilakukan.

Data latih yang terdiri atas tiga fitur pitch dari tiga sensor akan dihitung *gradient* di setiap baris data yang hasilnya bisa dilihat pada Gambar 4.4. Nilai g*radient* digunakan untuk membantu menghapus data latih yang merekam ketika subjek sedang tidak bergerak, dengan begini proses klasifikasi tidak akan diganggu oleh data yang tidak bersangkutan dengan kegiatan berjalan dan naik-turun tangga.

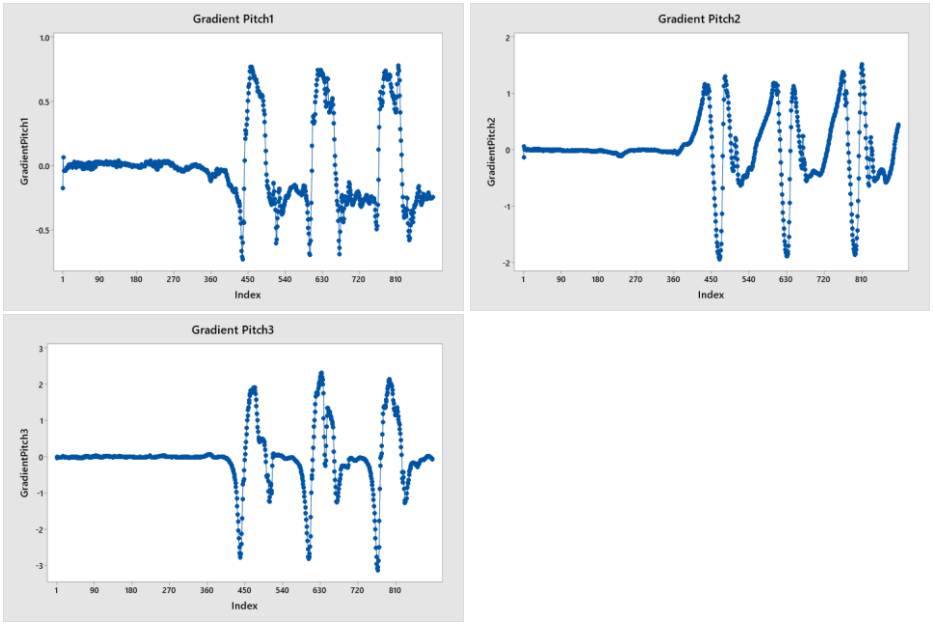


Gambar 4.2 Contoh potongan data pada fitur Pitch1(kiri atas), Pitch2 (kanan atas), Pitch3 (kiri bawah)

Pada Gambar 4.2 terlihat ada bagian grafik yang cenderung mendatar atau tidak memiliki lonjakan-lonjakan data, data tersebut merupakan data hasil rekaman ketika subjek sedang tidak bergerak. Subjek memang untuk berdiri diam di tempat dan diminta tidak bergerak sebelum dan sesudah perekaman. Pada gerakan berjalan subjek diminta berdiri diam di titik mulai seperti terlihat di Gambar 4.3

Gambar 4.3 Subjek berhenti di akhir lintasan

Gambar 4.4 adalah contoh hasil *gradient* di setiap baris data. Ketiga gambar di bawah masing-masing mewakili sensor pertama, kedua, dan ketiga pada subjek dan kegiatan yang sama.



Gambar 4.4 Grafik nilai g*radient* setiap data

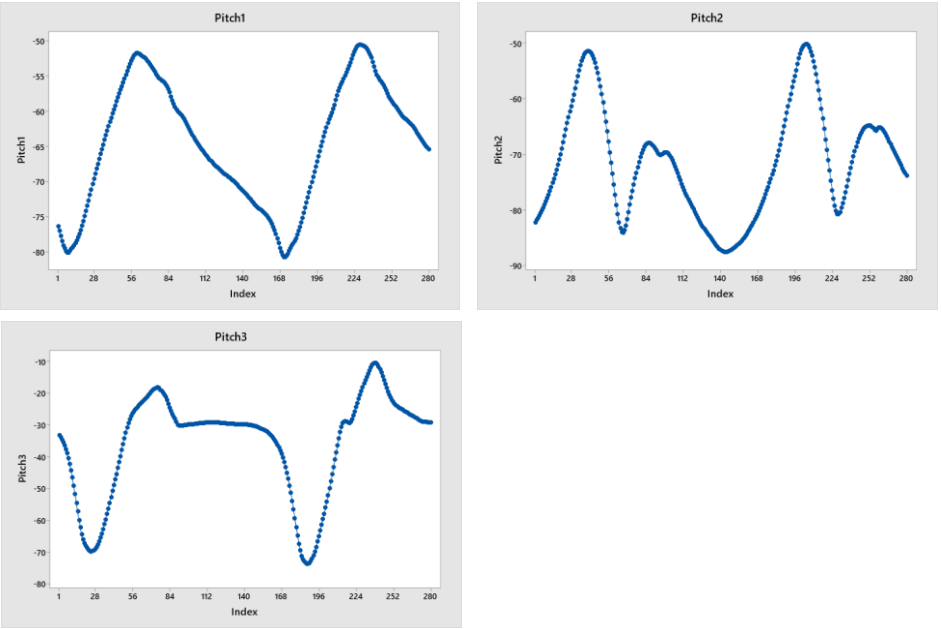
Gambar 4.4 di atas merupakan hasil penghitungan nilai *gradient* dari masing-masing baris data. Gambar 4.4 terlihat sebelum terjadinya lonjakan pertama, terdapat kumpulan nilai *gradient* yang membentuk garis mendatar. Kemudian menjelang terjadi puncak pertama, *gradient* akan secara bertahap berubah nilainya menjadi lebih besar atau lebih kecil. Perubahan nilai *gradient* ini menunjukkan subjek mulai bergerak sehingga nilai perekaman sensor mulai bervariasi, ini memberikan petunjuk kapan sebaiknya data mulai diambil untuk dijadikan data latih. Nilai awal *gradient* sebelum terjadinya penurunan atau kenaikan akan diambil sebagai *threshold.*

Tabel 4.1 berikut ini merupakan daftar *threshold* yang sudah dikumpulkan di setiap fitur dan di setiap kegiatan. Nilai *threshold* ini selanjutnya digunakan untuk membantu dalam memotong data latih lainnya. Nilai rata-rata dari semua *threshold* digunakan untuk membantu proses klasifikasi secara *realtime*.

Tabel 4.1 Daftar *threshold*

	Jalan	Naik	Turun
Pitch Sensor CC2	0,5	0,6	0,4
Pitch Sensor C42	0,35	0,5	0,6
Pitch Sensor C62	0,5	0,55	0,6
Rata - rata		0,51	

Nilai *threshold* di atas dikumpulkan dengan melihat berbagai subjek di kegiatan yang berbeda. Semua nilai *threshold* tersebut digunakan untuk memotong semua data latih yang sudah dikumpulkan. Hasil data latih yang sudah terpotong bisa dilihat pada Gambar 4.5



Gambar 4.5 Contoh data latih setelah dipotong

Data latih yang sudah dipotong dengan menggunakan *threshold* akan memiliki grafik yang tidak memiliki bagian yang mendatar lagi, melainkan langsung dimulai dari awal terjadinya gelombang pertama.

Satu subjek di satu percobaan akan menghasilkan 280 baris data yang diambil dari data yang sudah terpotong, sehingga total dari jumlah data latih yang terkumpulkan dirangkum pada Tabel 4.2.

Tabel 4.2 Jumlah data latih setelah dipotong

Kegiatan	Jumlah Data Latih
Jalan	5 orang × 7 perulangan × 280 baris × 3 fitur
Naik	5 orang × 7 perulangan × 280 baris × 3 fitur
Turun	5 orang × 7 perulangan × 280 baris × 3 fitur
Total	5 orang × 7 perulangan × 280 baris × 3 fitur × 3 kegiatan

Tabel 4.2 merupakan data latih dimana setiap subjek di satu percobaan memberikan 280 baris data dengan jumlah fitur sebanyak tiga yaitu Pitch sensor CC2, Pitch sensor C42, dan Pitch sensor C62. Sehingga data latih akan terbentuk atas lima subjek, dimana setiap subjek melakukan tiga kegiatan yaitu berjalan, naik tangga, dan turun tangga yang masing-masing kegiatan diulang sebanyak tujuh kali sehingga hasil akhir setiap perulangan berupa 280 baris data. Hasil gabungan dari lima subjek sebanyak 29400 baris dengan tiga kolom yaitu Pitch1, Pitch2, dan

Pitch3. 29400 merupakan gabungan dari semua kelas, sehingga setiap kelasnya masing-masing memiliki 9800 baris data.

Tabel 4.3 menjelaskan jumlah dimensi data latih setelah *feature extraction*.

Tabel 4.3 Jumlah data latih setelah *feature extraction*

Kegiatan	Jumlah Data Latih
Jalan	35 × 6 fitur
Naik	35 × 6 fitur
Turun	35 × 6 fitur
Total	105 × 6 fitur

Nilai dari *wave length* dan *mean absolute deviation* dihitung setiap 280 baris data. Total 29400 baris data sensor berisi nilai pitch dari tiga sensor berubah menjadi 105 baris data berisi nilai *wave length* dan *mean absolute deviation*. 105 baris data ini merupakan gabungan dari ketiga kelas, setiap kelasnya memiliki 35 baris data seperti yang sudah dirangkum pada Tabel 4.3

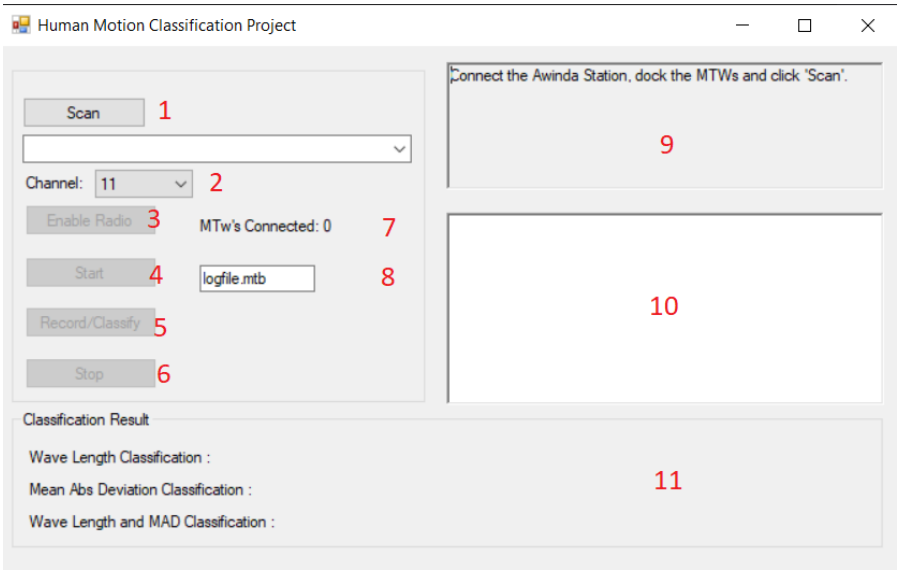
Proses *feature extraction* dilakukan di setiap fitur dengan menghitung nilai *wave length* dan *mean absolute deviation* terangkum pada Tabel 4.4 berikut. *Feature extraction* mengubah satu fitur menjadi dua fitur dimana kedua fitur tersebut masing-masing merupakan hasil dari menghitung *wave length* dan *mean absolute deviation*.

Tabel 4.4 Perubahan fitur setelah *feature extraction*

Fitur sebelum *feature extraction*	Fitur setelah *feature extraction*
Pitch1	waveLengthPitch1 madPitch1
Pitch2	waveLengthPitch2 madPitch2
Pitch3	waveLengthPitch3 madPitch3

4.3 Tampilan Antarmuka Aplikasi

Gambar 4.6 merupakan tampilan awal yang terlihat ketika program dijalankan dan menjadi satu-satunya halaman yang dimiliki. Aplikasi klasifikasi dibuat dengan menggunakan bahasa C# dan *framework Windows Form Application*.



Gambar 4.6 Tampilan antarmuka aplikasi

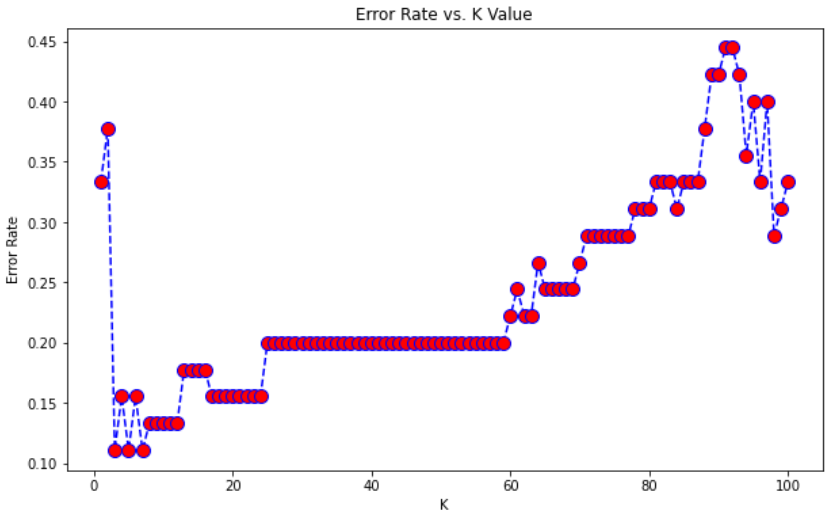
Tombol bernomor satu digunakan untuk mendeteksi Awinda Station yang terhubung melalui *port usb* dengan perangkat peneliti. Awinda station yang terhubung akan ditampilkan pada dropdown pada bagian nomor dua. Tombol tiga digunakan setelah Awinda Station berhasil dideteksi. Tombol ini berfungsi untuk menghubungkan antara semua sensor yang aktif dengan Awinda Station, jumlah sensor yang berhasil terhubung akan ditampilkan pada area bernomor tujuh. Tombol empat digunakan untuk memulai proses penerimaan data yang dikirim oleh sensor bersamaan dengan menampilkan datanya. Data yang diterima akan ditampilkan di area kotak bernomor sepuluh.

Tombol lima berfungsi untuk memulai proses klasifikasi. Pastikan untuk menentukan nama *file* yang ingin disimpan dengan mengganti teks pada kotak input bernomor delapan. Data yang dikirim oleh sensor akan disimpan sebanyak 1120 baris. Aplikasi akan berhenti ketika data yang dikumpulkan sudah mencapai 1120 baris atau ketika tombol nomor enam ditekan. Hasil klasifikasi akan ditampilkan pada area kotak bernomor sebelas. Area kotak bernomor sembilan menjadi tempat untuk menampilkan sedikit panduan dalam mengoperasikan aplikasi dan kotak bernomor delapan untuk mengatur nama berkas dari hasil perekaman data.

4.4 Hasil Pencarian K-NN Terbaik

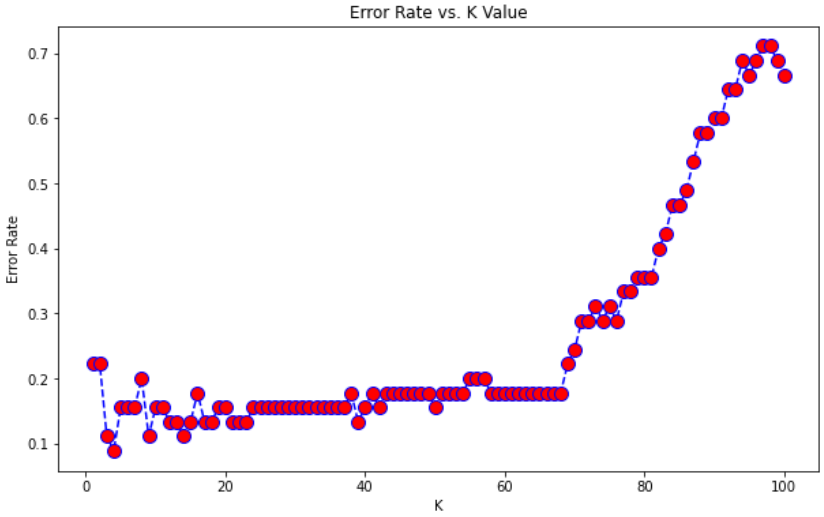
Mencari nilai k terbaik dilakukan dengan cara mencoba nilai k = 1 sampai dengan k = 100. Pencarian nilai k dilakukan terhadap ketiga metode klasifikasi,

sehingga akan didapatkan tiga nilai k dengan nilai *error* yang terkecil. Gambar 4.7 merupakan hasil *error rate* yang diujikan terhadap model klasifikasi dengan menggunakan fitur *wave length* saja.



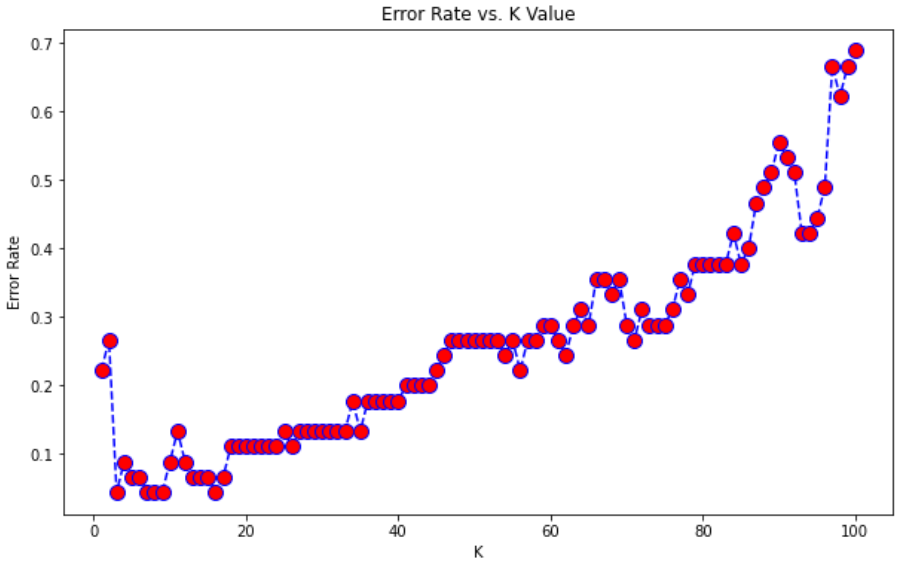
Gambar 4.7 *Error rate* metode *wave length*

Gambar 4.8 berikut ini menunjukkan hasil pengujian tingkat kegagalan klasifikasi mulai dari k = 1 sampai dengan k = 100 dengan menggunakan fitur *mean absolute deviation* saja. Berdasarkan grafiknya, semakin besar nilai k, semakin besar pula tingkatan kesalahan klasifikasi terjadi.



Gambar 4.8 *Error rate* metode *mean absolute deviation*

Gambar 4.9 menguji tingkat kesalahan klasifikasi dengan menggunakan fitur *wave length* dan *mean absolute deviation*



Gambar 4.9 *Error rate* metode gabungan *wave length* dan *mean absoulete deviation*

Nilai k terbaik pada metode klasifikasi menggunakan *Wave Length* didapatkan pada k = 3, 5, 7. Ketiga nilai k ini memberikan nilai *error rate* yang sama senilai 0,111. Keputusan akhir untuk metode klasifikasi menggunakan *wave length* akan dipilih nilai k = 3, dari hasil pengujian, nilai k = 3 menjadi yang pertama dalam menghasilkan tingkat terjadinya *error* yang paling kecil sehingga tidak perlu memilih nilai k = 5 atau k = 7. Metode klasifikasi menggunakan *mean absolute deviation* mendapatkan nilai k dengan *error rate* terkecil sebesar 0,088 pada k = 4, sedangkan nilai *error rate* terkecil pada proses klasifikasi menggunakan kedua fitur sebesar 0,0444 berhasil dicapai dengan k sebesar 3.

Tabel 4.5 merangkum hasil akurasi dan *error rate* dari ketiga metode beserta dengan nilai k yang paling optimal.

Tabel 4.5 Rangkuman *akurasi* dan *error rate*

Fitur	*Error rate*	Akurasi	Jumlah k
Wave Length	0,11	0,88	3
Mean Absolute Deviation	0,08	0,91	4
Gabungan Wave Length dan Mean Absolute Deviation	0,04	0,95	3

4.5 Hasil Klasifikasi Realtime

Klasifikasi *realtime* dilakukan dengan tiga metode klasifikasi. Pada Tabel 4.6 adalah rangkuman akurasi dari ketiga metode yang dihitung dari jumlah klasifikasi benar yang didapatkan.

Penjelasan dari masing-masing metode akan dijelaskan pada subbab berikut ini.

Tabel 4.6 Rangkuman akurasi secara keseluruhan

Fitur	Akurasi Keseluruhan
Wave Length	0,31
Mean Absolute Deviation	0,5
Keduanya	0,65

4.5.1 Hasil Klasifikasi Realtime Wave Length

Tabel 4.7 adalah hasil rangkuman keluaran klasifikasi data tes secara *realtime* menggunakan fitur *wave length*.

Tabel 4.7 Hasil klasifikasi *realtime wave length*

		PREDIKSI		
		Jalan	Naik	Turun
AKTUAL	Jalan	**27**	3	6
	Naik	24	**5**	7
	Turun	26	8	**2**

Tabel 4.8 menjelaskan hasil *confusion matrix* dari data pada Tabel 4.7 beserta besaran nilai akurasi, *precision*, *recall*, dan *specificity*.

Tabel 4.8 *Confusion matrix wave length*

	TP	TN	FP	FN	Akurasi	Precision	Recall	Specificity
Jalan	27	22	50	9	0,45	0,35	0,75	0,30
Naik	5	61	11	31	0,61	0,31	0,13	0,84
Turun	2	59	13	34	0,56	0,13	0,05	0,81

Pada Tabel 4.8 merangkum penghitungan *confusion matrix* dari hasil klasifikasi *wave length* pada Tabel 4.7. Dari Tabel 4.8 tersebut dapat diketahui besar akurasi yang didapatkan dari klasifikasi *wave length*. Dapat diketahui pula kemampuan model untuk membedakan kelas jalan terhadap kelas negatif cukup

tinggi, sebaliknya kemampun model kurang baik dalam membedakan kelas naik dan turun dengan kelas lainnya.

Model juga tidak memberikan performa yang baik dalam mengklasifikasikan kelas naik dan turun terlihat dari nilai *recall* yang kecil, sebaliknya model berhasil membedakan antara kelas jalan dan kelas lainnya dengan baik.

Tabel 4.9 Hasil klasifikasi setiap subjek fitur *wave length*

Subjek	Kegiatan	Klasifikasi		Akurasi
		Benar	Salah	
1	Jalan	4	2	66,66%
	Naik	0	6	0%
	Turun	0	6	0%
2	Jalan	5	1	83,33%
	Naik	4	2	66,66%
	Turun	1	5	16,66%
3	Jalan	4	2	66,66%
	Naik	0	6	0%
	Turun	1	5	16,66%
4	Jalan	4	2	66,66%
	Naik	1	5	16,66%%
	Turun	0	6	0%
5	Jalan	4	2	66,66%
	Naik	0	6	0%
	Turun	0	6	0%
6	Jalan	6	0	100%
	Naik	0	6	0%
	Turun	0	6	0%

Data Tabel 4.9 menunjukkan akurasi klasifikasi terhadap setiap subjek di setiap kegiatan. Terlihat klasifikasi terhadap gerakan naik dan turun tidak memberikan akurasi yang baik.

4.5.2 Hasil Klasifikasi *Realtime Mean Absolute Deviation*

Tabel 4.10 adalah hasil rangkuman keluaran klasifikasi data tes secara *realtime* menggunakan fitur *mean absolute deviation.*

Tabel 4.10 Hasil klasifikasi *realtime mean absolute deviation*

		PREDIKSI		
		Jalan	Naik	Turun
AKTUAL	Jalan	**18**	18	0
	Naik	11	**25**	0
	Turun	24	1	**11**

Tabel 4.11 menjelaskan hasil *confusion matrix* dari data pada Tabel 4.10 beserta besaran nilai akurasi, *precision*, *recall*, dan *specificity*.

Tabel 4.11 *Confusion matrix mean absolute deviation*

	TP	TN	FP	FN	Akurasi	Precision	Recall	Specificity
Jalan	18	37	35	18	0,50	0,33	0,5	0,51
Naik	25	53	19	11	0,72	0,56	0,69	0,73
Turun	11	72	0	25	0,76	1	0,30	1

Klasifikasi dengan menggunakan *mean absolute deviation* sedikit lebih baik dibandingkan dengan klasifikasi menggunakan *wave length*. Pada Tabel 4.11 terlihat ada sedikit peningkatan pada akurasi, dari nilai *precision* dapat diketahui model juga lebih baik dalam mengklasifikasikan kelas dan mampu membedakan setiap kelas, kecuali pada kelas jalan. Nilai *precision* pada kegiatan jalan lebih kecil dibandingkan dengan kelas lainnya, menunjukkan banyak kelas naik atau turun yang terklasifikasi sebagai kelas jalan. Kelas turun memiliki nilai *precision* satu sehingga model mampu membedakan kelas lainnya dengan kelas turun.

Data Tabel 4.12 berisi rangkuman hasil klasifikasi terhadap setiap subjek di setiap kegiatan.

Tabel 4.12 Hasil klasifikasi setiap subjek fitur *mean absolute deviation*

Subjek	Kegiatan	Klasifikasi		Akurasi
		Benar	Salah	
1	Jalan	6	0	100%
	Naik	0	6	0%
	Turn	0	6	0%
2	Jalan	5	1	83,33%
	Naik	4	2	66,66%
	Turun	1	5	20%
3	Jalan	0	6	0%
	Naik	6	0	100%
	Turun	0	6	0%
4	Jalan	6	0	100%
	Naik	1	5	20%
	Turun	5	1	83,33%
5	Jalan	0	6	0%
	Naik	6	0	100%
	Turun	1	5	20%

Tabel 4.12 Lanjutan

Subjek	Kegiatan	Klasifikasi		Akurasi
		Benar	Salah	
6	Jalan	0	6	0%
	Naik	6	0	100%
	Turun	0	6	0%

Pada klasifikasi menggunakan fitur ini sedikit lebih baik dibandingkan dengan hanya menggunakan *wave length* saja.

4.5.3 Hasil Klasifikasi Gabungan Dua Fitur

Tabel 4.13 adalah hasil rangkuman keluaran klasifikasi secara *realtime* menggunakan gabungan kedua fitur.

Tabel 4.13 Hasil klasifikasi gabungan dua fitur

		PREDIKSI		
		Jalan	Naik	Turun
AKTUAL	Jalan	**31**	5	0
	Naik	9	**27**	0
	Turun	0	23	**13**

Tabel 4.14 menjelaskan hasil *confusion matrix* dari data pada Tabel 4.13 beserta besaran nilai akurasi, *precision*, *recall*, dan *specificity*.

Tabel 4.14 *Confusion matrix* gabungan dua fitur

	TP	TN	FP	FN	Akurasi	Precision	Recall	Specificity
Jalan	31	63	9	5	0,87	0,77	0,86	0,87
Naik	27	44	28	9	0,65	0,49	0,75	0,61
Turun	13	72	0	23	0,78	1	0,36	1

Tabel 4.13 merangkum hasil klasifikasi setiap kegiatan. Bila dibandingkan dengan dua metode lainnya, klasifikasi dengan menggabungkan dua fitur memberikan hasil yang lebih baik. Terlihat jumlah kesalahan klasifikasi di kelas jalan ada lima, berbeda dengan metode *wave length* dimana kesalahan klasifikasi terjadi sebanyak sembilan kali, sedangkan pada metode *mean absolute deviation* kesalahan klasifikasi pada gerakan berjalan ada 18 data.

Klasifikasi pada gerakan naik tangga pada metode gabungan dua fitur berhasil mengklasifikasi dengan benar sejumlah 27 data dan salah mengklasifikasi sebanyak sembilan data. Peningkatan yang besar dibandingkan dengan melakukan klasifikasi menggunakan *wave length* saja yang mana berhasil mengklasifikasi sebanyak lima data saja dan salah mengklasifikasi gerakan tersebut sebagai gerakan berjalan sebanyak 24 data. Metode klasifikasi *mean absolute deviation* cukup baik dalam mengklasifikasi gerakan naik tangga dibandingkan dengan metode sebelumnya.

Klasifikasi menggunakan *wave length*, *mean absolute deviation*, dan gabungan kedua fitur tidak berhasil mengklasifikasi gerakan turun. Perbedaan dari ketiga fitur ini pada proses klasifikasi gerakan turun tangga adalah metode *wave length* dan *mean absolute deviation* menganggap gerakan tersebut sebagai gerakan berjalan, sedangkan klasifikasi dengan menggunakan kedua fitur mengklasifikasikan gerakan turun tangga sebagai gerakan naik tangga.

Tabel 4.14 memiliki akurasi yang lebih baik dibandingkan dengan kedua metode lainnya. Model masih memiliki banyak kesalahan dalam mengklasifikasi gerakan naik dan turun.

Tabel 4.15 berikut ini merangkum jumlah klasifikasi yang benar dan salah disetiap subjek.

Tabel 4.15 Hasil klasifikasi setiap subjek gabungan dua fitur

Subjek	Kegiatan	Klasifikasi		Akurasi
		Benar	Salah	
1	Jalan	6	0	100%
	Naik	6	0	100%
	Turun	6	0	100%
2	Jalan	6	0	100%
	Naik	0	6	0%
	Turun	0	6	0%
3	Jalan	5	1	83,33%
	Naik	6	0	100%
	Turun	0	6	0%
4	Jalan	6	0	100%
	Naik	3	3	50%
	Turun	6	0	100%
5	Jalan	5	1	83,33%
	Naik	6	0	100%
	Turun	1	5	20%

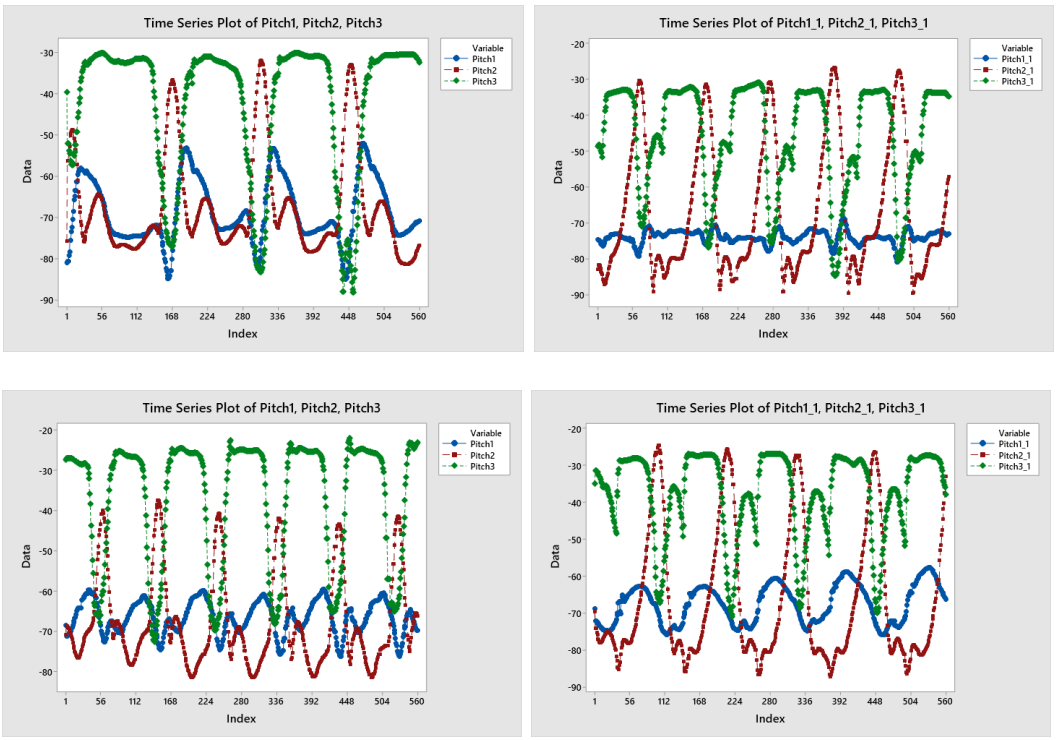
Tabel 4.15 Lanjutan

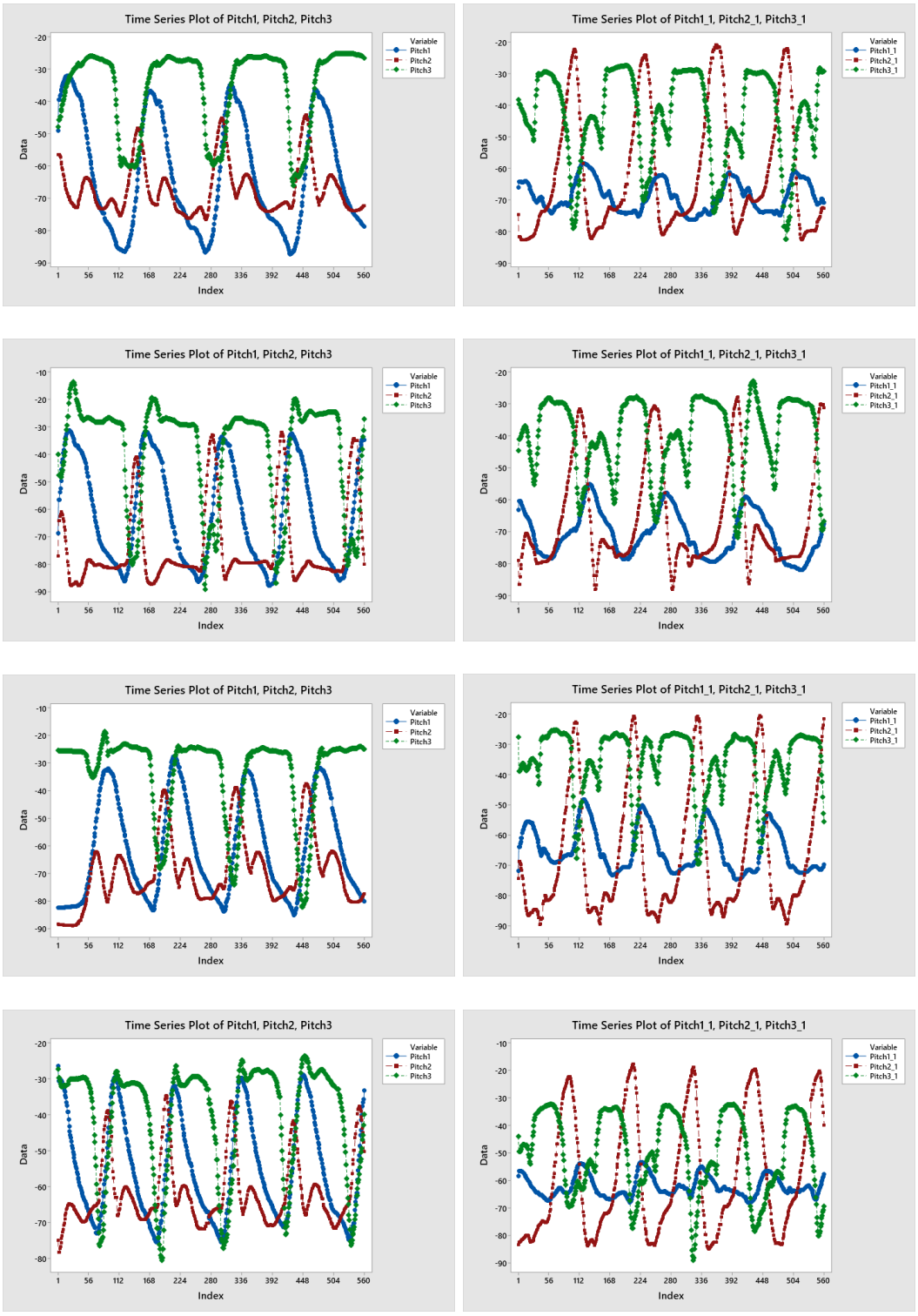
Subjek	Kegiatan	Klasifikasi		Akurasi
		Benar	Salah	
6	Jalan	3	3	50%
	Naik	6	0	100%
	Turun	0	6	0%

Tabel 4.15 menunjukkan peningkatan akurasi yang lebih baik. Lebih banyak ditemukan hasil klasifikasi yang sesuai dengan kenyataannya, tapi masih banyak ditemukan ada kesalahan klasifikasi pada gerakan naik-turun tangga.

4.6 Pengujian Naik-Turun Tangga

Hasil klasifikasi terhadap gerakan naik-turun tangga menunjukkan banyak terjadi kesalahan. Hal ini mengarah pada perlunya mengetahui bagaimana pola data yang dihasilkan dari kedua gerakan tersebut. Gambar berikut ini adalah grafik dari rekaman data sensor terhadap kedua gerakan tersebut.





(Naik) (Turun)

Gambar 4.10 Grafik perekaman data naik-turun tangga

Gambar 4.10 merupakan perwakilan dari setiap subjek di satu percobaan. Dari gambar tersebut terlihat ada sedikit kemiripan antara gerakan naik dan turun

tangga. Tabel 4.16 ini berisikan pengujian terhadap gerakan tersebut dengan membandingkan setiap fitur yang digunakan di setiap subjek menggunakan *cross correlation.*

Tabel 4.16 Pengujian kemiripan

Subjek	Percobaan		Lag (k)	CCF(x)	$\frac{2}{\sqrt{n-\lvert k\rvert}}(y)$	Status $x > y$
1	1	Pitch1	13	0,14	0,08	Korelasi
		Pitch2	17	0,17	0,08	Korelasi
		Pitch3	29	0,11	0,08	Korelasi
	2	Pitch1	11	0,26	0,08	Korelasi
		Pitch2	-10	-0,54	0,08	Korelasi
		Pitch3	-33	0,68	0,08	Korelasi
	3	Pitch1	-30	0,25	0,08	Korelasi
		Pitch2	-12	-0,38	0,08	Korelasi
		Pitch3	-33	0,42	0,08	Korelasi
2	1	Pitch1	2	-0,06	0,08	Tidak
		Pitch2	4	-0,17	0,08	Korelasi
		Pitch3	-20	0,19	0,08	Korelasi
	2	Pitch1	-6	-0,30	0,08	Korelasi
		Pitch2	-14	-0,42	0,08	Korelasi
		Pitch3	0	-0,22	0,08	Korelasi
	3	Pitch1	-29	-0,42	0,08	Korelasi
		Pitch2	16	0,47	0,08	Korelasi
		Pitch3	32	0,35	0,08	Korelasi
3	1	Pitch1	4	-0,45	0,08	Korelasi
		Pitch2	-18	-0,32	0,08	Korelasi
		Pitch3	17	-0,34	0,08	Korelasi
	2	Pitch1	15	0,39	0,08	Korelasi
		Pitch2	-14	0,43	0,08	Korelasi
		Pitch3	33	0,24	0,08	Korelasi
	3	Pitch1	9	-0,79	0,08	Korelasi
		Pitch2	-17	-0,57	0,08	Korelasi
		Pitch3	33	-0,54	0,08	Korelasi
4	1	Pitch1	-26	0,62	0,08	Korelasi
		Pitch2	-25	0,48	0,08	Korelasi
		Pitch3	-13	0,33	0,08	Korelasi
	2	Pitch1	33	-0,82	0,08	Korelasi
		Pitch2	-33	0,62	0,08	Korelasi
		Pitch3	-28	0,50	0,08	Korelasi
	3	Pitch1	-13	-0,28	0,08	Korelasi
		Pitch2	-29	-0,22	0,08	Korelasi
		Pitch3	-17	-0,21	0,08	Korelasi
5	1	Pitch1	6	0,48	0,08	Korelasi
		Pitch2	-8	0,47	0,08	Korelasi
		Pitch3	13	0,43	0,08	Korelasi

Tabel 4.16 Lanjutan

Subjek	Percobaan		Lag (k)	CCF(x)	$\frac{2}{\sqrt{n-\lvert k\rvert}}$ (y)	Status $x > y$
5	2	Pitch1	-33	0,79	0,08	Korelasi
		Pitch2	-33	0,66	0,08	Korelasi
		Pitch3	-23	0,85	0,08	Korelasi
	3	Pitch1	2	-0,45	0,08	Korelasi
		Pitch2	-12	-0,48	0,08	Korelasi
		Pitch3	23	-0,38	0,08	Korelasi
6	1	Pitch1	8	0,77	0,08	Korelasi
		Pitch2	-1	0,72	0,08	Korelasi
		Pitch3	21	0,72	0,08	Korelasi
	2	Pitch1	28	-0,17	0,08	Korelasi
		Pitch2	7	-0,42	0,08	Korelasi
		Pitch3	-11	0,34	0,08	Korelasi
	3	Pitch1	21	0,47	0,08	Korelasi
		Pitch2	2	0,61	0,08	Korelasi
		Pitch3	-24	-0,59	0,08	Korelasi

x = Cross Correlation Function (CCF)
n = 560 (2 stride)
k = nilai lag dengan nilai CCF tertinggi
y = 2 / √(|n – k |)

Tabel 4.16 menunjukkan hampir semua fitur yang dipakai berkorelasi kecuali subjek kedua, artinya fitur antara gerakan naik-turun tangga memiliki data mirip. Hal ini merupakan salah satu kemungkinan nilai akurasi pada klasifikasi gerakan naik dan turun tangga nilainya rendah seperti telah ditunjukkan pada Tabel 4.15. Total data yang berkorelasi ada 53 data, sehingga persentase data yang berkorelasi adalah 98,14%.

4.7 Perbandingan Hasil Klasifikasi

Hasil dari penelitian ini dilakukan perbandingan terhadap penelitian serupa lainnya yang terangkum dalam Tabel 4.17.

Tabel 4.17 Perbandingan akurasi hasil klasifikasi

	Metode	Praproses	Sensor	Posisi	Kegiatan	Akurasi
Li et al	Triplet Markov Model	-	1	Metatarsal	Jalan, lari, naik, turun	99,2%, data subjek lain
Lewi et al	KNN (Manhattan)	Penghilangan *noise*	-	-	Tidur, berdiri, lari	99,4% Data validasi

Tabel 4.17 Lanjutan

	Metode	Praproses	Sensor	Posisi	Kegiatan	Akurasi
Calisan dan Talu	KNN, w*indowing*, 128 baris	Feature extraction (mean, deviation, maximum eigenvalues)	1	Dada	Naik, turun, duduk, berdiri, lari, jalan	98,5% data subjek lain
Ranakoti et al	SVM	Akselerasi, perubahan akselerasi, rata-rata akselerasi, standard deviasi, Z-score	-	-	Berdiri, jalan, lari, naik-turun tangga, jatuh	78,04% data subjek lain
Penelitian ini	KNN (Euclidean) ,w*indowing* 280 baris	*Feature extraction (wave length, mean absolute deviation)*	3	Paha, di atas mata kaki sisi medial, *metatarsal*	Jalan, naik-turun tangga	65,74% data subjek lain

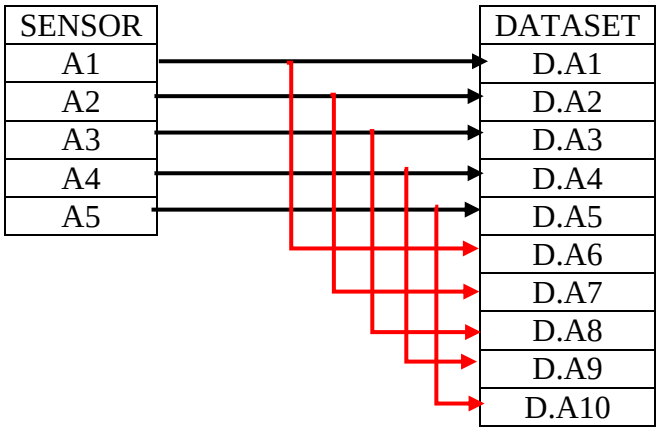
Tabel diatas dapat diketahui hasil penelitian ini tidak memberikan akurasi setinggi penelitian lainnya. Didalam penelitian ini mengadopsi beberapa teknik pengolahan data namun hasil yang didapatkan masih kurang dibandingkan dengan teknik-teknik pengolahan data serta metode klasifikasi lainnya.

4.8 Riwayat Pembuatan Model

Hasil penelitian yang dilakukan melewati banyak percobaan dalam pembuatan data latih untuk KNN. Berikut ini beberapa percobaan yang telah dilakukan dalam mencari bentuk data latih untuk klasifikasi

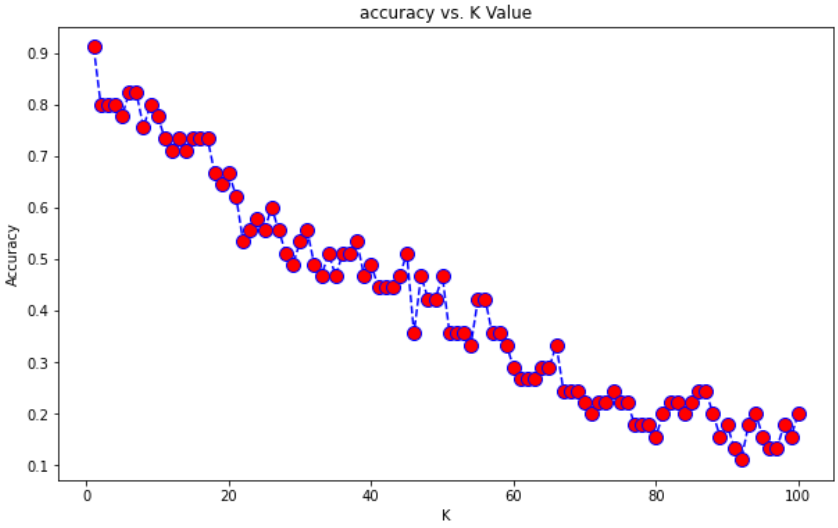
4.8.1 Klasifikasi *Windowed*

Metode klasifikasi ini dilakukan tanpa melakukan *feature extraction*. Fitur yang digunakan adalah pitch dan roll. Klasifikasi dilakukan dengan menghitung setiap baris data *realtime* berukuran 280 baris terhadap data latih dengan memperhatikan posisi *index* dari setiap datanya. Proses klasifikasi dilakukan dengan ilustrasi pada gambar berikut



Gambar 4.11 Contoh data latih dan data *realtime*

Klasifikasi dilakukan dengan menghitung Euclidean setiap baris sesuai dengan posisinya di setiap ukuran potongan data. Contoh bila klasifikasi dilakukan setiap lima data sensor maka Euclidean dihitung antara A1 dengan D.A1, A2 dengan D.A2, A3 dengan D.A3 dan seterusnya sampai D.A5. Setelah kelima data dihitung didapatkan satu buah nilai Euclidean. Proses ini diulang kembali mulai dari A1 sampai dengan A5 terhadap D.A6 sampai D.A10. Hasil menghitung Euclidean tersebut menjadi nilai Euclidean yang kedua. Gambar 4.11 menjelaskan proses klasifikasi pada metode Klasifikasi Windowed dimana sesungguhnya klasifikasi atau proses menghitung Euclidean dilakukan setiap 280 baris data.



Gambar 4.12 Akurasi klasifikasi *windowed*

Gambar 4.12 adalah hasil akurasi menggunakan data latih dimana akurasi terbaik sebesar 91% dengan k bernilai 1. Akurasi k = 2, 3, dan 4 bernilai 80%. Dan akurasi k = 6 dan 7 sebesar 82%. Metode ini memberikan akurasi yang cukup baik, tetapi dari alur proses klasifikasi yang sudah dijelaskan pada Gambar 4.12 menyebabkan banyak perulangan yang harus dilakukan untuk mendapatkan satu klasifikasi, hal ini dapat mencegah klasifikasi menjadi lambat.

4.8.2 Klasifikasi dengan Feature Extraction

Percobaan pada subbab ini memiliki kesamaan dengan metode klasifikasi final yang digunakan pada penelitian ini, hanya saja pembedanya adalah hasil dari *feature extraction* digabung menjadi satu nilai. Hasil dari *feature extraction* terangkum dalam Tabel 4.18 berikut.

Tabel 4.18 Rangkuman hasil *feature extraction wave length*

Fitur Awal	Fitur Hasil Feature Extraction
Pitch1	Wave length gabungan
Pitch2	
Pitch3	

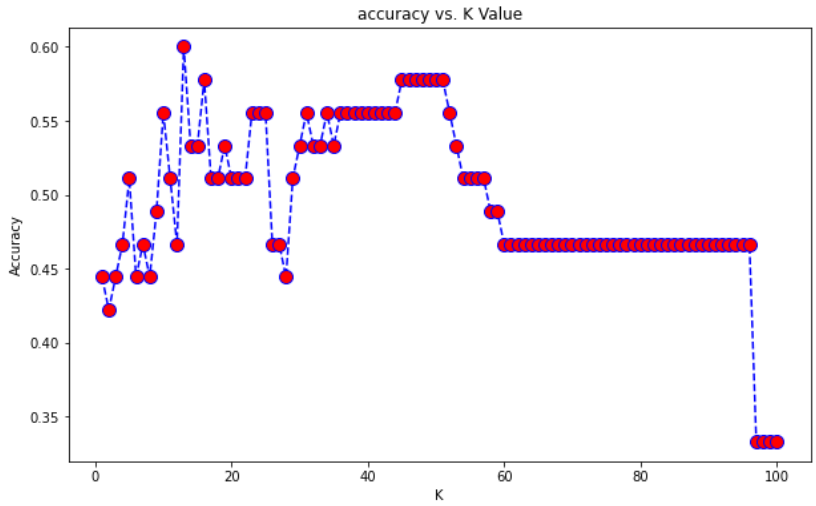
Ketiga fitur yang dipakai yaitu Pitch1, Pitch2, dan Pitch3 akan dihitung menjadi satu nilai *wave length* saja. Proses ini dilakukan serupa dalam mendapatkan nilai *mean absolute deviation*, ketiga variabel tersebut akan terangkum menjadi satu nilai. Rangkuman hasil *feature extraction mean absolute deviation* dapat dilihat pada Tabel 4.19 berikut.

Tabel 4.19 Rangkuman hasil *feature extraction mean absolute deviation*

Fitur Awal	Fitur Hasil Feature Extraction
Pitch1	*Mean absolute deviation* gabungan
Pitch2	
Pitch3	

Sehingga dari metode *feature extraction* di atas total fitur yang digunakan hanya ada dua yaitu satu nilai *wave length* dan satu nilai *mean absolute deviation*. Proses pengujian terhadap metode ini juga dilakukan di setiap fitur dan gabungan dari kedua fitur.

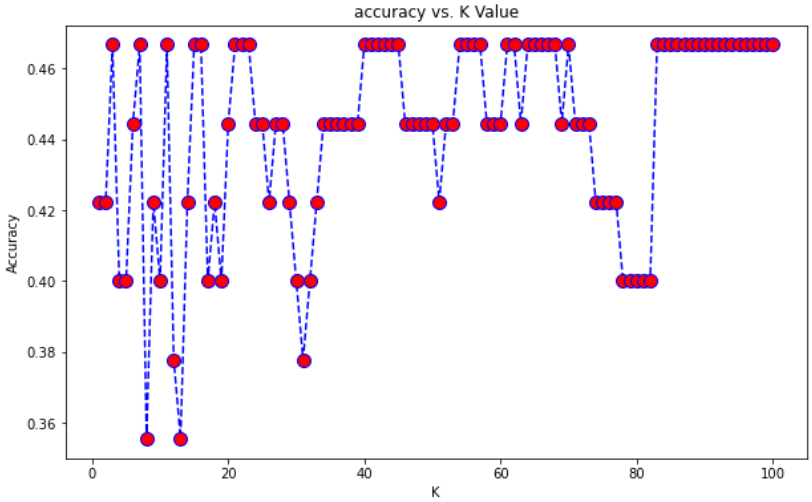
Hasil pengujian akurasi klasifikasi menggunakan fitur *wave length* dapat dilihat pada Gambar 4.13 berikut.



Gambar 4.13 Pengujian akurasi menggunakan *wave length*

Klasifikasi dengan menggunakan wave length saja memberikan akurasi tertinggi sebesar 60% pada k bernilai 13, sedangkan akurasi pada nilai k lainnya cenderung lebih kecil.

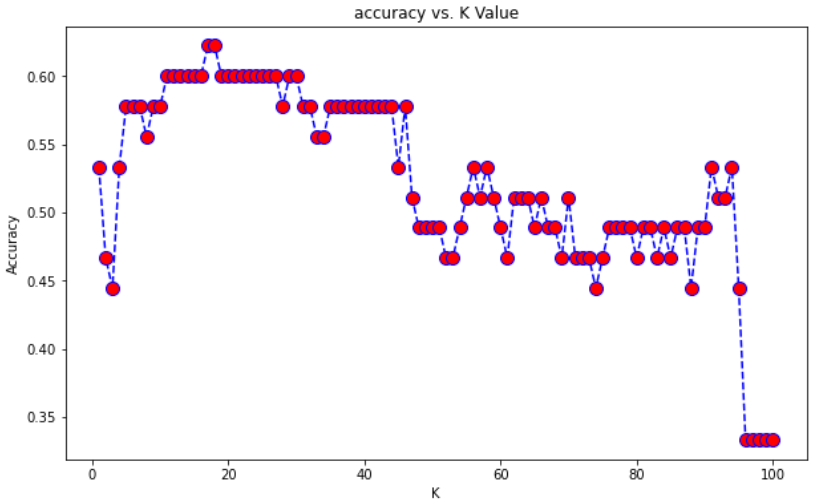
Hasil pengujian akurasi pada klasifikasi menggunakan mean absolute deviation dapat dilihat pada Gambar 4.14 berikut.



Gambar 4.14 Pengujian akurasi menggunakan *mean absolute deviation*

Akurasi menggunakan fitur ini kurang memuaskan dengan akurasi tertinggi sebesar 46,67%. Selain akurasi yang kecil, nilai yang didapatkan juga tidak konstan sehingga sulit untuk menentukan nilai k yang terbaik.

Akurasi yang didapatkan bila klasifikasi dilakukan dengan menggunakan kedua fitur dapat dilihat pada Gambar 4.15.



Gambar 4.15 Pengujian akurasi menggunakan kedua fitur

Klasifikasi pada metode ini sedikit lebih baik dibandingkan dengan ketika menggunakan *wave length* saja. Akurasi tertinggi sebesar 62% dengan k bernilai 17. Semakin tinggi nilai k, akurasi cenderung semakin menurun.

Bab V
Kesimpulan dan Saran

5. 1 Kesimpulan

Menggunakan KNN dalam mengklasifikasikan gerakan secara *realtime* tidak menimbulkan masalah dari segi kecepatannya. Hal ini disebabkan karena data yang digunakan sebagai data latih dan data uji tidak banyak, sehingga perulangan yang harus dilakukan untuk mendapatkan satu klasifikasi tidak memakan banyak waktu.

KNN tidak berhasil memberikan klasifikasi yang baik pada metode *wave length* dan *mean absolute deviation*. Banyak gerakan-gerakan naik dan turun tangga diklasifikasikan sebagai berjalan. Klasifikasi menggunakan gabungan dari kedua fitur memberikan peningkatan akurasi tapi hasil yang diberikan masih belum memuaskan. Pengujian data gerakan naik-turun tangga membuktikan kedua gerakan tersebut memiliki kemiripan yang tinggi sehingga klasifikasi masih sulit untuk membedakan gerakan tersebut.

Proses *training* untuk mencari nilai k terbaik berhasil mendapatkan akurasi yang memuaskan, tetapi pada proses klasifikasi secara *realtime* tidak memberikan persentase keberhasilan yang sama dengan proses *training*. Hal ini membuktikan bahwa nilai k optimal dalam proses *training* tidak dapat digunakan dalam klasifikasi gerakan secara *realtime*.

5.2 Saran

Ada beberapa variabel yang dapat mempengaruhi besar akurasi dalam penelitian ini

1. Letak pemasangan sensor
2. Metode klasifikasi yang digunakan
3. Metode ekstraksi fitur yang digunakan
4. Tidak menggunakan fitur reset ketika pengambilan data

Berdasarkan variabel-variabel di atas, maka saran yang dapat diberikan adalah mengganti letak pemasangan sensor di tulang kering sisi medial menjadi sisi

lateral atau mengkombinasikan letak pemasangan sensor seperti di paha dan betis di kedua kaki atau di dada untuk mendapatkan *center of gravity* tubuh.

Mencari metode klasifikasi lain yang lebih akurat dan mampu untuk mendeteksi kemunculan pola dari suatu data. Selain itu perlu mencari metode ekstraksi fitur lain sehingga bisa diketahui metode mana yang lebih cocok bila memang metode klasifikasi yang digunakan membutuhkan fitur ekstraksi. Saran terakhir adalah mengingat adanya lonjakan data yang tidak terkontrol maka sebaiknya dipertimbangkan menggunakan fitur reset orientasi sensor bila memang API yang disediakan memungkinkan untuk melakukan itu.

Daftar Pustaka

Ahmad, A., 2017. Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning. *Jurnal Teknologi Indonesia.*

Brownlee, J., 2020. *What is a Confusion Matrix in Machine Learning.* [Online] Available at: https://machinelearningmastery.com/confusion-matrix-machine-learning/ [Diakses 8 10 2021].

Calisan, M. & Talu, M. F., 2020. Comparison of Methods for Determining Activity from Physical Movements. *Journal of Polytechnic.*

Ismail, A. M., 2018. *Cara Kerja Algoritma k-Nearest Neighbor (k-NN).* [Online] Available at: https://medium.com/bee-solution-partners/cara-kerja-algoritma-k-nearest-neighbor-k-nn-389297de543e [Diakses 5 Mei 2021].

Kim, S., Nozaki, T. & Murakami, T., 2016. An Approach to Categorization Analysis for Human Motion by Kinect and IMU. *IEEEXplore.*

Kulikov, P., Dietrich, E., Wagner, B. & Victor, Y., 2021. *The history of C#.* [Online] Available at: https://docs.microsoft.com/en-us/dotnet/ csharp /whats-new/csharp-version-history [Diakses 9 Oktober 2021].

Lewi, E. B., Wijayanto, I. & Patmasari, R., 2021. Studi Penggunaan Tapis Wavelet Untuk Penghilangan Derau Sinyal Sensor Unit Pengukuran Inersia Pada Sistem Pengenalan Postur. *e-Proceeding of Engineering,* Volume 8, p. 292.

Li, H., Derrode, S. & Pieczynski, W., 2019. An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model. *Elsevier,* Volume 362, pp. 94-105.

Manchola, M. D. S., Bernal, M. J. P., Munera, M. & Cifuentes, C. A., 2019. Gait Phase Detection for Lower-Limb Exoskeletons using Foot Motion Data from a Single Inertial Measurement Unit in Hemiparetic Individuals. *Sensors,* Volume 19.

Matplotlib, 2002. [Online] Available at: https://matplotlib.org/ [Diakses 7 Agustus 2021].

Muhardian, A., 2018. *Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal.* [Online] Available at: https://www.petanikode.com /python-linux/ [Diakses 5 Mei 2021].

NumPy, 2019. *About Us.* [Online] Available at: https://numpy.org/about/[Diakses 7 Agustus 2021].

Pandas Team, 2014. *About Pandas.* [Online] Available at: https://pandas.pydata.org/about/index.html [Diakses 4 Agustus 2021].

Ranakoti, S. et al., 2019. Human Fall Detection System over IMU Sensors Using Triaxial Accelerometer. Volume 1, pp. 495-507.

Sathya, R. & Abraham, A., 2013. Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. *International Journal of Advanced Research in Artificial Intelligence,* Volume 2, pp. 29-33.

Scikit Learn, 2007. *About us.* [Online] Available at: https://scikit-learn.org/stable/about.html [Diakses 7 Agustus 2021].

Sohom, P., 2019. *History of Python.* [Online] Available at: https://www.geeksforgeeks.org/history-of-python/ [Diakses 5 Mei 2021].

Soleha, N., Syauqy, D. & Setiawan, E., 2019. Sistem Deteksi dan Klasifikasi Pergerakan Kepala Menggunakan K Nearest neighbor. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.*

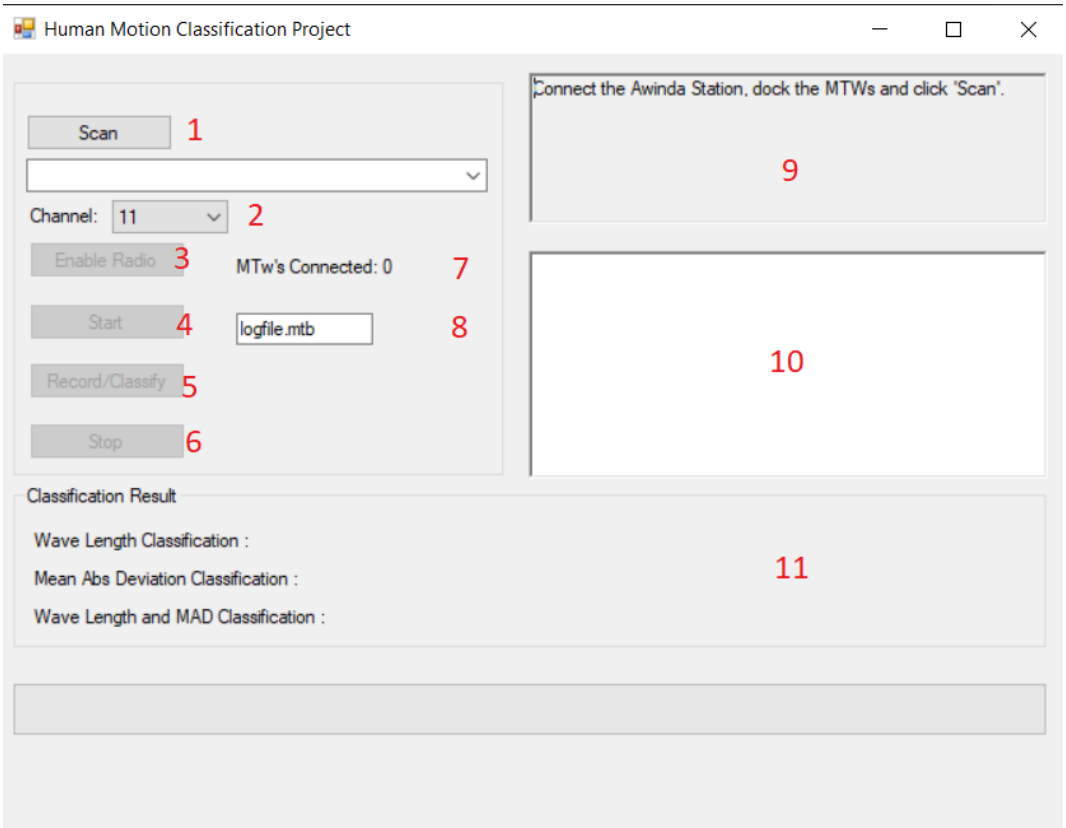
W3School, t.thn. *C# Tutorial.* [Online] Available at: https://www.w3schools.com /cs/index.php [Diakses 30 September 2021].

Waskom, M., 2012. *seaborn: statistical data visualization.* [Online] Available at: https://seaborn.pydata.org/ [Diakses 7 Agustus 2021].

Widodo, R. B. & Wada, C., 2016. Artificial Neural Network Based Step-Length Prediction Using Ultrasonic Sensors from Simulation to Implementation in Shoe-Type Measurement Device. *Fuji Technology Press,* Volume 21, pp. 321-329.

Lampiran

A



Gambar A. 1 UI aplikasi klasifikasi

1. Pasang antena pada Awinda Station. Hubungkan Awinda Station dengan dengan perangkat *desktop* menggunakan usb.

Gambar A.2 Awinda station dengan antena

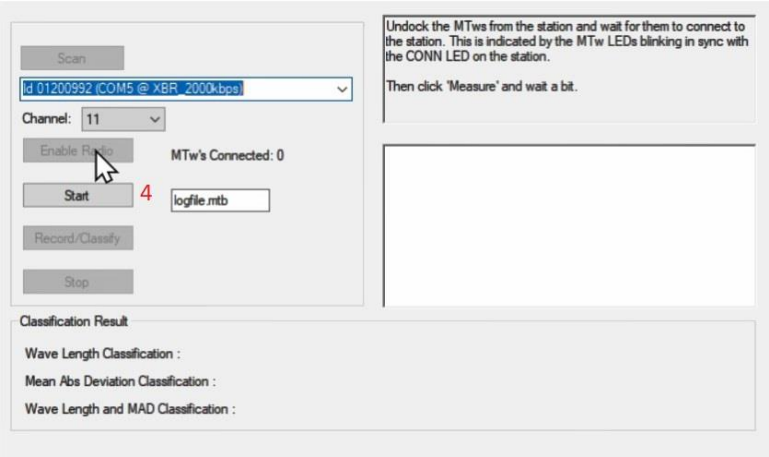
2. Tekan tombol Scan bertanda satu. Tombol ini berfungsi untuk mendeteksi apakah ada Awinda Station yang terhubung. Awinda Station yang terdeteksi akan ditampilkan *dropdown* seperti pada gambar .

Gambar A.3 Awinda station terdeteksi

3. Channel digunakan untuk memilih gelombang yang ingin digunakan. Fitur ini dipakai bila di area sekitar terdapat perangkat-perangkat lain yang memiliki panjang gelombang yang sama.
4. Nyalakan sensor IMU dengan menekan tombol *power* sampai lampu led menyala.
5. Tombol Enable Radio bertanda nomor tiga akan aktif setelah proses mendeteksi Awinda Station berhasil. Tekan tombol tersebut untuk menyambungkan sensor dengan Awinda Station.

Gambar A.4 Tombol Enable Radio

6. Setelah menekan tombol Enable Radio, tombol Start akan aktif dan sudah bisa ditekan. Tetapi jangan ditekan dahulu, lanjutkan pada langkah berikutnya.

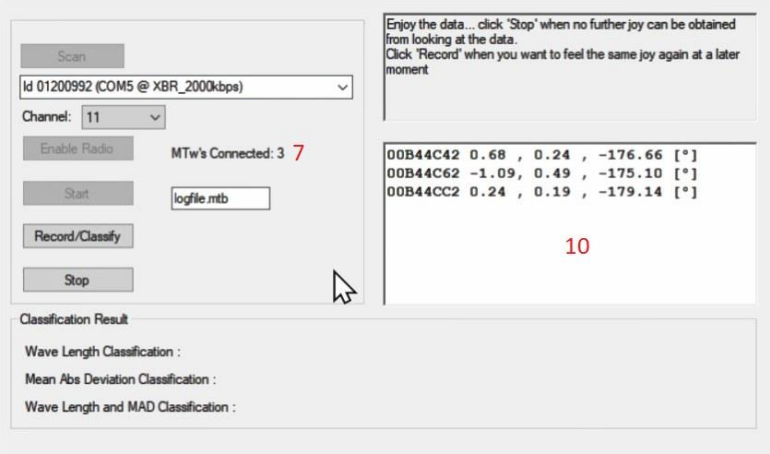


Gambar A. 5

7. Sebelum menekan tombol Start. Pastikan lampu led pada sensor berkelip-kelip selaras dengan nyala lampu pada Awinda Station.

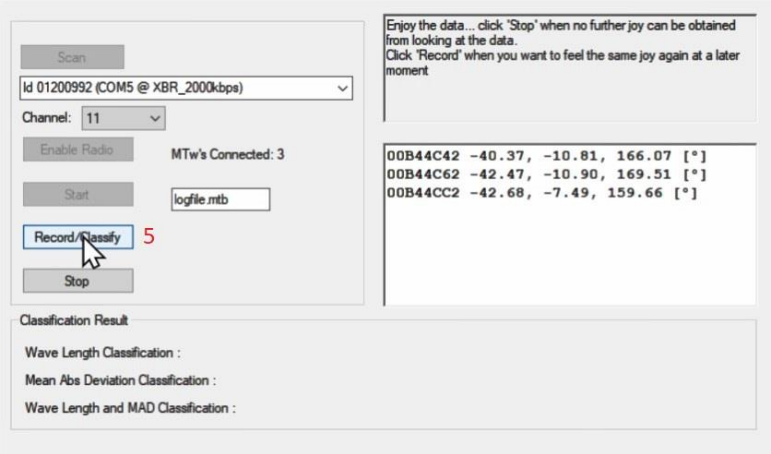
Gambar A. 6 Nyala lampu sensor selaras dengan Awinda Station

8. Tekan tombol Start untuk memulai pengiriman data dari sensor ke Awinda Station. Hasil data sensor yang diterima akan ditampilkan di area kotak bernomor sepuluh. Jumlah sensor yang terhubung akan ditampilkan pada label bernomor tujuh.



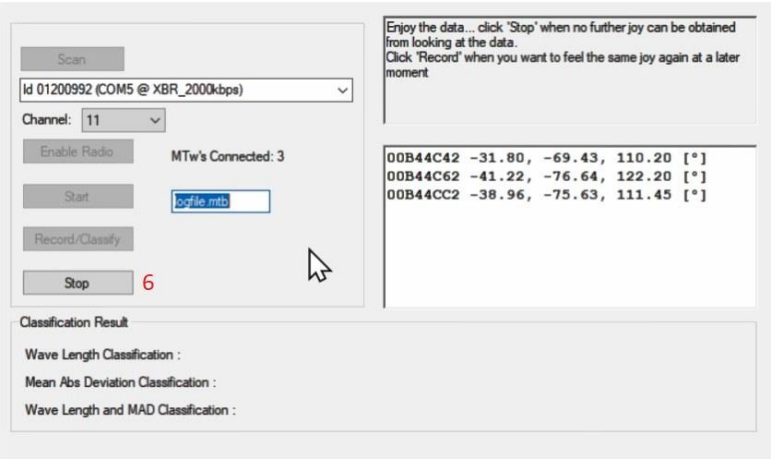
Gambar A.7 Hasil menekan tombol Start

9. Input *text* bernomor tujuh digunakan untuk mengatur nama berkas yang akan tersimpan ketika tombol Classify ditekan. Pastikan untuk mengubah isi dari input *text* bila ingin menggunakan nama yang lain sebelum melakukan klasifikasi.
10. Tombol Classify berfungsi untuk menyimpan data sekaligus mengumpulkan data untuk dilakukan klasifikasi. Setelah tombol Record ditekan, hanya menyisakan tombol Stop saja yang aktif. Proses klasifikasi dan mengumpulkan data akan memakan waktu selama beberapa detik



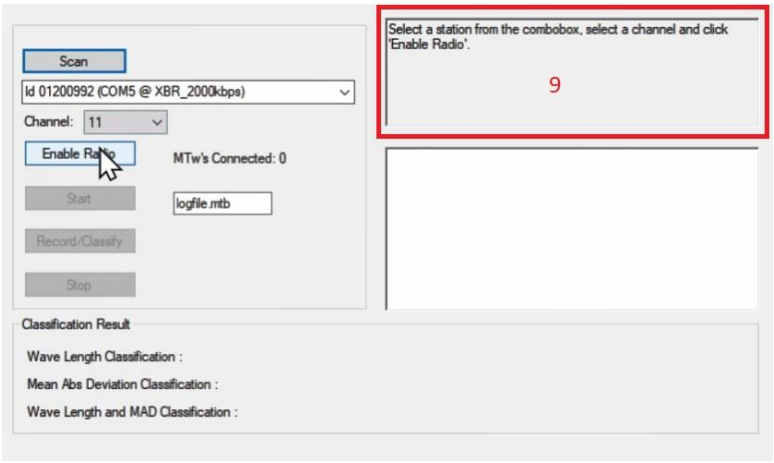
Gambar A.8 Tombol Record

11. Tombol Stop, ditandai dengan nomor enam berfungsi untuk menghentikan aplikasi.



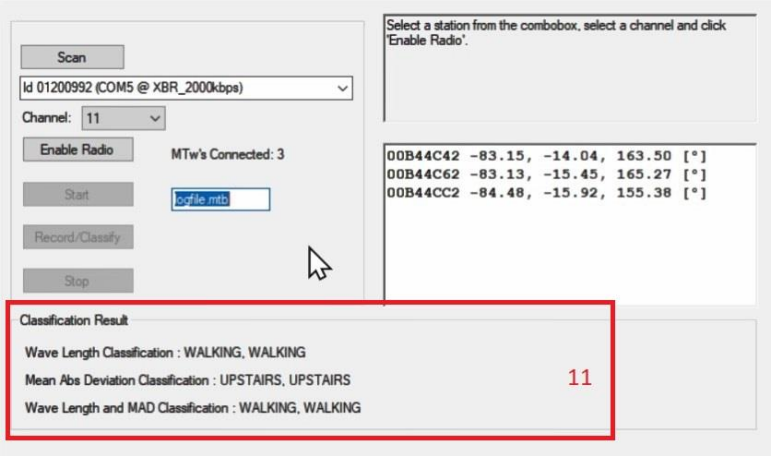
Gambar A.9 UI ketika sedang merekam

12. Kotak di kanan atas ditandai dengan nomor sembilan berfungsi untuk menampilkan sedikit panduan dalam penggunaan aplikasi



Gambar A.10 Panduan aplikasi

13. Kotak bernomor sebelas adalah teks yang digunakan untuk menampilkan hasil klasifikasi.



Lampiran

B

Table B.1 Hasil klasifikasi subjek 1

		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Jalan	Jalan	Jalan	Jalan	Jalan	Jalan
	2	Turun	Jalan	Jalan	Jalan	Jalan	Jalan
	3	Jalan	Naik	Jalan	Jalan	Jalan	Jalan
Naik	1	Jalan	Jalan	Naik	Naik	Naik	Naik
	2	Turun	Jalan	Naik	Naik	Naik	Naik
	3	Turun	Jalan	Naik	Naik	Naik	Naik
Turun	1	Jalan	Jalan	Turun	Turun	Turun	Turun
	2	Naik	Jalan	Turun	Turun	Turun	Turun
	3	Jalan	Jalan	Turun	Jalan	Turun	Turun

Table B.2 Hasil klasifikasi subjek 2

		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Jalan	Jalan	Jalan	Jalan	Jalan	Jalan
	2	Jalan	Turun	Jalan	Jalan	Jalan	Jalan
	3	Jalan	Jalan	Jalan	Jalan	Jalan	Jalan
Naik	1	Turun	Naik	Jalan	Jalan	Jalan	Jalan
	2	Naik	Naik	Jalan	Jalan	Jalan	Jalan
	3	Turun	Naik	Jalan	Jalan	Jalan	Jalan
Turun	1	Jalan	Jalan	Jalan	Jalan	Naik	Naik
	2	Naik	Jalan	Jalan	Jalan	Naik	Naik
	3	Jalan	Turun	Jalan	Jalan	Naik	Naik

Table B.3 Hasil klasifikasi subjek 3

		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Jalan	Jalan	Naik	Naik	Jalan	Jalan
	2	Jalan	Turun	Naik	Naik	Jalan	Jalan
	3	Jalan	Turun	Naik	Naik	Naik	Jalan
Naik	1	Jalan	Jalan	Naik	Naik	Naik	Naik
	2	Jalan	Jalan	Naik	Naik	Naik	Naik
	3	Jalan	Jalan	Naik	Naik	Naik	Naik
Turun	1	Naik	Turun	Jalan	Jalan	Naik	Naik
	2	Jalan	Naik	Jalan	Jalan	Naik	Naik
	3	Naik	Naik	Jalan	Jalan	Naik	Naik

Table B.4 Hasil klasifikasi subjek 4

		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Naik	Jalan	Jalan	Jalan	Jalan	Jalan
	2	Naik	Jalan	Jalan	Jalan	Jalan	Jalan
	3	Jalan	Jalan	Jalan	Jalan	Jalan	Jalan
Naik	1	Turun	Jalan	Jalan	Jalan	Naik	Naik
	2	Jalan	Jalan	Naik	Jalan	Naik	Jalan
	3	Jalan	Naik	Jalan	Jalan	Jalan	Jalan
Turun	1	Jalan	Jalan	Turun	Jalan	Turun	Turun
	2	Jalan	Jalan	Turun	Turun	Turun	Turun
	3	Jalan	Jalan	Turun	Turun	Turun	Turun

Table B.5 Hasil klasifikasi subjek 5

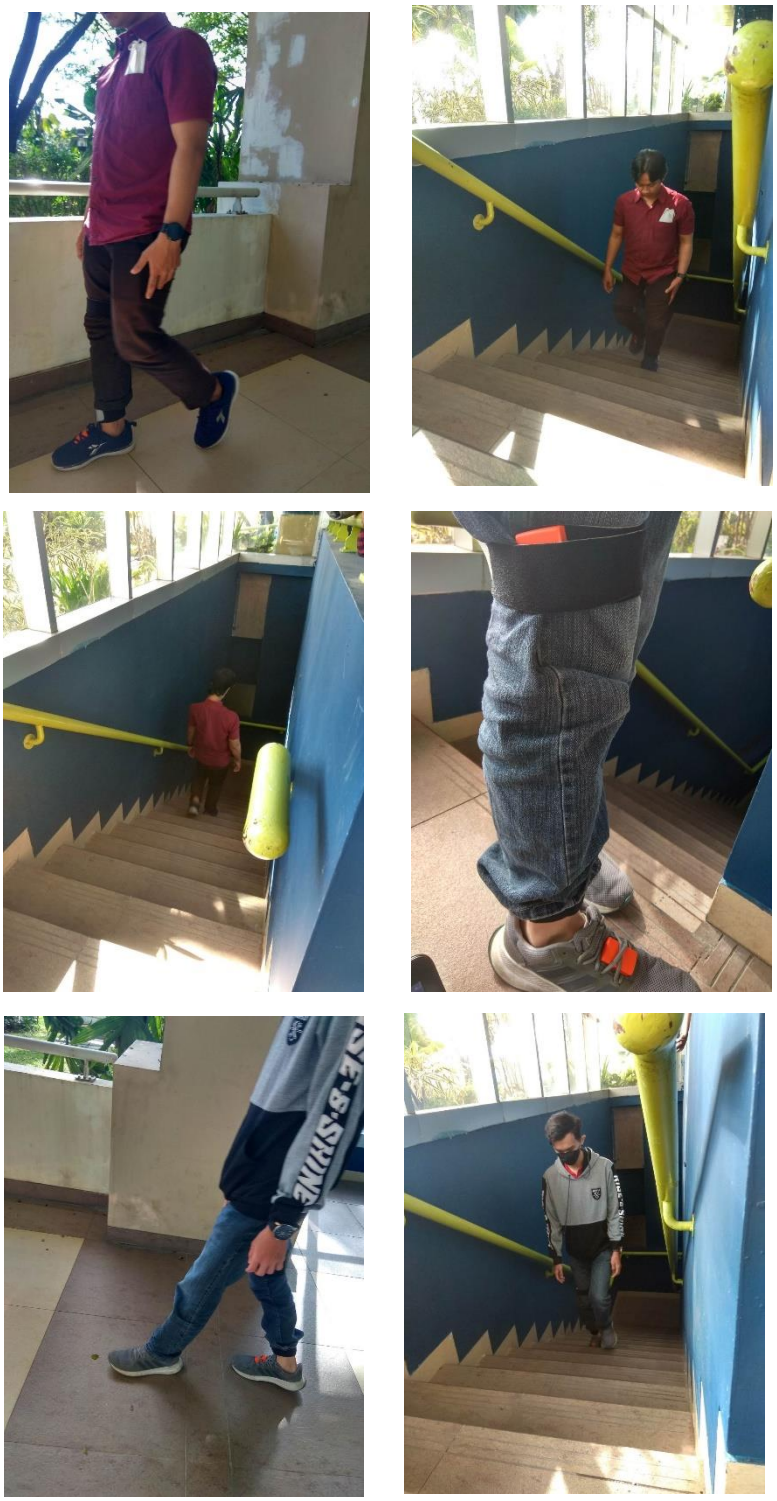
		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Jalan	Jalan	Naik	Naik	NAIK	Jalan
	2	Jalan	Turun	Naik	Naik	Jalan	Jalan
	3	Jalan	Turun	Naik	Naik	Jalan	Jalan
Naik	1	Jalan	Jalan	Naik	Naik	Naik	Naik
	2	Turun	Jalan	Naik	Naik	Naik	Naik
	3	Jalan	Jalan	Naik	Naik	Naik	Naik
Turun	1	Jalan	Jalan	Jalan	Jalan	Naik	Naik
	2	Jalan	Jalan	Jalan	Turun	Naik	Turun
	3	Jalan	Jalan	Jalan	Jalan	Naik	Naik

Table B.6 Hasil klasifikasi subjek 6

		Wave Length		MAD		Gabungan	
		Stride 1	Stride 2	Stride 1	Stride 2	Stride 1	Stride 2
Jalan	1	Jalan	Jalan	Naik	Naik	Naik	Jalan
	2	Jalan	Jalan	Naik	Naik	Naik	Jalan
	3	Jalan	Jalan	Naik	Naik	Naik	Jalan
Naik	1	Jalan	Jalan	Naik	Naik	Naik	Naik
	2	Turun	Jalan	Naik	Naik	Naik	Naik
	3	Jalan	Jalan	Naik	Naik	Naik	Naik
Turun	1	Jalan	Jalan	Jalan	Jalan	Naik	Naik
	2	Jalan	Jalan	Jalan	Naik	Naik	Naik
	3	Naik	Naik	Jalan	Jalan	Naik	Naik

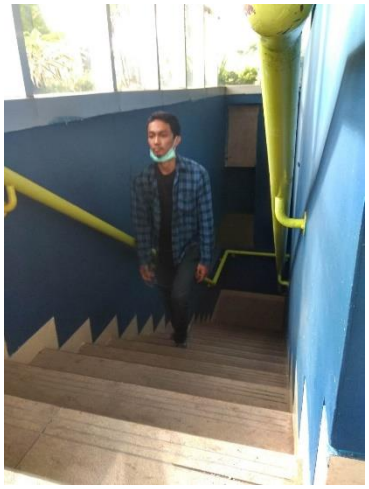
Lampiran

C



Gambar C.1 Dokumentasi pengujian jalan, naik-turun tangga



Gambar C.2 Dokumentasi Pengujian Subjek