

**DETEKSI KEJATUHAN DENGAN SENSOR ACCELEROMETER DAN  
GYROSCOPE BERBASIS ARDUINO MENGGUNAKAN JARINGAN  
SYARAF TIRUAN**

**(Skripsi)**

**Oleh**

**EZZA AHMAD FATUROHMAN**



**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG**

**2025**

## **ABSTRAK**

# **DETEKSI KEJATUHAN DENGAN SENSOR ACCELEROMETER DAN GYROSCOPE BERBASIS ARDUINO MENGGUNAKAN JARINGAN SYARAF TIRUAN**

**Oleh**

**EZZA AHMAD FATUROHMAN**

Jatuh adalah kejadian di mana seseorang tiba-tiba kehilangan keseimbangan dan jatuh ke tanah atau permukaan lain tanpa kontrol, terutama pada lansia, karena bisa menyebabkan cedera serius atau bahkan kematian jika tidak segera ditangani. Penelitian ini bertujuan membuat sistem deteksi kejatuhan menggunakan sensor *accelerometer* dan *gyroscope* yang terdapat pada modul MPU6050 dan dipasang pada sebuah rompi alat yang berbasis Arduino. Data dari sensor ini kemudian diproses menggunakan Jaringan Syaraf Tiruan (JST) dan dengan metode backpropagation untuk mengenali jenis gerakan. Data diambil dari 23 orang dengan total 392 pola gerakan yang terdiri dari delapan jenis gerakan termasuk gerakan jatuh dan gerakan aktivitas sehari-hari, diantaranya yaitu jatuh ke depan, jatuh ke samping, jatuh ke belakang, berdiri ke duduk, berdiri ke tidur, berdiri ke rukuk, berdiri ke sujud, dan rukuk ke sujud. Penelitian ini menggunakan dua skenario pengujian, skenario pertama membedakan delapan jenis gerakan secara spesifik, sedangkan skenario kedua membagi gerakan hanya menjadi dua kategori, yaitu jatuh dan tidak jatuh. Pelatihan dan pengujian model dilakukan menggunakan *software* Matlab. Dari hasil pengujian, sistem menunjukkan bahwa pada skenario pertama memperoleh nilai akurasi sebesar 93,75%, presisi 96%, dan recall 80%. Sedangkan pada skenario kedua, sistem mencapai akurasi 98,44%, presisi 96%, dan recall 100% dalam mendeteksi gerakan jatuh.

Kata kunci; Jatuh, MPU6050, Jaringan Syaraf Tiruan, Backpropagation, Matlab.

## **ABSTRACT**

### **FALL DETECTION WITH ACCELEROMETER AND GYROSCOPE SENSORS BASED ON ARDUINO USING ARTIFICIAL NEURAL NETWORKS**

**By**

**EZZA AHMAD FATUROHMAN**

*Fall is an incident where a person suddenly loses balance and falls to the ground or another surface uncontrollably. This is especially dangerous for the elderly, as it can cause serious injuries or even death if not immediately addressed. This research aims to develop a fall detection system using an accelerometer and gyroscope sensor embedded in the MPU6050 module, which is mounted on a vest-based device powered by Arduino. The sensor data is processed using an Artificial Neural Network (ANN) with the backpropagation method to recognize types of movements. The data was collected from 23 participants, resulting in a total of 392 motion patterns consisting of eight types of movements, including fall-related and daily activity movements. Among them are falling forward, falling sideways, falling backward, standing to sitting, standing to lying down, standing to bowing, standing to prostration, and bowing to prostration. This research was conducted using two testing scenarios. The first scenario classifies eight specific types of movements, while the second scenario categorizes movements into two classes, fall and non-fall. The training and testing processes were carried out using MATLAB software. The testing results show that in the first scenario, the system achieved an accuracy of 93.75%, a precision of 96%, and a recall of 80%. In the second scenario, the system achieved an accuracy of 98.44%, a precision of 96%, and a recall of 100% in detecting fall movements.*

*Keywords: Fall, MPU6050, Artificial Neural Network, Backpropagation, MATLAB.*

**DETEKSI KEJATUHAN DENGAN SENSOR ACCELEROMETER DAN  
GYROSCOPE BERBASIS ARDUINO MENGGUNAKAN JARINGAN  
SYARAF TIRUAN**

**Oleh**

**EZZA AHMAD FATUROHMAN**

**Skripsi**

**Sebagai Salah Satu Syarat Untuk Memperoleh Gelar  
SARJANA TEKNIK**

**Pada**

**Jurusan Teknik Elektro  
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK  
JURUSAN TEKNIK ELEKTRO  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

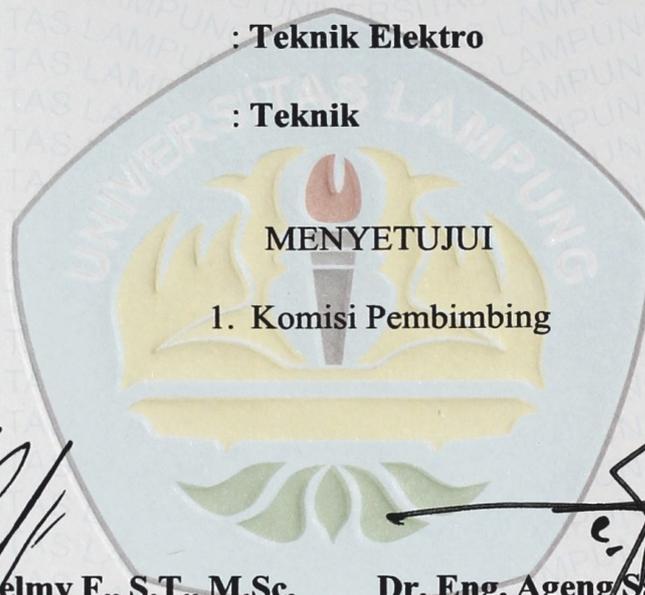
Judul Skripsi : **DETEKSI KEJATUHAN DENGAN SENSOR ACCELEROMETER DAN GYROSCOPE BERBASIS ARDUINO MENGGUNAKAN JARINGAN SYARAF TIRUAN**

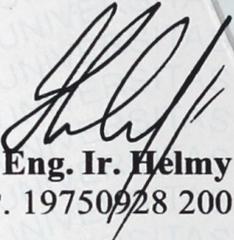
Nama Mahasiswa : **Ezza Ahmad Faturohman**

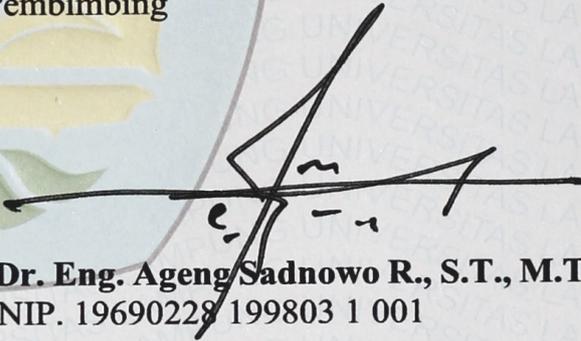
Nomor Pokok Mahasiswa : **1815031038**

Jurusan : **Teknik Elektro**

Fakultas : **Teknik**



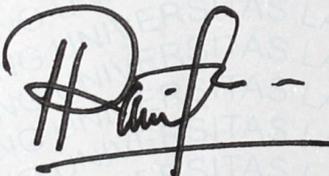
  
**Dr. Eng. Ir. Helmy F., S.T., M.Sc.**  
NIP. 19750928 200112 1 002

  
**Dr. Eng. Ageng Sadnowo R., S.T., M.T.**  
NIP. 19690228 199803 1 001

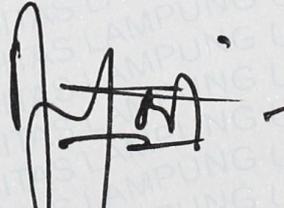
## 2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik Elektro



**Herlinawati, S.T., M.T.**  
NIP. 19710314 199903 2 001

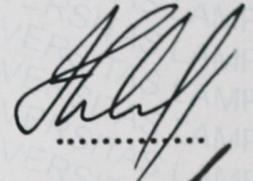


**Sumadi, S.T., M.T.**  
NIP. 19731104 200003 1 001

## MENGESAHKAN

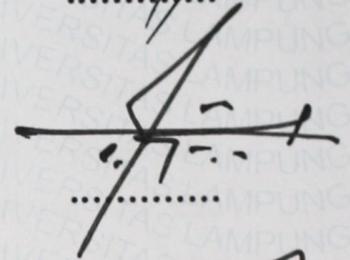
### 1. Tim Penguji

Ketua : **Dr. Eng. Ir. Helmy F., S.T., M.Sc.**



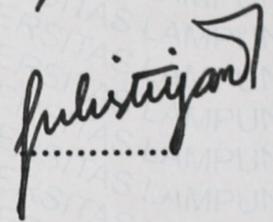
.....

Sekretaris : **Dr.Eng.Ageng Sadnowo R., S.T., M.T.**



.....

Penguji : **Dr. Sri Ratna S., S.T., M.T**



.....

### 2. Dekan Fakultas Teknik

**Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. }**  
NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi: **4 Juni 2025**

## SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Ezza Ahmad Faturohman

NPM : 1815031038

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 17 Juni 2025

   
EZZA AHMAD FATUROHMAN

NPM. 1815031038

## RIWAYA HIDUP



Penulis dilahirkan di Desa Bumi Agung pada tanggal 5 november 2000. Penulis merupakan anak pertama dari empat bersaudara dari pasangan Bapak Jamaluddin dan Ibu Hellani. Penulis memulai pendidikan di Madrasah Ibtidaiyah (MI) Da'arul Ma'arif pada tahun 2006 hingga 2012, Sekolah Menengah Pertama (SMP) Negeri 1 Natar pada tahun 2012 hingga 2015, dan Sekolah Menengah Kejuruan (SMK) Negeri 1 Natar pada tahun 2015 hingga 2018. Penulis menjadi mahasiswa Jurusan Teknik Elektro, Universitas Lampung pada tahun 2018 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN). Selama menjadi mahasiswa, penulis berkesempatan tergabung dalam Forum Komunikasi Bidikmisi Universitas Lampung sebagai Anggota Departemen Pembinaan Dan Pengembangan Sumber Daya Mahasiswa (PPSDM) pada tahun 2020 dan menjadi Ketua Departemen Kesekretariatan pada tahun 2021. Selain itu, penulis juga tergabung dalam lembaga kemahasiswaan yang ada di Jurusan Teknik Elektro (HIMATRO) sebagai Anggota Divisi Sosial dan Kewirausahaan pada tahun 2019 hingga 2020 dan sebagai Anggota Divisi Pendidikan pada tahun 2020 hingga 2021. Penulis mengikuti proyek jangka panjang dari dosen tentang sistem deteksi kejatuhan berbasis Arduino menggunakan Jaringan Syaraf Tiruan (JST) yang kemudian salah satu tahap pengerjaannya diangkat menjadi skripsi penulis

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

**Alhamdulillah, Atas Izin Allah yang Maha Kuasa**

**KUPERSEMBAHKAN KARYA INI UNTUK**

*Ayah dan Ibu Tercinta*

Jamaluddin dan Hellani

*Adik-adik Tersayang*

Vicky Maulina Alifya  
Raffa Ahmad Al-Aziz  
Gilang Ahmad Ramadhan

**Keluarga Besar, Dosen, Teman dan Almamater**



## MOTTO

"Janganlah engkau bersedih, sesungguhnya Allah bersama kita."

(QS. At-Taubah : 40)

"Jagalah salatmu. Karena saat kamu kehilangan salat, maka kamu akan kehilangan segalanya"

(Umar Bin Khattab)

" Saya Siap! Saya Siap! Saya Siap! "

(SpongeBob SquarePants)

"Kalau terus ragu tidak akan menghasilkan apa-apa. Ini bukan waktunya untuk ragu."

(Nara Shikamaru)

## SANWACANA

Segala puji bagi Allah SWT, atas limpahan nikmat-Nya yang diberikan kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini. Shalawat dan salam senantiasa dicurahkan kepada Nabi Muhammad saw. suri teladan yang mampu membuka sesuatu yang terkunci, penutup dari semua yang terdahulu, penolong kebenaran dengan jalan yang benar, dan petunjuk kepada jalan-Mu yang lurus.

Tugas Akhir dengan judul **“Deteksi Kejatuhan Dengan Sensor Accelerometer Dan Gyroscope Berbasis Arduino Menggunakan Jaringan Syaraf Tiruan”** ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung dan selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis dengan baik.
2. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung dan selaku dosen penguji yang telah memberikan masukan, kritik, dan saran yang membangun kepada penulis.
3. Wahyu Eko S, S.T., M.Sc. selaku Sekretaris Jurusan Teknik Elektro Universitas Lampung.
4. Bapak Sumadi, S.T., M.T. selaku Ketua Prodi Teknik Elektro Universitas Lampung.
5. Bapak Dr. Eng. Ageng Sadnowo R., S.T., M.T. selaku pembimbing pendamping yang telah memberikan bimbingan, arahan, dan nilai-nilai kehidupan kepada penulis dengan baik.
6. Ibu Yetti Yuniati, S.T., M.T. selaku dosen pembimbing akademik (PA) yang telah memberikan nasihat, arahan, dan bimbingan bagi penulis dalam mempersiapkan diri menjadi seorang Sarjana Teknik.

7. Segenap Dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat, wawasan, dan pengalaman bagi penulis
8. Segenap Staf di Jurusan Teknik Elektro dan Fakultas Teknik yang telah membantu penulis baik dalam hal administrasi dan hal-hal lainnya.
9. Rekan-rekan mahasiswa Teknik Elektro Universitas Lampung Angkatan 2018 (ELTICS 2018), SILAMPARI SQUAD, yang telah banyak memberi dukungan moril untuk saya.
10. Sahabat seperjuangan Edo, Raja, Adit, Pian, Doyog, Erik, Ucok, Steven, Ridho, Dimas, Gilang, Hamid yang telah banyak membantu, memberikan semangat, pengalaman, cerita, suka dan duka, nasehat dan saran.
11. Saya juga ingin mengucapkan terima kasih kepada kekasih saya, Novita Rahayu, yang telah memberikan dukungan, perhatian, dan semangat selama proses penyusunan skripsi ini. Kehadiranmu sangat berarti dan membuat semuanya terasa lebih ringan.
12. Semua pihak yang telah membantu selama pelaksanaan skripsi dan penyusunan laporan namun tidak dapat disebutkan satu persatu

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Penulis mengharapkan kritik dan saran konstruktif dari semua pihak demi kemajuan bersama. Penulis berharap skripsi ini dapat bermanfaat bagi kita semua

Bandar Lampung, 17 Juni 2025



**Ezza Ahmad Faturrohman**

NPM 181503138

## DAFTAR ISI

|                                |      |
|--------------------------------|------|
| ABSTRAK.....                   | ii   |
| ABSTRACT.....                  | iii  |
| MENGESAHKAN.....               | vi   |
| SURAT PERNYATAAN.....          | vii  |
| RIWAYA HIDUP.....              | viii |
| MOTTO.....                     | x    |
| SANWACANA.....                 | xi   |
| DAFTAR ISI.....                | xiii |
| DAFTAR GAMBAR.....             | xvi  |
| DAFTAR TABEL.....              | xvi  |
| BAB I PENDAHULUAN.....         | 1    |
| 1.1 Latar Belakang.....        | 1    |
| 1.2 Tujuan Penelitian.....     | 2    |
| 1.3 Manfaat Penelitian.....    | 2    |
| 1.4 Rumusan Masalah.....       | 3    |
| 1.5 Batasan Masalah.....       | 3    |
| 1.6 Hipotesis.....             | 3    |
| 1.7 Sistematika Penulisan..... | 3    |
| BAB II TINJAUAN PUSTAKA.....   | 5    |
| 2.1 Tinjauan Terdahulu.....    | 5    |
| 2.2 Arduino.....               | 5    |

|                                    |   |    |
|------------------------------------|---|----|
| 2.2.1                              | <i>Hardware</i> Arduino .....                                   | 6  |
| 2.2.2                              | <i>Software</i> Arduino .....                                   | 7  |
| 2.3                                | Bahasa Pemograman Arduino .....                                 | 9  |
| 2.3.1                              | Struktur Pemograman.....  | 9  |
| 2.3.2                              | Variabel.....   | 11 |
| 2.3.3                              | Fungsi.....   | 11 |
| 2.4                                | Sensor MPU-6050 .....   | 12 |
| 2.5                                | Jaringan Syaraf Tiruan ( <i>Artifical Neural Network</i> )..... | 15 |
| 2.6                                | Metode Backpropagation JST .....                                | 17 |
| 2.7                                | Fase Dalam Jaringan Syaraf Tiraun Backpropagation .....         | 19 |
| 2.8                                | MATLAB .....  | 19 |
| BAB III METODOLOGI PENELITIAN..... |   | 19 |
| 3.1                                | Waktu dan Tempat Penelitian .....                               | 19 |
| 3.2                                | Alat dan Bahan .....  | 19 |
| 3.3                                | Prosedur Penelitian.....  | 20 |
| 3.4                                | Perancangan Kerja Jaringan Syaraf Tiruan .....                  | 21 |
| 3.4.1                              | Perakitan Sistem Deteksi Kejatuhan .....                        | 22 |
| 3.4.2                              | Pemograman Arduino IDE.....                                     | 23 |
| 3.4.3                              | Pengambilan Data Uji .....                                      | 26 |
| 3.4.4                              | Menyimpan Data Uji Ke Excel.....                                | 27 |
| 3.4.5                              | Membuat Program Matlab .....                                    | 29 |
| 3.4.5.1                            | Perancangan Program Pelatihan Matlab.....                       | 29 |
| 3.4.5.2                            | Perancangan Program Pengujian Matlab.....                       | 32 |
| 3.4.6                              | Menghitung Nilai Akurasi .....                                  | 35 |
| BAB IV HASIL DAN PEMBAHASAN .....  |   | 38 |
| 4.1                                | Dataset .....   | 38 |

|                                 |                      |    |
|---------------------------------|----------------------|----|
| 4.2                             | Pelatihan Model..... | 50 |
| 4.3                             | Hasil Pengujian..... | 56 |
| BAB V KESIMPULAN DAN SARAN..... |                      | 62 |
| 5.1                             | Kesimpulan.....      | 62 |
| 5.2                             | Saran.....           | 63 |
| DAFTAR PUSTAKA .....            |                      | 64 |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Arduino Uno.....   | 7  |
| Gambar 2.2 Tampilan <i>toolbar</i> arduino.....   | 8  |
| Gambar 2.3 Sensor MPU-6050.....   | 13 |
| Gambar 2.4 Ilustrasi Akselerasi dan Sudut Sistem Deteksi Kejatuhan .....  | 14 |
| Gambar 2.5 Jaringan <i>Single Layer Network</i> .....   | 16 |
| Gambar 2.6 Jaringan <i>Multi Layer Network</i> .....  | 16 |
| Gambar 2.7 Arsitektur jaringan syaraf tersembunyi lapisan 1.....  | 18 |
| Gambar 3.1 Diagram Alir Penelitian .....  | 20 |
| Gambar 3.2 Prosedur Kerja Jaringan Syaraf Tiruan .....  | 21 |
| Gambar 3.3 Tampilan Sistem Pada Rompi Responden .....   | 22 |
| Gambar 3.4 Pemasangan Sensor Pada Bagian Bahu Dari Responden.....   | 23 |
| Gambar 3.5 Proses Pengambilan Data Pada Responden .....   | 27 |
| Gambar 3.6 Tampilan Data Mentah Pada Layar Kerja Arduino IDE.....   | 28 |
| Gambar 4.1 Grafik Perubahan Akselerasi dan Sudut Saat Jatuh Ke Depan .....  | 39 |
| Gambar 4.2 Grafik Perubahan Akselerasi dan Sudut Saat Jatuh Ke Samping .....                                      | 41 |
| Gambar 4.3 Grafik Perubahan Akselerasi dan Sudut Saat Jatuh Ke Belakang .....                                     | 42 |
| Gambar 4.4 Grafik Perubahan Akselerasi dan Sudut Saat Berdiri Ke Duduk.....                                       | 44 |
| Gambar 4.5 Grafik Perubahan Akselerasi dan Sudut Saat Berdiri Ke Tidur.....                                       | 45 |
| Gambar 4.6 Grafik Perubahan Akselerasi dan Sudut Saat Berdiri Ke Rukuk.....                                       | 46 |
| Gambar 4.7 Grafik Perubahan Akselerasi dan Sudut Saat Berdiri Ke Sujud .....                                      | 48 |
| Gambar 4.8 Grafik Perubahan Akselerasi dan Sudut Saat Duduk Ke Sujud.....   | 49 |
| Gambar 4.9 (A)Arsitektur Jaringan Syaraf Tiruan Skenario 1, (B) Arsitektur Jaringan Syaraf Tiruan Skenario 2..... | 51 |
| Gambar 4.10 Grafik Perbandingan Keluaran dan Target Skenario Pertama .....  | 52 |
| Gambar 4.11 Grafik Training State Skenario Pertama .....  | 53 |
| Gambar 4.12 Grafik perbandingan Keluaran dan Target Skenario Kedua .....  | 54 |

|  |    |
|--|----|
| Gambar 4.13 Grafik Training State Skenario Kedua.....                | 55 |
| Gambar 4.14 Grafik Keluaran Uji dan Target Uji Skenario Pertama..... | 57 |
| Gambar 4.15 Grafik Keluaran Uji dan Target Uji Skenario Kedua.....   | 59 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 3.1 Tabel Data Gerakan Jatuh Ke Depan.....                              | 28 |
| Tabel 4.1 Pembagian Dataset Pola Gerakan .....                                | 38 |
| Tabel 4.2 Tabel Data Jatuh Ke Depan .....                                     | 39 |
| Tabel 4.3 Tabel Data Jatuh Ke Samping.....                                    | 40 |
| Tabel 4.4 Tabel Data Jatuh Ke Belakang.....                                   | 42 |
| Tabel 4.5 Tabel Data Gerakan Berdiri Ke Duduk.....                            | 43 |
| Tabel 4.6 Tabel Data Gerakan Berdiri Ke Tidur.....                            | 44 |
| Tabel 4.7 Tabel Data Gerakan Berdiri Ke Rukuk.....                            | 46 |
| Tabel 4.8 Tabel Data Gerakan Berdiri Ke Sujud .....                           | 47 |
| Tabel 4.9 Tabel Data Gerakan Duduk Ke Sujud.....                              | 49 |
| Tabel 4.10 Tabel Nilai Aktual Dan Prediksi Uji Pada Skenario Pertama.....     | 58 |
| Tabel 4.11 Tabel Nilai Aktual dan Hasil Prediksi Uji Pada Skenario Kedua..... | 60 |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam dunia kesehatan kondisi dari keselamatan pasien adalah yang utama, dan diantara banyaknya kasus mengenai kecelakaan pasien salah satu yang paling sering terjadi adalah jatuh. Bagi pasien lanjut usia, mengalami jatuh yang tidak teramati bisa menjadi hal yang sangat berbahaya. Kemungkinan yang jelas dari efek jatuh adalah cedera awal dan dapat diperparah lagi dengan konsekuensi yang mungkin terjadi jika pertolongan tidak diperoleh dalam waktu singkat. Statistik menunjukkan bahwa mayoritas konsekuensi serius bukan akibat langsung dari terjatuh, namun karena penundaan bantuan dan perawatan [1].

Dengan demikian konsekuensi pasca jatuh bisa sangat berkurang jika bantuan dapat diberikan tepat pada waktunya. Untuk dapat memberikan bantuan dengan cepat pada saat peristiwa kejatuhan terjadi, diperlukan suatu alat yang dapat memberikan peringatan apabila peristiwa kejatuhan terjadi. Pengembangan perangkat untuk deteksi kejatuhan telah menjadi topik hangat. Dalam beberapa tahun terakhir, kemajuan teknologi sensor akuator *micro electro mechanical - system* (MEMS) telah memungkinkan untuk merancang detektor jatuh berdasarkan *accelerometer* MEMS. Teknik ini didasarkan pada prinsip mendeteksi perubahan gerakan dan posisi tubuh seseorang. Dengan memakai sensor yang dapat melacak perubahan akselerasi dalam tiga arah ortogonal. Data tersebut terus dianalisis secara algoritmik untuk mengetahui apakah tubuh individu tersebut jatuh atau tidak. Jika seseorang jatuh, perangkat tersebut dapat menggunakan GPS dan pemancar nirkabel untuk menentukan lokasi dan mengeluarkan peringatan untuk mendapatkan bantuan. Maka dari itu diciptakanlah sebuah alat yang dapat mendeteksi kejatuhan seseorang pasien lansia secara *real time*. Alat ini menggunakan telemetri GSM modul SIM900

yang dapat digunakan pada tempat yang tidak memiliki akses internet. Selain itu, alat ini juga menggunakan GPS yang dapat memberikan data letak pengguna melalui ponsel. Pada penelitian sebelumnya sistem yang dibuat akan memberikan notifikasi pada ponsel pengguna yang terdaftar saat sistem mendeteksi terjadinya kejatuhan. Namun pada hasil penelitian yang sebelumnya sudah dilakukan, data yang dihasilkan masih belum dapat membuktikan apakah pasien benar-benar mengalami kejatuhan atau tidak, ini karena saat terdeteksi adanya perubahan sudut yang melebihi tiga derajat, sistem akan mengaggap bahwa kondisi tersebut adalah kondisi kejatuhan [2].

Oleh karena itu, pada penelitian kali ini akan berfokus pada pengolahan data yang dihasilkan oleh sistem. Penelitian akan menggunakan salah satu metode jaringan syaraf tiruan yaitu metode *Backpropagation*, yang diharapkan dapat mengenali aktivitas lansia dan mendapatkan akurasi yang tinggi dalam mendeteksi jatuh.

## **1.2 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah :

1. Merancang arsitektur dan algoritma program pendeteksi pola gerakan jatuh dan tidak jatuh menggunakan metode *backpropagation* pada jaringan syaraf tiruan.
2. Mengimplementasikan *tools software Matlab* yang bisa membantu dalam merancang sistem deteksi yang tepat dalam mengklasifikasikan pola gerakan

## **1.3 Manfaat Penelitian**

Manfaat yang diharapkan dari penelitian ini adalah mendapatkan program dan sistem deteksi kejatuhan yang mampu membedakan antara gerakan jatuh dan tidak jatuh dengan memanfaatkan algoritma *backpropagation* pada jaringan syaraf tiruan, sehingga dapat meningkatkan akurasi dalam pengenalan pola gerakan.

#### **1.4 Rumusan Masalah**

Adapun yang menjadi perumusan masalah dalam penelitian ini adalah :

1. Bagaimana merancang arsitektur dan algoritma jaringan syaraf tiruan dengan metode backpropagation yang mampu membedakan pola gerakan jatuh dan tidak jatuh secara akurat?
2. Bagaimana implementasi perangkat lunak Matlab dapat digunakan untuk membantu proses pelatihan dan pengujian model deteksi kejatuhan dengan hasil klasifikasi yang efektif?

#### **1.5 Batasan Masalah**

Adapun yang menjadi batasan masalah dalam penelitian ini adalah :

1. Pengujian data hanya diuji pada data yang dihasilkan oleh sensor *Accelerometer* dan *Gyroscope* dari modul MPU6050.
2. Mikrokontroler yang digunakan adalah Arduino Uno R3 untuk pengolahan data.
3. Pemrosesan data hasil menggunakan *software Matlab*

#### **1.6 Hipotesis**

Dapat meningkatkan akurasi sistem deteksi kejatuhan sehingga lebih efektif dalam penggunaannya, yaitu untuk memantau tubuh seseorang lansia dari jarak jauh dan *real time*, sehingga dapat ditangani secara langsung jika terjadi kecelakaan yang dimaksud adalah kejatuhan.

#### **1.7 Sistematika Penulisan**

Dalam rangka penulisan skripsi ini, disusun suatu sistematika penulisan dengan membaginya menjadi beberapa bab. Susunan sistematika tersebut adalah:

##### **BAB I. PENDAHULUAN**

Menjelaskan tentang latar belakang permasalahan, tujuan dilakukannya penelitian, manfaat yang dapat diberikan dari penelitian, perumusan masalah, batasan masalah, hipotesis, dan sistematika penulisan.

## BAB II. TINJAUAN PUSTAKA

Bagian ini akan berisi tentang teori – teori sistem mikrokontroler Arduino UNO, sensor *gyroscope* MPU6050, metode *Backpropagation* dalam Jaringan Syaraf Tiruan (JST) yang digunakan dalam menyelesaikan penelitian ini.

## BAB III. METODE PENELITIAN

Bagian ini akan menjelaskan metode yang digunakan dalam proses perancangan dan perancangan alat, dan pengolahan data hasil, diantaranya waktu dan tempat penelitian, alat dan bahan, prosedur pembuatan alat dan pengujian sistem.

## BAB IV. HASIL DAN PEMBAHASAN

Bagian ini berisi tentang hasil pengujian dan pembahasan data – data yang diperoleh dari pengujian.

## BAB V. SIMPULAN DAN SARAN

Bagian ini akan menyimpulkan semua kegiatan dan hasil – hasil yang diperoleh selama proses perancangan dan pembuatan alat. Diberikan juga saran – saran yang perlu dipertimbangkan dalam upaya pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Terdahulu**

Beberapa penelitian relevan telah dilakukan sebelumnya seperti yang dilakukan oleh Ma'ruf Fajar Santoso dari Teknik Elektro, Fakultas Teknik, Universitas Lampung [2], Fahmi Idris dari Fakultas Teknik, Teknik Elektro Universitas Negeri Padang [3] dan juga penelitian oleh Mardi Hardjianto dari Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur [4] yang sama-sama meneliti deteksi kejatuhan menggunakan *accelerometer* pada *smartphone*. Berdasarkan penelitian dapat diambil sebuah kesimpulan bahwa, sistem ini dapat mengirim sebuah SMS yang berisikan sebuah pesan peringatan bahwa pasien terjatuh, kordinat pasien terjatuh dan sistem akan memberi respon dengan menyambungkan kepada nomor yang telah didaftarkan. Lalu ada juga penelitian yang dilakukan oleh Adlian Jefiza dari jurusan Teknik Elektro, Fakultas Teknik, Institut Teknologi Sepuluh November [5].

Penelitian menggunakan metode *Bacpropagation* dalam JST, berdasarkan peneilitian dapat diambil kesimpulan bahwa metode *Backpropagation* meningkatkan akurasi dari sistem deteksi kejatuhan. Peneliti menggunakan arduino nano sebagai mikrokontroler-nya dan data hasilnya tidak ditampilkan dengan notifikasi pada *smartphone*. Perbedaan peneilitian ini pada penelitian-penelitian sebelumnya adalah pada pengolahan data yang dihasilkan oleh alat deteksi kejatuhan, terutama pada data yang dihasilkan oleh sensor *Accelerometer* yang akan mengidentifikasi bahwa, apakah pasien benar-benar terjatuh atau tidak menggunakan metode Jaringan Syaraf Tiruan (JST) yang berbasis Arduino Uno R3.

#### **2.2 Arduino**

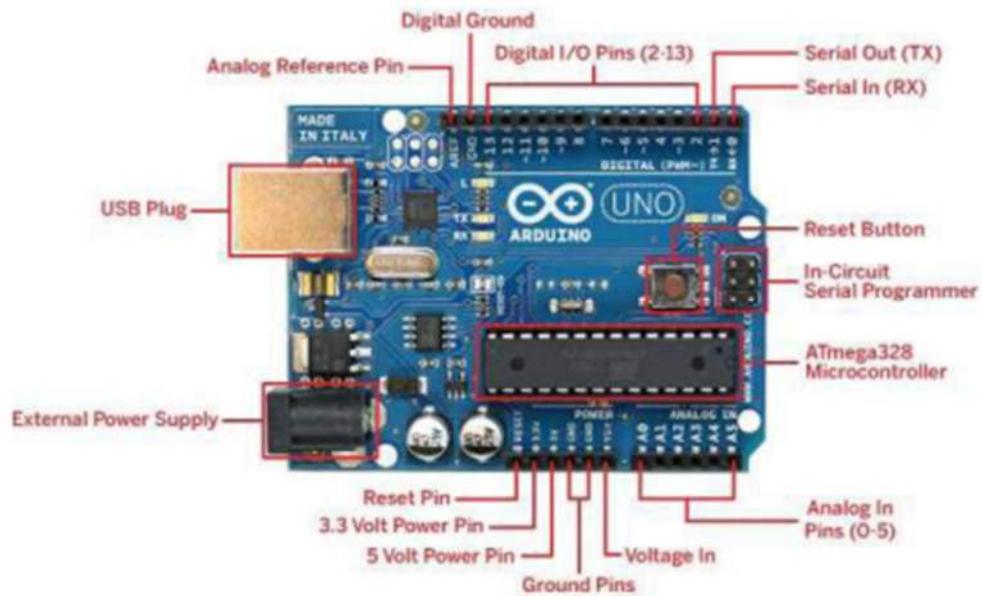
Arduino adalah sebuah *microcontroller single board* yang sifatnya *open source*

yang mempunyai fleksibilitas tinggi baik dari segi perangkat keras (*hardware*) dan perangkat lunak (*software*). Arduino menggunakan IC ATmega sebagai IC program dan bahasa pemrograman yang digunakan pada arduino adalah JAVA dilengkapi dengan *library C/C++* yang penulisannya sendiri hampir mirip dengan penulisan bahasa manusia. Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada papan masukan dan keluaran sederhana. Yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan perangkat lunak dan perangkat keras yang dapat mendeteksi dan merespon situasi dan kondisi. Kelebihan arduino dari *platform* hardware mikrokontroler lain adalah:

1. IDE Arduino dibuat berdasarkan pada IDE *Processing* sederhana sehingga mudah digunakan.
2. Pemrograman Arduino menggunakan kabel yang terhubung dengan *port* USB bukan *port serial*. Fitur ini berguna karena banyak komputer sekarang ini tidak memiliki *port serial*.
3. Perangkat lunak dan perangkat keras arduino berbasis *open source*, pembaca bisa mengunduh perangkat lunak dan gambar rangkaian arduino tanpa harus membayar ke pembuat arduino.
4. Proyek arduino ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.
5. Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi terutama oleh programmer pemula [6].

### **2.2.1 Hardware Arduino**

Arduino uno R3 merupakan papan pengembangan mikrokontroler berbasis ATmega328P yang berukuran kecil atau dapat diartikan juga dengan suatu rangkaian berukuran kecil yang didalamnya terdapat komputer berbentuk *chip* yang kecil. Pada Gambar 2.1 adalah sebuah gambar papan arduino UNO yang digunakan dalam penelitian ini.



Gambar 2.1 Arduino Uno

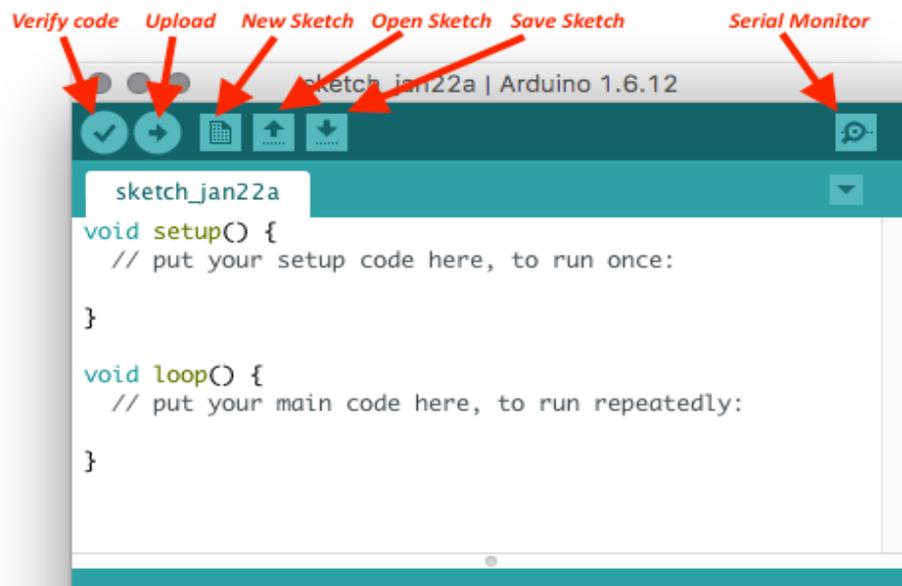
Pada perangkat keras arduino uno R3 terdapat 20 pin yang meliputi :

1. 14 pin IO digital (pin 0-13)  
dimana 14 pin diantaranya dapat digunakan sebagai output PWM antara lain pin 0 sampai 13 yang dapat dijadikan masukan atau keluaran yang diatur dengan cara membuat program IDE.
2. 6 pin masukan Analog (pin A0-A5)  
menggunakan crystal 16 MHz antara lain pin A0 sampai A5 yang dapat digunakan untuk membaca nilai input.
3. 6 pin keluaran analog (pin 3, 5, 6, 9, 10, dan 11)  
Sejumlah pin yang sebenarnya merupakan pin digital tetapi sejumlah pin tersebut dapat diprogram kembali menjadi pin keluaran analog dengan cara membuatnya pada program IDE.
4. koneksi USB, jack listrik, *header* ICSP dan tombol reset.[7]

### 2.2.2 Software Arduino

Dalam penelitian ini *software* yang digunakan dalam pemrograman arduino adalah Arduino IDE, Arduino IDE adalah perangkat lunak (*Software*) yang biasanya dipakai untuk membuat suatu *sketch* pemrograman, yang biasa dipakai pengguna

untuk mengedit, membuat, meng-*coding*, meng-*upload* ke board arduino tertentu yang akan digunakan. Arduino IDE terbuat dari bahasa pemrograman JAVA, yang dilengkapi dengan *library C/C++ (wiring)*, yang membuat penggunaan *input/output* menjadi lebih mudah. Gambar 2.2 merupakan IDE arduino yaitu perangkat lunak sangat canggih yang ditulis dengan menggunakan bahasa pemrograman java. IDE arduino terdiri dari :



Gambar 2.2 Tampilan *toolbar* arduino

Bagian-bagian tampilan *toolbar* arduino Gambar 2.2 dapat diuraikan sebagai berikut :

1. *Verify*  
Digunakan untuk mengecek atau meng-*compile* suatu program apakah masih ada kesalahan atau tidak
2. *Upload*  
Digunakan untuk mengirimkan program ke dalam *board* arduino.
3. *New*  
Digunakan untuk membuat halaman pemrograman yang baru.
4. *Open*  
Digunakan untuk membuka program tersimpan yang sudah pernah dikerjakan.
5. *Save*

Digunakan untuk menyimpan file program yang sudah selesai.

#### 6. *Serial Monitor*

Digunakan untuk menampilkan data yang telah selesai di upload ke *board* arduino, yang sudah dijalankan supaya bisa dilihat pada serial monitor [7].

### **2.3 Bahasa Pemrograman Arduino**

Program komputer atau sering kali disingkat sebagai program adalah serangkaian instruksi yang ditulis untuk melakukan suatu fungsi spesifik pada komputer. Komputer pada dasarnya membutuhkan keberadaan program agar bisa menjalankan fungsinya sebagai komputer, biasanya hal ini dilakukan dengan cara mengeksekusi serangkaian instruksi program tersebut pada prosesor. Bahasa pemrograman atau sering diistilahkan juga dengan bahasa komputer adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Contohnya adalah program yang digunakan oleh kasir di mal-mal atau swalayan, penggunaan lampu lalu lintas di jalan raya, dll. Bahasa pemrograman komputer yang kita kenal antara lain adalah Java, *Visual Basic*, C++, C, Cobol, PHP, .Net, dan bahasa lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya, dan pada penelitian kali ini yang digunakan adalah bahasa pemrograman Arduino. Bahasa pemrograman arduino terdiri dari tiga bagian utama yaitu struktur, variabel, dan fungsi [6].

#### **2.3.1 Struktur Pemrograman**

Struktur dasar bahasa pemrograman arduino sangatlah mudah dan sederhana. Agar program dapat berjalan dengan baik maka perlu setidaknya dua bagian atau fungsi yaitu *setup()* yang dipanggil hanya satu kali, biasanya untuk inisialisasi program (mengatur masukan dan lain-lain). Serta *loop()* tempat untuk mengeksekusi program secara berulang-ulang, biasanya untuk membaca input atau men-*trigger* keluaran. Berikut ini bentuk penulisannya:

```

Void setup ()
{
    //Isi Program;
}
Void loop()
{
    //Isi Program;
}

```

- **Setup()**

Fungsi *setup* () hanya dipanggil satu kali saja saat program mulai berjalan. Fungsi *setup* () berguna untuk melakukan inisialisasi mode pin atau memulai komunikasi serial. *Setup* () ini harus ada meskipun tidak ada program yang akan dieksekusi. Berikut ini bentuk penulisannya:

```

Void setup()
{
    pinMode(led, OUTPUT); //set led sebagai keluaran
}
Void loop()
{
    //Isi Program;
}

```

- **Loop()**

Setelah menyiapkan inisialisasi pada *setup()*, berikut membuat fungsi *loop()*. Sesuai namanya, fungsi ini akan mengulang program yang ada secara terus-menerus, sehingga program akan berubah dan merespon sesuai masukan. Fungsi *loop()* ini akan secara aktif mengontrol papan arduino.

Contoh penggunaan fungsi *loop()* seperti berikut:

```

Void setup()
{

```

```

pinMode(led, OUTPUT);           //set led sebagai keluaran
}
Void loop()
{
digitalWrite(led, HIGH);        //set led on
delay (500);                     // tunda untuk ½ detik
digitalWrite(led, LOW);         //set led off
delay (500);                     //tunda untuk ½ detik
}

```

### 2.3.2 Variabel

Fungsi variable adalah untuk memberikan nama yang sesuai dengan kebutuhan dan menampung nilai angka dalam membuat program. Sebuah variabel perlu dideklarasikan terlebih dahulu, dan bisa digunakan sebagai penampung pembaca masukan yang akan disimpan atau diberi nilai awal. Dengan menggunakan variabel, maka nilai yang ada dapat diubah dengan leluasa.

### 2.3.3 Fungsi

*Function* atau Fungsi terdiri dari :

1. Fungsi Digital I/O
2. Fungsi untuk digital I/O ada tiga buah yaitu *pinMode (pin, mode)*, *digitalWrite(pin, value)*, dan *int digitalRead (pin)*.
3. Fungsi Analog I/O
4. Fungsi untuk analog I/O ada tiga buah yaitu *analogReference (type)*, *int analogRead(pin)*, dan *analogWrite(pin, value)-PWM*.
5. Fungsi Waktu
6. Fungsi waktu terdiri dari *unsigned long millis()*, *delay(ms)* Dan *delayMicroseconds (us)*.
7. Fungsi Matematika
8. Fungsi matematika terdiri dari *min(x,y)*, *max(x,y)*, *abs(x)*, *sqrt(x)*
9. dan *pow (base, exponent)*.

## 2.4 Sensor MPU-6050

Sensor akselereasi dan sudut MPU6050 merupakan sensor yang berperan penting pada penelitian ini. sensor ini digunakan untuk mendeteksi pergerakan harian pasien yang dalam hal ini adalah mendeteksi perubahan nilai akselerasi dan sudut yang terjadi pada setiap pergerakan yang sudah di tentukan. *Accelerometer* adalah sensor yang digunakan untuk mengukur percepatan suatu objek, yaitu mengukur percepatan statis dan dinamisnya. Pengukuran dinamis adalah pengukuran percepatan pada objek bergerak, sedangkan pengukuran statis adalah pengukuran terhadap gravitasi bumi, lebih tepatnya untuk mengukur sudut kemiringan. Prinsip kerja dari sensor *accelerometer* berdasarkan hukum fisika bahwa apabila suatu konduktor digerakkan melalui suatu medan magnet, atau jika suatu medan magnet digerakkan melalui suatu konduktor, maka akan timbul suatu tegangan induksi pada konduktor tersebut. *Accelerometer* MPU6050 adalah sebuah modul berinti MPU6050 yang merupakan 6 *axis Motion Processing Unit* dengan penambahan regulator tegangan dan beberapa komponen pelengkap lainnya yang membuat modul ini siap pakai dengan tegangan *supply* sebesar 3-5 VDC. Modul ini memiliki interface I2C yang dapat disambungkan langsung ke MCU (*Microcontrolle Units*) yang memiliki fasilitas I2C. Sensor MPU-6050 berisi sebuah *Micro electro mechanical Systems* (MEMS) *Accelerometer* dan sebuah MEMS *Gyro* yang saling terintegrasi. Sensor ini sangat akurat dengan fasilitas *hardware* internal 16bit ADC untuk tiap kanalnya.

Sedangkan *Gyroscope* adalah alat berupa cakram yang sumbunya berputar antara dua penopang dan tetap dalam posisinya apabila tidak ada pengaruh kekuatan dari luar. Sedangkan dalam dunia instrumentasi, alat ini biasa digunakan untuk mengukur orientasi berdasarkan prinsip momentum sudut. Sensor ini akan mengukur kecepatan sudut dari suatu rotasi yang satuannya adalah radian per detik (rad/s). *Gyroscope* yang digunakan dalam penelitian ini adalah *gyroscope* elektrik yang ada dalam sensor MPU6050. *Output* yang dihasilkan dari sensor *gyroscope* ini adalah berupa data kecepatan sudut. Kecepatan sudut sendiri merupakan besaran vektor yang menyatakan frekuensi sudut suatu benda dan sumbu putarnya, Satuan Internasional (SI) untuk kecepatan sudut adalah radian per detik (rad/s) [8].

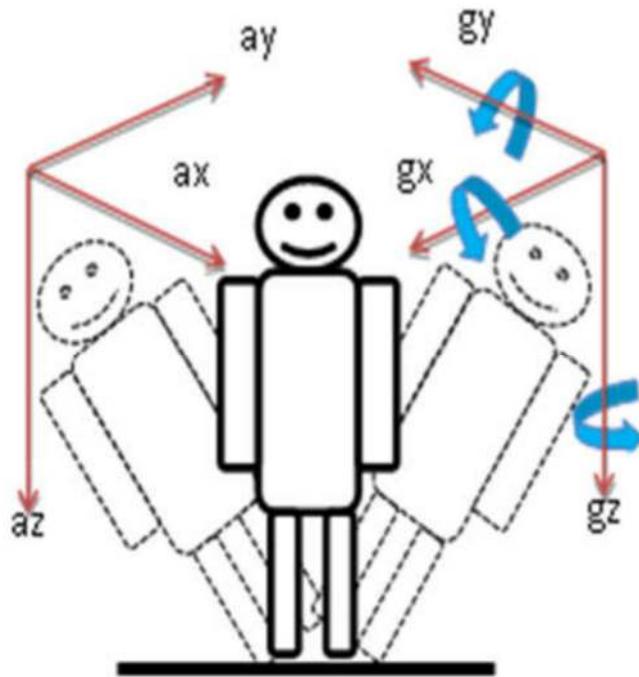


Gambar 2.3 Sensor MPU-6050

Gambar 2.3 merupakan gambar dari sensor *accelerometer* MPU6050. Beberapa fitur dari sensor *accelerometer* sebagai berikut :

1. *Gyroscope Range* + 250 500 1000 2000 / s
2. *Acceleration Range*: + 2 + 4 + 8 + 16 g
3. Berbasis *Chip* MPU-6050, berteknologi *Motion Fusion* yang mengoptimalkan kinerja sensor dan adanya *Digital Motion Processor* modul dapat diintegrasikan dengan sensor lainnya melalui komunikasi I2C dan bekerja tanpa membebani mikrokontrolernya.
4. Tegangan supply sekitar 3-5 VDC dan pada modul ini sudah dilengkapi LDO (*Low Drop-out*) *Voltage Regulator*. Jadi, untuk mendapat sumber tegangan hanya perlu tersambung dengan sumber Vcc pengolah data seperti Arduino.
5. Tersedianya *pull-up* resistor pada pin SDA dan SCL tanpa resistor eksternal tambahan.
6. Data keluaran *Motion Fusion* sebanyak 6 atau 9 sumbu dalam format matriks rotasi, *quaternion*, sudut Euler, atau *raw* data format.
7. Memori penampung data (*buffer memory*) sebesar 1KB, FIFO (*First-In-First-Out*).

8. Gabungan antara *accelerometer* dan *gyroscope* dalam satu sirkuit menyebabkan pendeteksian gerakan dan gravitasi menjadi lebih akurat.



Gambar 2.4 Ilustrasi Akselerasi dan Sudut Sistem Deteksi Kejatuhan

Gambar 2.4 menunjukkan ilustrasi bagaimana sensor *accelerometer* dan *gyroscope* mendeteksi perubahan posisi dan orientasi tubuh. Sensor accelerometer mengukur percepatan linier dalam tiga sumbu utama yaitu,  $a_x$  (sumbu X),  $a_y$  (sumbu Y), dan  $a_z$  (sumbu Z), yang ditunjukkan oleh panah merah ke arah masing-masing sumbu. Sementara itu, gyroscope mengukur kecepatan sudut terhadap sumbu rotasi yang sama, yakni  $g_x$ ,  $g_y$ , dan  $g_z$ . Perubahan nilai dari sensor-sensor ini mencerminkan rotasi tubuh atau perubahan orientasi akibat gerakan seperti membungkuk, terjatuh ke samping, atau ke belakang. Pada posisi tegak (normal), nilai percepatan dan kecepatan sudut relatif stabil. Namun, ketika terjadi kejatuhan, terjadi perubahan signifikan pada nilai  $a_x$ ,  $a_y$ ,  $a_z$  (karena percepatan gravitasi dan tumbukan), serta  $g_x$ ,  $g_y$ ,  $g_z$  (karena gerakan rotasi tubuh). Pola perubahan nilai-nilai inilah yang kemudian dianalisis oleh sistem deteksi kejatuhan untuk menentukan apakah suatu peristiwa merupakan jatuh atau bukan.

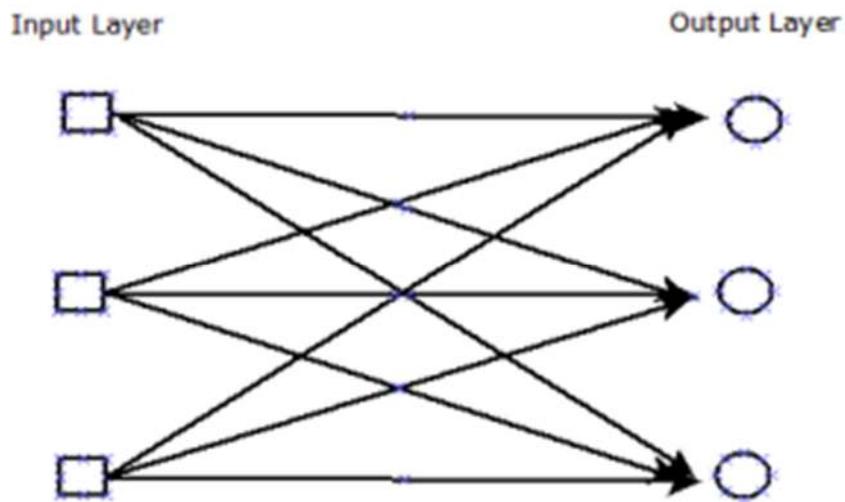
## 2.5 Jaringan Syaraf Tiruan ( *Artificial Neural Network* )

Jaringan saraf tiruan (*Artificial Neural Network*) adalah suatu sistem pemrosesan informasi yang dibuat dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah, yaitu dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Sejak pertama kali ditemukan oleh Mc.Culloch dan Pitts sistem jaringan syaraf tiruan sudah berkembang pesat dan banyak di gunakan oleh banyak aplikasi pada penelitian maupun penggunaannya dalam berbagai teknologi dimasa kini, jaringan syaraf tiruan (*Artificial Neural Network*) adalah suatu jaringan untuk memodelkan cara kerja sistem syaraf manusia (otak) dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasi sel-sel penyusunan (*neuron*), sehingga memiliki kemampuan untuk melaksanakan tugas-tugas tertentu khususnya pengenalan pola dengan efektifitas jaringan tertiggi.

Jaringan saraf tiruan mampu melakukan pengenalan kegiatan berbasis data masa lalu. Data masa lalu akan di pelajari oleh jaringan syaraf tiruan sehingga mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum pernah dipelajari. Keunggulan yang utama dari sistem *Artificial Neural Network* (ANN) adalah adanya kemampuan untuk belajar dari contoh yang di berikan atau data training, sedangkan untuk kelemahan dari *Artificial Neural Network* (ANN) yakni di butuhkan pelatihan untuk pengoperasiannya dan di butuhkan waktu yang lama untuk memproses-nya, metode ini sangat jauh lebih sederhana dibandingkan dengan syaraf tiruan manusia yang sebenarnya. Suatu jaringan syaraf tiruan ditentukan oleh 3 hal yaitu, pola antara hubungan neuron yang disebut arsitektur jaringan, penentuan bobot penghubung yang disebut metode *training* atau *learning* dan aktivasi yang digunakan.

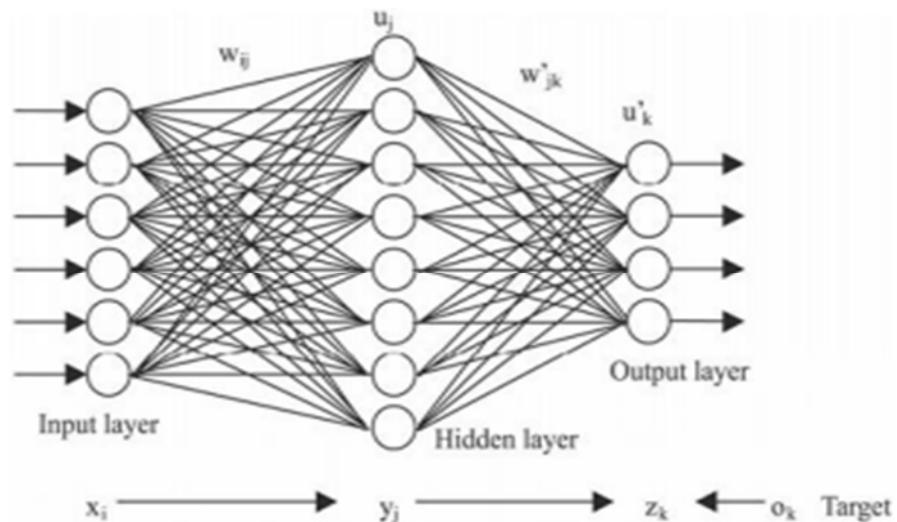
Berdasarkan jumlah Layer arsitektur jaringan *Artificial Neural Network* (ANN) dapat di klasifikasikan menjadi dua kelas yang berbeda yaitu :

1. Jaringan layer tunggal (*single layer network*) merupakan semua unit *input* dalam jaringan ini dihubungkan dalam semua unit *output*, meskipun dengan bobot yang berbeda - beda, Berikut contoh jaringan layer tunggal dapat di lihat pada Gambar 2.5.



Gambar 2.5 Jaringan *Single Layer Network*

2. Jaringan layer jamak ( *multi layer network* ) adalah jaringan layer jamak yang merupakan perluasan dari layer tunggal jaringan layer jamak ini memperkenalkan satu atau layer tersembunyi ( *hidden layer* ) yang mempunyai simpun yang disebut *neuron* tersembunyi ( *hidden layer* ), berikut contoh jaringan layer jamak dapat di lihat pada Gambar 2.6.



Gambar 2.6 Jaringan *Multi Layer Network*

Suatu sistem jaringan syaraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi hal ini terinspirasi oleh kinerja oleh model kerja otak

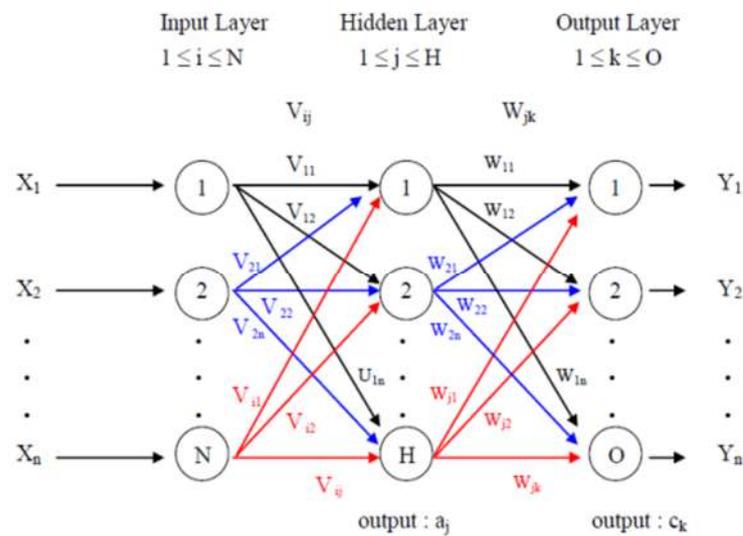
sungguhan. Hecht-Nielsen mendefinisikan sistem saraf buatan sebagai berikut, "Suatu *Neural Network* (NN), adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang dikoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (*fan out*) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut). Keluaran dari elemen proses tersebut dapat merupakan sebuah jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen proses harus benar - benar dilakukan secara lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal" [9].

## **2.6 Metode Backpropagation JST**

Terdapat berbagai macam metode dalam penerapan Jaringan Syaraf Tiruan (JST), salah satu yang paling umum digunakan adalah metode Backpropagation. Dalam penelitian ini, metode Backpropagation dipilih karena kemampuannya yang efektif dalam melakukan pelatihan jaringan untuk meminimalkan kesalahan output. Arsitektur dari jaringan syaraf tiruan dengan algoritma Backpropagation umumnya terdiri dari beberapa lapisan, yaitu lapisan masukan (*input layer*), setidaknya satu lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Dalam implementasinya, satu lapisan tersembunyi sering kali sudah cukup untuk menyelesaikan berbagai jenis permasalahan, meskipun dalam beberapa kasus yang lebih kompleks, penggunaan lebih dari satu lapisan tersembunyi mungkin diperlukan untuk mencapai hasil yang optimal.

Setiap neuron atau unit pada suatu lapisan dalam jaringan ini terhubung secara penuh (*fully connected*) dengan neuron pada lapisan berikutnya, membentuk struktur yang memungkinkan aliran informasi dan perhitungan bobot secara menyeluruh. Selain itu, baik pada lapisan tersembunyi maupun lapisan keluaran, setiap unit atau neuron dilengkapi dengan sebuah bias yang bernilai satu, yang berfungsi untuk meningkatkan fleksibilitas model dalam proses pembelajaran.

Dalam jaringan ini, terdapat dua jenis bobot yang digunakan: pertama, bobot yang menghubungkan lapisan input dengan lapisan tersembunyi, dan kedua, bobot yang menghubungkan lapisan tersembunyi dengan lapisan output. Kedua jenis bobot ini berperan penting dalam proses penyesuaian nilai selama pelatihan agar jaringan mampu menghasilkan keluaran yang sesuai dengan target yang diinginkan. Berikut Contoh Gambar arsitektur jaringan Syaraf tersembunyi lapisan 1 dapat di lihat pada gambar 2.7.



Gambar 2.7 Arsitektur jaringan syaraf tersembunyi lapisan 1

Keterangan:

- $X_1 \dots X_n$  : masukan pada jaringan
- $i, j, k$  : variabel indeks pada masukan
- $V_{ij}$  : bobot koneksi sel ke  $i$  pada lapisan *input layer* dengan sel ke  $j$  pada *hidden layer*.
- $W_{jk}$  : bobot koneksi sel ke  $j$  pada lapisan *hidden layer* dengan sel ke  $k$  pada *output layer*.
- $N$  : jumlah sel pada *input layer*
- $H$  : jumlah sel pada *hidden layer*
- $O$  : jumlah sel pada *output layer*
- $a_j$  : keluaran pada *hidden layer*
- $c_k$  : keluaran pada *output layer*.

## 2.7 Fase Dalam Jaringan Syaraf Tiraun Backpropagation

Model yang dikembangkan oleh Paul Werbos ini mempunyai dua fase (tahap) dalam proses pelatihannya, yaitu fase propagasi maju (*forward propagation*) dan fase propagasi mundur (*backward propagation*).

### a. Fase propagasi maju

Fase dimana pola masukan pada lapisan masukan akan diteruskan (*propagated*) ke masing-masing sel pada lapisan berikutnya (lapisan tersembunyi dan lapisan keluaran) sampai dihasilkan suatu keluaran akhir. Keluaran ini kemudian dibandingkan dengan nilai keluaran idealnya sehingga dihasilkan kesalahan bagi masing-masing keluaran.

### b. Fase propagasi mundur

Fase ini terjadi saat kesalahan yang dihasilkan pada fase maju masih lebih besar dari nilai toleransi kesalahan (*epsilon*) yang diberikan. Dimana kesalahan ini kemudian diumpun balik (*backward*) dari lapisan keluaran ke masing-masing sel pada lapisan sebelumnya.

## 2.8 MATLAB

MATLAB atau (*Matrix Laboratory*) adalah salah satu dari sekian banyak software yang digunakan untuk memudahkan pengolahan suatu data, MATLAB telah menjadi sebuah perangkat lunak yang di peruntukkan pemrograman, analisis, dan komputasi teknis dan matematis berbasis matriks. Sejak pertama kali dirilis oleh *Cleve Moler* pada tahun 1970, MATLAB digunakan secara khusus untuk menangani permasalahan yang berkaitan dengan persamaan aljabar linear. Namun, seiring berjalannya waktu, sistem MATLAB terus berkembang dengan peningkatan fungsi dan kinerja komputasinya. Saat ini, MATLAB tidak hanya mampu menyelesaikan berbagai macam permasalahan matematika dan teknis, tetapi juga menyediakan beragam alat untuk pemodelan, simulasi, visualisasi data, dan pengembangan aplikasi. Pada Penelitian ini *Software* MATLAB digunakan dalam proses pelatihan data latih, dan juga pengujian dari data yang sudah dilatih dengan menggunakan data uji yang bukan termasuk dalam data latih tersebut [10].

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Waktu dan Tempat Penelitian**

Penelitian ini dilakukan di Laboratorium Elektronika Fakultas Teknik Universitas Lampung yang dilaksanakan mulai dari bulan Februari 2024 sampai Februari 2025.

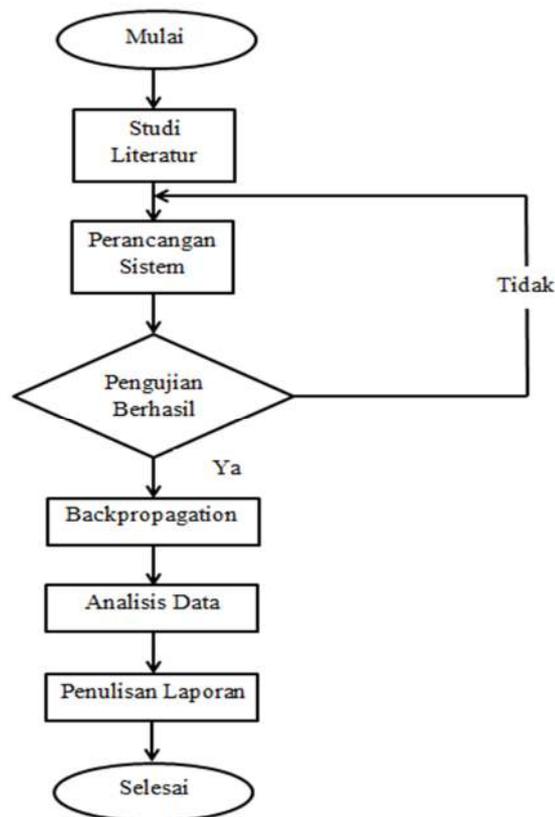
#### **3.2 Alat dan Bahan**

Adapun alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut.

1. Laptop HP ENVY 13-aqxxx dengan *Prosesor Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz*  
Laptop digunakan sebagai media pengolahan data hasil pengujian baik pelatihan maupun pengujian.
2. Arduino Uno dan Sensor MPU-6050  
Arduino dan sensor MPU-6050 digunakan untuk proses pengambilan data yang dilakukan pada responden untuk mendapatkan data mentah untuk diolah.
3. Arduino IDE  
Arduino IDE digunakan sebagai media menulis dan mengedit program yang di upload ke Arduino Uno.
4. MATLAB  
MATLAB digunakan sebagai media pelatihan dan pengujian data hasil dari sensor.

### 3.3 Prosedur Penelitian

Pada rancang bangun sistem deteksi kejatuhan untuk pasien lanjut usia ini terdapat beberapa tahapan yang dilakukan diantaranya adalah studi literasi yang bertujuan untuk menemukan informasi yang relevan dengan masalah yang akan diteliti dan supaya penelitian dilandaskan oleh teori yang relevan dengan masalah yang akan diteliti. Selanjutnya adalah perancangan sistem, selain untuk memberikan gambaran umum dari sistem yang akan dibuat, langkah ini juga ditujukan supaya dalam mengerjakan penelitian bisa lebih efektif dan efisien. Semua rancangan dan alur penelitian ini dapat dibaca dan diamati melalui diagram alir di bawah ini.



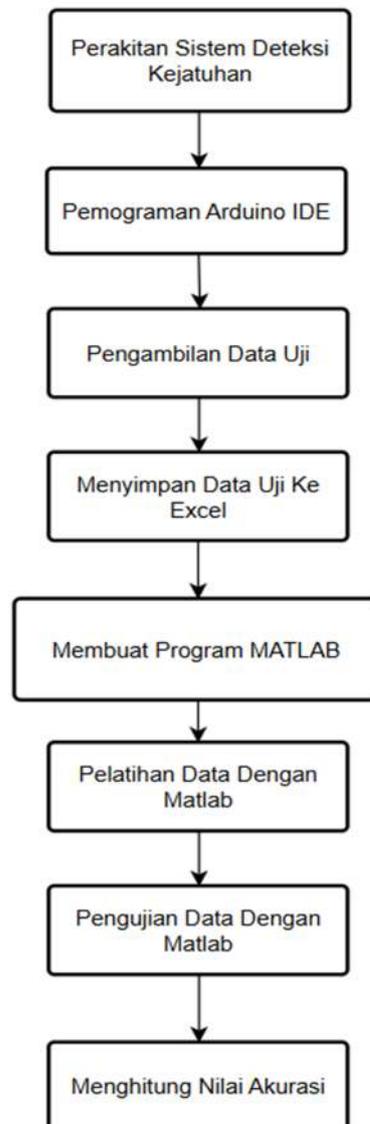
Gambar 3.1 Diagram Alir Penelitian

Pada Gambar 3.1 Setelah sistem selesai dirancang maka sistem akan di uji apakah sesuai dengan yang diinginkan, jika tidak maka akan kembali memperbaiki perancangan sistem dan jika sudah sesuai dengan yang diinginkan, maka data yang dihasilkan akan diolah menggunakan Jaringan Syaraf Tiruan (JST) yang kali ini menggunakan metode *Backpropagation* menggunakan software Matlab. Setelah dilatih, lalu menghasilkan data yang diinginkan, data selanjutnya akan dianalisis

dan dibandingkan dengan sistem tanpa menggunakan Jaringan Syaraf Tiruan (JST). Terakhir jika semua langkah penelitian sudah dikerjakan, maka langkah selanjutnya adalah penulisan laporan.

### 3.4 Perancangan Kerja Jaringan Syaraf Tiruan

Fokus utama pada penelitian ini adalah pada pengolahan data akselerasi dan sudut yang dihasilkan Sistem Deteksi Kejatuhan dengan Modul MPU6050 dan di olah menggunakan *tool* MATLAB. Pengolahan data ini akan melatih sistem untuk mengenal pola gerakan harian dan gerakan jatuh dari lansia, supaya nantinya dapat mengirimkan informasi yang akurat pada pengguna.



Gambar 3.2 Prosedur Kerja Jaringan Syaraf Tiruan

Gambar 3.2 di atas merupakan tahapan kerja Jaringan Syaraf Tiruan (JST) pada system deteksi kejatuhan. Tahapan akan dimulai dengan perakitan sistem deteksi kejatuhan, tahap selanjutnya saat sistem telah selesai dirakit adalah membuat program pada Arduino IDE yang akan digunakan untuk melakukan pengambilan data uji pada responden. Setelah pengambilan data uji selesai, data mentah yang diperoleh akan disimpan ke dalam *Microsoft excel* berupa tabel-tabel supaya memudahkan dalam proses pelatihan yang akan dilakukan JST, setelah dataset siap tahapan berikutnya adalah pembuatan program matlab yang terdiri dari program pelatihan dan program pengujian. Selanjutnya tahapan terakhir dalam JST pada sistem deteksi kejatuhan adalah menghitung nilai akurasi apakah JST pada sistem sudah bekerja dengan yang diharapkan atau tidak.

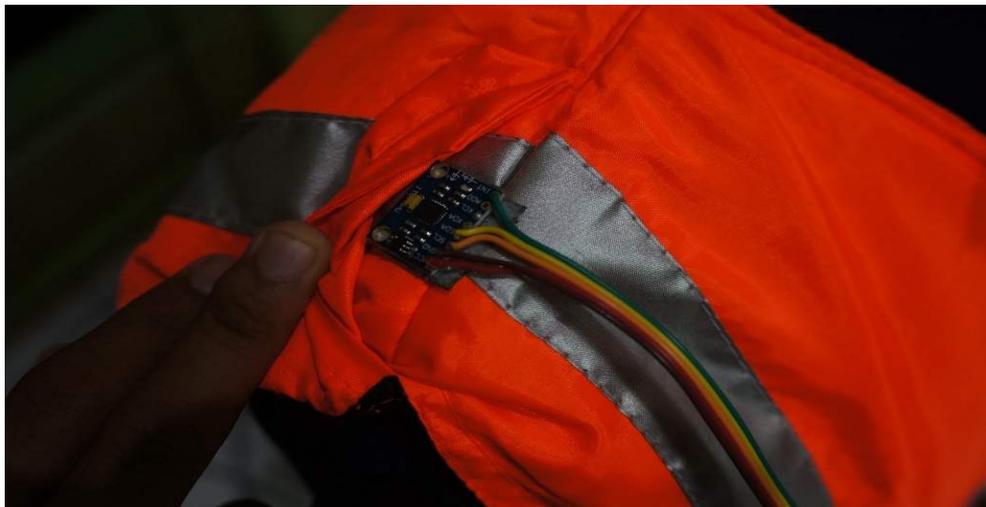
#### 3.4.1 Perakitan Sistem Deteksi Kejatuhan

Tahap ini melibatkan perancangan dan pengintegrasian perangkat keras serta perangkat lunak yang digunakan dalam sistem deteksi kejatuhan. Perangkat utama yang digunakan meliputi sensor *Accelerometer* dan *Gyroscope* MPU6050, serta mikrokontroler Arduino UNO untuk pengambilan data uji yang nantinya akan di proses menggunakan JST. Penempatan komponen alat yang digunakan dapat dilihat pada Gambar 3.3.



Gambar 3.3 Tampilan Sistem Pada Rompi Responden

Pengambilan data sistem deteksi kejatuhan dilakukan melalui koneksi langsung antara Arduino dan laptop menggunakan perangkat lunak Arduino. Sensor *accelerometer* dan *gyroscope* dipasang langsung pada bahu pasien seperti Gambar 3.4, ini karena bahu merupakan bagian tubuh yang sejajar dengan permukaan tanah, pemasangan pada posisi ini memungkinkan konfigurasi awal nilai sudut dan akselerasi dilakukan dengan lebih baik, sehingga hasil deteksi dapat lebih optimal. Untuk memperoleh data perubahan nilai akselerasi dan sudut, yang menjadi variabel penting dalam proses deteksi kejatuhan, pemantauan serta pengambilan data dilakukan secara real-time dengan menghubungkan perangkat secara langsung melalui kabel dari laptop ke Arduino, sehingga informasi yang diperoleh dapat diproses dan dianalisis dengan akurat.



Gambar 3.4 Pemasangan Sensor Pada Bagian Bahu Dari Responden

### 3.4.2 Pemograman Arduino IDE

Program yang digunakan pada sistem deteksi kejatuhan adalah sebagai berikut :

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <MPU6050_tockn.h>
```

```
Adafruit_MPU6050 mpu;
```

```

MPU6050 mpu6050(Wire);

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); //
  Serial.println("Adafruit MPU6050 test!");
  // Try to initialize Adafruit MPU6050
  if (!mpu.begin()) {
    Serial.println("Failed to find Adafruit MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("Adafruit MPU6050 Found!");
}

```

Kode program diatas merupakan inisialisasi awal yang biasa disebut pemanggilan *library* pada Arduino IDE dan merupakan kode yang memberikan perintah untuk mengkonfirmasi jika sensor MPU6050 sudah terdeteksi atau belum. Jika modul tidak terdeteksi maka tulisan yang muncul adalah "*Failed to find Adafruit MPU6050 chip*", namun jika modul terdeteksi maka layar kerja akan memunculkan tulisan "*Adafruit MPU6050 Found!*".

```

// Set up motion detection for Adafruit MPU6050
mpu.setHighPassFilter(MPU6050_HIGHPASS_0_63_HZ);
mpu.setMotionDetectionThreshold(1);
mpu.setMotionDetectionDuration(20);
mpu.setInterruptPinLatch(true); //
mpu.setInterruptPinPolarity(true);
mpu.setMotionInterrupt(true);
// Initialize Tockn MPU6050

```

```

Serial.println("Initializing Tockn MPU6050...");
mpu6050.begin();
mpu6050.calcGyroOffsets(false); // Gyro calibration
Serial.println("MPU6050 ready!");
delay(1000);
}

void loop() {
  mpu6050.update();
  // Get Pitch, Roll, Yaw values from Tockn MPU6050
  float pitch = mpu6050.getAngleX();
  float roll = mpu6050.getAngleY();
  float yaw = mpu6050.getAngleZ();
  // Get Adafruit MPU6050 event data
  sensors_event_t accel, gyro, temp;
  mpu6050.getEvent(&accel, &gyro, &temp);
  float ax = accel.acceleration.x;
  float ay = accel.acceleration.y;
  float az = accel.acceleration.z;
  float gx = gyro.gyro.x;
  float gy = gyro.gyro.y;
  float gz = gyro.gyro.z;

```

Kode diatas merupakan penentuan nilai waktu dan fitur yang akan di ekstraksi dari dalam modul yang akan digunakan pada sistem, dan fitur yang akan digunakan adalah memunculkan nilai perubahan akselerasi pada sumbu Ax, Ay, Az, nilai perubahan sudut pada sumbu Gx, Gy, Gz, dan nilai Pitch, Roll dan Yaw. Dan kode dibawah merupakan kode yang menunjukkan perintah untuk menampilkan nilainya pada layer kerja Arduino IDE.

```

// Output values
Serial.print("AccelX:");

```

```

Serial.print(ax);
Serial.print(", AccelY:");
Serial.print(ay);
Serial.print(", AccelZ:");
Serial.print(az);
Serial.print(", ");
Serial.print("GyroX:");
Serial.print(gx);
Serial.print(", GyroY:");
Serial.print(gy);
Serial.print(", GyroZ:");
Serial.print(gz);
Serial.print(", ");
Serial.print("Pitch:");
Serial.print(pitch);
Serial.print(", Roll:");
Serial.print(roll);
Serial.print(", Yaw:");
Serial.println(yaw);
delay(600);
}

```

### 3.4.3 Pengambilan Data Uji

Pada penelitian ini pengambilan data dilakukan oleh 23 orang responden, dimana 10 orang responden melakukan pengambilan data sebanyak 1x set gerakan, dengan setiap 1 set gerakan terdapat 8 gerakan yang berbeda. Sedangkan untuk 13 orang sisanya melakukan 3x set gerakan, yang secara keseluruhan data yang dihasilkan sebanyak 39200 data. Jika pada 1 gerakan memiliki 10x10 variabel yaitu terdiri dari nilai Ax, Ay, Az, SV, Gx, Gy, Gz, Pitch, Roll, Yaw, dan untuk rentang 10 data dalam 1 kali pengambilan.

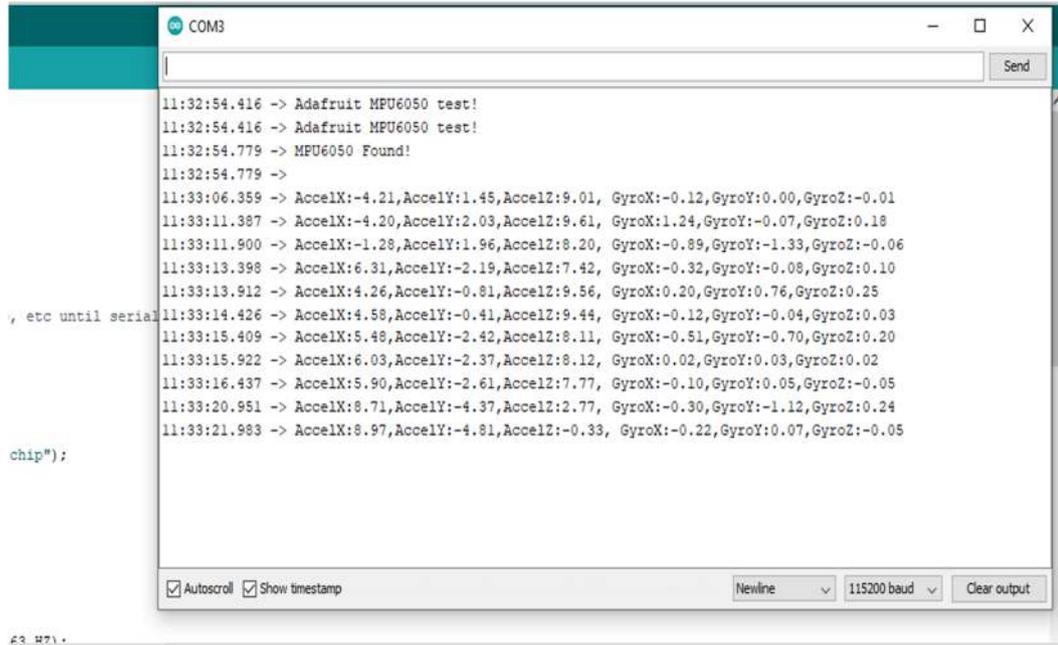
Penelitian ini dilakukan dengan mengumpulkan 8 jenis gerakan kejatuhan serta gerakan sehari-hari, diantaranya adalah jatuh ke depan, jatuh ke samping, jatuh ke belakang, berdiri ke duduk, berdiri ke tidur, berdiri ke rukuk, berdiri ke sujud, dan rukuk ke sujud. Proses pengambilan data gerakan jatuh dan gerakan sehari-hari pada responden yang dilakukan di atas sebuah matras, dan sistem yang terpasang pada rompi yang digunakan oleh responden terhubung secara langsung dengan laptop seperti pada gambar dibawah.



Gambar 3.5 Proses Pengambilan Data Pada Responden

#### 3.4.4 Menyimpan Data Uji Ke Excel

Proses ini merupakan proses memindahkan data mentah yang telah disalin sebelumnya dari layer kerja Arduino IDE ke note, dipindahkan dan dirapihkan menjadi tabel dalam *Microsoft excel* supaya data uji dapat terbaca dengan baik oleh program JST pada Matlab nantinya.



Gambar 3.6 Tampilan Data Mentah Pada Layar Kerja Arduino IDE

Berikut merupakan contoh data gerakan jatuh ke depan yang sudah disusun pada excel :

Tabel 3.1 Tabel Data Gerakan Jatuh Ke Depan

| <b>Ax</b> | <b>Ay</b> | <b>Az</b> | <b>Gx</b> | <b>Gy</b> | <b>Gz</b> | <b>Pitch</b> | <b>Roll</b> | <b>Yaw</b> |
|-----------|-----------|-----------|-----------|-----------|-----------|--------------|-------------|------------|
| -0,74     | -1,89     | 9,98      | -0,3      | -0,52     | -0,02     | 18,83        | 62,14       | -71,03     |
| 7         | -3,6      | 8,04      | 0,33      | 0,25      | 0,32      | 17,8         | 67,12       | -76,02     |
| 9,44      | 0,86      | 3,69      | -0,25     | -0,05     | 0,08      | 21,99        | 59,48       | -69,88     |
| 10,45     | 1,01      | 3,15      | -0,02     | -0,26     | -0,03     | 17,8         | 67,12       | -76,02     |
| 10,36     | 1,69      | 2,33      | -0,02     | -0,22     | 0,01      | 21,99        | 59,48       | -69,88     |
| 9,64      | 1,61      | 2,69      | -0,14     | 0,18      | -0,02     | 19,77        | 57,6        | -65,4      |
| 9,53      | 1,01      | 3,11      | -0,11     | 0,11      | -0,02     | 23,14        | 44,94       | -53,9      |
| 9,36      | 1,22      | 3,23      | -0,14     | 0,03      | 0         | 32,49        | 46,69       | -64,68     |
| 9,89      | 1,03      | 2,64      | -0,11     | -0,05     | 0,03      | 24,17        | 52,72       | -63,18     |
| 9,28      | 1,55      | 2,59      | -0,07     | 0,12      | 0,05      | 20,06        | 43,74       | -50,49     |

### 3.4.5 Membuat Program Matlab

Matlab atau (*Matrix Laboratory*) merupakan bahasa pemrograman yang dibuat dengan tujuan sebagai alat bantu perhitungan yang rumit atau simulasi dari suatu sistem yang ingin di simulasikan. Dalam penelitian ini matlab digunakan sebagai *software* utama baik dalam pelatihan maupun pengujian data.

#### 3.4.5.1 Perancangan Program Pelatihan Matlab

Berikut merupakan program yang digunakan dalam proses pelatihan data hasil pada Matlab.

```
clc; clear; close all;

% Baca data dari file Excel
filename = 'Data_kejatuhan.xlsx';
data = readtable(filename);

% kita asumsikan data memiliki 3280 sampel (baris) dan 10 fitur (kolom)
input_raw = table2array(data(:, 1:10)); % Tidak ditranspose

% Jumlah total sampel
total_samples = size(input_raw, 1);
sample_size = 10;

% Restrukturisasi data menjadi kelompok 10x10
input = zeros(sample_size^2, total_samples / sample_size);
for i = 1:total_samples/sample_size
    segment = input_raw((i-1)*sample_size+1:i*sample_size, :);
    input(:, i) = segment(:);
end
```

Berdasarkan kode matlab diatas merupakan tahap awal perancangan koding yang memberikan perintah pemanggilan data yang berupa file.xlsx yang sudah disiapkan. Pada 2 paragraf selanjutnya merupakan inialisai jumlah sampel data latih yang dalam penelitian ini terdapat 3280 jumlah baris dan 10 jumlah kolom. Dari data yang telah didapatkan masih harus di kelompokkan menjadi sebuah data yang berpola mewakili dari sebuah gerakan dengan pola matriks 10x10 untuk 1 pola gerakan seperti pada tabel data sebelumnya.

```
% Normalisasi input
input_min = min(input, [], 2);
```

```

input_max = max(input, [], 2);
input_norm = (input - input_min) ./ (input_max - input_min);

%menyimpan nilai min dan max
save('normalization_params.mat','input_min','input_max');

% Jumlah total sampel yang telah dikelompokkan
total_samples_grouped = size(input_norm, 2);

% Target Pelatihan (2 skenario)

% Skenario 1
% target = zeros(1, total_samples_grouped);
% target(1:41) = 1;
% target(42:82) = 2;
% target(83:123) = 3;
% target(124:164) = 4;
% target(164:205) = 5;
% target(206:246) = 6;
% target(247:287) = 7;
% target(288:328) = 8;

% Skenario 2
target = zeros(1, total_samples_grouped);
target(1:123) = 1;
target(124:328) = 2;

```

Berdasarkan kode matlab diatas merupakan kode yang memberikan perintah untuk menormalisasi data latih yang sudah di konstruksi menjadi matrik pola gerakan dan menyimpan nilai *minimum dan maximum* yang akan digunakan kembali saat pengujian, ini bertujuan agar parameter pelatihan dan pengujian memiliki nilai yang sama. Melabeli setiap pola gerakan dengan inisial yang mewakili masing-masing gerakan, diantaranya adalah label angka 1 untuk jatuh kedepan, angka 2 untuk jatuh ke samping, angka 3 untuk jatuh ke belakang, angka 4 untuk gerakan berdiri ke duduk, angka 5 untuk gerakan berdiri ke tidur, angka untuk gerakan berdiri ke rukuk, angka 7 untuk gerakan berdiri ke sujud, angka 8 untuk gerakan duduk ke sujud.

```

% Bangun jaringan saraf tiruan
net = feedforwardnet([10 5 10], 'trainlm');
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'logsig';

```

```

net.layers{3}.transferFcn = 'logsig';

% Tentukan parameter pelatihan

net.trainParam.epochs = 1000; % Jumlah epoch
net.trainParam.goal = 1e-5; % Tujuan MSE (Mean Squared Error)

% Latih jaringan saraf
net = train(net, input_norm, target);

% Simulasikan jaringan saraf dan bulatkan hasilnya
output = round(net(input_norm));

% Simpan jaringan saraf terlatih
save net.mat net

```

Berdasarkan kode diatas merupakan perintah untuk membangun sebuah jaringan syaraf tiruan yang digunakan untuk melatih data yang telah siap dinormalisasi sebelumnya. Pada jaringan syaraf tiruan yang digunakan untuk melatih data latih ini memiliki 3 buah layer tersembunyi yang disetiap layernya memiliki masing-masing 10, 5, dan 10 neuron. Selama proses pelatihan, jaringan syaraf tiruan akan mengalami banyak iterasi (*epoch*). Setiap *epoch*, bobot jaringan akan diperbarui untuk meminimalkan *error* yang menggunakan MSE (*Mean Squared Error*). Proses ini akan terus berlanjut sampai jumlah epoch yang telah ditentukan tercapai atau hingga MSE mencapai nilai yang diinginkan, dalam pelatihan ini batas jumlah *epoch* nya adalah 1000, dan nilai MSE nya adalah 0,00001.

```

% Menampilkan grafik keluaran vs target

figure;
plot(target, 'b');
hold on;
plot(output, 'r');
legend('Target', 'Output');
title('Grafik Keluaran Jaringan vs Target');
xlabel('Data Index');
ylabel('Nilai');
hold off;

% Menghitung error
nilai_error = target - output;

% Menampilkan grafik error

```

```

figure;
plot(nilai_error, 'k');
title('Grafik Error Jaringan Saraf Tiruan');
xlabel('Data Index');
ylabel('Error');

% Tampilkan hasil

disp('Input Normalized:');
disp(input_norm);
disp('Target:');
disp(target);
disp('Output:');
disp(output);

% Evaluasi hasil
accuracy = sum(output == target) / total_samples_grouped * 100;
fprintf('Akurasi: %.2f%%\n', accuracy);

```

Program di atas merupakan bagian akhir yang berisi perintah untuk menampilkan nilai error, grafik error, hasil pelatihan, serta persentase akurasi dari hasil pelatihan.

Pada tahap ini program matlab akan dibuat menjadi 2 skenario, yaitu pada skenario pertama, parameter pelatihan menggunakan data berlabel yang mewakili setiap pola gerakan dengan angka 1 hingga 8, di mana angka-angka ini digunakan untuk menunjukkan delapan pola gerakan yang dilatih. Sedangkan pada skenario kedua, setiap pola gerakan dikelompokkan menjadi dua kategori berdasarkan pelabelan angka 1 untuk "jatuh" dan angka 2 untuk "tidak jatuh". Perbedaannya, skenario pertama membedakan berbagai pola gerakan secara spesifik, sementara skenario kedua hanya membedakan antara gerakan yang mengindikasikan jatuh atau tidak.

#### 3.4.5.2 Perancangan Program Pengujian Matlab

Setelah dilakukan pelatihan data langkah selanjutnya adalah pengujian data hasil yang sudah dilatih sebelumnya pada matlab, pengujian data latih juga dilakukan dengan menggunakan 2 skenario yang sama pada proses pelatihan. Berikut adalah program pengujian data latih pada Matlab.

```

clc; clear; close all;

% Baca data uji dari file Excel

```

```

filename_test = 'Data_uji.xlsx'; % Ganti dengan file data uji yang
sesuai
data_test = readtable(filename_test);

% Asumsikan data uji memiliki 3280 sampel (baris) dan 10 fitur
(kolom)
% Ekstrak data menjadi matriks input
input_raw_test = table2array(data_test(:, 1:10)); % Tidak
ditranspose

% Jumlah total sampel uji
total_samples_test = size(input_raw_test, 1);
sample_size = 10; % Sesuaikan dengan ukuran sampel pada pelatihan

% Restrukturisasi data uji menjadi kelompok 10x10
input_test = zeros(sample_size^2, total_samples_test / sample_size);
for i = 1:total_samples_test/sample_size
    segment = input_raw_test((i-1)*sample_size+1:i*sample_size, :);
    input_test(:, i) = segment(:);
end

% Muat nilai min dan max untuk normalisasi dari pelatihan
load('normalization_paramsSK2.mat', 'input_min', 'input_max');

```

Berdasarkan kode diatas merupakan tahapan awal yaitu pemanggilan data latih yang sudah tersimpan, dan 2 paragraf selanjutnya merupakan inisialisasi jumlah sampel yang sudah di panggil. Pada akhir paragraph kode diatas merupakan pemanggilan nilai *minimum* dan nilai *maximum* yang digunakan pada proses pelatihan sebelumnya.

```

% Normalisasi data uji menggunakan min dan max dari data pelatihan
% input_min = min(input_test, [], 2);
% input_max = max(input_test, [], 2);
input_test_norm = (input_test - input_min) ./ (input_max -
input_min);
% Jumlah total sampel uji setelah pengelompokan
total_samples_test_grouped = size(input_test_norm, 2);

% Muat jaringan saraf yang telah dilatih
load('netSK1.mat', 'net');

% Simulasikan jaringan saraf pada data uji dan hitung output
(logits)

```

```

output_test_raw = net(input_test_norm);
output_test = round(output_test_raw);

% Menampilkan grafik keluaran prediksi dan target

% Target uji
target_uji = zeros(1, total_samples_test_grouped);
target_uji(1:8) = 1;
target_uji(9:16) = 2;
target_uji(17:24) = 3;
target_uji(25:32) = 4;
target_uji(33:40) = 5;
target_uji(41:48) = 6;
target_uji(49:56) = 7;
target_uji(57:64) = 8;

% target_uji(1:24) = 1;
% target_uji(25:64) = 2;

```

Pada paragraf pertama kode diatas merupakan tahapan normalisasi nilai, dan perhitungan jumlah total sampel uji setelah pengelompokan data pada program sebelumnya, pada paragraf selanjutnya yaitu perintah untuk memanggil jaringan syaraf tiruan yang telah dilatih sebelumnya dan mengujinya pada data uji untuk menghitung nilai *output* yang akan dibandingkan dengan target uji yang ditetapkan pada paragraf selanjutnya.

```

% Tampilkan grafik perbandingan antara target dan output prediksi
figure;
subplot(2, 1, 1);
plot(output_test, 'r', 'LineWidth', 1.5);
title('Grafik Keluaran Jaringan Saraf pada Data Uji');
xlabel('Index Data Uji');
ylabel('Kelas Prediksi');
legend('Output Prediksi');
grid on;

% Pastikan target_uji sudah didefinisikan dengan benar.
subplot(2, 1, 2);
plot(target_uji, 'b', 'LineWidth', 1.5);
title('Grafik Target Kelas pada Data Uji');
xlabel('Index Data Uji');
ylabel('Kelas Target');
legend('Target Kelas');
grid on;

% Hitung akurasi

```

```

accuracy_test = sum(output_test == target_uji) / length(target_uji)
* 100;
fprintf('Akurasi pada data uji: %.2f%%\n', accuracy_test);
fprintf('Akurasi Data Uji: %.2f%%\n', accuracy_test);

% Hitung jumlah prediksi yang benar dan salah
correct_predictions = sum(output_test == target_uji);
wrong_predictions = sum(output_test ~= target_uji);

% Indeks data yang salah
wrong_indices = find(output_test ~= target_uji);

% Tampilkan hasil
fprintf('Jumlah Prediksi Benar: %d\n', correct_predictions);
fprintf('Jumlah Prediksi Salah: %d\n', wrong_predictions);
disp(output_test);
fprintf('Indeks Data yang Salah: %s\n', mat2str(wrong_indices));

% Simpan hasil prediksi
save('output_test.mat', 'output_test');

```

Kode diatas merupakan bagian akhir program yang pada paragraf pertama dan kedua program memiliki perintah untuk menampilkan grafik hasil pengujian dan grafik target, untuk melihat perbandingan diantara keduanya. Pada paragraph selanjutnya memiliki perintah untuk menghitung akurasi nilai hasil pengujian berdasarkan target hasil yang sudah ditentukan dan menemukan jumlah prediksi yang salah oleh sistem, dan untuk paragraf terakhir program merupakan perintah untuk menyimpan hasil prediksi data uji.

### 3.4.6 Menghitung Nilai Akurasi

Pada tahap ini nilai yang sudah di latih dan nilai yang telah diuji dengan program matlab selanjutnya akan dihitung nilai akurasi, presisi, *recall*, dan akurasinya untuk melihat apakah proses pelatihan data latih sudah berjalan dengan yang diharapkan baik atau belum. Salah satu metode evaluasi yang umum digunakan dalam sistem

klasifikasi biner adalah *confusion matrix*. Metode ini memberikan gambaran menyeluruh mengenai performa sistem dalam mengklasifikasikan data masukan berdasarkan kondisi aktual dan hasil prediksi. *Confusion matrix* menyajikan hasil prediksi model dalam bentuk matriks dua dimensi yang membandingkan antara label aktual (*ground truth*) dengan label prediksi. Matriks ini terdiri dari empat komponen utama, yaitu:

1. *True Positive* (TP) adalah jumlah data yang diprediksi sebagai "jatuh" dan benar-benar merupakan data kejadian jatuh.
2. *True Negative* (TN) adalah jumlah data yang diprediksi sebagai "tidak jatuh" dan memang benar merupakan data tidak jatuh.
3. *False Positive* (FP) adalah jumlah data yang diprediksi sebagai "jatuh", padahal sebenarnya merupakan data tidak jatuh (kesalahan positif).
4. *False Negative* (FN) adalah jumlah data yang diprediksi sebagai "tidak jatuh", padahal sebenarnya merupakan data kejadian jatuh (kesalahan negatif).

Dengan menggunakan keempat nilai tersebut, maka dapat dihitung beberapa metrik evaluasi kinerja sistem sebagai berikut:

#### 1. Akurasi

Akurasi menunjukkan sejauh mana sistem mampu mengklasifikasikan data dengan benar, baik data jatuh maupun tidak jatuh. Nilai akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (3.1)$$

#### 2. Presisi

Presisi mengukur tingkat ketepatan sistem dalam mendeteksi kejadian jatuh, yaitu seberapa besar prediksi "jatuh" yang benar-benar merupakan kejadian jatuh. Rumus presisi adalah:

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (3.2)$$

### 3. *Recall*

*Recall* mengukur sensitivitas sistem terhadap kejadian jatuh, yaitu seberapa banyak data jatuh yang berhasil terdeteksi dengan benar. Rumus recall adalah:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3.3)$$

### 4. *F1-Score*

*F1-score* merupakan metrik gabungan dari presisi dan *recall* yang digunakan untuk memberikan penilaian yang lebih seimbang, khususnya jika terdapat ketidakseimbangan jumlah data antar kelas. Rumus *F1-score* adalah :

$$\text{F1-score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (3.4)$$

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Adapun kesimpulan dari penelitian ini adalah sebagai berikut :

1. Penelitian ini berhasil merancang arsitektur dan algoritma program untuk mendeteksi pola gerakan jatuh dan tidak jatuh menggunakan metode *backpropagation* pada jaringan syaraf tiruan. Sistem yang dibangun mampu melakukan pelatihan dan prediksi pola gerakan berdasarkan data dari sensor *accelerometer* dan *gyroscope*. Hasil pelatihan menunjukkan bahwa skenario pertama menghasilkan akurasi pelatihan sebesar 83,23%, sedangkan skenario kedua memberikan hasil yang lebih tinggi dengan akurasi pelatihan sebesar 97,87%, yang menunjukkan bahwa klasifikasi dua kelas di skenario kedua lebih efektif dari pada skenario pertama dalam proses pelatihan jaringan syaraf tiruan.
2. Implementasi sistem deteksi dalam perangkat lunak MATLAB memberikan hasil yang memiliki nilai akurat yang tinggi proses pengujian. Berdasarkan pengujian terhadap 64 pola gerakan, skenario pertama berhasil memprediksi 60 pola gerakan dengan benar sehingga menghasilkan akurasi pengujian sebesar 93%, sedangkan skenario kedua berhasil memprediksi 63 pola gerakan dengan benar dan menghasilkan akurasi pengujian sebesar 98,4%. Hasil ini menunjukkan bahwa MATLAB mendukung pelatihan dan evaluasi sistem secara optimal, serta bahwa klasifikasi dua kelas memberikan performa deteksi yang lebih unggul dibandingkan klasifikasi *multi*-kelas.

## 5.2 Saran

Adapun saran untuk pengembangan penelitian di masa kedepannya adalah dengan memperbanyak jumlah dataset serta menambah variasi dari pola gerakan yang sudah ditentukan. Misalnya, untuk gerakan jatuh ke depan, dapat divariasikan menjadi jatuh langsung, jatuh perlahan, atau jatuh dengan gerakan tertentu seperti duduk dahulu dan kemudian tersungkur. Hal ini bertujuan agar model dapat mengenali berbagai kondisi gerakan dengan lebih akurat. Selain itu, proses pengambilan data disarankan untuk konsisten dalam menentukan kurun gerakan, juga menambahkan fitur penyimpanan memori yang memungkinkan sistem menyimpan data pelatihan sebelumnya, sehingga mempermudah pengelolaan dataset dan meningkatkan efisiensi dalam proses pelatihan model.

## DAFTAR PUSTAKA

- [1] S. S. Jun et al., "Development of fall detection and activity recognition using threshold based method and neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1338–1347, 2019, doi: 10.11591/ijeecs.v17.i3.pp1338-1347.
- [2] R. Fajar Santoso, "Rancang Bangun Sistem Deteksi Kejatuhan Untuk Pasien Lansia Berbasis Sensor Gyroscope Mpu6050," 2018.
- [3] F. Idris, D. T. P. Yanto, A. Aswardi, and M. Yuhendri, "Rancang Bangun Pemantauan Pergerakan Orang Lanjut Usia Berbasis Mikrokontroler," *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 4, no. 2, Aug. 2023, doi: 10.24036/jtein.v4i2.450.
- [4] M. Hardjianto, M. A. Rony, and G. S. Trengginas, "Prosiding Sentia 2016- Politeknik Negeri Malang Deteksi Jatuh Pada Lansia Dengan Menggunakan Akselerometer Pada Smartphone". 2016.
- [5] *Proceeding, 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics: September 19-21, 2017, Grand Mercure, Yogyakarta, Indonesia. IEEE, 2017.*
- [6] R. R. Saragih, "Pemrograman Dan Bahasa Pemrograman."
- [7] S. J. Sokop, D. J. Mamahit, M. Eng, S. R. U. A. Sompie, Mahasiswa, and Pembimbing, "Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno," *Journal Teknik Elektro dan Komputer*, vol. 5, no. 3, 2016.
- [8] Kurniawan, D. *Projek Akhir*, "Penerapan Sensor Akselometer Mpu6050 Sebagai Sensor Bidang Miring Dengan Tampilan Visual Grafik Berbasis Atmega 328."
- [9] "Sudarsono, A. "Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation (Studi

- Kasus Di Kota Bengkulu)”. *Jurnal Media Infotama*, Vol. 12, no. 1, Juni 2016.
- [10] A. Atina, “Aplikasi Matlab pada Teknologi Pencitraan Medis,” *Jurnal Penelitian Fisika dan Terapannya (JUPITER)*, vol. 1, no. 1, p. 28, Aug. 2019, doi: 10.31851/jupiter.v1i1.3123.
- [11] Liu J, Li X, Huang S, Chao R, Cao Z, Wang S, et al. “A review of wearable sensors based fall-related recognition systems”. *Eng Appl Artif Intell*. 2023;121:105993. [doi: 10.1016/j.engappai.2023.105993]
- [12] Jain R, Semwal VB. “A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors”. *IEEE Sensors J*. 2022;22(23):22943-22951. [doi: 10.1109/jsen.2022.3213814]
- [13] Sadreazami H, Bolic M, Rajan S. “Fall detection using standoff radar-based sensing and deep convolutional neural network”. *IEEE Trans Circuits Syst II Express Briefs*. 2020;67(1):197-201. [doi: 10.1109/tcsii.2019.2904498]
- [14] Kerdegari H, Mokaram S, Samsudin K, Ramli AR. “A pervasive neural network based fall detection system on smart phone”. *J Ambient Intell Smart Environ*. 2015;7(2):221-230. [doi: 10.3233/ais-150306]
- [15] R. N. Shabrina, W. N. Widiana, N. Sarah, R. W. T. Hartono, and K. Kunci, “e -Motion : Smart Remote Internet of Things Based Of Elderly Body Movements,” pp. 4–5, 2021.
- [16] Sumijan, S., Windarto, A. P., Muhammad, A., & Budiharjo, B. Implementation of Neural Networks in Predicting the Understanding Level of Students Subject. *International Journal of Software Engineering and Its Applications*, 2016. 10(10), 189-204.
- [17] H. Alpin, “Hubungan Fungsi Gerak Sendi Dengan Tingkat Kemandirian Lansia Di Panti Sosial Tresna Werdha Gau Mabaji Kabupaten Gowa,” *J. Ilm. Kesehat. Sandi Husada*, vol. 4, no. 1, pp. 43–49, 2016.

- [18] Ramadhana . M. Satrio Y, Waskitho W, Henning Titi C, “Rancang Bangun Sistem Fall Detection dengan Analisa Percepatan sudut rotasi (Quaternion) secara Real-time menggunakan Mikrokontroler Arduino”,2017.
- [19] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian and B. Q. Tran, "Optimization of an Accelerometer and Gyroscope-Based Fall," Journal of Sensors, pp. 2-8, 2015.
- [20] Rahma. Desti, “Rancang Bangun Alat Bantu Pendeteksi Kejatuhan Pada Lansia Menggunakan Accelerometer Dan Gyroscope”, Politeknik Negeri Sriwijaya, 2023.
- [21] Effendy, Ryan. Perancangan Prototipe pendeteksi Gerakan Jatuh Pada Lansia Menggunakan Sensor Accelerometer Berbasis IoT. Yogyakarta, Universitas Islam Indonesia. 2020.
- [22] Helmy. F, Sri. Purwiyanti, Ezza A. F, Ma’ruf F. S, Anisa U. D, Teddy S. G, “Development of a Low-Cost Fall Detection System for the Elderly with Accurate Detection and Real-Time Alerts”, Universitas Lampung, 2024