



TUGAS AKHIR (AE43250)

**KLASIFIKASI AKTIVITAS *WELDER* BERBASIS DATA
ORIENTASI *QUATERNION* MENGGUNAKAN METODE
CNN**

Dela Almal Liani
0922040009

Dosen Pembimbing
AULIA RAHMA ANNISA, S.ST., M.T.
MOHAMMAD BASUKI RAHMAT, S.T., M.T.

PROGRAM STUDI D4 TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR (AE43250)

**KLASIFIKASI AKTIVITAS WELDER BERBASIS DATA
ORIENTASI QUATERNION MENGGUNAKAN METODE
CNN**

Dela Almal Liani
NRP. 0922040009

Dosen Pembimbing
AULIA RAHMA ANNISA, S.ST., M.T.
MOHAMMAD BASUKI RAHMAT, S.T., M.T.

PROGRAM STUDI D4 TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

KLASIFIKASI AKTIVITAS *WELDER* BERBASIS DATA ORIENTASI *QUATERNION* MENGGUNAKAN METODE CNN

Disusun Oleh:
Dela Almal Liani
0922040009

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi D4 Teknik Otomasi
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim penguji Tugas Akhir Tanggal Ujian : 27 Juli 2026
Periode Wisuda : Oktober 2026

Menyetujui,

Dosen Penguji

1. Noorman Rinanto, S.T., M.T., Ph.D.

NIDN/NUPTK

(0014107607)

Tanda Tangan

(.....)

2. Aulia Rahma Annisa, S.ST., M.T.

(0303069102)

(.....)

3. Hendro Agus Widodo, S.ST., M.T.

(0013076902)

(.....)

4. Sholahuddin Muhammad Irsyad, S.Tr.T., M.Tr.T.

(0337772673130343)

(.....)

Dosen Pembimbing

1. Aulia Rahma Annisa, S.ST., M.T.

NIDN/NUPTK

(0303069102)

Tanda Tangan

(.....)

2. Mohammad Basuki Rahmat, S.T., M.T.

(0022057304)

(.....)

Menyetujui

Ketua Jurusan,

Mengetahui

Koordinator Program Studi,



Isa Rahman, S.T., M.T.

NIP. 198008162008121001

Agus Khumaidi, S.ST., M.T.

NIP. 199308172020121004

(Halaman ini sengaja dikosongkan)

 PPNS	<u>PERNYATAAN BEBAS PLAGIAT</u>	No. : F.WD I. 021 Date : 3 Nopember 2015 Rev. : 01 Page : 1 dari 1
---	---	--

Yang bertandatangan dibawah ini :

Nama : Dela Almal Liani

NRP. : 0922040009

Jurusan/Prodi : Teknik Kelistrikan Kapal/Teknik Otomasi

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

KLASIFIKASI AKTIVITAS *WELDER* BERBASIS DATA ORIENTASI *QUATERNION* MENGGUNAKAN METODE CNN

Adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain.

Apabila di kemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut, maka saya bersedia menerima **sanksi** sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 18 Agustus 2026

Yang membuat pernyataan,



(Dela Almal Liani)

NRP. 0922040009

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Dengan penuh rasa syukur, segala puji dan syukur ke hadirat Allah SWT atas limpahan rahmat dan pertolongan-Nya, sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir yang berjudul “Klasifikasi Aktivitas *Welder* Brebasis Data Orientasi *Quaternion* Menggunakan Metode CNN” dengan baik.

Tugas Akhir ini disusun oleh penulis dengan tujuan sebagai salah satu syarat untuk menyelesaikan Program Studi D4 - Teknik Otomasi di Politeknik Perkapalan Negeri Surabaya, penulis menyadari bahwa keberhasilan yang dicapai berkat dukungan, bimbingan, dan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang telah melimpahkan rahmat, karunia, dan pertolongan-Nya yang tak terhenti-henti sehingga penulis dapat menyelesaikan perkuliahan pada jarak 800km dari rumah hingga akhir.
2. Teruntuk Ayahanda dan Ibunda, yang merupakan alasan terbesar di balik setiap langkah dan pencapaian penulis. Dua permata hati yang dititipkan Tuhan untuk membantu mengurangi arus deras kehidupan. Terasa sulit hidup ini tanpa kehadiranmu. Selalu ku rayu tuhan agar kalian diberikannya umur yang panjang dan sehat. Beliau memang tidak sempat merasakan bangku perkuliahan, namun tidak pernah sekalipun keterbatasan tersebut menjadi penghalang bagi anak-anaknya untuk mengenyam pendidikan setinggi mungkin. Terima kasih hari ini, kuhadiahkan gelar S.Tr.T. sebagai bentuk bakti dan syukurku.
3. Almarhum Bapak Mukarrom yang sudah penulis anggap seperti orang tua kedua penulis di Surabaya, terima kasih sudah merawat dan memberikan tempat tinggal yang nyaman tanpa mengeluarkan uang sepeserpun, semoga menjadi amal ibadah bapak diakhirat, aamiin.
4. Terima kasih untuk diri saya sendiri. Apresiasi sebesar-besarnya karena sudah berjuang menyelesaikan apa yang sudah dimulai. Sulit bisa bertahan sampai di titik ini, terima kasih untuk tetap hidup dan merayakan dirimu sendiri, Tetaplah jadi manusia yang berusaha dan tidak kenal lelah untuk mencoba “*One Piece 988 20:35*”.

5. Adik adikku tercinta Indra, Silvi, dan Rama telah menjadi penyemangat penulis di setiap perjuangan untuk menyelesaikan perkuliahan. Terima kasih atas dukungan dan semangatmu yang selalu hadir, bahkan dalam cara sederhana sekalipun, kehadiranmu membuat perjalanan lebih berwarna.
6. Teruntuk teman-teman tersayang, utamanya kepada Fadel Muhammad Ilman, Cindy Nur Laily S, Ariba, Chika, Ega, Edo dan teman-teman yang tidak bisa disebut satu persatu, telah kebersamai penulis menyelesaikan penyusunan Tugas Akhir di tengah suka, duka, dan pasang surut kehidupan. Terima kasih selalu memberikan dukungan dan mendengarkan keluh kesah penulis selama proses penyusunan berlangsung.
7. Bapak Isa Rachman, S.T., M.T. selaku Ketua Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya.
8. Bapak Agus Khumaidi, S.ST., M.T. selaku Koordinator Program Studi Teknik Otomasi Politeknik Perkapalan Negeri Surabaya.
9. Ibu Aulia Rahma Annisa, S.ST., M.T. selaku Dosen Pembimbing I, terima kasih telah sabar membimbing, memberikan dukungan dan solusi sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
10. Bapak Mohammad Basuki Rahmat, S.T., M..T. selaku Dosen Pembimbing II, terima kasih atas bimbingan, dukungan, dana, solusi, serta kepercayaan yang telah Bapak berikan selama proses penyusunan Tugas Akhir ini, sehingga dapat terselesaikan dengan baik. Terima kasih pula karena telah menganggap saya seperti anak sendiri.
11. Bapak Noorman Rinanto, S.T., M.T., Ph.D., Bapak Hendro Agus Widodo, S.ST., M.T., Bapak Sholahuddin Muhammad Irsyad, S.Tr.T., M.Tr.T. selaku Dosen Penguji, terima kasih atas bimbingan, memberikan dukungan dan solusi sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
12. Bapak Ryan Yudha Adhitya, S.ST., M.T. selaku Koordinator Tugas Akhir Teknik Otomasi yang telah memberikan pengarahan dan motivasi dalam mengerjakan Tugas Akhir ini.
13. Seluruh Dosen Jurusan Teknik Kelistrikan Kapal PPNS yang telah memberikan ilmu dan bimbingannya selama penulis melaksanakan studi.

14. Seluruh keluarga besar Teknik Otomasi khususnya Angkatan 2022 terutama kelas A yang kebersamaan penulis selama 4 tahun yang berkesan dan akan selalu terkenang dalam hidup penulis.
15. Team MRC CHAMPIONSHIPS tempat dimana penulis belajar banyak perihal robotika, pemrograman, dan banyak kenangan yang penulis rasakan selama bergabung di team championships kenangan yang tidak bisa di lupakan.
16. Serta pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis dalam proses penyusunan Tugas Akhir.

Penulis menyadari bahwa dalam pengerjaan dan penulisan Tugas Akhir ini masih banyak kekurangan. Sehingga masukan, kritik dan saran untuk evaluasi sangat dihargai. Penulis juga berharap semoga Tugas Akhir ini dapat bermanfaat dan berguna bagi penulis sendiri dan bagi pembacanya.

Penulis



(Dela Almal Liani)

(Halaman ini sengaja dikosongkan)

KLASIFIKASI AKTIVITAS *WELDER* BERBASIS DATA ORIENTASI *QUATERNION* MENGUNAKAN METODE CNN

Dela Almal Liani

ABSTRAK

Pemantauan aktivitas *welder* secara manual masih terbatas karena bergantung pada observasi manusia, sehingga kurang objektif dan tidak dapat dilakukan secara berkelanjutan. Penelitian ini merancang sistem *Human Activity Recognition* (HAR) berbasis *wearable embedded system* untuk mengklasifikasikan enam aktivitas *welder* (Duduk Aktif, Berdiri Aktif, Berbaring Aktif, Duduk Tidak Aktif, Berdiri Tidak Aktif, dan Berbaring Tidak Aktif) menggunakan sensor IMU ICM-20948. Data gerakan diproses melalui tahap *filtering*, normalisasi, dan segmentasi dengan metode *sliding window* sebagai masukan untuk model *Convolutional Neural Network* 1D (CNN-1D), dengan hasil klasifikasi divalidasi menggunakan kamera sebagai *ground truth*. Pengujian menunjukkan bahwa model CNN-1D dengan data *quaternion* mencapai akurasi 99,86%, lebih tinggi dibandingkan dengan model tanpa *quaternion* yang mencapai 96,03%. Hal ini membuktikan bahwa informasi orientasi *quaternion* meningkatkan kemampuan model dalam mengenali pola gerakan *welder*. Hasil *classification report* menunjukkan *precision*, *recall*, dan *F1-score* diperoleh rata-rata sebesar 1,00 pada kelas Berbaring Aktif, Berbaring Tidak Aktif, Berdiri Aktif, dan Duduk Tidak Aktif, sementara kelas Berdiri Tidak Aktif memperoleh *recall* 0,99 dan kelas Duduk Aktif memperoleh *precision* 0,99 dari total 724 data uji, menunjukkan model mampu mengenali keenam kelas aktivitas secara akurat tanpa *bias*. Sistem mampu melakukan klasifikasi secara *real-time* dengan waktu inferensi rata-rata 110,58 ms per *window*, serta menampilkan hasil pemantauan dan durasi aktivitas melalui *dashboard* berbasis web. Sistem *wearable* berbasis IMU dan CNN-1D terbukti dapat digunakan sebagai teknologi pendukung untuk memantau aktivitas kerja *welder* secara objektif dan berkelanjutan.

Kata kunci: CNN-1D, HAR, IMU, *Quaternion*, *Wearable Embedded System*.

(Halaman ini sengaja dikosongkan)

CLASSIFICATION OF WELDER ACTIVITIES BASED ON QUATERNION ORIENTATION DATA USING THE CNN METHOD

Dela Almal Liani

ABSTRACT

Manual monitoring of welder activity remains limited because it relies on human observation, making it less objective and unsustainable. This study designed a Human Activity Recognition (HAR) system based on a wearable embedded system to classify six welder activities (Active Sitting, Active Standing, Active Lying Down, Inactive Sitting, Inactive Standing, and Inactive Lying Down) using an ICM-20948 IMU sensor. Motion data is processed through filtering, normalization, and segmentation using the sliding window method as input for a 1D Convolutional Neural Network (CNN-1D) model, with classification results validated using a camera as ground truth. Testing showed that the CNN-1D model using quaternion data achieved an accuracy of 99.86%, higher than the model without quaternions, which achieved 96.03%. This proves that quaternion orientation information enhances the model's ability to recognize welder movement patterns. The classification report results show that precision, recall, and F1-score averaged 1.00 for the Active Lying Down, Inactive Lying Down, Active Standing, and Inactive Sitting classes, while the "Inactive Standing" class achieved a recall of 0.99 and the "Active Sitting" class achieved a precision of 0.99 out of a total of 724 test data points, indicating that the model can accurately recognize all six activity classes without bias. The system is capable of real-time classification with an average inference time of 110.58 ms per window, and displays monitoring results and activity duration via a web-based dashboard. The IMU- and CNN-1D-based wearable system has proven to be a viable supporting technology for objectively and continuously monitoring welders' work activities.

Keywords: CNN-1D, HAR, IMU, Quaternion, Wearable Embedded System.

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
PERNYATAAN BEBAS PLAGIAT	v
KATA PENGANTAR.....	vii
ABSTRAK	xi
ABSTRACT	xiii
DAFTAR ISI.....	xv
DAFTAR TABEL	xix
DAFTAR GAMBAR.....	xxi
DAFTAR NOTASI.....	xxv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 Kajian Penelitian Terdahulu	7
2.2 Kajian Pustaka	9
2.2.1 Human Activity <i>Recognition</i> (HAR)	9
2.2.2 <i>Multi-Sensor Inertial Measurement Unit</i> (IMU)	10
2.2.3 Orientasi Gerak <i>Quaternion</i> pada sensor IMU	11
2.2.4 <i>Metode Convolutional Neural Network (CNN)</i>	15
2.3 <i>Hardware dan Software</i>	21
2.3.1 Sensor IMU ICM-20948	21
2.3.2 Baterai Lipo 3.7V 4000mAh	22
2.3.3 Baterai Lipo 3.7V 3.400mAh	22
2.3.4 <i>Step Down XY-3606 DC-DC Buck converter</i>	23
2.3.5 Baterai Level Indikator Modul	24
2.3.6 Jaket Kerja <i>Welder</i>	24

2.3.7 SD Card.....	25
2.3.8 ESP32.....	25
2.3.9 Raspberry Pi.....	26
2.3.10 Modul Kamera USB.....	27
2.3.11 Arduino IDE.....	28
2.3.12 Visual Studio Code.....	29
2.3.13 Python.....	30
2.3.14 MongoDB.....	30
2.3.15 MQTT (<i>Message Queuing Telemetry Transport</i>).....	31
BAB 3 METODE PENELITIAN	33
3.1 Konsep Penelitian	33
3.2 Tahapan Penelitian.....	33
3.2.1 Diagram Blok Sistem	34
3.2.2 Diagram Aliran Data Proses.....	35
3.3 Desain dan Perancangan Sistem	37
3.3.1 <i>Hardware Electric</i>	37
3.3.2 <i>Hardware Mekanik</i>	39
3.3.3 Ketahanan Panas Komponen dengan Lingkungan Las	43
3.3.4 Perancangan Desain UI/UX <i>Dashboard</i>	44
3.3.5 Perancangan Metode <i>Convolutional Neural Network</i>	44
3.4 Pengambilan Data	47
3.4.1 Objek dan Perangkat Pengambilan Data	47
3.4.2 Jenis Data yang Diambil	47
3.4.3 Durasi Pengambilan Data.....	48
3.4.4 Periode Pengambilan Data	48
3.5 Pengolahan Data IMU	48
3.6 Pengolahan Data Kamera.....	50
3.7 Skenario Pengujian Sistem	51
BAB 4 HASIL DAN PEMBAHASAN	55
4.1 Pengujian Sensor IMU.....	55
4.1.1 Kalibrasi <i>Bias</i> Sensor IMU ICM-20948.....	59
4.1.2 Hasil Pengujian Sensor <i>Accelerometer</i>	63
4.1.3 Hasil Pengujian Sensor <i>Gyroscope</i>	64

4.1.4 Hasil Pengujian Sensor <i>Magnetometer</i>	66
4.1.5 Pengujian Nilai Orientasi <i>Quaternion</i>	69
4.2 Pengujian SD Card	72
4.3 Pengujian Kamera	74
4.4 Pengujian Baterai	75
4.5 Berat Keseluruhan Komponen	80
4.6 Pengujian Komunikasi MQTT	82
4.7 Pengujian <i>Dashboard</i>	83
4.7.1 Pengujian sistem <i>login dashboard</i>	84
4.7.2 Pengujian Gagal <i>Login</i> dan <i>User</i> Berhasil Dibuat	85
4.7.3 Hasil Pengujian Tampilan <i>Homepage Supervisor</i>	85
4.7.4 Hasil Pengujian <i>Database</i> Mongo DB	87
4.8 Pengujian <i>Latency</i> Sistem	88
4.8.1 Pengujian <i>Latency</i> Inferensi CNN	88
4.8.2 Pengujian <i>Latency</i> Penyimpanan Basis Data	89
4.8.3 Pengujian <i>Latency</i> Penyajian <i>Dashboard</i>	89
4.9 Penerapan Metode CNN	90
4.9.1 Pengambilan dan Pengumpulan <i>Dataset</i>	90
4.9.2 Pelabelan <i>Dataset</i>	92
4.9.3 <i>Pre-Processing</i> Data	93
4.9.4 <i>Sliding Window</i>	99
4.9.5 <i>Split</i> Data <i>Training</i> dan Data <i>Test</i>	100
4.10 Penerapan Metode CNN Dengan <i>Quaternion</i>	101
4.11 Penerapan Metode CNN Tanpa <i>Quaternion</i>	107
4.12 Perbandingan CNN Dengan <i>Quaternion</i> dan Tanpa <i>Quaternion</i>	113
4.13 Hasil Pengujian <i>Real-time</i>	114
BAB 5 KESIMPULAN DAN SARAN	117
5.1 Kesimpulan	117
5.2 Saran	117
DAFTAR PUSTAKA	119
LAMPIRAN	123
BIODATA PENULIS	139

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2.1 Spesifikasi ESP32	26
Tabel 2.2 Spesifikasi Raspberry Pi 4 Model B	27
Tabel 2.3 Spesifikasi Kamera Mini USB	28
Tabel 3.1 Ketahanan Panas Komponen.....	43
Tabel 3.2 Architecture CNN.....	45
Tabel 3.3 Skenario Pengujian Sistem.....	51
Tabel 4.1 Spesifikasi Sensor IMU ICM-20948.....	55
Tabel 4.2 Konektivitas ICM-20948.....	57
Tabel 4.3 Hasil Offset Kalibrasi Accelerometer.....	60
Tabel 4.4 Hasil Offset Kalibrasi Gyroscope.....	61
Tabel 4.5 Hasil Offset Kalibrasi Magnetometer.....	62
Tabel 4.6 Uji Sensor Accelerometer.....	64
Tabel 4.7 Uji Sensor Gyroscope.....	66
Tabel 4.8 Uji Sensor Magnetometer.....	68
Tabel 4.9 Hasil Pengujian Quaternion	71
Tabel 4.10 Konektivitas SD Card.....	73
Tabel 4.11 Hasil Pengujian Waktu Operasi Baterai ESP32	77
Tabel 4.12 Hasil Pengujian Waktu Operasi Baterai Raspberry Pi	79
Tabel 4.13 Berat Keseluruhan Komponen	80
Tabel 4.14 Hasil Pengujian Latency Inferensi CNN	88
Tabel 4.15 Hasil Pengujian Latency Penyimpanan MongoDB.....	89
Tabel 4.16 Hasil Pengujian Latency PHP ke Browser	90
Tabel 4.17 Durasi dan Jumlah Dataset yang digunakan	91
Tabel 4.18 Contoh Raw Data dalam format CSV	91
Tabel 4.19 Contoh Labeling Kelas Berdiri Aktif.....	93
Tabel 4.20 Hasil Pembagian Data	101
Tabel 4.21 Hasil Model Sequential	102
Tabel 4.22 Classification Report	104
Tabel 4.23 Hasil Evaluasi.....	105

Tabel 4.24 Hasil Model Sequential Tanpa Quaternion	107
Tabel 4.25 Classification Report Tanpa Quaternion	109
Tabel 4.26 Hasil Evaluasi	110
Tabel 4.27 Hasil Pengujian Real-time	114

DAFTAR GAMBAR

Gambar 1.1 Dokumentasi di PT Adiluhung Saranasegara Indonesia	2
Gambar 2.1 Aktivitas Welder a.) Berdiri, b.) Berbaring, c.) Duduk	10
Gambar 2.2 Arah Deteksi.....	11
Gambar 2.3 Gimbal Lock.....	12
Gambar 2.4 Architecture of CNN	15
Gambar 2.5 Operasi Konvolusi Pada Convolutional Layer.....	17
Gambar 2.6 <i>Max</i> Pooling dan Average Pooling	17
Gambar 2.7 Full Connected Layer	18
Gambar 2.8 <i>Dropout</i> Layer	19
Gambar 2.9 <i>Softmax</i> Layer.....	20
Gambar 2.10 Sensor ICM-20948 Module.....	21
Gambar 2.11 Baterai Lipo 3.7V 4000 mAh	22
Gambar 2.12. Baterai Lipo 3.7V 1200 mAh.....	22
Gambar 2.13 Modul Charger LX LCBST.....	23
Gambar 2.14 XY-3606 Buck converter Step Down.....	24
Gambar 2.15 Baterai Level Indikator Modul	24
Gambar 2.16 Jaket Kerja Welder	25
Gambar 2.17 SD Card	25
Gambar 2.18 Chip ESP32	26
Gambar 2.19 Raspberry Pi	27
Gambar 2.20 Modul kamera USB Mini.....	28
Gambar 2.21 Arduino IDE.....	29
Gambar 2.22 Tampilan Software Visual Studio Code	29
Gambar 2.23 Logo Python	30
Gambar 2.24 Logo MongoDB	30
Gambar 2.25 Topologi Protokol MQTT	31
Gambar 3.1 Diagram Penelitian.....	34
Gambar 3.2 Diagram Blok Sistem	35
Gambar 3.3 Diagram Aliran Data Proses	36

Gambar 3.4 Perancangan Hardware electric	37
Gambar 3.5 a) Hasil Routing PCB dan b) Rangkaian Skematik PCB	38
Gambar 3.6 a) Desain PCB Tampak Atas b) Desain PCB Tampak Bawah.....	39
Gambar 3.7 a) Hasil Perakitan PCB Tampak Atas	39
Gambar 3.8 Design Hardware	40
Gambar 3.9 a) Hasil perancangan jaket welder tampak depan.....	41
Gambar 3.10 Desain 3D Casing PCB.....	41
Gambar 3.11 Hasil 3D Casing PCB	42
Gambar 3.12 Desain 3D Casing Baterai Raspberry Pi.....	42
Gambar 3.13 Hasil Desain 3D Casing Baterai Raspberry Pi	43
Gambar 3.14 Tampilan Welder Activity Dashboard.....	44
Gambar 3.15 Architecture CNN	45
Gambar 3.16 Ilustrasi Pengolahan Data IMU	49
Gambar 3.17 Ilustrasi Pengolahan Data Camera.....	50
Gambar 4.1 Testing sensor a) tiga sensor b) accelerometer dan gyroscope c) magnetrometer.....	56
Gambar 4.2 Rangkaian Wiring ICM-20948	57
Gambar 4.3 Kalibrasi Accelerometer	59
Gambar 4.4 Kalibrasi Gyroscope	60
Gambar 4.5 Kalibrasi Magnetometer	61
Gambar 4.6 Pengambilan Data Parsial Acc.....	63
Gambar 4.7 Pengambilan Data Parsial Gyro.....	65
Gambar 4.8 Pengambilan Data Parsial Magne.....	67
Gambar 4.9 Wiring SD Card	72
Gambar 4.10 Hasil Pengujian SD Card	73
Gambar 4.11 a) Pengambilan Tangkapan Cahaya Las b) Tampilan Ketika Cahaya Las terdeteksi c) Tampilan Ketika Cahaya Las Tidak Terdeteksi.....	74
Gambar 4.12 a) Raspberry Pi yang Tersambung dengan Kamera.....	75
Gambar 4.13 ESP32 Publish data ke MQTT.....	82
Gambar 4.14 Laptop Subscribe data dari ESP32	82
Gambar 4.15 Raspberry Pi Publish data ke MQTT.....	83
Gambar 4.16 Laptop Subscribe data dari Raspberry Pi.....	83

Gambar 4.17 Pengujian Halaman <i>Login</i>	84
Gambar 4.18 Tampilan Gagal Login dan b)user berhasil dibuat	85
Gambar 4.19 Tampilan Dashboard Setelah Login	86
Gambar 4.20 Tampilan Dashboard Setelah Login	86
Gambar 4.21 Tampilan Dashboard Setelah Login	86
Gambar 4.22 Database Menyimpan Status Device.....	87
Gambar 4.23 Database Menyimpan Hasil Prediction.....	87
Gambar 4.24 Database Menyimpan Hasil Active_session	88
Gambar 4.25 Hasil Filtering Accelerometer	95
Gambar 4.26 Filtering Gyroscope.....	96
Gambar 4.27 Filtering Magnetometer.....	97
Gambar 4.28 Confusion matrix.....	103
Gambar 4.29 Hasil Accuracy CNN	106
Gambar 4.30 Hasil Loss CNN	106
Gambar 4.31 Confusion matrix Tanpa Quaternion	108
Gambar 4.32 Hasil Accuracy CNN Tanpa Quaternion	111
Gambar 4.33 Hasil Loss CNN Tanpa Quaternion.....	112
Gambar 4.34 Hasil Perbandingan performa Model	113

(Halaman ini sengaja dikosongkan)

DAFTAR NOTASI

$x(t)$: sinyal masukan sensor

$w(k)$: bobot kernel konvolusi

K : ukuran kernel

b : *bias*

$y(t)$: keluaran hasil konvolusi

x_i : fitur masukan

w_i : bobot

b : *bias*

C : jumlah kelas aktivitas

z_j : keluaran neuron ke- j

α : learning rate

$\nabla L(w_t)$: gradien fungsi *loss* terhadap bobot

T : panjang *window* waktu

N : jumlah kanal sensor

f_s : *sampling* rate (Hz)

t_w : durasi waktu *window* (detik)

$\hat{q}t$: *quaternion* orientasi pada waktu t

$\omega_x, \omega_y, \omega_z$: kecepatan sudut *gyroscope* (rad/s)

β : gain filter untuk mengontrol kecepatan konvergensi

∇f : gradien fungsi objektif

Δt : interval waktu *sampling*

$|H(j\omega)|$: respons magnitude filter

ω : frekuensi sinyal *input* (rad/s)

ω_c : frekuensi *cutoff* (rad/s)

n : orde filter

f_c : frekuensi *cutoff*

f_s : frekuensi *sampling*

z : nilai setelah normalisasi

x : nilai asli

μ : nilai rata – rata (mean) *dataset*

σ : standar deviasi *dataset*

w_i : *window* ke- i

L : panjang *window* (jumlah sampel per segmen)

s : langkah geser

t : indeks waktu

TP : *True positive*

TN : *True Negative*

FP : *False Positive*

FN : *False Negative*

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengelasan merupakan aktivitas kerja yang bersifat intensif secara fisik dan banyak dijumpai pada berbagai sektor industri, seperti fabrikasi logam, manufaktur, perkapalan, konstruksi, maupun bengkel pengelasan. Dalam pelaksanaannya, seorang *welder* melibatkan gerakan lengan dan tangan yang dilakukan secara berulang selama proses kerja. Selain itu, pekerjaan *welder* juga mencakup kondisi ketika tidak terjadi proses pengelasan, seperti saat menunggu, berpindah posisi, atau melakukan persiapan kerja, yang ditandai dengan pola gerakan tubuh yang berbeda. Variasi tingkat keterlibatan gerakan tersebut membentuk perbedaan karakteristik aktivitas kerja yang dapat diamati melalui pola gerakan tubuh. Oleh karena itu, pemahaman terhadap aktivitas *welder* berdasarkan intensitas dan pola gerakan menjadi aspek penting sebagai dasar pengelompokan aktivitas kerja dalam lingkungan industri pengelasan (Baek & Nam, 2021; Pribadi & Shinoda, 2022).

Dalam praktik operasional di lingkungan kerja pengelasan, pemantauan dan identifikasi aktivitas *welder* masih banyak dilakukan melalui observasi langsung oleh *supervisor* atau berdasarkan laporan pekerjaan secara manual. Metode tersebut memiliki keterbatasan karena proses pemantauan sangat bergantung pada keberadaan *supervisor* dan sulit dilakukan secara kontinu pada area kerja yang luas. Akibatnya, perbedaan aktivitas kerja, baik ketika *welder* sedang melakukan pengelasan maupun ketika tidak melakukan pengelasan, sering kali tidak teridentifikasi secara sistematis. Selain itu, informasi mengenai durasi waktu pada setiap aktivitas *welder* yang teridentifikasi selama proses kerja berlangsung. Oleh karena itu, diperlukan sistem pemantauan aktivitas yang mampu mengidentifikasi jenis aktivitas sekaligus merekam durasi aktivitas *welder* secara objektif dan kontinu tanpa bergantung pada pengamatan manual (Zhou et al., 2024).

Permasalahan tersebut juga dijumpai pada lingkungan kerja pengelasan di PT Adiluhung Saranasegara Indonesia yang digunakan sebagai studi kasus dalam penelitian ini. Berdasarkan hasil observasi lapangan, aktivitas *welder* dilakukan pada area kerja terbuka dengan variasi posisi kerja yang beragam sehingga proses

pemantauan aktivitas masih dilakukan secara langsung oleh *supervisor*. Kondisi tersebut menunjukkan perlunya suatu sistem *monitoring* aktivitas *welder* yang bersifat *wearable*, fleksibel, dan mampu bekerja secara kontinu tanpa bergantung pada pengamatan manual. Temuan pada studi kasus ini digunakan sebagai dasar dalam perancangan sistem pada penelitian, namun tidak dilakukan implementasi langsung pada proses produksi perusahaan.



Gambar 1.1 Dokumentasi di PT Adiluhung Saranasegara Indonesia
(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan Gambar 1.1, terlihat bahwa proses fabrikasi kapal berlangsung pada area kerja terbuka dengan variasi posisi kerja *welder*. Kondisi tersebut menunjukkan perlunya sistem *monitoring* yang bersifat *wearable*, fleksibel, dan mampu melakukan pemantauan aktivitas secara *real-time*. Lingkungan kerja di PT Adiluhung Saranasegara Indonesia digunakan sebagai studi kasus dalam penelitian ini sebagai acuan perancangan sistem, tanpa dilakukan implementasi langsung pada proses produksi perusahaan. Untuk mewujudkan sistem *monitoring* tersebut, diperlukan metode yang mampu mengenali dan mengklasifikasikan aktivitas *welder* secara otomatis berdasarkan pola gerakan tubuh, Salah satu pendekatan yang banyak digunakan adalah *Human Activity Recognition* (HAR).

Human Activity Recognition (HAR) merupakan pendekatan untuk mengenali dan mengklasifikasi aktivitas manusia berdasarkan pola gerakan tubuh yang direpresentasikan melalui data sensor. Dalam pekerjaan *welder* di industri pengelasan, HAR menjadi relevan karena aktivitas pengelasan melibatkan rangkaian gerakan lengan dan tangan yang berulang serta bervariasi sesuai dengan kondisi kerja. Selain itu, pengelasan memiliki karakteristik pola gerakan yang

berbeda. Dengan menerapkan HAR, aktivitas *welder* dapat dikelompokkan secara sistematis berdasarkan intensitas dan dinamika gerakan, sehingga memungkinkan identifikasi perbedaan antara kondisi mengelas dan tidak mengelas secara objektif dan kontinu (Tseng & Wen, 2023).

Untuk memperoleh representasi gerakan tubuh *welder* secara langsung, diperlukan sensor yang mampu menangkap perubahan posisi. *Inertial Measurement Unit* (IMU) merupakan sensor berbasis *Microelectromechanical system* (MEMS) yang banyak digunakan pada sistem *wearable* karena memiliki ukuran kecil, konsumsi daya rendah, serta mampu mengukur percepatan linear menggunakan *accelerometer*, kecepatan sudut *gyroscope*, serta orientasi menggunakan *magnetometer* pada tidak sumbu koordinat. Data keluaran dari sensor IMU selanjutnya dapat direpresentasikan dalam bentuk *quaternion* untuk menggambarkan orientasi dan rotasi anggota tubuh secara lebih stabil serta mengurangi permasalahan *gimbal lock*. Pada aktivitas pengelasan, representasi data IMU dalam domain *quaternion* dapat digunakan untuk menggambarkan dinamika gerakan lengan *welder* selama melakukan proses pengelasan maupun aktivitas lainnya. (Fan et al., 2025; Farahan et al., 2022).

Seiring dengan pemanfaatan IMU, diperlukan metode pengolahan data yang mampu mengenali pola gerakan secara efektif. Pendekatan konvensional dalam HAR umumnya mengandalkan ekstraksi fitur manual, pendekatan tersebut sering kali kurang optimal dalam menangani variasi pola gerakan yang kompleks dan dinamis, seperti pada aktivitas kerja *welder*. Perkembangan metode *deep learning* menawarkan pendekatan yang lebih adaptif, salah satunya melalui penggunaan *Convolutional Neural Network* (CNN). CNN mampu mengekstraksi fitur secara otomatis dari data sensor berbasis waktu serta menangkap pola lokal spasial dan temporal, sehingga sesuai untuk mengklasifikasi aktivitas *welder* yang memiliki karakteristik gerakan kompleks dan bervariasi antar individu, dengan hasil klasifikasi yang selanjutnya dapat divalidasi menggunakan data visual sebagai ground truth. (Alarfaj et al., 2024).

Berdasarkan permasalahan dan kesenjangan penelitian tersebut, penelitian ini bertujuan untuk merancang sistem *monitoring* aktivitas *welder* berbasis *embedded system* dengan memanfaatkan sensor IMU yang terintegrasi pada jaket

welder, serta menerapkan metode *Convolutional Neural Network* (CNN) untuk pengenalan dan pemantauan aktivitas *welder*. Sistem yang dikembangkan mampu memantau aktivitas *welder* secara *real-time* serta mencatat durasi waktu pada setiap aktivitas *welder* yang teridentifikasi selama proses kerja berlangsung. Penelitian ini mencakup perancangan sistem, implementasi algoritma CNN berbasis data IMU, serta evaluasi performa menggunakan data visual sebagai *ground truth*. Hasil penelitian diharapkan dapat memberikan informasi objektif mengenai jenis dan durasi aktivitas *welder* sebagai pendukung proses *monitoring* aktivitas kerja di industri pengelasan.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian tugas akhir ini sebagai berikut:

1. Bagaimana merancang *embedded system* dengan sensor IMU yang terintegrasi pada jaket *welder* untuk memperoleh data pola gerakan tubuh *welder*?
2. Bagaimana kemampuan sistem dalam mengklasifikasi aktivitas *welder* menggunakan metode CNN-1D?
3. Bagaimana tingkat kesesuaian antara status aktif atau tidak aktif hasil klasifikasi sistem dengan kondisi aktual proses pengelasan yang direkam melalui deteksi cahaya las sebagai *ground truth*?

1.3 Tujuan Penelitian

Berdasarkan pertanyaan penelitian, terdapat tujuan penelitian sebagai berikut:

1. Merancang dan mengembangkan *embedded system* berbasis sensor IMU yang terintegrasi pada jaket *welder* untuk memperoleh data pola gerakan tubuh *welder* selama aktivitas kerja.
2. Menerapkan metode CNN-1D untuk melakukan klasifikasi aktivitas *welder* berdasarkan data sensor IMU serta mengetahui kemampuan sistem dalam mengenali aktivitas tersebut.
3. Melakukan validasi hasil klasifikasi sistem dalam membedakan aktif mengelas dan tidak mengelas berdasarkan perbandingan data prediksi sistem dengan kondisi aktual menggunakan data visual sebagai *ground truth*.

1.4 Batasan Masalah

Agar permasalahan yang diangkat di dalam kajian penelitian ini lebih fokus dan dapat menyelesaikan permasalahan. Adapun Batasan pada proses perancangan, pengerjaan, dan pengujian sistem ini sehingga diberikan batasan penelitian antara lain yaitu sebagai berikut:

1. Aktivitas *welder* yang diklasifikasikan dibatasi pada enam kelas aktivitas, yaitu duduk aktif, berdiri aktif, berbaring aktif, duduk tidak aktif, berdiri tidak aktif, berbaring tidak aktif.
2. Data yang digunakan dalam penelitian ini merupakan data gerakan lengan atas *welder* yang direkam oleh sensor IMU yang tertanam pada jaket *welder*, dengan pemasangan sensor dibatasi pada bagian tangan kanan lengan atas.
3. Pengambilan data dilakukan secara langsung dari *welder* dengan jumlah partisipan dibatasi sebanyak empat orang, dengan kriteria menggunakan tangan kanan tidak kidal.
4. Kamera digunakan hanya sebagai alat validasi visual aktivitas pengelasan dan tidak digunakan sebagai *input* utama sistem klasifikasi.
5. Konektivitas jaringan pada Raspberry Pi dibatasi menggunakan kabel LAN untuk menjaga stabilitas koneksi selama proses pengambilan dan pengiriman data.

1.5 Manfaat Penelitian

Penelitian Tugas Akhir ini diharapkan dapat memberikan manfaat:

1. Meningkatkan efektivitas pemantauan aktivitas *welder* dengan menyediakan sistem yang mampu mengidentifikasi aktivitas mengelas dan tidak mengelas secara kontinu tanpa bergantung sepenuhnya pada observasi manual.
2. Memberikan kontribusi dalam pengembangan sistem *Human Activity Recognition* (HAR) berbasis sensor IMU dengan penerapan metode CNN-1D untuk klasifikasi aktivitas kerja *welder*.
3. Sebagai referensi ilmiah untuk dijadikan pengembangan pada tugas akhir selanjutnya.

(Halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Jaramillo et al., (2022) menunjukkan bahwa sistem *Human Activity Recognition* (HAR) menggunakan sensor *Inertial Measurement Unit* (IMU) dan *Convolutional Neural Network* (CNN) pada perangkat *embedded* untuk klasifikasi aktivitas manusia secara *real-time*. Sistem mencapai akurasi 90,4% dalam pengujian langsung. CNN unggul dalam efisiensi komputasi dan deteksi pola lokal, ideal untuk perangkat dengan sumber daya terbatas. Penggunaan beberapa IMU dengan sensitivitas berbeda meningkatkan performa. Namun, kompleksitas model tinggi, sehingga generalisasi terhadap data baru terbatas. Aktivitas yang dikenali masih umum, belum spesifik untuk kerja industri. Penelitian selanjutnya diperlukan untuk menerapkan pada aktivitas seperti pengelasan berbasis *embedded*.

Pribadi & Shinoda, (2022) mengembangkan sistem analisis gerakan tangan *welder* untuk membedakan *welder* berkualifikasi (Q) dan tidak berkualifikasi (UQ) menggunakan sensor IMU 9-DOF. Sistem memantau aktivitas *real-time*, merekam pola gerakan selama pengelasan. Data IMU membangun basis pola Q dan UQ, diklasifikasikan dengan SVM. Hasil efektif membedakan karakteristik gerakan, sebagai alternatif evaluasi performa. Sistem *cost-effective*, berpotensi di industri konstruksi baja dan galangan kapal. *Monitoring* web memungkinkan pengawas memantau jarak jauh, relevan saat pandemi. Namun, fokus SVM konvensional, belum *deep learning*. penelitian ini menjadi dasar penting bagi pengembangan sistem pengenalan aktivitas *welder* menggunakan sensor IMU 9-DOF dan membuka peluang pengembangan dengan CNN untuk pengenalan aktivitas *welder* secara *real-time*.

Penelitian yang dilakukan oleh Pribadi & Shinoda, (2025) adalah penelitian lanjutan dari Pribadi & Shinoda, (2022) membahas tentang pengembangan sistem pengenalan aktivitas dan keterampilan tangan *welder* pada proses *Flux-Cored Arc Welding* (FCAW) dengan metode CNN–LSTM. Sistem memanfaatkan CNN untuk fitur spasial dan LSTM untuk ketergantungan temporal data sensor. Hasil pengujian

true positive rate 97,69% dalam kondisi terkontrol. Dibanding SVM, CNN–LSTM lebih unggul secara keseluruhan, meski SVM lebih cepat di beberapa kasus. Namun, akurasi turun hingga 60,10% di lingkungan berbeda, menunjukkan sensitivitas dan *overfitting*. Diperlukan peningkatan *robustness* via optimasi, fitur tambahan, dan augmentasi data. Penelitian ini relevan sebagai dasar HAR berbasis CNN yang ringan untuk *embedded* sistem, mendeteksi aktivitas *welder*.

Virmansyah et al., (2024) melakukan penelitian mengenai identifikasi pola gerakan tangan *welder* menggunakan perangkat *wearable* berbasis sensor *Inertial Measurement Unit* (IMU) yang dipasang pada pergelangan tangan. Data yang dikumpulkan meliputi percepatan, kecepatan sudut, dan medan magnet yang merepresentasikan karakteristik gerakan tangan selama proses pengelasan FCAW. Penelitian tersebut menerapkan metode *deep learning Convolutional Neural Network–Long Short Term Memory* (CNN-LSTM) untuk mengklasifikasikan identitas *welder* dan posisi pengelasan. Hasil penelitian menunjukkan bahwa model CNN-LSTM mampu mencapai akurasi sebesar 99,73% dalam mengidentifikasi *welder* secara individual dan 97,07% dalam mengklasifikasikan posisi pengelasan. Temuan ini membuktikan bahwa pola gerakan tangan *welder* memiliki karakteristik yang khas dan dapat dikenali secara efektif menggunakan data sensor IMU dan pendekatan *deep learning*. Oleh karena itu, penelitian ini menjadi landasan kuat bagi pengembangan *embedded system* untuk klasifikasi aktivitas *welder*, khususnya yang memanfaatkan data IMU.

Kim & Lee, (2023) menunjukkan sebuah algoritma *data valuation* berbasis *meta-reinforcement learning* untuk meningkatkan performa *Human Activity Recognition* (HAR) berbasis sensor *Inertial Measurement Unit* (IMU). Penelitian tersebut menitikberatkan pada permasalahan kualitas data latih, khususnya pada data IMU berbentuk *multivariate time-series* yang rentan terhadap *noise* dan kesalahan pelabelan. Algoritma yang diusulkan mampu menilai kontribusi setiap sampel data latih dengan menerima data mentah (*raw-level input*) melalui integrasi struktur ekstraksi fitur berbasis 1D *Convolutional Neural Network* (1D-CNN) dan *transformer encoder*. Evaluasi dilakukan pada beberapa *dataset* IMU-based HAR, yaitu BBS, UCI-HAR, WISDM, dan PAMAP2, dengan hasil menunjukkan bahwa lebih dari 96% data rusak dapat teridentifikasi serta terjadi peningkatan akurasi

klasifikasi setelah data bernilai rendah dihilangkan. Penelitian ini menunjukkan bahwa seleksi data latih berdasarkan nilai kontribusi dapat meningkatkan performa sistem HAR secara signifikan. Namun, fokus penelitian tersebut masih terbatas pada pengenalan aktivitas umum, belum mengkaji penerapan algoritma serupa pada aktivitas *welder* di galangan kapal berbasis sensor IMU.

(Martínez-zarzuela et al., 2023) memperkenalkan VIDIMU, yaitu sebuah *dataset* multimodal yang mengombinasikan data kinematik dari sensor *Inertial Measurement Unit* (IMU) dengan rekaman video untuk aktivitas manusia dalam kehidupan sehari-hari. Hasil penelitian menunjukkan bahwa sinkronisasi data IMU dan video mampu meningkatkan keandalan pelabelan aktivitas serta mengurangi subjektivitas dalam proses anotasi data. Meskipun demikian, penelitian ini masih berfokus pada aktivitas kehidupan sehari-hari dan belum diarahkan pada aktivitas *welder* serta belum mengembangkan sistem *monitoring* yang menyajikan informasi durasi aktivitas kerja. Oleh karena itu, penelitian ini menjadi acuan penting dalam penggunaan kamera sebagai *ground truth* untuk validasi sistem HAR berbasis IMU, yang selanjutnya diadaptasi pada penelitian ini untuk pemantauan aktivitas *welder*.

2.2 Kajian Pustaka

Bab II berisikan sub-bab yang menjelaskan teori-teori yang digunakan dalam penelitian, meliputi objek penelitian, metode, serta perangkat lunak dan perangkat keras yang digunakan. Kajian pustaka berfungsi sebagai landasan konseptual dan prinsip dasar pengembangan penelitian melalui tinjauan literatur dan konsep utama yang mendukung perumusan hipotesis, analisis data, serta penyusunan kerangka penelitian.

2.2.1 Human Activity Recognition (HAR)

Human Activity Recognition merupakan suatu pendekatan dalam bidang kecerdasan buatan yang bertujuan untuk mengenali dan mengklasifikasikan aktivitas manusia berdasarkan pola gerakan yang dihasilkan dari aktivitas tersebut. Selain mampu mengenali aktivitas umum seperti aktivitas fisik sehari-hari berjalan, duduk, berdiri, tidur, serta berbagai aktivitas transisi lainnya, HAR juga dapat digunakan untuk mengenali aktivitas dan postur kerja spesifik (Hakim, 2021; Pribadi & Shinoda, 2022), Sebagaimana ditunjukkan pada Gambar 2.1, aktivitas

pengelasan melibatkan kombinasi gerakan lengan dan postur tubuh tertentu yang membentuk pola gerakan khas *welder*.

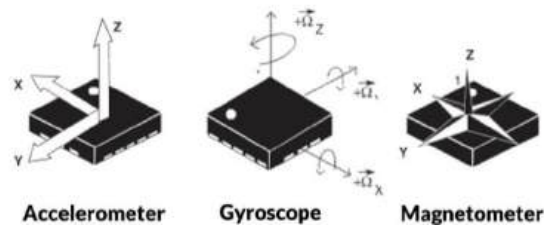


Gambar 2.1 Aktivitas *Welder* a.) Berdiri, b.) Berbaring, c.) Duduk
(Dokumentasi Pribadi Penulis, 2026)

Secara fungsional, HAR berperan dalam memantau dan mengidentifikasi dinamika aktivitas manusia secara sistematis. Informasi aktivitas yang diperoleh dapat digunakan untuk mengevaluasi kondisi fisik, mendeteksi perubahan perilaku, serta memberikan umpan balik yang berguna bagi pengguna maupun sistem yang mengelolanya (Arifien et al., 2022). Oleh karena itu, dalam penelitian ini HAR digunakan untuk mengklasifikasi aktivitas *welder* berdasarkan pola gerakan yang dihasilkan selama proses kerja berlangsung.

2.2.2 Multi-Sensor Inertial Measurement Unit (IMU)

Inertial Measurement Unit (IMU) merupakan perangkat sensor yang terdiri dari *accelerometer*, *gyroscope*, dan *magnetometer* yang digunakan untuk mengukur gerakan dan orientasi suatu objek dalam ruang tiga dimensi. *Accelerometer* berfungsi untuk mengukur percepatan linear, *gyroscope* mengukur kecepatan sudut, sedangkan *magnetometer* digunakan untuk menentukan arah orientasi terhadap medan magnet bumi. Kombinasi ketiga sensor tersebut memungkinkan sistem memperoleh informasi gerakan yang lebih lengkap dan akurat dibandingkan penggunaan satu jenis sensor secara terpisah. Penggunaan multi sensor IMU banyak diterapkan di dalam pengenalan aktivitas pengelasan dan menjadi komponen penting dalam pengembangan sistem pengenalan aktivitas manusia yang membutuhkan pemantauan gerakan secara presisi (Pribadi & Shinoda, 2022).



Gambar 2.2 Arah Deteksi

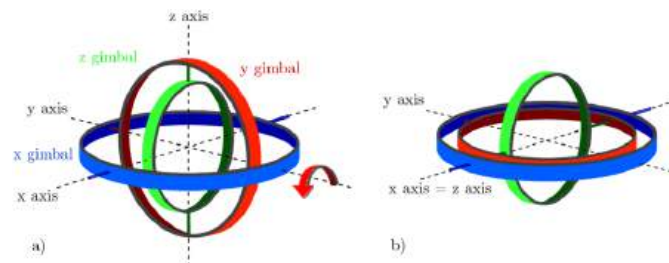
(Sumber : <https://www.globalperformanceinsights.com/>)

Pada gambar 2.2 menunjukkan arah deteksi *accelerometer*, *gyroscope* dan *magnetometer* dimana *accelerometer*, sensor mengukur percepatan linier lengan *welder* pada tiga sumbu, yaitu sumbu Ax yang merepresentasikan gerakan maju mundur mengikuti jalur pengelasan, sumbu Ay yang menggambarkan gerakan kanan kiri tangan terhadap area las, serta sumbu Az yang menunjukkan gerakan naik turun lengan saat mengatur jarak elektroda dengan permukaan material. Informasi ini digunakan untuk mengidentifikasi intensitas gerakan dan pola aktivitas *welder* yang bersifat repetitif. *Gyroscope* mengukur kecepatan perubahan sudut (*angular velocity*) gerakan lengan *welder* terhadap sumbu Gx (*roll*), Gy (*pitch*), dan Gz (*yaw*). Sumbu roll merepresentasikan rotasi pergelangan tangan saat memiringkan elektroda, pitch menunjukkan gerakan naik turun sudut tangan selama mengikuti kampuh las, sedangkan yaw menggambarkan perubahan arah gerakan tangan sepanjang jalur pengelasan. Informasi kecepatan sudut ini digunakan untuk menganalisis dinamika gerakan, stabilitas sudut pengelasan, serta perubahan postur kerja *welder*. Lalu *magnetometer*, sensor digunakan untuk menentukan orientasi absolut lengan *welder* terhadap medan magnet bumi. *Magnetometer* digunakan untuk mengukur arah medan magnet bumi pada sumbu Mx, My, dan Mz, yang berfungsi sebagai referensi arah global. Informasi ini dimanfaatkan dalam proses estimasi orientasi untuk membantu menentukan arah hadap (*yaw*) lengan *welder* serta mengoreksi *drift* pada data *gyroscope*, khususnya pada pemantauan jangka panjang. Dengan adanya referensi arah global, kestabilan estimasi orientasi dapat dipertahankan, sehingga sistem lebih mampu mengidentifikasi arah gerakan lengan *welder* saat berpindah posisi kerja.

2.2.3 Orientasi Gerak *Quaternion* pada sensor IMU

Sensor IMU tidak hanya digunakan untuk merekam tetapi juga untuk

merepresentasikan orientasi gerak suatu objek. Representasi orientasi umumnya dinyatakan dalam bentuk *Euler angle* yang terdiri dari sudut roll, pitch, dan yaw. Representasi ini bersifat intuitif dan mudah dipahami, sehingga banyak digunakan dalam sistem *embedded* dan aplikasi pemantauan gerakan. Namun, penggunaan *Euler angle* memiliki keterbatasan ketika diaplikasikan pada gerakan dengan rotasi yang kompleks dan multidimensional. Salah satu permasalahan utama yang dapat muncul adalah fenomena gimbal lock, yaitu kondisi ketika dua sumbu rotasi sejajar sehingga sistem kehilangan satu derajat kebebasan rotasi. Kondisi ini menyebabkan representasi orientasi menjadi tidak stabil dan kurang akurat.



Gambar 2.3 Gimbal Lock
(Sumber : <https://www.researchgate.net/>)

Pada gambar 2.3 tersebut mengilustrasikan prinsip terjadinya gimbal lock. Pada kondisi awal, ketiga sumbu rotasi berada dalam posisi saling tegak lurus. Namun, setelah terjadi rotasi sebesar 90° pada salah satu sumbu, dua sumbu lainnya dapat menjadi sejajar sehingga sistem kehilangan kemampuan untuk membedakan rotasi pada sumbu yang berbeda. Dalam konteks aktivitas *welder*, gerakan lengan sering melibatkan kombinasi rotasi dan perubahan orientasi yang kompleks, seperti saat pengelasan pada posisi miring atau di atas kepala. Kondisi ini berpotensi memicu terjadinya gimbal lock apabila orientasi direpresentasikan menggunakan *Euler angle*.

Sebagai alternatif, *quaternion* digunakan sebagai representasi orientasi berbasis empat parameter yang terdiri dari satu komponen skalar dan tiga komponen vektor. *Quaternion* mampu merepresentasikan rotasi tiga dimensi secara kontinu tanpa mengalami permasalahan gimbal lock. Selain itu, *quaternion* memiliki kestabilan numerik yang lebih baik dan efisien secara komputasi,

sehingga sesuai untuk implementasi pada sistem *embedded* berbasis sensor IMU. *Quaternion* dinyatakan sebagai persamaan 2.1:

$$q = [q_w q_x q_y q_z] \quad (2.1)$$

Keterangan:

q : Vektor *quaternion* yang merepresentasikan orientasi atau rotasi suatu objek dalam tuangan tiga dimensi

q_w : Komponen skalar dari *quaternion*

q_x : Komponen vector pada sumbu x

q_y : Komponen vector pada sumbu y

q_z : Komponen vector pada sumbu z

dengan q_w sebagai komponen skalar, sedangkan q_x, q_y, q_z merupakan komponen vektor yang menyatakan sumbu rotasi. *Quaternion* yang digunakan adalah unit *quaternion*, dengan persamaan 2.2:

$$|q| = \sqrt{q_w^2 + q_x^2 + q_y^2 + q_z^2} = 1 \quad (2.2)$$

Keterangan :

$|q|$: Norma atau magnitudo *quaternion*

q : Quaternion yang merepresentasikan orientasi

q_x : Komponen vector pada sumbu x

q_y : Komponen vector pada sumbu y

q_z : Komponen vector pada sumbu z

Dalam sistem IMU, *quaternion* umumnya diperoleh melalui proses sensor *fusion* yang menggabungkan data *accelerometer*, *gyroscope*, dan *magnetometer* digunakan untuk menghitung perubahan rotasi berdasarkan kecepatan sudut $\omega = (G_x, G_y, G_z)$. Perubahan orientasi awal dihitung melalui integrasi rotasi pada persamaan 2.3:

$$q_{gyro} = q_t - 1 + \frac{1}{2} q_t - 1 \otimes \omega \Delta t \quad (2.3)$$

Keterangan :

q_{gyro} : *Quaternion* hasil prediksi orientasi berdasarkan data giroskop pada waktu t

$q_t - 1$: *Quaternion* orientasi pada waktu sebelumnya (t - 1)

$\frac{1}{2}$: Faktor yang berasal dari persamaan diferensial kinematika *quaternion*

$\otimes$: Operator perkalian *quaternion*

ω : Vektor kecepatan sudut dari sensor giroskop

Δt : Selang waktu antara dua pengukuran berturut-turut

dengan Δt sebagai interval waktu *sampling* dan $\otimes$ menyatakan perkalian *quaternion*. Namun, integrasi *gyroscope* dapat menimbulkan *drift*. Oleh karena itu, arah gravitasi dari *accelerometer* dan referensi medan magnet dari *magnetometer* digunakan sebagai koreksi orientasi.

Pada penelitian ini, digunakan madgwick filter sebagai algoritma sensor *fusion*. Filter ini mengoreksi *error* orientasi dengan pendekatan *gradient descent*, sehingga pembaruan *quaternion* menjadi persamaan 2.4:

$$q_t = q_{gyro} - \beta \nabla f \quad (2.4)$$

Keterangan :

q_t : *Quaternion* hasil estimasi orientasi pada waktu t

q_{gyro} : *Quaternion* hasil prediksi orientasi berdasarkan integrasi data sensor giroskop

β : Parameter penguatan (gain) laju konvergensi algoritma yang mengatur besar koreksi terhadap estimasi orientasi

∇f : Vektor gradien dari fungsi objektif

di mana β adalah parameter koreksi, dan ∇f selisih antara arah gravitasi/magnet yang terukur dengan estimasi orientasi. Proses ini menghasilkan sinyal *quaternion time-series* yang merepresentasikan perubahan orientasi lengan atas secara kontinu dan stabil pada persamaan 2.5:

$$q(t) = [q_w(t), q_x(t), q_y(t), q_z(t)] \quad (2.5)$$

Keterangan :

$q(t)$: *Quaternion* yang merepresentasikan orientasi objek pada waktu t

$q_w(t)$: Komponen skalar (*scalar part*) *quaternion* pada waktu t

$q_x(t)$: Komponen vektor (*vector part*) *quaternion* pada sumbu x pada waktu t

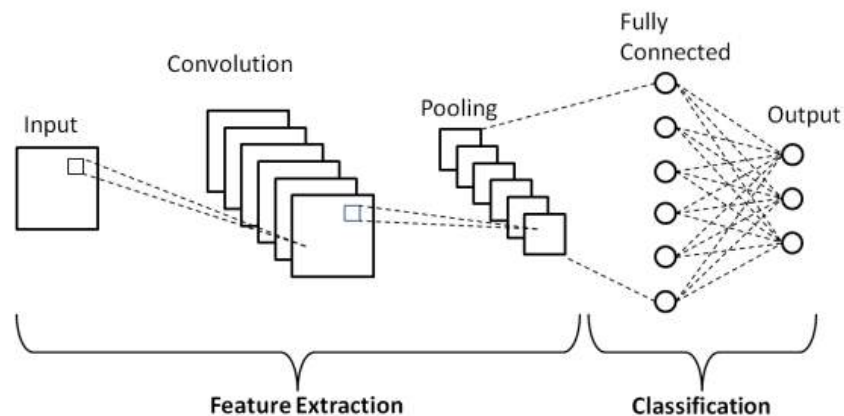
$q_y(t)$: Komponen vektor (*vector part*) *quaternion* pada sumbu y pada waktu t

$q_z(t)$: Komponen vektor (*vector part*) *quaternion* pada sumbu z pada waktu t

Oleh karena itu, penggunaan representasi orientasi berbasis *quaternion* dipilih untuk meminimalkan pengaruh gimbal lock pada data IMU. *Quaternion* digunakan sebagai hasil estimasi orientasi yang menjadi representasi utama gerakan lengan atas *welder* sebelum tahap segmentasi dan pelatihan model klasifikasi aktivitas.

2.2.4 Metode Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu metode *deep learning* yang banyak digunakan untuk pengenalan pola pada data berbasis grid, termasuk sinyal *time-series*. Pada penelitian ini digunakan *Convolutional Neural Network* satu dimensi (1D-CNN) untuk mengolah data sinyal *time-series* dari sensor *Inertial Measurement Unit* (IMU). Berbeda dengan CNN pada citra dua dimensi, 1D-CNN melakukan proses konvolusi sepanjang dimensi waktu sehingga mampu mengekstraksi pola perubahan sinyal yang berkaitan dengan gerakan lengan *welder*. CNN juga memiliki kemampuan mengekstraksi fitur secara otomatis tanpa memerlukan ekstraksi fitur manual (Himanshi Singh, 2022; MATLAB, 2024). Secara umum, arsitektur CNN terdiri atas *input layer*, *convolution layer*, *activation layer*, *pooling layer*, *fully connected layer*, dan *Output layer*. Pada penelitian ini, *convolution* layer berfungsi mengekstraksi fitur lokal dari sinyal IMU, seperti pola percepatan, rotasi, dan orientasi yang menjadi karakteristik setiap aktivitas *welder*. Arsitektur dasar CNN ditunjukkan pada Gambar 2.4.



Gambar 2.4 Architecture of CNN
(Sumber : <https://www.analyticsvidhya.com>)

1. Convolutional Layer

Convolutional layer merupakan komponen inti dari CNN yang berfungsi untuk mengekstraksi fitur dari data sinyal masukan. Lapisan ini menggunakan operasi matematika konvolusi untuk mendeteksi pola pada sinyal. Dalam pengoperasiannya, lapisan ini menggunakan filter (kernel) yang digeser sepanjang dimensi waktu pada sinyal masukan untuk menghasilkan feature map. Pada penelitian ini, *convolutional* layer berfungsi mengekstraksi fitur lokal dari sinyal IMU, seperti pola percepatan, rotasi, dan orientasi yang menjadi karakteristik setiap aktivitas *welder*.

Operasi konvolusi 1D dilakukan dengan menggeser kernel pada sinyal masukan untuk menghasilkan *feature map*. yang merepresentasikan respons kernel terhadap pola sinyal. Secara matematis, operasi konvolusi 1D dinyatakan sebagai persamaan 2.6:

$$y(t) = \sum_{k=0}^{K-1} x(t+k).w(k) + b \quad (2.6)$$

Keterangan:

$x(t)$: sinyal masukan (data sensor IMU)

$w(k)$: bobot kernel konvolusi

K : ukuran kernel

b : bias

$y(t)$: keluaran hasil konvolusi

Hasil konvolusi berupa *feature map* satu dimensi kemudian dilewatkan ke fungsi aktivasi untuk memperkenalkan sifat non-linear pada model. Fungsi aktivasi yang digunakan adalah *Rectified Linear Unit* (ReLU), yang didefinisikan sebagai persamaan 2.7:

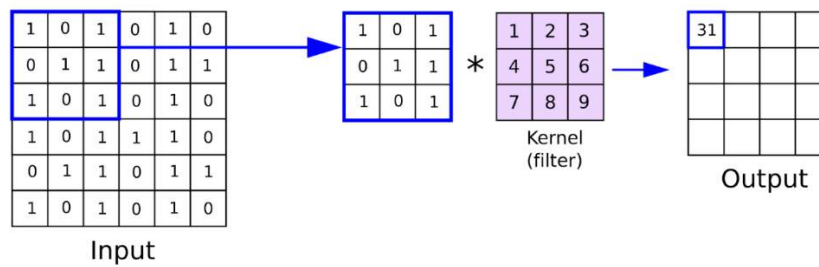
$$f(x) = \text{Max}(0, x) \quad (2.7)$$

$f(x)$: Fungsi aktivasi *Rectified Linear Unit* (ReLU)

x : Nilai masukan yang diterima oleh fungsi aktivasi

$\text{Max}(0, x)$: Operator yang memilih nilai terbesar antara 0 dan x

Fungsi ini mempertahankan nilai positif dan mengubah nilai negatif menjadi nol, sehingga membantu model mempelajari pola non-linear dari sinyal sensor. Operasi konvolusi ditunjukkan pada Gambar 2.5.



Gambar 2.5 Operasi Konvolusi Pada *Convolutional Layer*
(Sumber : <https://www.analyticsvidhya.com/>)

2. Pooling Layer

Setelah melalui lapisan konvolusi, *feature map* diteruskan ke lapisan *pooling*. *Pooling* layer digunakan untuk mereduksi dimensi *feature map* dan meningkatkan ketahanan model terhadap variasi kecil pada sinyal *input*, tanpa menghilangkan informasi penting. Proses ini digunakan untuk mengurangi kompleksitas komputasi dan risiko *overfitting*. Jenis *pooling* yang digunakan pada penelitian ini adalah *Max Pooling*, yang mengambil nilai maksimum dalam suatu jendela tertentu sebagaimana dinyatakan pada persamaan 2.8:

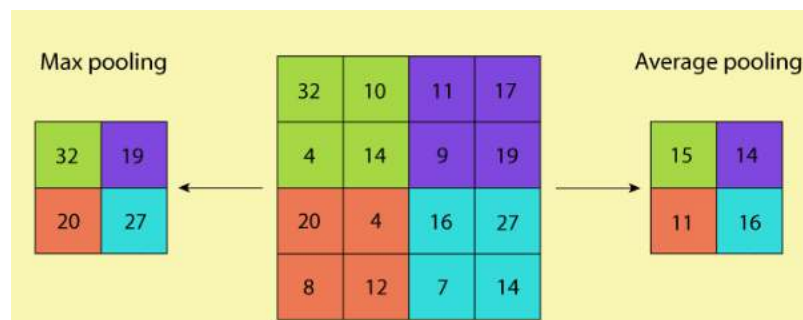
$$y = \text{Max}(x_1, x_2, \dots, x_n) \quad (2.8)$$

y : Nilai keluaran

$\text{Max}(\cdot)$: *Max Pooling*

$x_1, x_2, \dots, x_n$: Nilai masukan

Pada HAR berbasis IMU, *pooling* membantu CNN menjadi lebih robust terhadap variasi kecil pada sinyal sensor akibat perbedaan gaya gerak antar individu. *Max Pooling* dan *Average Pooling* ditunjukkan pada Gambar 2.6.



Gambar 2.6 *Max Pooling* dan *Average Pooling*
(Sumber : <https://developersbreach.com/>)

3. Fully Connected Layer

Feature map hasil *pooling* kemudian diteruskan ke *fully connected layer* untuk proses klasifikasi. Pada lapisan ini, seluruh fitur hasil ekstraksi digabungkan dan dipetakan ke ruang kelas aktivitas. *Full Connected Layer* ditunjukkan pada Gambar 2.7. Secara matematis, *Full Connected Layer* dapat dinyatakan sebagai persamaan 2.9:

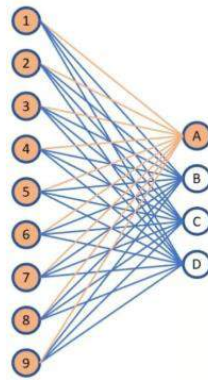
$$z = \sum_{i=1}^n w_i x_i + b \quad (2.9)$$

Keterangan:

x_i : fitur masukan

w_i : bobot

b : bias



Gambar 2.7 Full Connected Layer
(Sumber : <https://developersbreach.com/>)

4. Dropout Layer

Untuk mencegah terjadinya *overfitting* pada proses pelatihan model, setelah *fully connected* layer diterapkan *Dropout* layer dengan *Dropout rate* sebesar 0,3. *Dropout* bekerja dengan menonaktifkan sebagian neuron secara acak pada setiap iterasi pelatihan, sehingga model tidak bergantung secara berlebihan pada neuron tertentu dan lebih mampu melakukan generalisasi terhadap data baru (Srivastava et al., 2014). Secara matematis, *Dropout* dinyatakan sebagai persamaan 2.10:

$$\hat{h}_i = r_i \cdot h_i, r_i \sim \text{Bernoulli}(1 - p) \quad (2.10)$$

Keterangan:

h_i : keluaran neuron ke- i sebelum *Dropout*

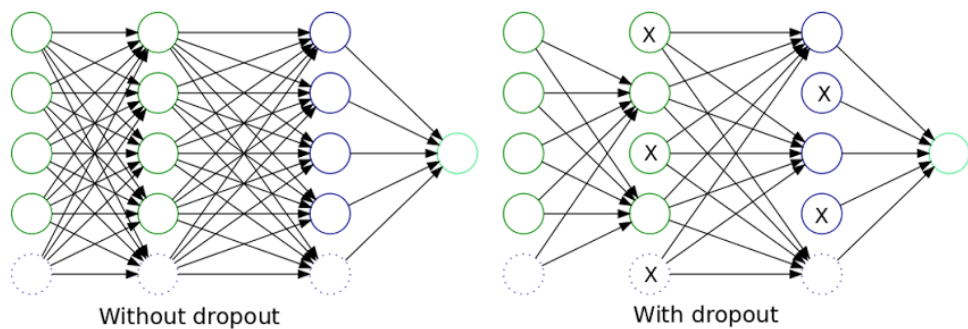
$\hat{h}_i$: keluaran neuron ke- i setelah *Dropout*

r_i : nilai mask acak (0 atau 1) yang mengikuti distribusi Bernoulli

p : *Dropout* rate (probabilitas neuron dinonaktifkan), pada penelitian ini $p = 0,3$

Neuron yang mendapat nilai mask $r_i = 0$ akan dinonaktifkan sementara pada iterasi pelatihan tersebut, sedangkan neuron dengan $r_i = 1$ tetap aktif dan diteruskan ke lapisan berikutnya. Proses ini hanya diterapkan pada tahap pelatihan, sedangkan pada tahap prediksi seluruh neuron diaktifkan kembali secara penuh.

Dibawah ini Gambar 2.8 adalah *Dropout Layer*.



Gambar 2.8 *Dropout Layer*
(Sumber : <https://dotnettutorials.net/>)

5. Output Layer

Lapisan *Output* merupakan lapisan terakhir pada arsitektur CNN yang menghasilkan hasil akhir klasifikasi. Untuk klasifikasi multi-kelas aktivitas, CNN menggunakan fungsi *Softmax* pada lapisan *Output*. Fungsi *Softmax* mengubah nilai keluaran menjadi probabilitas setiap kelas aktivitas, di mana jumlah seluruh probabilitasnya bernilai satu. Secara matematis, fungsi *Softmax* dinyatakan sebagai persamaan 2.11:

$$P(y = j) = \frac{e^{z_j}}{\sum_{k=1}^C e^{z_k}} \quad (2.11)$$

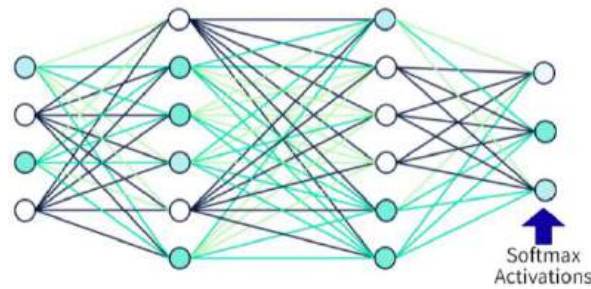
Keterangan:

C : jumlah kelas aktivitas

z_j : keluaran neuron ke- j

Kelas dengan probabilitas tertinggi dipilih sebagai hasil klasifikasi aktivitas

welder. *Softmax Layer* ditunjukkan pada Gambar 2.9.



Gambar 2.9 *Softmax Layer*
(Sumber : <https://blog.gopenai.com/>)

6. Proses Pembelajaran CNN

Proses pembelajaran CNN terdiri dari dua tahap utama, yaitu *feedforward* dan *backpropagation*. *Feedforward* adalah proses propagasi data dari lapisan *input* hingga menghasilkan keluaran prediksi aktivitas. *Backpropagation* digunakan untuk memperbarui bobot jaringan berdasarkan nilai kesalahan (*loss*) antara hasil prediksi dan label sebenarnya. Fungsi *loss* yang umum digunakan untuk klasifikasi multi-kelas adalah *categorical cross-entropy* persamaan 2.12:

$$L = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (2.12)$$

Keterangan :

L : Nilai fungsi *loss*

C : Jumlah kelas

y_i : Label aktual (*ground truth*) untuk kelas ke- i

$\log(\hat{y}_i)$: Logaritma dari probabilitas prediksi kelas ke- i

Bobot jaringan diperbarui menggunakan algoritma optimasi yaitu *Adam Optimizer*, dengan persamaan 2.13 pembaruan bobot:

$$w_{t+1} = w_t - \alpha \cdot \nabla L(w_t) \quad (2.13)$$

Keterangan:

w_{t+1} : Bobot (*weight*) model pada iterasi berikutnya

w_t : Bobot model pada iterasi ke- t

α : *learning rate*

$\nabla L(w_t)$: gradien fungsi *loss* terhadap bobot

Dengan arsitektur ini, CNN 1D mampu mengekstraksi pola temporal dari sinyal IMU dan mengklasifikasikan aktivitas *welder* secara otomatis berdasarkan karakteristik gerakan lengan yang terekam.

2.3 Hardware dan Software

Pada Tugas Akhir ini digunakan sejumlah *hardware* dan *software* sebagai komponen utama dalam perancangan dan implementasi sistem. Komponen tersebut berfungsi untuk mendukung proses akuisisi data sensor, pemrosesan data, komunikasi antar perangkat, serta pengujian sistem. Pemilihan *hardware* dan *software* disesuaikan dengan kebutuhan sistem, karakteristik lingkungan kerja, serta aspek keandalan dan kompatibilitas. *Hardware* yang digunakan dalam penelitian ini adalah sebagai berikut:

2.3.1 Sensor IMU ICM-20948

ICM20948 merupakan sensor *Inertial Measurement Unit* (IMU) 9 *Degree of Freedom* (DoF) yang dikembangkan oleh TDK InvenSense dan mengintegrasikan *accelerometer* 3-sumbu, *gyroscope* 3-sumbu, serta *magnetometer* 3-sumbu dalam satu modul berukuran 3×3 mm. Sensor ini memiliki resolusi ADC 16-bit dengan rentang pengukuran hingga $\pm 16g$ pada *accelerometer*, ± 2000 dps pada *gyroscope*, dan $\pm 4900 \mu T$ pada *magnetometer*, sehingga mampu mengakuisisi data gerak dan orientasi secara akurat. Selain berukuran kecil dan ringan, ICM20948 juga mudah diintegrasikan dengan mikrokontroler 3,3 V maupun 5 V sehingga sesuai untuk aplikasi *embedded system* berbasis IMU (Adafruit, 2023). Tampilan sensor ICM20948 ditunjukkan pada Gambar 2.10.



Gambar 2.10 Sensor ICM-20948 Module
(Sumber : <https://trueetrainingbox.com/>)

2.3.2 Baterai Lipo 3.7V 4000mAh

Baterai Lithium Polymer (Li-Po) 3,7 V berkapasitas 4000 mAh memiliki energi sekitar 18,5 Wh sehingga mampu menyediakan pasokan daya yang stabil untuk sistem *embedded*. Dengan ukuran $10 \times 54 \times 60$ mm dan bobot sekitar 85 gram, baterai ini cukup ringan sehingga tidak mengganggu kenyamanan maupun mobilitas pengguna selama proses pengambilan data. Tampilan baterai Li-Po 3,7 V 4000 mAh ditunjukkan pada Gambar 2.11.



Gambar 2.11 Baterai Lipo 3.7V 4000 mAh
(Sumber : <https://www.amazon.com/>)

2.3.3 Baterai Lipo 3.7V 3.400mAh

Karakteristik baterai *Lithium Polymer* (Li-Po) menunjukkan bahwa baterai ini memiliki massa yang ringan, kapasitas penyimpanan energi yang besar, serta mampu menghasilkan *discharge rate* yang tinggi sehingga sesuai digunakan pada perangkat elektronik (Lamablawa & Aritonang, 2024). Pada penelitian ini, baterai Li-Po 3,7 V 1200mAh digunakan sebagai sumber daya utama untuk menyuplai Raspberry Pi 4 Model B sebesar 5V 3A, dengan konfigurasi 3 *cell* yang menghasilkan tegangan nominal sekitar 11,1 V. Untuk visualisasinya dapat dilihat pada Gambar 2.12.



Gambar 2.12. Baterai Lipo 3.7V 1200 mAh
(Sumber : <https://core-electronics.com.au/>)

Modul *Charger* LX LCBST merupakan modul pengisian daya yang digunakan untuk mengisi baterai lithium melalui antarmuka USB Type-C. Modul ini mendukung proses pengisian baterai secara aman karena telah dilengkapi dengan rangkaian proteksi terhadap kondisi *overcharge*, *over-discharge*, *overcurrent*, dan *short circuit*. Pada penelitian ini, modul *charger* LX LCBST digunakan sebagai pengisi daya untuk baterai Li-ion yang terdiri atas tiga sel berkapasitas 1200 mAh yang disusun secara seri (3S). Susunan tersebut menghasilkan tegangan nominal sebesar 11,1 V dengan tegangan penuh sekitar 12,6 V. Energi yang tersimpan pada baterai selanjutnya digunakan sebagai sumber daya utama sistem *wearable*, khususnya untuk menyuplai Raspberry Pi melalui rangkaian konverter tegangan. Visualisasi modul *charger* LX LCBST dapat dilihat pada Gambar 2.13.



Gambar 2.13 Modul *Charger* LX LCBST
(Sumber: <https://abra-electronics.com/>)

2.3.4 Step Down XY-3606 DC-DC Buck converter

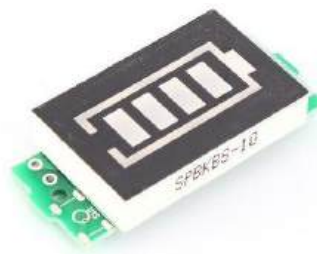
Super Step Down XY-3606 DC-DC *Buck converter* merupakan modul konverter DC-DC tipe buck yang berfungsi untuk menurunkan tegangan DC menjadi tegangan yang lebih rendah dengan efisiensi konversi yang tinggi. Modul ini memiliki tegangan masukan sekitar 9–36 V DC dan mampu menghasilkan tegangan keluaran yang dapat diatur hingga 5 V dengan kemampuan arus mencapai 3 A, sehingga sesuai digunakan sebagai catu daya Raspberry Pi. Pada penelitian ini, modul XY-3606 digunakan untuk menurunkan tegangan keluaran baterai Li-ion 3S yang berkisar antara 11,1–12,6 V menjadi tegangan stabil sebesar 5 V yang dibutuhkan oleh Raspberry Pi 4 Model B. Penggunaan modul *buck converter* ini bertujuan menjaga kestabilan tegangan suplai sehingga Raspberry Pi dapat beroperasi secara optimal tanpa mengalami undervoltage maupun kerusakan akibat tegangan yang terlalu tinggi. Untuk visualisasinya dapat dilihat pada Gambar 2.14.



Gambar 2.14 XY-3606 *Buck converter* Step Down
(Sumber: <https://abra-electronics.com/>)

2.3.5 Baterai Level Indikator Modul

Baterai Level Indikator Modul merupakan modul elektronik yang berfungsi untuk memantau dan menampilkan tingkat kapasitas baterai berdasarkan nilai tegangan yang terukur. Modul ini menggunakan indikator LED sebagai penunjuk persentase daya baterai. Pada penelitian ini, Baterai Level Indikator Modul digunakan untuk memantau kondisi baterai yang menyuplai Raspberry Pi 4 Model B. Baterai yang digunakan memiliki tegangan nominal sekitar 12 V, sehingga indikator baterai berfungsi memberikan informasi mengenai sisa daya selama sistem beroperasi. Dengan adanya modul ini, pengguna dapat mengetahui kapan baterai perlu diisi ulang sehingga sistem *monitoring* aktivitas *welder* dapat bekerja secara berkelanjutan tanpa mengalami gangguan akibat kehabisan daya. Untuk visualisasinya dapat dilihat pada Gambar 2.15.



Gambar 2.15 Baterai Level Indikator Modul
(Sumber: <https://shopee.co.id/>)

2.3.6 Jaket Kerja *Welder*

Jaket kerja *welder* adalah jaket pelindung khusus untuk proses pengelasan yang melindungi tubuh bagian atas dari panas, percikan api, dan risiko mekanis. Terbuat dari kulit suede tahan panas dan tahan gores, jaket ini kuat namun tetap

cukup nyaman digunakan dalam durasi kerja yang lama di lingkungan industri. Seperti yang terlihat pada Gambar 2.16.



Gambar 2.16 Jaket Kerja *Welder*
(Sumber: <https://kartamulyasafety.com/>)

2.3.7 SD Card

Sd Card merupakan media penyimpanan eksternal yang digunakan untuk menyimpan data digital secara berkelanjutan sehingga data dapat diakses kembali (Bieproduction, 2026). Pada penelitian ini, SD Card digunakan untuk menyimpan data CSV dari sensor IMU ICM-20948 serta data video dari kamera sebagai media penyimpanan awal dalam jumlah besar dan memudahkan proses transfer data. Tampilan dapat dilihat pada Gambar 2.17.



Gambar 2.17 SD Card
(Sumber : <https://www.tokopedia.com/>)

2.3.8 ESP32

ESP32 merupakan mikrokontroler yang dikembangkan oleh *Espressif Systems* sebagai penerus dari ESP8266. Dibandingkan dengan *chip* ESP sebelumnya, ESP32 menawarkan keunggulan seperti kecepatan pemrosesan yang lebih tinggi, arsitektur 32 bit, kapasitas memori yang lebih besar, serta integrasi modul *Wi-Fi* dan Bluetooth dalam satu *chip* dengan konsumsi daya yang *relative* rendah, menjadikannya pilihan yang sangat baik untuk berbagai proyek elektronik.

Integrasi modul komunikasi nirkabel ini menghilangkan kebutuhan penggunaan modul *wireless* eksternal tambahan yang umumnya memiliki harga yang lebih tinggi. (Zenhadi, 2022). Tampilan dapat dilihat pada Gambar 2.18.



Gambar 2.18 *Chip ESP32*
(Sumber : <https://www.studiopieters.nl/>)

Adapun spesifikasi dari ESP32 dapat diamati pada Tabel 2.1 di bawah ini.

Tabel 2.1 Spesifikasi ESP32

Parameter	Spesifikasi
CPU	<i>Tensilica Xtensa Dual-Core 32-bit LX6 microprocessor 160MHz/240MHz</i>
Tegangan Suplai	2.3V – 3.6V
Konsumsi Arus	500mA
<i>Wireless connectivity</i>	<i>Wi-Fi, Bluetooth legacy, dan Bluetooth Low Energy (BLE)</i>
<i>Bandwidth</i>	7.8 kHz – 500kHz
ROM	448kB
SRAM	520KB
<i>Peripheral</i>	SPI, UART, ADC, DAC, I2C, 2s, <i>Touch Capacitive</i> , PWM
Sensitivitas	-111dBm – 148dBm
Jumlah I/O	34
Ukuran	19mm x 25.5mm

(Sumber: <https://documentation.espressif.com/>)

2.3.9 Raspberry Pi

Raspberry Pi adalah sebuah *single-board computer* (SBC) yang banyak digunakan dalam penelitian dan pengembangan sistem *embedded* serta *Internet of Things* (IoT). Raspberry Pi menawarkan kemampuan komputasi yang lebih tinggi dibandingkan mikrokontroler sederhana, dukungan terhadap sistem operasi Linux, serta berbagai antarmuka *input/Output* seperti *General Purpose Input Output* (GPIO), USB, dan jaringan yang memudahkan integrasi dengan sensor dan

perangkat eksternal. Raspberry Pi sering dimanfaatkan sebagai unit pemrosesan dalam sistem yang membutuhkan akuisisi data, pemrosesan lokal, serta komunikasi jaringan, seperti aplikasi IoT dan sistem *monitoring* berbasis sensor (Foundation, 2023). Dapat dilihat pada Gambar 2.19.



Gambar 2.19 Raspberry Pi
(Sumber : <https://id.wikipedia.org/>)

Adapun spesifikasi Raspberry Pi dapat diamati pada Tabel 2.2 di bawah ini.

Tabel 2.2 Spesifikasi Raspberry Pi 4 Model B

Parameter	Spesifikasi
Model	Raspberry Pi 4 Model B
Prosesor (CPU)	Broadcom BCM2711, Quad-core ARM Cortex-A72 (ARMv8) 64-bit
Memori (RAM)	1 GB / 2 GB / 4 GB / 8 GB LPDDR4 SDRAM
Media Penyimpanan	MicroSD card
Konektivitas Nirkabel	Wi-Fi 2.4 GHz & 5 GHz (IEEE 802.11 b/g/n/ac), Bluetooth 5.0 + BLE
Port USB	2 × USB 3.0, 2 × USB 2.0
GPIO	40-pin General Purpose <i>Input Output</i> (GPIO)
Daya	5 V DC via USB-C (3 A)

(Sumber: <https://www.raspberrypi.com/>)

2.3.10 Modul Kamera USB

Kamera USB merupakan perangkat akuisisi citra digital yang menggunakan antarmuka USB sebagai media transmisi data dan suplai daya. Kamera ini mendukung konsep plug and play sehingga tidak memerlukan instalasi *driver* tambahan ketika dihubungkan dengan Raspberry Pi. Kamera memiliki resolusi video sebesar 640×480 piksel (VGA) dengan lensa *fixed focus* F6.0 mm dan jarak fokus minimum sekitar 20 mm. Dengan kebutuhan catu daya sebesar 5 V DC,

kamera dapat dihubungkan langsung ke port USB Raspberry Pi 4 Model B untuk melakukan proses pengambilan citra secara kontinu. Tampilan kamera mini USB ditunjukkan pada Gambar 2.20.



Gambar 2.20 Modul kamera USB Mini
(Sumber : <https://www.aliexpress.com/>)

Adapun spesifikasi dari modul kamera USB mini dapat dilihat pada Tabel 2.3 di bawah ini.

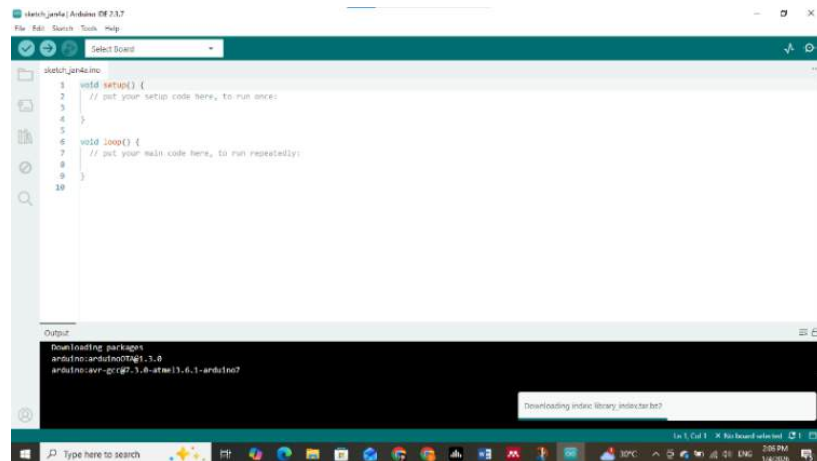
Tabel 2.3 Spesifikasi Kamera Mini USB

Parameter	Spesifikasi
<i>Interface</i>	USB
Sensor Gambar	CMOS
Resolusi Video	640 × 480 piksel (VGA)
Format Video	JPEG
Panjang Fokus Lensa	F6.0 mm
Jarak Fokus Minimum	20mm
Tegangan Operasi	DC 5V

(Sumber : <https://www.aliexpress.com/>)

2.3.11 Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang digunakan untuk lingkungan pengembangan program pada berbagai jenis board. Software ini berfungsi untuk membuat, mengedit, mengunggah program yang dikenal sebagai *sketch* ke board yang dipilih, dan melakukan pengkodean untuk berbagai kebutuhan. Arduino IDE dikembangkan menggunakan bahasa pemrograman JAVA dan dilengkapi dengan *library C/C++ (wiring)*, yang mempermudah pengelolaan operasi *input* dan *Output*. Berikut adalah tampilan *software* Arduino IDE pada Gambar 2.21 berikut.

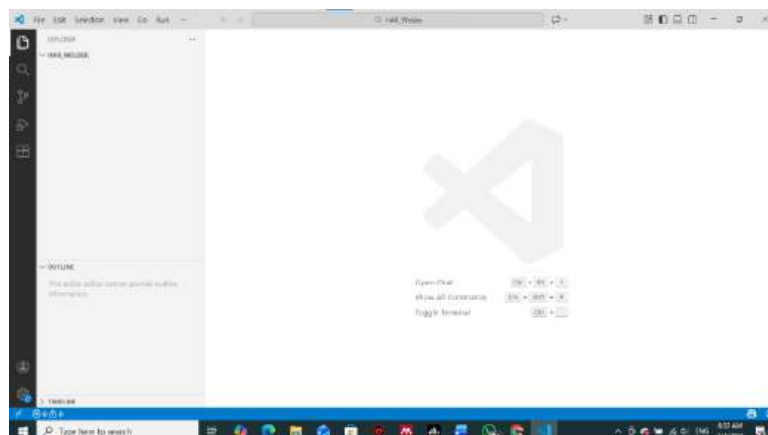


Gambar 2.21 Arduino IDE
(Dokumentasi Pribadi Penulis, 2026)

Arduino IDE terdiri atas tiga komponen utama, yaitu *editor* untuk menulis *sketch*, *compiler* untuk mengubah kode program menjadi kode biner yang dapat dipahami mikrokontroler, dan *uploader* untuk mengunggah kode biner ke memori mikrokontroler.

2.3.12 Visual Studio Code

Visual Studio Code merupakan editor kode sumber *open source* yang dikembangkan oleh Microsoft dan dapat dijalankan pada berbagai sistem operasi, seperti *Windows*, *Linux*, dan *macOS*. Perangkat lunak ini mendukung berbagai bahasa pemrograman, seperti *Python*, *C++*, *Java*, *JavaScript*, dan *Go*, serta dilengkapi fitur *syntax highlighting*, *code completion*, *debugging*, dan integrasi *Git* yang mendukung proses pengembangan perangkat lunak secara efisien (Santi, 2025). Tampilan Visual Studio Code ditunjukkan pada Gambar 2.22.



Gambar 2.22 Tampilan *Software* Visual Studio Code
(Dokumentasi Pribadi Penulis, 2026)

2.3.13 Python

Python merupakan bahasa pemrograman berbasis *interpreter* yang dapat dijalankan pada berbagai platform, seperti *Windows* dan *Linux*. Python mendukung berbagai paradigma pemrograman, antara lain prosedural, berorientasi objek, dan fungsional, sehingga fleksibel dalam pengembangan berbagai jenis aplikasi. Selain itu, Python banyak digunakan dalam pengembangan aplikasi web, desktop, *Internet of Things* (IoT), pengolahan data, serta *machine learning* karena kemampuannya dalam mengelola data, terintegrasi dengan berbagai sistem basis data, dan mendukung perhitungan matematis yang kompleks (Rahman et al., 2023). Logo Python ditunjukkan pada Gambar 2.23.



Gambar 2.23 Logo Python
(Sumber: <https://www.python.org/community/logos/>)

2.3.14 MongoDB

MongoDB merupakan sistem manajemen basis data NoSQL berbasis dokumen yang menyimpan data dalam format BSON. Basis data ini memiliki struktur yang fleksibel sehingga sesuai untuk menyimpan data sensor dan aplikasi *Internet of Things* (IoT). Pada penelitian ini, MongoDB digunakan untuk menyimpan hasil klasifikasi aktivitas *welder* beserta *timestamp*, sehingga mendukung penyimpanan dan penyajian data secara *real-time* pada *dashboard* web (GeeksforGeeks, 2026). Tampilan MongoDB ditunjukkan pada Gambar 2.24.



Gambar 2.24 Logo MongoDB
(Sumber : <https://opencv.org/media-kit/>)

2.3.15 MQTT (*Message Queuing Telemetry Transport*)

Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi ringan berbasis mekanisme *publish-subscribe* yang melibatkan *client* dan *broker*. Pada mekanisme ini, *publisher* mengirimkan data ke *broker* melalui suatu *topic*, kemudian *broker* mendistribusikan data tersebut kepada *subscriber* yang berlangganan pada *topic* yang sama. Karena memiliki kebutuhan bandwidth yang rendah dan latensi kecil, MQTT banyak digunakan pada aplikasi *Internet of Things* (IoT) (HiveMQ, 2024). Ilustrasi mekanisme MQTT ditunjukkan pada Gambar 2.25.



Gambar 2.25 Topologi Protokol MQTT
(Sumber : <https://iotstudio.labs.telkomuniversity.ac.id/>)

(Halaman ini sengaja dikosongkan)

BAB 3

METODE PENELITIAN

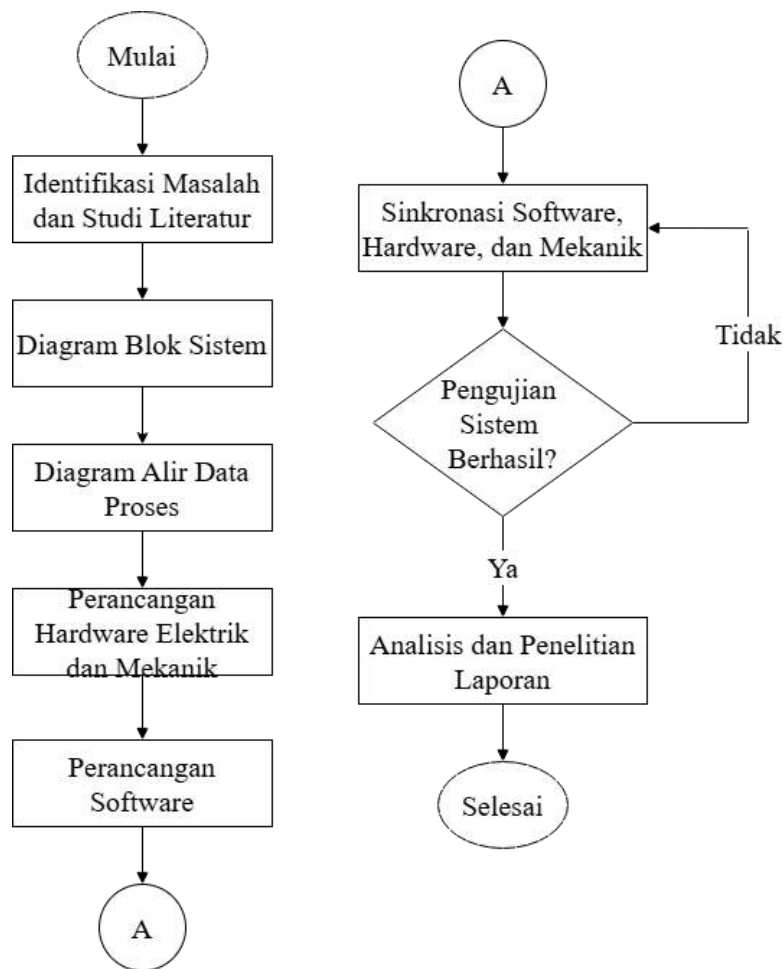
Metode penelitian adalah suatu rangkaian langkah yang menjadi pedoman dalam menyelesaikan proses penelitian. Konsep penelitian berfungsi sebagai dasar untuk merancang dan mengembangkan sistem yang akan dibangun. Dengan memberikan gambaran yang jelas tentang sistem yang akan dibuat, penelitian ini akan memiliki struktur yang kokoh.

3.1 Konsep Penelitian

Penelitian Tugas Akhir ini berfokus pada perancangan sistem, mencakup struktur *hardware* dan *software* yang digunakan. Tahapan penelitian dimulai dengan pembuatan diagram blok sistem untuk menggambarkan keterkaitan antar komponen secara keseluruhan, dilanjutkan dengan penyusunan diagram aliran data proses, dan diakhiri dengan penerapan metode perancangan agar hasil penelitian sesuai dengan tujuan yang ditetapkan.

3.2 Tahapan Penelitian

Pada Gambar 3.1 ditunjukkan alur penelitian yang menggambarkan tahapan pelaksanaan Tugas Akhir secara sistematis mulai dari identifikasi permasalahan hingga penyusunan laporan penelitian. Tahap awal penelitian diawali dengan identifikasi masalah dan studi literatur untuk memahami permasalahan yang dihadapi serta memperoleh landasan teori yang relevan sebagai dasar dalam perancangan sistem. Selanjutnya dilakukan penyusunan diagram blok sistem untuk menggambarkan hubungan antar komponen utama, kemudian dilanjutkan dengan penyusunan diagram alir data proses yang menjelaskan aliran data sistem secara keseluruhan. Berdasarkan hasil perancangan tersebut, dilakukan perancangan hardware elektrik dan mekanik sebagai media implementasi sistem *wearable*, serta perancangan *software* yang meliputi program pada ESP32, Raspberry Pi, komunikasi data, proses pengolahan data, model klasifikasi, hingga *dashboard* web sebagai media *monitoring* aktivitas *welder*. Setelah seluruh komponen selesai dirancang, dilakukan proses sinkronisasi antara *software*, *hardware*, dan mekanik untuk memastikan seluruh subsistem dapat bekerja secara terintegrasi.



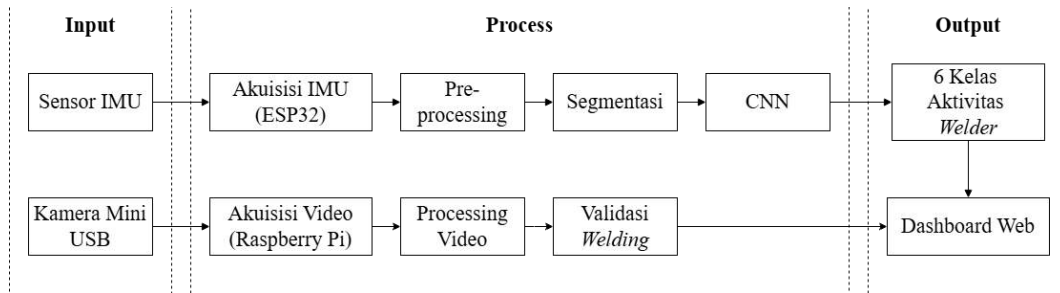
Gambar 3.1 Diagram Penelitian
(Dokumentasi Pribadi Penulis, 2026)

Tahap berikutnya adalah pengujian sistem guna mengevaluasi kesesuaian fungsi setiap komponen terhadap tujuan penelitian. Apabila hasil pengujian menunjukkan sistem belum bekerja sesuai dengan yang diharapkan, maka dilakukan perbaikan pada tahap sinkronisasi *hardware* maupun *software* hingga diperoleh sistem yang berfungsi dengan baik. Sebaliknya, apabila sistem telah memenuhi seluruh kebutuhan fungsional, penelitian dilanjutkan pada tahap analisis hasil dan penyusunan laporan sebagai bentuk evaluasi terhadap kinerja sistem yang telah dikembangkan. Seluruh rangkaian penelitian diakhiri setelah seluruh tahapan selesai dilaksanakan dan tujuan penelitian berhasil dicapai.

3.2.1 Diagram Blok Sistem

Pada Gambar 3.2 ditunjukkan diagram blok sistem yang menggambarkan struktur dan keterkaitan antar komponen dalam sistem *embedded* berbasis sensor

Inertial Measurement Unit (IMU) untuk mengklasifikasikan aktivitas *welder*. Diagram ini memberikan gambaran menyeluruh mengenai alur kerja sistem yang dirancang dalam tiga blok utama, yaitu blok *input*, blok *process*, dan blok *Output*, yang saling terintegrasi satu sama lain.



Gambar 3.2 Diagram Blok Sistem
(Dokumentasi Pribadi Penulis, 2026)

Pada bagian *input*, sistem menerima data dari dua sumber utama, yaitu sensor IMU dan kamera mini USB. Data IMU diakuisisi menggunakan ESP32 untuk merekam gerakan dan orientasi tubuh *welder* selama proses berlangsung, sedangkan kamera USB pada raspberry pi digunakan untuk merekam cahaya busur las sebagai data *ground truth* untuk memvalidasi kondisi pengelasan.

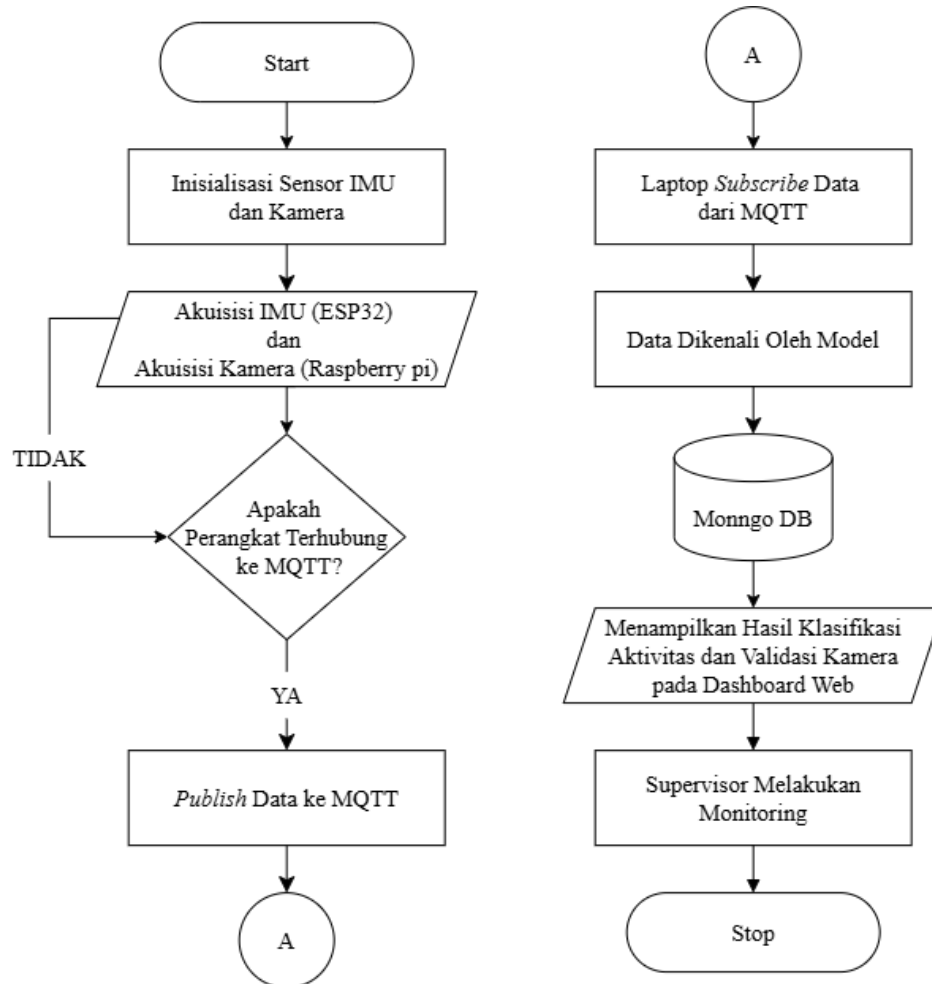
Pada blok *process*, data IMU melalui tahap *pre-processing* yang meliputi pembersihan dan normalisasi, kemudian disegmentasi menggunakan metode *sliding window* sebelum diproses oleh model *Convolutional Neural Network* (CNN) untuk mengklasifikasikan aktivitas *welder*. Sementara itu, data dari kamera di proses untuk mendeteksi tingkat *brightness* sebagai validasi apakah *welder* sedang melakukan pengelasan atau tidak.

Pada bagian *Output*, sistem menghasilkan enam kelas aktivitas *welder*, yaitu duduk aktif, berdiri aktif, berbaring aktif, duduk tidak aktif, berdiri tidak aktif, dan berbaring tidak aktif, serta menampilkan hasil pemantauan melalui *dashboard web*. *Dashboard web* tersebut digunakan sebagai media visualisasi untuk menampilkan hasil klasifikasi aktivitas secara *real-time*.

3.2.2 Diagram Aliran Data Proses

Diagram aliran data proses pada Gambar 3.3 menunjukkan alur sistem *monitoring* aktivitas *welder*. Diagram aliran data proses menggambarkan

bagaimana data aktivitas *welder* diperoleh, diproses, disimpan, dan ditampilkan kepada pengguna data aktivitas *welder* secara *real-time*.



Gambar 3.3 Diagram Aliran Data Proses
(Dokumentasi Pribadi Penulis, 2026)

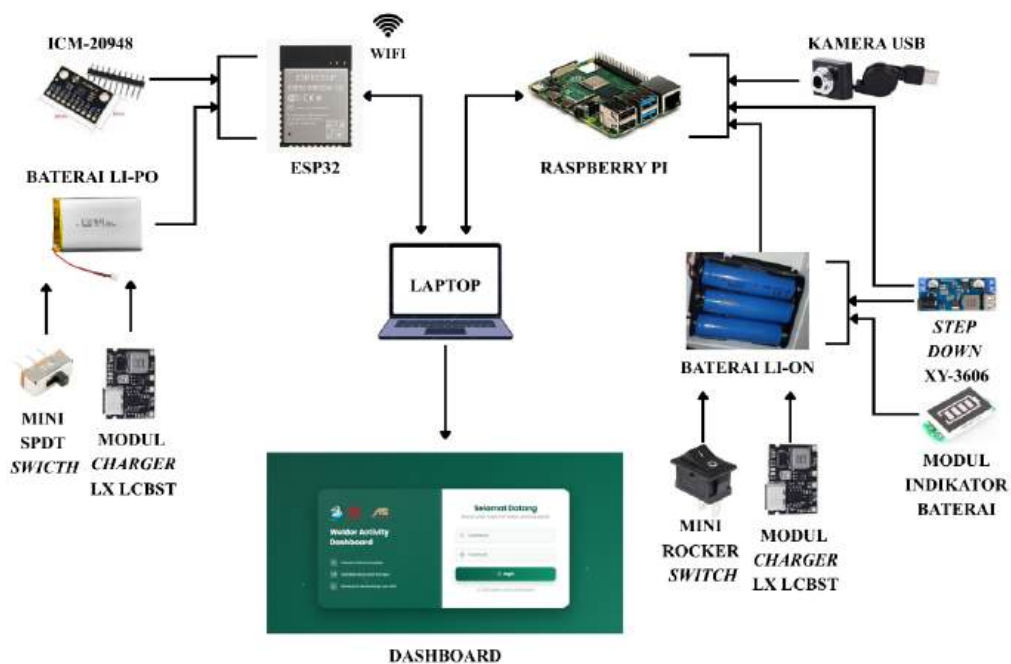
Data gerakan *welder* diperoleh dari sensor IMU yang terintegrasi pada jaket *welder* dan diakuisisi menggunakan ESP32, sedangkan kamera yang terhubung dengan Raspberry Pi digunakan untuk merekam cahaya busur las sebagai data pendukung validasi. Setelah proses akuisisi, sistem memeriksa koneksi perangkat ke broker MQTT. Apabila koneksi berhasil, data dipublikasikan melalui MQTT dan diproses oleh model klasifikasi untuk mengenali aktivitas *welder*. Selanjutnya, laptop melakukan *subscribe* terhadap data MQTT, kemudian menyimpan hasil klasifikasi ke dalam *database* MongoDB. Data aktivitas beserta validasi kamera ditampilkan pada *dashboard* web sehingga *supervisor* dapat melakukan *monitoring* aktivitas *welder* secara *real-time* dan berkelanjutan.

3.3 Desain dan Perancangan Sistem

Tahap desain dan perancangan sistem menjelaskan rancangan keseluruhan *embedded system* yang dipasang langsung pada jaket kerja *welder* dan digunakan untuk pemantauan aktivitas *welder*. Tahap ini sistem dibagi menjadi dua bagian, yaitu. Perancangan *hardware* elektrik meliputi pembuatan skema rangkaian elektronik, Perancangan *hardware* mekanik mencakup desain fisik perangkat sistem yang dipasang pada jaket kerja *welder*.

3.3.1 Hardware Electric

Pada sub bab ini menunjukkan perancangan *hardware* elektrik yang merupakan proses perencanaan rangkaian elektrik yang akan dipasang di jaket kerja *welder*.

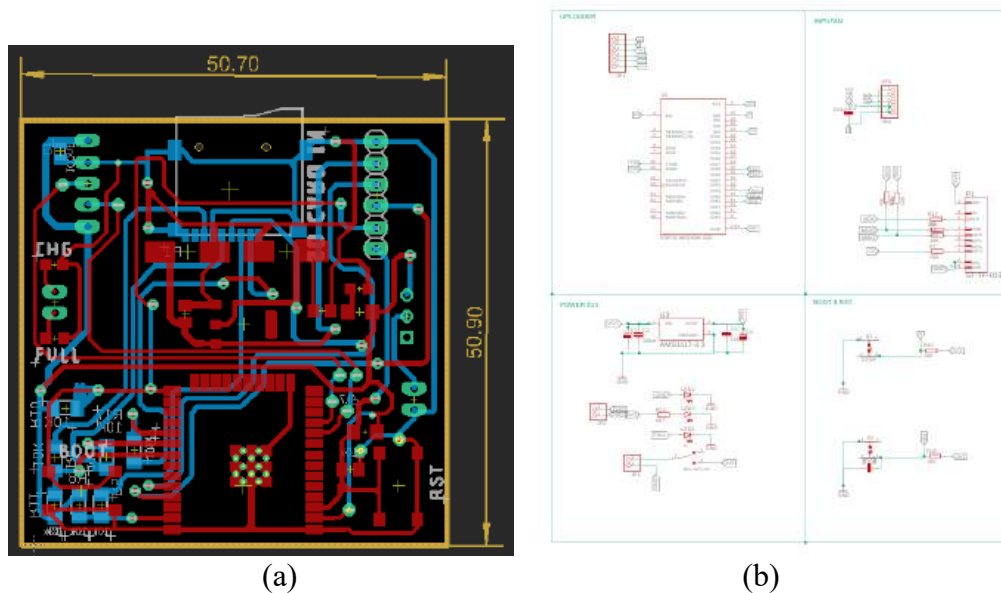


Gambar 3.4 Perancangan *Hardware electric*
(Dokumentasi Pribadi Penulis, 2026)

Pada penelitian ini, perancangan *hardware electric* dilakukan secara terintegrasi untuk mendukung kebutuhan akuisisi data dan komunikasi sistem sebagaimana ditunjukkan pada Gambar 3.4. Sistem menggunakan dua perangkat pemrosesan utama, yaitu ESP32 dan Raspberry Pi 4 Model B, dengan fungsi yang berbeda. ESP32 digunakan sebagai mikrokontroler untuk melakukan akuisisi data

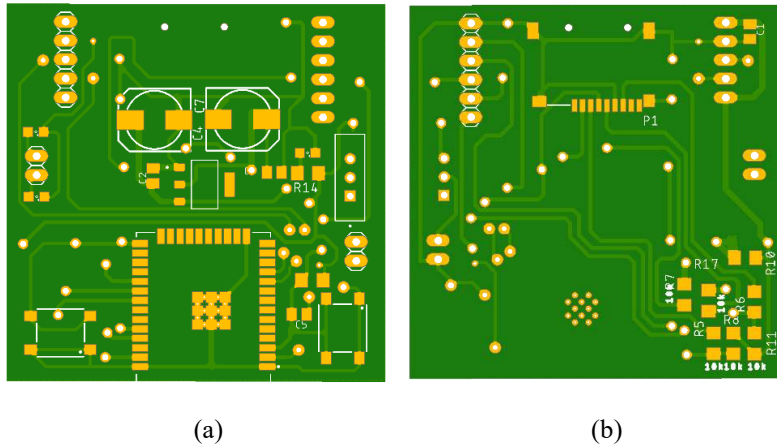
dari sensor IMU ICM-20948, sedangkan Raspberry Pi 4 digunakan untuk melakukan akuisisi data visual melalui kamera USB. Kedua perangkat pemrosesan terpasang pada jaket *welder*.

Kedua perangkat bekerja secara independen dengan sumber daya masing-masing dan melakukan proses perekaman data secara *real-time*. ESP32 memperoleh suplai daya dari baterai LiPo melalui rangkaian pengatur tegangan, sedangkan Raspberry Pi menggunakan sumber daya terpisah untuk mendukung proses *welding detection* dan komunikasi data. Dengan rancangan *hardware* elektrik tersebut, sistem mampu melakukan pengambilan data gerakan dan visual secara simultan untuk mendukung proses klasifikasi aktivitas *welder*.



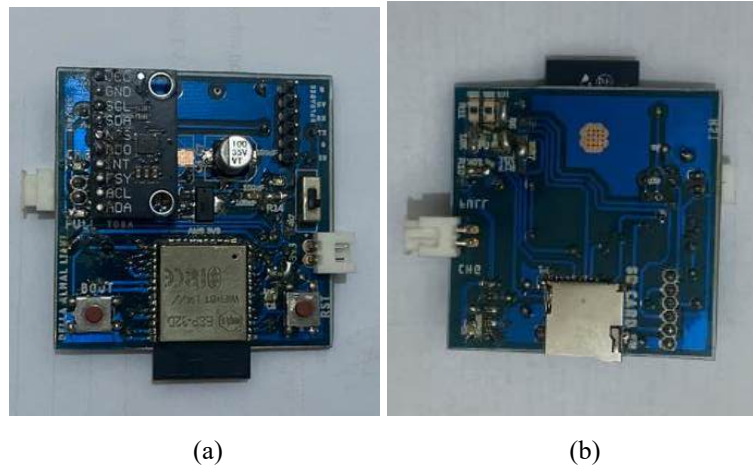
Gambar 3.5 a) Hasil *Routing* PCB dan b) Rangkaian Skematik PCB
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 3.5 penelitian ini juga dibuat desain PCB (*Printed Circuit Board*) khusus yang berfungsi sebagai penghubung antara komponen utama. Perancangan ini bertujuan untuk merapikan koneksi antar perangkat elektronik, mengurangi penggunaan kabel yang berlebihan, serta meningkatkan stabilitas dan efisiensi sistem secara keseluruhan. Proses desain dilakukan menggunakan perangkat lunak Eagle PCB, yang memungkinkan peneliti untuk membuat skematik rangkaian dan tata letak jalur.



Gambar 3.6 a) Desain PCB Tampak Atas b) Desain PCB Tampak Bawah
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 3.6 a) dan Gambar 3.6 b) adalah tampak atas dan bawah papan PCB yang telah dirancang dengan mempertimbangkan efisiensi ruang dan pengaturan jalur sinyal yang baik untuk menghindari interferensi antar komponen.



Gambar 3.7 a) Hasil Perakitan PCB Tampak Atas
b) Hasil Perakitan PCB Tampak Bawah

Pada Gambar 3.7 a) dan Gambar 3.7 b) menunjukkan hasil perakitan perancangan papan PCB tampak atas dan tampak bawah yang digunakan sebagai media integrasi rangkaian elektronik pada sistem.

3.3.2 Hardware Mekanik

Hardware mekanik ini dirancang untuk mendukung *embedded system* berbasis sensor IMU yang terintegrasi pada jaket kerja *welder*. Desain mekanik dibuat dengan mempertimbangkan kenyamanan, fleksibilitas gerak, posisi

pemasangan, serta keamanan komponen selama proses kerja. Desain *hardware* mekanik dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Design Hardware*
(Dokumentasi Pribadi Penulis, 2026)

Gambar tersebut menunjukkan rancangan penempatan sistem *embedded* berbasis sensor IMU yang terintegrasi pada jaket kerja *welder*. Keterangan:

1. Sensor ditempatkan pada lengan atas pada jaket *welder*. Posisi ini dipilih agar terlindungi dari panas dan percikan las.
2. Kamera USB diletakkan dibagian dada atas menghadap ke depan. Posisi ini memungkinkan kamera menghadap aktivitas pengelasan dari sudut pandang yang mendekati pandangan pengguna, sehingga data visual yang diperoleh relevan sebagai pendukung analisis aktivitas.
3. Raspberry Pi ditempatkan pada saku bagian bawah jaket. Posisi ini memberikan ruang yang cukup untuk mikro, membantu distribusi berat agar tetap nyaman dipakai, serta memudahkan pengolahan kabel.

Selanjutnya, pada Gambar 3.9 (a) dan (b) ditunjukkan hasil realisasi desain *hardware* mekanik *wearable* yang telah terintegrasi pada jaket *welder*. Perancangan mekanik ini berfungsi sebagai media pemasangan komponen utama sistem agar seluruh komponen dapat terpasang dengan rapi dan terlindungi selama proses pengoperasian. Selain itu, desain *wearable* dirancang dengan mempertimbangkan distribusi beban yang merata, fleksibilitas gerak saat melakukan aktivitas pengelasan.

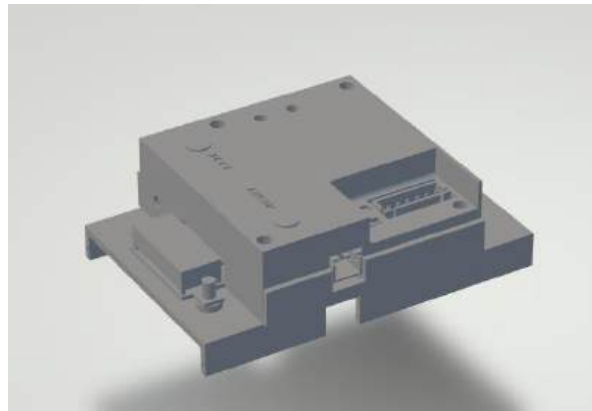


(a)

(b)

Gambar 3.9 a) Hasil perancangan jaket *welder* tampak depan
b) Hasil perancangan jaket *welder* tampak samping

Pada Gambar 3.10 Desain *casing* PCB dibuat menggunakan pemodelan 3D sebagai pelindung komponen elektronik pada sistem *wearable*. *Casing* dirancang menyesuaikan dimensi PCB dan komponen pendukung agar pemasangan lebih rapi, aman, serta mudah diintegrasikan pada jaket *welder*. Hasil desain 3D kemudian direalisasikan menggunakan metode 3D *printing* untuk menghasilkan *casing* fisik yang mampu melindungi komponen selama proses penggunaan sistem.



Gambar 3.10 Desain 3D *Casing* PCB
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 3.11 menunjukkan hasil realisasi *casing* PCB menggunakan metode 3D *printing*. *Casing* ini digunakan sebagai pelindung komponen elektronik sekaligus media pemasangan PCB pada sistem *wearable*. Desain *casing* dibuat menyesuaikan dimensi komponen agar PCB dapat terpasang dengan stabil, terlindungi, dan mudah diintegrasikan pada jaket *welder*.



Gambar 3.11 Hasil 3D *Casing* PCB
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 3.12 Desain *casing* baterai Raspberry Pi dibuat menggunakan pemodelan 3D sebagai wadah pelindung baterai LiPo yang digunakan sebagai sumber daya sistem pemrosesan. *Casing* dirancang menyesuaikan dimensi baterai. Hasil desain kemudian direalisasikan menggunakan metode 3D *printing* untuk menghasilkan *casing* yang mampu melindungi baterai selama sistem *wearable* digunakan.



Gambar 3.12Desain 3D *Casing* Baterai Raspberry Pi
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 3.13 menunjukkan hasil desain 3D *casing* baterai Raspberry Pi yang berfungsi sebagai pelindung baterai LiPo pada sistem *wearable*. *Casing* dirancang menyesuaikan ukuran baterai dengan mempertimbangkan aspek keamanan, kestabilan pemasangan, serta kemudahan integrasi pada jaket *welder*. Desain ini digunakan sebagai acuan dalam proses pembuatan *casing* fisik menggunakan metode 3D *printing*.



Gambar 3.13 Hasil Desain 3D *Casing* Baterai Raspberry Pi
(Dokumentasi Pribadi Penulis, 2026)

3.3.3 Ketahanan Panas Komponen dengan Lingkungan Las

Dibawah ini adalah ketahanan panas sistem pada tabel 3.1.

Tabel 3.1 Ketahanan Panas Komponen

No	Komponen	Suhu Operasi	Sumber	Mitigasi Desain
1.	ESP32	-40°C hingga +85 °C	https://www.alldatasheet.com/	Ditempatkan dalam jaket berbahan kulit
2.	Raspberry Pi 4 Model B	0°C hingga +50°C		
3.	Sensor IMU	-40°C hingga +85 °C		
4.	Baterai Li-Po	0°C hingga +60 °C		
5.	Baterai Li-On	0°C hingga +60°C		
6.	Kamera USB	0°C hingga +60°C		
7.	<i>Step Down</i> XY-3606	-40°C hingga +85 °C		
8.	Model <i>Charger</i> LX LCBST	-40°C hingga +85°C		
9.	<i>Mini Switch</i>	-20°C hingga +70 °C		
10.	Modul Indikator Baterai	-20°C hingga +50°C		

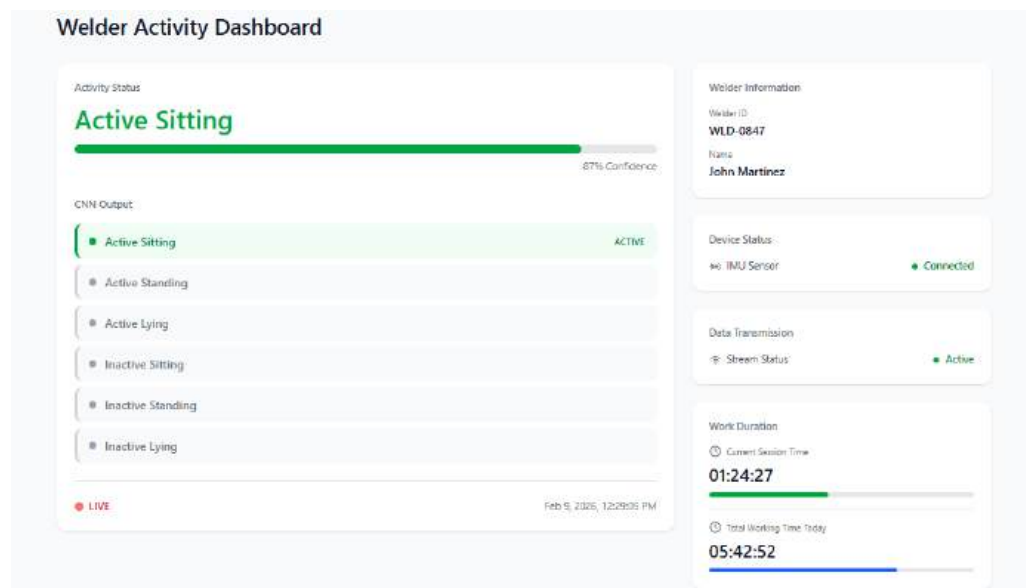
(Dokumentasi Pribadi Penulis, 2026)

Pada Tabel 3.1 ini menunjukkan analisis kelayakan termal komponen sistem pada lingkungan kerja pengelasan. Seluruh komponen bekerja dalam batas suhu operasional yang aman dan dilindungi melalui desain seperti penempatan di dalam

jaket berbahan kulit, sekat pelindung, serta ventilasi panas. Dari sisi *device safety*, mitigasi ini mencegah overheating, *drift* sensor, dan kerusakan rangkaian. Dari sisi *user safety*, penempatan komponen yang terlindung memastikan tidak ada risiko panas atau listrik yang membahayakan *welder*. Dengan demikian, sistem dinilai layak secara operasional dan aman secara termal untuk digunakan pada kondisi kerja pengelasan.

3.3.4 Perancangan Desain UI/UX *Dashboard*

Perancangan Desain UI/UX *Dashboard* adalah proses perencanaan rangkaian tampilan web yang akan digunakan sebagai media pemantauan aktivitas *welder* secara *real-time*. Pada tahap ini, desain difokuskan untuk menghasilkan tampilan yang informatif serta mudah dipahami. Perancangan ini mencakup beberapa tampilan utama, yang menunjukkan aktivitas status *activity welder*, informasi identitas pengguna *id welder* dan *device* status, dapat dilihat pada Gambar 3.14.

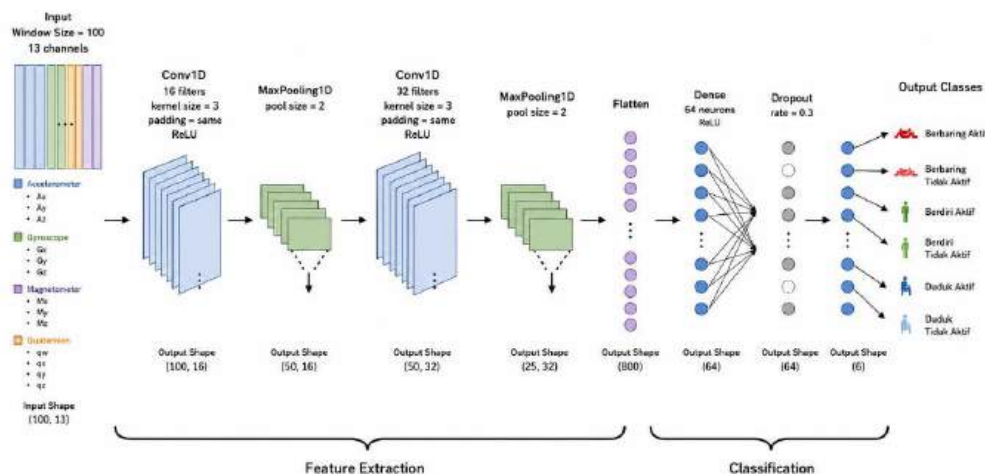


Gambar 3.14 Tampilan *Welder Activity Dashboard*
(Dokumentasi Pribadi Penulis, 2026)

3.3.5 Perancangan Metode *Convolutional Neural Network*

Arsitektur *Convolutional Neural Network* (CNN) pada penelitian ini digunakan untuk mengekstraksi fitur dari data sensor *Inertial Measurement Unit* (IMU) dalam bentuk sinyal *time-series* serta melakukan klasifikasi aktivitas *welder*.

Hasil klasifikasi aktivitas ini selanjutnya digunakan sebagai dasar analisis kondisi kerja *welder* dalam sistem yang dikembangkan.



Gambar 3.15 *Architecture CNN*
(Dokumentasi Pribadi Penulis, 2026)

Pada gambar 3.15 adalah CNN *architecture* untuk klasifikasi aktivitas *welder* dimana data *input* yang digunakan dalam penelitian ini terdiri dari *accelerometer*, *gyroscope*, *magnetometer*, dan *quaternion*. Data tersebut diproses untuk melakukan klasifikasi aktivitas *welder*. Konfigurasi lengkap tiap lapisan ditunjukkan pada Tabel 3.2 berikut.

Tabel 3.2 *Architecture CNN*

Lapisan	Konfigurasi	Output Shape
<i>Input</i>	13 kanal (accel, gyro, mag, <i>quaternion</i>), <i>window</i> 100 sampel	(100, 13)
Conv1D Layer 1	16 filter, kernel 3, <i>padding same</i> , aktivasi ReLU	(100, 16)
<i>Max Pooling</i> 1	Pool size 2	(50, 16)
Conv1D Layer 2	32 filter, kernel 3, <i>padding same</i> , aktivasi ReLU	(50, 32)
<i>Max Pooling</i> 2	Pool size 2	(25, 32)
<i>Flatten</i>	Vektorisasi	(800)
<i>Dense (Fully Connected)</i>	64 neuron, Aktivasi ReLU	(64)
<i>Dropout</i>	Rate 0,3	(64)
<i>Output</i>	6 neuron, aktivasi <i>Softmax</i>	(6)

Tabel 3.2 menunjukkan spesifikasi lengkap arsitektur CNN yang dirancang pada penelitian ini. Berikut penjelasan fungsi dan konfigurasi tiap lapisan.

1. *Input* menerima data dengan 13 kanal, terdiri dari *accelerometer* (3 sumbu), *gyroscope* (3 sumbu), *magnetometer* (3 sumbu), dan *quaternion* (4 komponen), dengan panjang *window* 100 sampel yaitu *sampling rate* 50 Hz $\times$ durasi *window* 2 detik.
2. Conv1D Layer 1 berfungsi mengekstraksi fitur dasar dari sinyal IMU, seperti pola percepatan dan perubahan orientasi awal, menggunakan operasi konvolusi sesuai Persamaan 2.6. Digunakan 16 filter dengan kernel size 3 dan *padding* same mempertahankan panjang *time-step*, diikuti fungsi aktivasi ReLU Persamaan 2.7 untuk memperkenalkan non-linearitas.
3. *Max Pooling* 1 mereduksi dimensi *feature map* menggunakan operasi *Max pooling* sesuai Persamaan 2.8, dengan pool size 2, sehingga panjang *time-step* berkurang setengah dari 100 menjadi 50 sambil mempertahankan fitur dominan.
4. Conv1D Layer 2 mengekstraksi pola gerakan yang lebih kompleks kombinasi gerak tangan dan orientasi, menggunakan 32 filter dua kali lipat dari lapisan pertama, mengikuti prinsip umum arsitektur CNN di mana lapisan yang lebih dalam menangkap representasi fitur yang lebih abstrak, dengan konfigurasi kernel dan aktivasi yang sama seperti Conv1D Layer 1.
5. *Max Pooling* 2 kembali mereduksi dimensi *feature map* dengan pool size 2, menghasilkan *Output* (25, 32).
6. *Flatten* mengubah feature map dua dimensi (25, 32) menjadi vektor satu dimensi berukuran 800 ($25 \times 32 = 800$), sebagai penghubung antara tahap ekstraksi fitur dan tahap klasifikasi.
7. *Dense (Fully Connected)* menggabungkan seluruh fitur hasil ekstraksi menggunakan operasi sesuai Persamaan 2.9, dengan 64 neuron dan aktivasi ReLU, mempelajari hubungan kompleks antar fitur sebelum tahap klasifikasi akhir.
8. *Dropout* diterapkan dengan rate 0,3 sesuai Persamaan 2.10, menonaktifkan 30% neuron secara acak pada setiap iterasi pelatihan untuk mencegah *overfitting*. Pemilihan rate 0,3 mempertimbangkan keseimbangan antara regularisasi yang

cukup dan mempertahankan informasi penting, mengingat jumlah sampel *dataset* yang terbatas.

9. *Output* menghasilkan hasil klasifikasi akhir menggunakan 6 neuron dengan aktivasi *Softmax* sesuai Persamaan 2.11, merepresentasikan probabilitas tiap kelas dari enam aktivitas *welder* yang diklasifikasikan Berbaring Aktif, Berbaring Tidak Aktif, Berdiri Aktif, Berdiri Tidak Aktif, Duduk Aktif, Duduk Tidak Aktif.

3.4 Pengambilan Data

Subbab ini membahas proses pengambilan data aktivitas *welder* sebagai dasar pengembangan sistem, meliputi objek dan perangkat penelitian, jenis data, serta durasi dan periode pengambilan data.

3.4.1 Objek dan Perangkat Pengambilan Data

Objek penelitian ini adalah aktivitas *welder* di lingkungan bengkel las, dengan fokus pengamatan pada gerakan lengan atas karena berperan dominan selama proses pengelasan, seperti mengarahkan elektroda dan menjaga kestabilan gerakan. Pengambilan data dilakukan berdasarkan enam skenario aktivitas, yaitu duduk aktif, berdiri aktif, berbaring aktif, duduk tidak aktif, berdiri tidak aktif, dan berbaring tidak aktif. Perangkat utama yang digunakan adalah sistem *wearable* berbasis mikrokontroler ESP32 dengan sensor IMU 9-DoF ICM-20948, yang terdiri atas *accelerometer*, *gyroscope*, dan *magnetometer* untuk merekam data percepatan, kecepatan sudut, dan medan magnet. Perangkat ditempatkan pada jaket kerja di area lengan atas agar pergerakan tubuh terekam stabil tanpa mengganggu aktivitas *welder*.

3.4.2 Jenis Data yang Diambil

Data utama penelitian berupa data sensor *time-series* multikanal yang diperoleh dari modul IMU ICM-20948. Data yang direkam terdiri atas sembilan parameter sensor, yaitu *accelerometer* (A_x, A_y, A_z), *gyroscope* (G_x, G_y, G_z), dan *magnetometer* (M_x, M_y, M_z), serta data orientasi dalam bentuk *quaternion* (q_w, q_x, q_y, q_z) yang dihitung langsung oleh mikrokontroler ESP32 melalui filter Madgwick pada saat proses akuisisi data. Data direkam dengan frekuensi *sampling* sebesar 50

Hz (interval 0,02 detik) dan disimpan dalam format CSV sebagai data mentah (*raw data*) untuk selanjutnya diproses pada tahap *pre-processing*.

3.4.3 Durasi Pengambilan Data

Pengambilan data dilakukan di bengkel las dengan skenario aktivitas yang dibagi menjadi dua kondisi utama, yaitu kondisi aktif saat *welder* melakukan proses pengelasan, dan kondisi tidak aktif saat *welder* menunggu, melakukan persiapan, atau berada pada jeda kerja. Pola gerakan yang direkam mencakup gerakan kontinu dan terarah pada kondisi aktif, serta gerakan minimal atau tidak terarah pada kondisi tidak aktif. Setiap sesi pengambilan data mencakup perekaman keenam kelas aktivitas secara berurutan pada satu subjek. Perekaman tiap aktivitas dilakukan secara bertahap dengan durasi sekitar 50 detik hingga 1 menit, mengikuti satu siklus penggunaan elektroda hingga habis. Proses perekaman diulang beberapa kali hingga diperoleh durasi efektif sekitar 10 menit untuk setiap kelas aktivitas. Dengan frekuensi *sampling* sebesar 50 Hz, durasi tersebut menghasilkan sekitar 30.000 sampel data mentah per kelas, sehingga total keseluruhan data mentah yang terkumpul dari enam kelas aktivitas adalah sekitar 180.000 sampel, sebelum dilakukan proses segmentasi (*windowing*) pada tahap *pre-processing*.

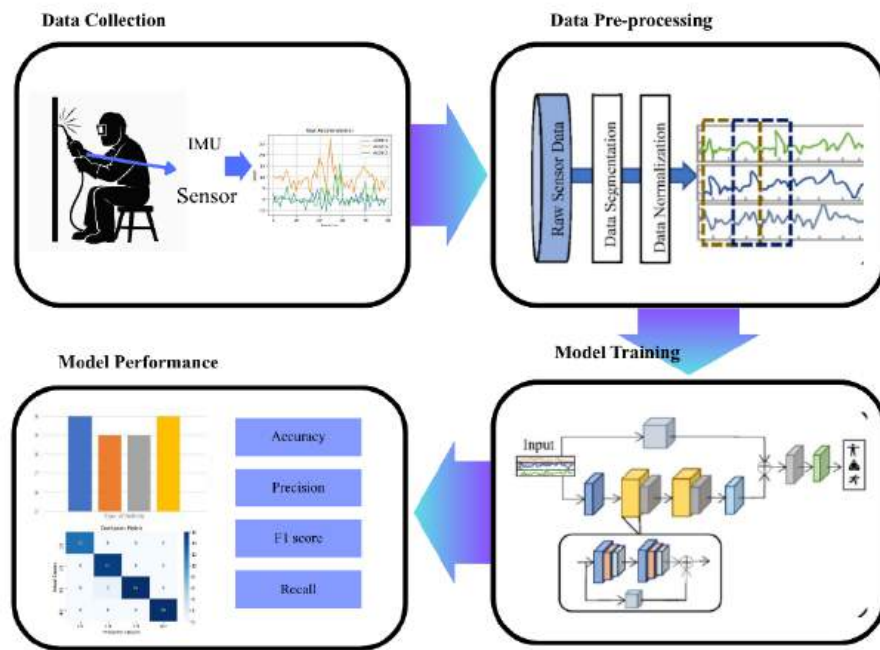
3.4.4 Periode Pengambilan Data

Pengambilan data dilakukan secara bertahap di 4 bengkel las yang berbeda, dengan kedua bengkel dikunjungi sebanyak dua kali dan 1 bengkel sebanyak 1 kali, sehingga total pengambilan data dilakukan dalam 5 kali kunjungan yang melibatkan 4 orang *welder* sebagai subjek penelitian. Pendekatan bertahap ini dilakukan untuk memperoleh variasi pola gerakan *welder* dari lingkungan kerja yang berbeda-beda, sekaligus mengantisipasi kendala teknis seperti gangguan komunikasi data, kehilangan sampel.

3.5 Pengolahan Data IMU

Pengolahan data dan permodelan merupakan tahapan penting dalam sistem klasifikasi aktivitas *welder* karena kualitas data yang dihasilkan sangat memengaruhi kinerja model. Data yang diperoleh dari sensor IMU tidak dapat langsung digunakan, melainkan harus melalui serangkaian proses untuk meningkatkan kualitas sinyal, mengurangi *noise*, serta memastikan konsistensi data

sebelum masuk ke tahap permodelan. Secara umum, tahapan pengolahan data dan permodelan dalam penelitian ini ditunjukkan pada Gambar 3.16:



Gambar 3.16 Ilustrasi Pengolahan Data IMU
(Dokumentasi Pribadi Penulis, 2026)

1. *Data Collection*

Tahap awal dimulai dengan proses pengambilan data gerakan lengan *welder* menggunakan sensor IMU. Sensor ini merekam data *accelerometer*, *gyroscope*, *magnetometer*, dan *quaternion* secara kontinu selama aktivitas pengelasan berlangsung. Data yang diperoleh masih berupa sinyal mentah raw data dari masing-masing sumbu sensor.

2. *Data Pre-processing*

Data hasil sensor *fusion* masuk ke tahap pra-pemrosesan. Pada tahap ini dilakukan beberapa proses, seperti segmentasi data menggunakan *sliding window*, normalisasi, serta pemilihan fitur yang relevan. Tujuan tahap ini adalah menyiapkan data agar sesuai dengan format *input* model pembelajaran mesin.

3. *Model Training*

Data yang telah dipra-proses digunakan sebagai *input* pada tahap pelatihan model, dalam hal ini model CNN 1D. Model dilatih untuk mengekstraksi fitur temporal dari sinyal sensor dan mempelajari pola gerakan yang merepresentasikan aktivitas pengelasan tertentu.

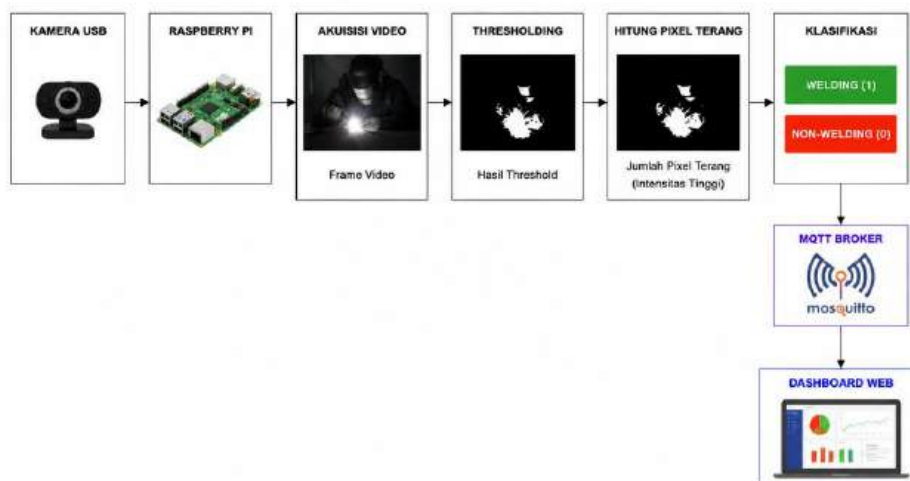
4. Model *Performance* (Evaluasi)

Tahap akhir adalah evaluasi kinerja model menggunakan metrik performa seperti *accuracy*, *precision*, *recall*, dan F1-score. Evaluasi ini bertujuan untuk menilai kemampuan model dalam mengenali dan mengklasifikasikan aktivitas *welder* secara akurat. Lalu, dilakukan analisis terhadap *confusion matrix* untuk melihat distribusi kesalahan klasifikasi pada setiap kelas aktivitas.

Hasil evaluasi ini kemudian digunakan sebagai dasar untuk melakukan perbaikan model agar diperoleh performa yang lebih optimal dan konsisten dalam berbagai kondisi pengujian.

3.6 Pengolahan Data Kamera

Dibawah pada Gambar 3.17 merupakan pengolahan data kamera.



Gambar 3.17 Ilustrasi Pengolahan Data Camera
(Dokumentasi Pribadi Penulis, 2026)

1. Kamera USB

Kamera USB digunakan untuk merekam aktivitas pengelasan secara visual. Kamera diarahkan ke area kerja *welder* untuk menangkap perubahan intensitas cahaya saat proses *welding*.

2. Akuisisi Video *Welder*

Video dari kamera diterima oleh Raspberry Pi sebagai unit pengolahan data visual secara *realtime*.

3. *Thresholding*

Frame video diproses menggunakan *thresholding* untuk memisahkan area cahaya tinggi akibat busur las dari background.

4. Hitung Pixel Terang

Sistem menghitung jumlah pixel putih hasil *thresholding* sebagai indikator intensitas cahaya pengelasan.

5. Klasifikasi *Welding/Non-Welding*

Jumlah pixel terang dibandingkan dengan nilai *threshold*. Jika melewati batas maka dikategorikan *Welding* (1), sedangkan jika tidak maka *Non-Welding* (0).

6. MQTT dan *Dashboard*

Hasil klasifikasi dikirim melalui MQTT menuju *dashboard* web untuk menampilkan status pengelasan secara *realtime*.

3.7 Skenario Pengujian Sistem

Pada tabel 3.3 ini dijelaskan skenario pengujian sistem yang dimulai dari tahap akuisisi sensor IMU serta perekaman video menggunakan kamera USB. Selanjutnya dilakukan proses sinkronisasi antara data IMU dan video. Data yang telah disinkronkan kemudian diolah hingga menghasilkan klasifikasi aktivitas. Tahap terakhir dalam pengujian sistem ini adalah pengujian *dashboard* web yang menampilkan enam aktivitas yang dilakukan oleh *welder*.

Tabel 3.3 Skenario Pengujian Sistem

No	Skenario Pengujian	Tujuan Pengujian	Input	Proses yang Diuji	Output yang Diharapkan	Parameter Evaluasi
1	Pengujian Akuisisi Data IMU	Memastikan sensor IMU dapat merekam data gerakan dengan benar	Gerakan lengan atas <i>welder</i>	Pembacaan sensor oleh ESP32	Data Ax, Ay, Az, Gx, Gy, Gz, Mx, My, Mz terekam	Kelengkapan data, stabilitas <i>sampling</i> 50 Hz

No	Skenario Pengujian	Tujuan Pengujian	Input	Proses yang Diuji	Output yang Diharapkan	Parameter Evaluasi
2	Pengujian Akuisisi Video	Memastikan kamera merekam cahaya dengan baik	Aktivitas <i>welder</i>	Perekaman video oleh Raspberry Pi	File video terekam tanpa drop frame signifikan	Kontinuitas rekaman
3	Pengujian MQTT	Memastikan MQTT dapat menerima data dari ESP32 dan Raspberry Pi	Data IMU dan Kamera	<i>Publish</i> data oleh ESP32 dan Raspberry Pi, <i>Subscribe</i> data oleh laptop	Data diterima laptop secara <i>real-time</i>	Keberhasilan koneksi dan kontinuitas data
4	Pengujian Pre-processing Data IMU	Memastikan data IMU siap digunakan sebagai <i>input</i> model	Data IMU mentah	Filtering, normalisasi, segmentasi	Data tersegmentasi dan ternormalisasi	Tidak ada data hilang, bentuk <i>window</i> sesuai
5	Pengujian Segmentasi Sliding Window	Memastikan pemotongan data sesuai parameter <i>window</i>	Data IMU hasil pre-processing	Sliding <i>window</i>	Potongan data berukuran tetap	Jumlah segmen, overlap sesuai konfigurasi
6	Pengujian Validasi Kamera (Deteksi Cahaya Las)	Memastikan sistem dapat menangkap dan mendeteksi cahaya busur las	Cahaya busur las	Deteksi piksel cahaya/nyala las oleh Raspberry Pi	Label <i>welding</i> 0 atau 1 pada setiap <i>timestamp</i>	Akurasi label dibanding observasi manual
7	Pengujian Klasifikasi CNN	Menguji performa model CNN dalam mengenali aktivitas <i>welder</i>	Data IMU tersegmentasi	Inferensi CNN	Kelas aktivitas <i>welder</i> terprediksi	Akurasi, <i>presisi</i> , <i>recall</i> , <i>confusion matrix</i>
9	Pengujian Dashboard Web	Memastikan data dapat ditampilkan ke <i>user</i>	Data dari server/database	Visualisasi di <i>dashboard</i>	status aktivitas tampil	Kesesuaian data, delay tampilan
10	Pengujian Sistem (Keseluruhan)	Menguji sistem secara keseluruhan	Aktivitas <i>welder</i> nyata	Akuisisi, Sinkronisasi, Klasifikasi, Visualisasi	Prediksi aktivitas tampil di <i>dashboard</i>	Akurasi sistem, <i>latency</i> , stabilitas

(Dokumentasi Pribadi Penulis, 2026)

Pengujian sistem dilakukan untuk memastikan bahwa setiap komponen, mulai dari proses akuisisi data klasifikasi aktivitas *welder*, dapat berjalan sesuai

dengan perancangan. Penyusunan skenario pengujian menjadi penting karena digunakan sebagai acuan untuk mengevaluasi fungsi, kinerja, dan keandalan sistem secara terstruktur, sehingga setiap potensi kesalahan dapat diidentifikasi sejak tahap awal. Skenario pengujian disusun secara sistematis untuk menguji kinerja masing-masing bagian sistem maupun kinerja sistem secara keseluruhan.

(Halaman ini sengaja dikosongkan)

BAB 4

HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil pengujian dan pembahasan seluruh komponen hardware maupun software yang digunakan pada Tugas Akhir ini. Tujuan dari bab ini adalah untuk memberikan kemudahan dalam memahami proses pengujian dan pembahasan yang dilakukan.

4.1 Pengujian Sensor IMU

Pada penelitian Tugas akhir ini sensor IMU yang digunakan adalah sensor ICM-20948. Sensor IMU digunakan untuk mengidentifikasi gerakan lengan atas *welder*. Proses pengujian dilakukan untuk mengetahui tingkat kesesuaian nilai sensor yang dihasilkan oleh sensor IMU ICM-20948 dan sensor IMU *Handphone*. Pembacaan X, Y, dan Z sumbu sensor diuji pada tiga sensor *accelerometer*, *gyroscope*, dan *magnetometer*. Pengujian dilakukan dengan membandingkan nilai hasil pembacaan sensor IMU ICM-20948 dan pembacaan sensor IMU *Handphone*.

Tabel 4.1 Spesifikasi Sensor IMU ICM-20948

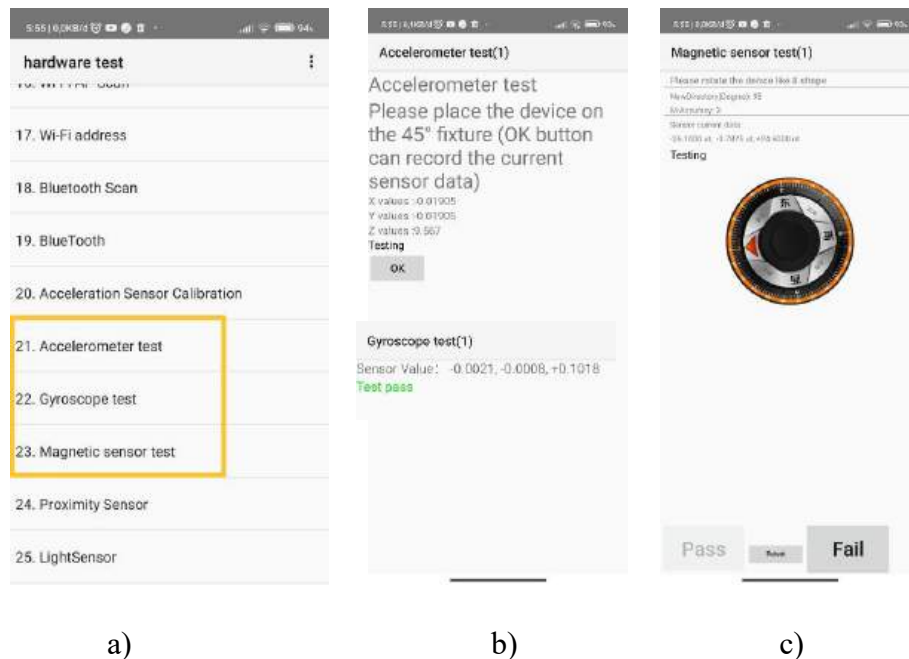
Spesifikasi ICM-20948		
<i>Accelerometer</i>	<i>Gyroscope</i>	<i>Magnetometer</i>
Sumbu Z 1g	X, Y, Z °/s	X, Y, Z μ T

(sumber : www.alldatasheet.com)

Tabel 4.1 menunjukkan spesifikasi sensor IMU ICM-20948 yang diperoleh dari datasheet resmi sensor yang diterbitkan oleh TDK InvenSense dan bersumber dari www.alldatasheet.com. Spesifikasi tersebut mencakup informasi karakteristik operasional dari ketiga sensor yang terdapat di dalam ICM-20948, yaitu *accelerometer*, *gyroscope*, dan *magnetometer*. Pemahaman terhadap spesifikasi sensor ini penting dilakukan sebelum proses pengujian agar dapat diketahui kemampuan dan batasan sensor dalam mengukur gerakan, sehingga hasil pengujian dapat dianalisis secara tepat dan sesuai dengan karakteristik sensor yang digunakan.

Handphone yang digunakan sebagai pembanding pada pengujian ini adalah Android Redmi 10. Sebelum digunakan, dilakukan pengecekan terlebih dahulu untuk memastikan bahwa ketiga sensor *accelerometer*, *gyroscope*, dan

magnetometer tersedia pada *handphone* tersebut. Pengecekan dilakukan dengan membuka aplikasi telepon dan mengetikkan kode *##64663##* pada *keypad*.



Gambar 4.1 Testing sensor a)tiga sensor b)*accelerometer* dan *gyroscope* c)*magnetrometer*

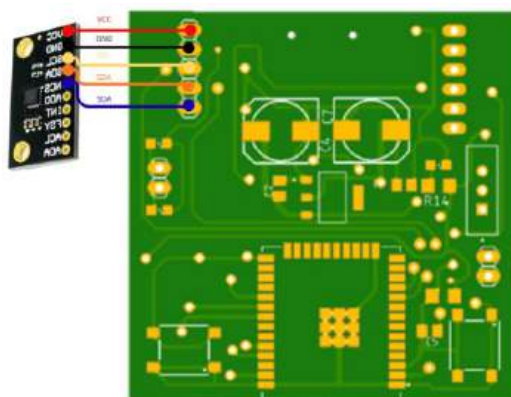
Berdasarkan Gambar 4.1, dapat dikonfirmasi bahwa *handphone* Redmi 10 memiliki ketiga sensor yang dibutuhkan. Sensor *accelerometer* menghasilkan pembacaan dengan satuan m/s^2 , *gyroscope* dalam satuan rad/s , dan *magnetometer* dalam satuan μT . Sebelum dilakukan perbandingan, diperlukan penyesuaian satuan antara kedua sensor karena sensor *accelerometer* ICM-20948 menggunakan satuan g , sedangkan sensor IMU *handphone* menggunakan satuan m/s^2 . Konversi dilakukan dengan membagi nilai *accelerometer* *handphone* dengan konstanta gravitasi $9,80665 \text{ m/s}^2$ sehingga diperoleh satuan g yang seragam dengan ICM-20948. Untuk *gyroscope* dan *magnetometer*, kedua sensor sudah menggunakan satuan yang sama yaitu rad/s dan μT sehingga tidak memerlukan konversi tambahan.

Perbandingan tersebut digunakan untuk mengetahui besar selisih pembacaan sensor dalam bentuk nilai *error*. Perhitungan nilai *error* dilakukan dengan mencari selisih antara nilai sensor IMU ICM-20948 dan nilai sensor IMU *Handphone*, kemudian dibandingkan dengan nilai sensor IMU *Handphone* dan

dikalikan 100% agar diperoleh nilai *error* dalam bentuk presentase. Adapun rumus perhitungan *error* yang digunakan pada persamaan 4.1:

$$Error = \left| \frac{\text{sensor IMU ICM20948} - \text{sensor IMU Handphone}}{\text{sensor IMU Handphone}} \right| \times 100 \quad (4.1)$$

Semakin kecil nilai *error* yang diperoleh, maka semakin kecil pula perbedaan antara hasil pembacaan sensor IMU ICM-20948 dengan pembanding. Dengan demikian, nilai *error* digunakan sebagai dasar untuk menilai tingkat akurasi pembacaan sensor IMU ICM-20948. Untuk mendukung proses pengujian dan pembacaan data sensor, diperlukan perancangan koneksi antar komponen agar komunikasi data dapat berjalan dengan baik. Rangkaian *wiring* ICM-20948 yang digunakan dalam penelitian ini ditunjukkan pada Gambar 4.2 dan penjelasan dapat dilihat pada Tabel 4.2.



Gambar 4.2 Rangkaian *Wiring* ICM-20948
Tabel 4.2 Konektivitas ICM-20948

ICM-20948	ESP32
VCC	3.3V
GND	GND
SCL	IO 26
SDA	IO 25
NCS	3.3V
ADO	3.3V
INT	-
FSY	-
ACL	-
ADA	-

(Dokumentasi Pribadi Penulis, 2026)

Sensor IMU ICM-20948 pada ESP32 diprogram menggunakan *software* Arduino IDE dan menggunakan *library* ICM_20948.h dan proses pembacaan data *accelerometer*, *gyroscope*, dan *magnetometer* dilakukan menggunakan fungsi *myICM.getAGMT()*.

```
void loop() {  
  // STABLE 50Hz  
  while ((micros() - prevMicros) < sampleInterval);  
  prevMicros = micros();  
  
  sampleCount++;  
  if (sampleCount > maxSamples) {  
    Serial.println("3000 DATA FINISHED");  
    while(1);  
  }  
  
  myICM.getAGMT();  
  delay(10);  
}
```

Kode Program 4.1 Fungsi Pembacaan Data

Pada Kode Program 4.1 ditunjukkan proses pembacaan data sensor IMU ICM-20948 menggunakan ESP32 dengan frekuensi *sampling* sebesar 50 Hz. Pada bagian awal program, terdapat perintah *while ((micros() - prevMicros) < sampleInterval);* yang berfungsi untuk menjaga interval pembacaan data agar tetap stabil setiap 20 ms. Selanjutnya, nilai waktu terbaru disimpan kembali ke dalam variabel *prevMicros* menggunakan fungsi *micros()*, sehingga proses pembacaan sensor dapat dilakukan secara periodik dan konsisten sesuai frekuensi *sampling* yang telah ditentukan.

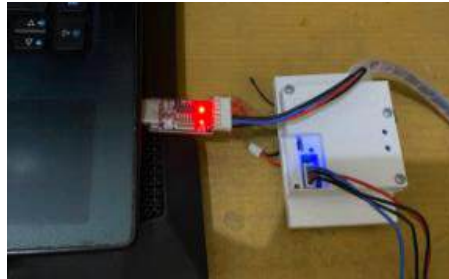
Pada program tersebut juga terdapat variabel *sampleCount* yang digunakan untuk menghitung jumlah data yang telah berhasil direkam. Ketika jumlah data melebihi *MaxSamples*, program akan menampilkan pesan "3000 DATA FINISHED" melalui serial monitor dan menghentikan proses pembacaan data menggunakan *while(1);*. Mekanisme ini digunakan agar jumlah data yang direkam sesuai dengan kebutuhan pengambilan *dataset*. Selanjutnya, proses pembacaan data sensor dilakukan menggunakan fungsi *myICM.getAGMT()*; dari *library* ICM_20948.h. Fungsi ini digunakan untuk mengambil data *accelerometer*, *gyroscope*, dan *magnetometer* secara bersamaan dari sensor ICM-20948. Setelah data berhasil dibaca, program memberikan jeda singkat menggunakan *delay(10);* untuk menjaga kestabilan komunikasi data sebelum proses dilakukannya kalibrasi.

4.1.1 Kalibrasi *Bias* Sensor IMU ICM-20948

Kalibrasi sensor merupakan tahap penting yang dilakukan sebelum proses pengambilan data untuk memastikan bahwa nilai yang dihasilkan oleh sensor ICM-20948 mendekati nilai sebenarnya dan meminimalkan *error* sistematis yang disebabkan oleh *bias* sensor. Proses kalibrasi dilakukan pada ketiga sensor yang terdapat di dalam ICM-20948 yaitu *accelerometer*, *gyroscope*, dan *magnetometer*, dengan metode dan prosedur yang berbeda untuk masing-masing sensor.

1. Kalibrasi *Accelerometer*

Kalibrasi *accelerometer* dilakukan dengan metode *bias removal* yaitu menghitung nilai *mean* pembacaan sensor saat berada dalam kondisi diam di atas permukaan yang rata seperti Gambar 4.3. Pada kondisi diam, *accelerometer* seharusnya membaca nilai gravitasi sebesar 1 g pada sumbu yang tegak lurus terhadap permukaan bumi, sedangkan dua sumbu lainnya seharusnya mendekati 0 g. Nilai *mean* yang diperoleh dari sejumlah sampel saat sensor diam inilah yang disebut sebagai nilai *bias accelerometer* yang selanjutnya dikurangkan dari data pembacaan sensor agar hasil pengukuran menjadi lebih akurat.



Gambar 4.3 Kalibrasi *Accelerometer*
(Dokumentasi Pribadi Penulis, 2026)

Proses kalibrasi *accelerometer* dilakukan selama ± 1 menit dengan frekuensi *sampling* 50 Hz dalam kondisi sensor diam total. Nilai *bias* dihitung menggunakan persamaan 4.2:

$$accBias_{x,y,z} = \frac{1}{N} \sum_{i=1}^N a_{x,y,z_i} \quad (4.2)$$

Dimana :

- $accBias$ = nilai *bias accelerometer* pada masing-masing sumbu

- N = jumlah sampel kalibrasi
- a_x, a_y, a_z = nilai pembacaan *accelerometer* mentah (g)
- -1 pada sumbu Z untuk menghilangkan komponen gravitasi

Setelah nilai *bias* diperoleh, koreksi diterapkan pada setiap pembacaan sensor menggunakan persamaan 4.3:

$$a_{x,y,z,cal} = a_{x,y,z} - accBias_{x,y,z} \quad (4.3)$$

Hasil Offset kalibrasi *accelerometer* ditunjukkan pada Tabel 4.3 berikut:

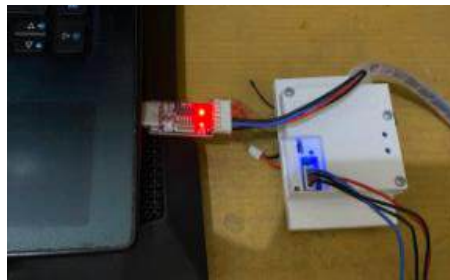
Tabel 4.3 Hasil Offset Kalibrasi *Accelerometer*

Parameter	Nilai
<i>accBiasX</i>	-0.060674 g
<i>accBiasY</i>	-0.039010 g
<i>accBiasZ</i>	0.023039 g

(Dokumentasi Pribadi Penulis, 2026)

2. Kalibrasi *Gyroscope*

Kalibrasi *gyroscope* dilakukan dengan metode *zero-rate offset* yaitu menghitung nilai rata-rata pembacaan *gyroscope* saat sensor dalam kondisi diam total tanpa adanya gerakan rotasi seperti pada Gambar 4.4.



Gambar 4.4 Kalibrasi *Gyroscope*
(Dokumentasi Pribadi Penulis, 2026)

Pada kondisi ideal, *gyroscope* seharusnya menghasilkan nilai 0 deg/s pada ketiga sumbu saat sensor tidak bergerak. Namun pada kenyataannya, terdapat nilai *offset* atau *bias* yang menyebabkan pembacaan *gyroscope* tidak tepat nol meskipun sensor diam. Nilai offset inilah yang dihilangkan melalui proses kalibrasi. Sama seperti kalibrasi *accelerometer*, proses kalibrasi *gyroscope* dilakukan dengan

mengambil 3000 sampel data selama ± 1 menit dalam kondisi sensor diam total. Nilai *bias gyroscope* dihitung menggunakan persamaan 4.4:

$$gyroBias_{x,y,z} = \frac{1}{N} \sum_{i=1}^N g_{x,y,z_i} \quad (4.4)$$

Dimana:

- *gyroBias* = nilai *bias gyroscope* pada masing-masing sumbu
- N = jumlah sampel kalibrasi = 3000
- g_x, g_y, g_z = nilai pembacaan *gyroscope* mentah (deg/s)

Setelah nilai *bias* diperoleh, koreksi diterapkan pada setiap pembacaan sensor menggunakan persamaan 4.5:

$$g_{x,y,z_{cal}} = g_{x,y,z} - gyroBias_{x,y,z} \quad (4.5)$$

Hasil Offset kalibrasi *accelerometer* ditunjukkan pada Tabel 4.4 berikut:

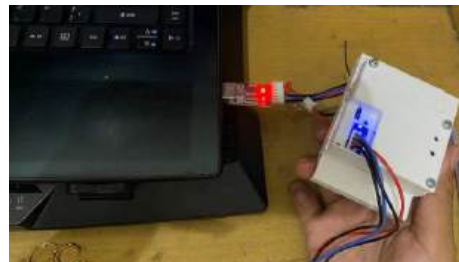
Tabel 4.4 Hasil Offset Kalibrasi *Gyroscope*

Parameter	Nilai
<i>gyroBiasX</i>	-0.409750 deg/s
<i>gyroBiasY</i>	-0.765565 deg/s
<i>gyroBiasZ</i>	0.122384 deg/s

(Dokumentasi Pribadi Penulis, 2026)

3. Kalibrasi *Magnetometer*

Kalibrasi *magnetometer* dilakukan menggunakan dua metode koreksi yaitu *hard-iron correction* dan *soft-iron correction*. *Hard-iron correction* dilakukan untuk menghilangkan *offset* yang disebabkan oleh distorsi magnetik permanen di sekitar sensor seperti komponen elektronik dan material ferromagnetik pada PCB. *Soft-iron correction* dilakukan untuk menormalisasi skala pembacaan setiap sumbu agar membentuk distribusi yang sferis (*spherical*) dan tidak berbentuk elipsoid.



Gambar 4.5 Kalibrasi *Magnetometer*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.5 menunjukkan proses kalibrasi *magnetometer* yang dilakukan dengan menggerakkan sensor ke seluruh arah tiga dimensi membentuk pola *figure-8* selama ± 1 menit dengan 3000 sampel. Selama proses ini, nilai maksimum dan minimum pada setiap sumbu dicatat untuk menghitung nilai koreksi. Nilai *hard-iron offset* dihitung menggunakan persamaan 4.6:

$$magOffset_{x,y,z} = \frac{mag_{x,y,z_{max}} + mag_{x,y,z_{min}}}{2} \quad (4.6)$$

Sedangkan *soft-iron scale factor* dihitung menggunakan persamaan 4.7-4.9:

$$range_{x,y,z} = mag_{x,y,z_{max}} - mag_{x,y,z_{min}} \quad (4.7)$$

$$angRange \frac{range_x + range_y + range_z}{3} \quad (4.8)$$

$$magScale_{x,y,z} \frac{avgRange}{range_{x,y,z}} \quad (4.9)$$

Koreksi kemudian diterapkan pada setiap pembacaan *magnetometer* menggunakan persamaan 4.10:

$$m_{x,y,z_{cal}} = (m_{x,y,z} - magOffset_{x,y,z}) \times magScale_{x,y,z} \quad (4.10)$$

Hasil Offset kalibrasi *magnetometer* ditunjukkan pada Tabel 4.5 berikut:

Tabel 4.5 Hasil Offset Kalibrasi *Magnetometer*

Parameter	Nilai
magOffsetX	-11.174999 μ T
magOffsetY	12.675000 μ T
magOffsetZ	-60.825001 μ T
magScaleX	0.8010
magScaleY	0.8885
magScaleZ	1.5973

(Dokumentasi Pribadi Penulis, 2026)

4. Stabilisasi Filter Madgwick

Setelah proses kalibrasi ketiga sensor selesai dilakukan, program ESP32 menjalankan proses stabilisasi filter Madgwick selama 20 detik sebelum pengambilan data dimulai. Proses stabilisasi ini diperlukan karena filter Madgwick

membutuhkan waktu konvergensi untuk menghasilkan estimasi orientasi *quaternion* yang stabil dan akurat. Selama 20 detik tersebut, sensor harus dalam kondisi diam agar filter dapat menginisialisasi orientasi referensi dengan benar serta meminimalkan error orientasi pada tahap awal pembacaan data. Setelah proses stabilisasi selesai, sensor siap digunakan untuk pengambilan data dan program secara otomatis mulai merekam data ke SD card. Dengan adanya tahap stabilisasi ini data yang diperoleh akan lebih stabil.

4.1.2 Hasil Pengujian Sensor *Accelerometer*

Pengujian sensor *accelerometer* dilakukan untuk mengetahui tingkat akurasi pembacaan sensor ICM-20948 yang terpasang pada ESP32 dibandingkan dengan sensor IMU yang terdapat pada *handphone*. Sensor *accelerometer* memiliki tiga sumbu pembacaan, yaitu sumbu X, Y, dan Z, yang digunakan untuk mendeteksi perubahan percepatan dan posisi selama aktivitas pengelasan berlangsung. Pada pengujian ini, nilai pembacaan dari sensor IMU pada *handphone* sebagai referensi. Perbandingan dilakukan untuk mengetahui tingkat kesesuaian hasil pembacaan sensor ICM-20948 setelah proses kalibrasi dilakukan. Data hasil pembacaan pada setiap sumbu kemudian dihitung nilai *error*-nya untuk mengetahui perbedaan.



Gambar 4.6 Pengambilan Data Parsial Acc
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.6 ditunjukkan proses pengambilan data sensor ICM-20948 dan sensor IMU *handphone* yang dilakukan secara bersamaan untuk memastikan kedua sensor merekam data pada kondisi dan waktu yang sama. Proses sinkronisasi dilakukan selama 20 detik agar data yang diperoleh dapat dibandingkan secara valid. Pengambilan data diawali dengan menjalankan program ESP32 yang melakukan stabilisasi filter Madgwick selama 20 detik dengan indikator countdown pada Serial Monitor. Setelah proses stabilisasi selesai, perekaman data IMU

handphone dimulai dan kedua sensor merekam gerakan pada kondisi yang sama. Data hasil pengukuran kemudian digunakan untuk analisis dan perhitungan error pembacaan sensor.

Tabel 4.6 Uji Sensor *Accelerometer*

No	Sensor IMU ICM-20948			Sensor IMU Handphone			Error		
	Ax	Ay	Az	Ax	Ay	Az	Ax	Ay	Az
1	0.00251	0.02113	0.99995	0.00256	0.02114	0.99228	1.72%	0.04%	0.72%
2	0.00382	0.01315	0.99979	0.00390	0.01312	0.99218	2.10%	0.24%	0.76%
3	0.00155	0.01820	0.99881	0.01555	0.01822	0.99402	0.39%	0.11%	0.48%
4	0.00186	0.01901	0.99881	0.00185	0.01919	0.99238	0.70%	0.94%	0.61%
5	0.00382	0.01592	0.99979	0.00372	0.01598	0.99402	2.49%	0.38%	0.58%
6	0.01364	0.01608	0.99816	0.01361	0.01606	0.99472	0.06%	0.15%	0.35%
7	0.01331	0.01787	0.99962	0.01333	0.01785	0.99259	0.10%	0.34%	0.71%
8	0.01348	0.01950	0.99930	0.01342	0.01914	0.99106	0.04%	0.13%	0.83%
9	0.01885	0.02145	0.99849	0.01882	0.02145	0.98984	0.16%	0.01%	0.87%
10	0.01852	0.02145	0.99881	0.01852	0.02182	0.99259	0.00%	1.70%	0.63%
Error Rata-Rata							0.78%	0.40%	0.65%

Berdasarkan hasil pembacaan pada Tabel 4.6, pengujian sensor *accelerometer* ICM-20948 terhadap sensor *accelerometer* handphone menghasilkan nilai error rata-rata sebesar 0,78% pada sumbu X, 0,40% pada sumbu Y, dan 0,65% pada sumbu Z. Nilai pembacaan *accelerometer* ICM-20948 pada sumbu Z berada pada rentang 0,998 g hingga 1,000 g, mendekati nilai gravitasi normal sebesar 1 g, sedangkan sensor pembanding memiliki rentang pembacaan 0,984 g hingga 0,993 g. Perbedaan pembacaan dipengaruhi oleh karakteristik sensor dan algoritma kalibrasi internal masing-masing perangkat. Secara keseluruhan, nilai error di bawah 1% menunjukkan bahwa proses kalibrasi sensor ICM-20948 telah berjalan dengan baik dan sensor dapat digunakan untuk mendukung sistem HAR pada aktivitas *welder*.

4.1.3 Hasil Pengujian Sensor *Gyroscope*

Pengujian sensor *gyroscope* dilakukan untuk mengetahui tingkat akurasi pembacaan sensor ICM-20948 dalam mengukur kecepatan sudut dibandingkan dengan sensor IMU yang terdapat pada *handphone*. Sensor *gyroscope* memiliki tiga sumbu pembacaan, yaitu sumbu X, Y, dan Z, yang digunakan untuk mendeteksi perubahan orientasi dan gerakan rotasi selama aktivitas pengelasan berlangsung.

Pada pengujian ini, nilai pembacaan *gyroscope* dari sensor ICM-20948 dibandingkan dengan nilai pembacaan dari Sensor IMU pada *handphone* sebagai referensi. Perbandingan dilakukan untuk mengetahui tingkat kesesuaian hasil pembacaan sensor ICM-20948 setelah proses kalibrasi *bias gyroscope* dilakukan. Data hasil pembacaan pada setiap sumbu kemudian dihitung nilai *error*-nya untuk mengetahui besar perbedaan antara sensor IMU ICM-20948 dan sensor IMU *handphone*.



Gambar 4.7 Pengambilan Data Parsial Gyro
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.7 ditunjukkan proses pengambilan data sensor ICM-20948 dan sensor IMU *handphone* yang dilakukan secara bersamaan untuk memastikan kedua sensor merekam data pada kondisi dan waktu yang sinkron. Sinkronisasi dilakukan selama 20 detik agar data yang dibandingkan merepresentasikan kondisi yang sama sehingga hasil perhitungan error menjadi valid. Proses dimulai dengan menyalakan sensor ICM-20948 dan menjalankan program ESP32 yang secara otomatis melakukan stabilisasi filter Madgwick selama 20 detik disertai *countdown* pada Serial Monitor. Setelah proses stabilisasi selesai, perekaman data pada sensor IMU *handphone* mulai dilakukan. Pengambilan data dilakukan pada kondisi sensor yang sama agar hasil pembacaan kedua sensor dapat dibandingkan secara akurat. Data yang diperoleh kemudian digunakan untuk proses analisis dan perhitungan nilai *error* masing-masing sensor.

Tabel 4.7 Uji Sensor *Gyroscope*

No	Sensor IMU ICM-20948			Sensor IMU Handphone			Error		
	Gx	Gy	Gz	Gx	Gy	Gz	Gx	Gy	Gz
1	0.00082	0.00013	0.00013	0.00082	0.00013	0.00013	0.12%	0.07%	0.36%
2	0.00055	0.00082	0.00082	0.00055	0.00082	0.00082	0.36%	0.00%	0.00%
3	0.00082	0.00041	0.00615	0.00082	0.00041	0.00618	0.12%	0.12%	0.79%
4	0.00082	0.00041	0.00279	0.00082	0.00041	0.00275	0.36%	0.12%	0.25%
5	0.00068	0.00027	0.00233	0.00068	0.00027	0.00233	0.15%	0.36%	0.23%
6	0.00082	0.00013	0.00388	0.00082	0.00013	0.00385	0.00%	0.36%	0.93%
7	0.00041	0.00027	0.00041	0.00041	0.00027	0.00041	0.12%	0.36%	0.36%
8	0.00027	0.00027	0.00206	0.00027	0.00027	0.00206	0.36%	0.72%	0.12%
9	0.00013	0.00411	0.00139	0.00013	0.00041	0.00137	0.36%	0.17%	1.74%
10	0.00041	0.00411	0.00082	0.00041	0.00041	0.00082	0.36%	0.17%	0.00%
Error Rata-Rata							0.23%	0.25%	0.48%

Berdasarkan hasil perhitungan yang ditunjukkan pada Tabel 4.7 di atas, diperoleh nilai *error rata-rata gyroscope* pada sumbu X sebesar 0,23%, pada sumbu Y sebesar 0,25%, dan pada sumbu Z sebesar 0,48%. Seluruh nilai error berada di bawah 0,5%, yang merupakan tingkat akurasi yang sangat baik untuk sensor *gyroscope* dalam aplikasi HAR. Nilai *gyroscope* pada kedua sensor menunjukkan angka yang sangat kecil dan berdekatan di seluruh sampel, yang mencerminkan kondisi sensor dalam keadaan relatif diam selama pengambilan data. Pada sampel ke-9, terdapat nilai error yang sedikit lebih besar pada sumbu Z yaitu sebesar 1,74%, yang kemungkinan disebabkan oleh getaran kecil atau perubahan posisi sesaat selama proses perekaman data berlangsung. Namun, nilai tersebut masih berada dalam batas toleransi dan tidak menunjukkan penyimpangan yang signifikan terhadap sensor pembanding. Secara keseluruhan, nilai error rata-rata *gyroscope* yang sangat kecil mengkonfirmasi bahwa kalibrasi *bias gyroscope* ICM-20948 telah berhasil meminimalkan *zero-rate offset* sehingga pembacaan sensor mendekati nilai referensi sensor IMU handphone dan sensor layak digunakan untuk sistem HAR pada aktivitas pengelasan.

4.1.4 Hasil Pengujian Sensor *Magnetometer*

Pengujian sensor *magnetometer* dilakukan untuk mengetahui tingkat akurasi pembacaan sensor ICM-20948 dalam mengukur medan magnet

dibandingkan dengan sensor IMU yang terdapat pada *handphone*. Sensor *magnetometer* memiliki tiga sumbu pembacaan, yaitu sumbu X, Y, dan Z, yang digunakan untuk mendeteksi besar medan magnet di sekitar perangkat selama aktivitas pengelasan berlangsung. Pada pengujian ini, nilai pembacaan *magnetometer* dari sensor ICM-20948 dibandingkan dengan nilai pembacaan dari Sensor IMU pada *handphone* sebagai referensi. Perbandingan dilakukan untuk mengetahui tingkat kesesuaian hasil pembacaan sensor ICM-20948 setelah proses kalibrasi dilakukan. Data hasil pembacaan pada setiap sumbu kemudian dihitung nilai *error*-nya untuk mengetahui besar perbedaan antara sensor ICM-20948 dan sensor IMU *handphone*.



Gambar 4.8 Pengambilan Data Parsial Magne
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.8 ditunjukkan proses pengambilan data sensor ICM-20948 dan sensor IMU *handphone* yang dilakukan secara bersamaan untuk memastikan kedua sensor merekam data pada kondisi dan waktu yang sinkron. Sinkronisasi dilakukan selama 20 detik agar data yang dibandingkan merepresentasikan kondisi yang sama sehingga hasil perhitungan *error* menjadi valid. Proses dimulai dengan menyalakan sensor ICM-20948 dan menjalankan program ESP32 yang secara otomatis melakukan stabilisasi filter Madgwick selama 20 detik disertai *countdown* pada Serial Monitor. Setelah proses stabilisasi selesai, perekaman data pada sensor IMU *handphone* mulai dilakukan. Pengambilan data dilakukan pada kondisi sensor

yang sama agar hasil pembacaan kedua sensor dapat dibandingkan secara akurat. Data yang diperoleh kemudian digunakan untuk proses analisis dan perhitungan nilai *error* masing-masing sensor.

Tabel 4.8 Uji Sensor *Magnetometer*

No	Sensor IMU ICM-20948			Sensor IMU Handphone			Error		
	Mx	My	Mz	Mx	My	Mz	Mx	My	Mz
1	19.3653	29.9042	12.5797	19.3875	29.8875	12.6562	0.11%	0.06%	0.60%
2	19.8977	29.9042	14.1744	19.3875	29.8875	14.1562	2.63%	0.06%	0.13%
3	19.4984	25.4490	14.1744	19.3875	25.3875	14.2875	0.57%	0.20%	0.08%
4	19.3653	25.9848	13.8200	19.3875	25.9875	13.8262	0.11%	0.01%	0.15%
5	18.9660	25.9848	13.8200	18.9562	25.9875	13.8262	0.05%	0.01%	0.15%
6	19.4984	24.7347	13.8200	19.4925	24.7875	13.8262	0.03%	0.21%	0.15%
7	19.6315	25.9848	13.8200	19.4925	24.9562	13.8562	0.71%	0.48%	0.41%
8	18.8329	25.2704	13.8200	18.8437	25.2562	13.8562	0.06%	0.06%	0.41%
9	18.6998	25.9848	13.8200	18.6562	25.8562	13.8562	0.23%	0.28%	0.41%
10	18.6998	25.6276	13.5542	18.6562	25.6125	13.5000	0.23%	0.06%	0.40%
Error Rata-Rata							0.47%	0.14%	0.29%

Berdasarkan hasil perhitungan yang ditunjukkan pada Tabel 4.8 di atas, diperoleh nilai error rata-rata *magnetometer* pada sumbu X sebesar 0,47%, pada sumbu Y sebesar 0,14%, dan pada sumbu Z sebesar 0,29%. Seluruh nilai *error* berada di bawah 0,5%, yang menunjukkan tingkat akurasi yang sangat baik untuk sensor *magnetometer* dalam aplikasi HAR. Nilai *magnetometer* ICM-20948 pada sumbu X berkisar antara 18,70 μ T hingga 19,90 μ T, sedangkan sensor IMU *handphone* membaca nilai antara 18,66 μ T hingga 19,49 μ T. Kedekatan nilai antara kedua sensor pada ketiga sumbu mengkonfirmasi bahwa kalibrasi *hard-iron correction* dan *soft-iron correction* pada sensor ICM-20948 telah berhasil diterapkan dengan baik sehingga pembacaan medan magnet total sangat mendekati nilai referensi sensor IMU *handphone*. Pada sampel ke-2 terdapat nilai *error* sumbu X yang sedikit lebih besar yaitu sebesar 2,63%, yang kemungkinan disebabkan oleh adanya gangguan medan elektromagnetik sesaat di sekitar area pengambilan data. Secara keseluruhan, nilai *error* rata-rata *magnetometer* yang berada di bawah 0,5% menunjukkan bahwa sensor ICM-20948 yang telah dikalibrasi layak digunakan untuk sistem HAR pada aktivitas pengelasan.

4.1.5 Pengujian Nilai Orientasi *Quaternion*

Madgwick filter merupakan algoritma sensor *fusion* yang digunakan untuk menggabungkan data *accelerometer*, *gyroscope*, dan *magnetometer* guna memperoleh orientasi dalam bentuk *quaternion*. Algoritma madgwick bekerja dengan mengkombinasikan data *gyroscope* sebagai orientasi prediksi dan *accelerometer* data serta *magnetometer* sebagai koreksi terhadap *error* akumulasi. *Quaternion* dipilih sebagai representasi orientasi karena mampu menghindari permasalahan gimbal lock yang sering terjadi pada representasi sudut *euler*, serta memberikan kestabilan numerik yang lebih baik dalam proses komputasi. Persamaan update *quaternion* pada madgwick filter ditunjukkan pada persamaan 4.11 dan persamaan 4.12:

$$\hat{q}t = \hat{q}t - 1 + \dot{\hat{q}} \cdot \Delta t \quad (4.11)$$

$$\dot{\hat{q}} = \frac{1}{2} \hat{q}t - 1 \otimes \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} - \beta \frac{\nabla f}{\|\nabla f\|} \quad (4.12)$$

Dimana:

- $\hat{q}t$ = *quaternion* orientasi pada waktu t
- $\omega_x, \omega_y, \omega_z$ = kecepatan sudut *gyroscope* (rad/s)
- β = gain filter untuk mengontrol kecepatan konvergensi
- ∇f = gradien fungsi objektif
- Δt = interval waktu *sampling* (0,02 detik untuk 50Hz)

Proses perhitungan *quaternion* dilakukan secara otomatis oleh *library MadgwickAHRS* pada program ESP32 selama pengambilan data. Hasil *quaternion* terdiri dari empat komponen Q_w, Q_x, Q_y, Q_z kemudian disimpan bersama data sensor lainnya ke dalam file CSV.

Sebagai ilustrasi penerapan persamaan 4.11 dan 4.12, berikut contoh perhitungan satu langkah *update quaternion* menggunakan data *gyroscope* pada Tabel 4.7 baris ke-2, yaitu $G_x = 0,00055$ rad/s, $G_y = 0,00082$ rad/s, dan $G_z =$

0,00082 rad/s, dengan asumsi *quaternion* awal $\hat{q}(t-1)$ merupakan *quaternion* identitas tanpa rotasi untuk menunjukkan mekanisme dasar perhitungan:

$$\hat{q}(t-1) = [1, 0, 0, 0]$$

Langkah 1 Membentuk *quaternion* kecepatan sudut *gyroscope*

Data *gyroscope* dinyatakan dalam bentuk *quaternion* murni bagian skalar nol:

$$\omega = [0, \omega_x, \omega_y, \omega_z] = [0; 0,00055; 0,00082; 0,00082]$$

Langkah 2 Menghitung turunan *quaternion* terhadap waktu persamaan 4.12, komponen *gyroscope*

$$\dot{\hat{q}} = \frac{1}{2} \cdot \hat{q}(t-1) \otimes \omega$$

Karena $\hat{q}(t-1)$ merupakan *quaternion* identitas, hasil perkalian *quaternion* ($\otimes$) disederhanakan menjadi:

$$\begin{aligned}\dot{\hat{q}} &= 1/2 \cdot [0; 0,00055; 0,00082; 0,00082] \\ \dot{\hat{q}} &= [0; 0,000275; 0,00041; 0,00041]\end{aligned}$$

Langkah 3 Update *quaternion* persamaan 4.11

Dengan interval waktu *sampling* $\Delta t = 0,02$ detik (sesuai frekuensi 50 Hz):

$$\begin{aligned}\hat{q}_t &= \hat{q}(t-1) + \dot{\hat{q}} \cdot \Delta t \\ \hat{q}_t &= [1; 0; 0; 0] + [0; 0,000275; 0,00041; 0,00041] \times 0,02 \\ \hat{q}_t &= [1,0000000; 0,0000055; 0,0000082; 0,0000082]\end{aligned}$$

Hasil perhitungan menunjukkan bahwa dalam satu langkah waktu *sampling* (20 ms), *quaternion* mengalami perubahan yang sangat kecil sesuai dengan kecepatan sudut *gyroscope* yang terukur pada saat itu. Perubahan kecil ini konsisten dengan karakteristik data *gyroscope* yang merepresentasikan kecepatan sudut sesaat, sehingga estimasi orientasi diperbarui secara bertahap pada setiap siklus *sampling*.

Perlu ditekankan bahwa nilai *quaternion* yang tercatat pada Tabel 4.9 $Q_w = 0,26743$; $Q_x = 0,00438$; $Q_y = 0,00532$; $Q_z = 0,96355$ pada baris ke-2 merupakan

hasil estimasi akumulatif dari filter Madgwick yang telah berjalan sejak inisialisasi sistem, termasuk periode *warmup* selama 20 detik sebelum pengambilan data dimulai.

Oleh karena itu, nilai *quaternion* pada suatu baris data tidak dapat direkonstruksi secara pasti hanya dari satu sampel data *gyroscope* pada waktu tersebut, melainkan merupakan hasil iterasi berkelanjutan yang dikombinasikan dengan koreksi dari *accelerometer* dan *magnetometer* pada setiap siklus *sampling* melalui suku $\beta \frac{v_f}{\|v_f\|}$ pada persamaan 4.12.

Perhitungan pada langkah 1 hingga 3 di atas mendemonstrasikan mekanisme satu langkah update berbasis *gyroscope* sebagai bagian dari proses tersebut. Validitas hasil estimasi orientasi *quaternion* selanjutnya dibuktikan melalui perbandingan dengan sensor referensi IMU *handphone* pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian *Quaternion*

	Sensor IMU ICM-20948				Sensor IMU Handphone				Error			
	Qw	Qx	Qy	Qz	Qw	Qx	Qy	Qz	Qw	Qx	Qy	Qz
1	0.37 493	0.00 579	0.00 532	0.99 993	0.37 538	0.00 574	0.00 533	0.99 999	0.12 %	1.01 %	0.01 %	0.00 %
2	0.26 743	0.00 438	0.00 532	0.96 355	0.26 757	0.00 434	0.00 533	0.96 932	0.01 %	1.24 %	0.02 %	0.31 %
3	0.26 710	0.00 419	0.00 487	0.96 355	0.26 948	0.04 195	0.00 489	0.96 918	0.09 %	0.00 %	0.04 %	0.39 %
4	0.26 548	0.00 411	0.00 490	0.96 355	0.26 565	0.00 416	0.00 498	0.96 903	0.00 %	0.99 %	0.20 %	0.31 %
5	0.25 971	0.00 392	0.00 496	0.96 355	0.25 989	0.00 412	0.00 498	0.96 882	0.06 %	4.71 %	0.20 %	0.32 %
6	0.25 628	0.00 366	0.00 504	0.96 355	0.25 798	0.00 361	0.00 506	0.96 843	0.06 %	1.63 %	0.02 %	0.31 %
7	0.25 079	0.00 397	0.00 504	0.96 355	0.25 030	0.00 394	0.00 506	0.96 827	0.06 %	0.99 %	0.10 %	0.32 %
8	0.25 548	0.00 418	0.00 474	0.96 355	0.25 414	0.00 415	0.00 475	0.96 809	0.06 %	0.82 %	0.02 %	0.32 %
9	0.25 520	0.00 395	0.00 476	0.96 355	0.25 222	0.00 399	0.00 475	0.96 789	0.12 %	1.08 %	0.02 %	0.32 %
10	0.26 179	0.00 401	0.00 474	0.96 355	0.26 181	0.00 401	0.00 475	0.96 679	0.01 %	0.02 %	0.04 %	0.32 %
Error Rata- Rata									0.06 %	1.25 %	0.07 %	0.29 %

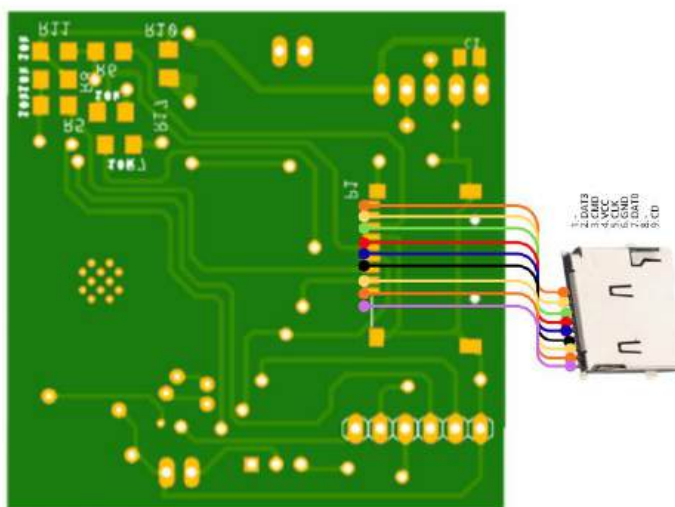
(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil perhitungan, diperoleh nilai *error* rata-rata *quaternion* pada Tabel 4.9 Qw sebesar 0,06%, Qx sebesar 1,25%, Qy sebesar 0,07%, dan Qz sebesar 0,29%. Nilai *error* yang berada di bawah 2% menunjukkan bahwa data

orientasi *quaternion* dari sensor ICM-20948 memiliki tingkat akurasi yang baik dibandingkan sensor IMU *handphone* sebagai referensi. Nilai *quaternion* pada kedua sensor menunjukkan pola yang hampir sama pada setiap sampel pengujian. Kedekatan nilai tersebut menunjukkan bahwa implementasi Madgwick Filter pada sensor ICM-20948 mampu menghasilkan estimasi orientasi yang stabil dan akurat. Secara keseluruhan, hasil pengujian menunjukkan bahwa sensor ICM-20948 layak digunakan pada sistem *Human Activity Recognition* (HAR) berbasis *wearable* karena mampu menghasilkan data orientasi yang mendekati sensor referensi *handphone*.

4.2 Pengujian SD Card

Pada penelitian ini, SD Card digunakan untuk menyimpan data *accelerometer*, *gyroscope*, *magnetometer*, serta *quaternion* dalam format CSV secara kontinu. Data yang tersimpan pada SD card selanjutnya digunakan sebagai *dataset* utama dalam tahapan pengolahan data, meliputi proses *filtering*, normalisasi, *sliding window*, hingga pelatihan model CNN untuk klasifikasi aktivitas *welder*. Penggunaan SD card memungkinkan data sensor tersimpan dalam jumlah besar secara portabel secara langsung tanpa bergantung pada komputer selama pengambilan data, sehingga memudahkan proses pemindahan dan pengolahan data pada komputer. Rangkaian *wiring* SD Card dapat dilihat pada Gambar 4.9 dan penjelasan konfigurasi pin ditunjukkan pada Tabel 4.10.



Tabel 4.10 Konektivitas SD Card

SD Card	ESP32
VDD	3.3V
CD	GND
CLK	IO 18
CMD	IO 23
DAT0	IO 19
DAT1	-
DAT2	-
DAT3	IO 5
VSS	GND
GND	GND

(Dokumentasi Pribadi Penulis, 2026)

SD Card pada ESP32 diprogram menggunakan *software* Arduino IDE dengan bantuan *library* *SD.h* dan *SPI.h*. Proses inialisasi dan komunikasi SD card dilakukan menggunakan fungsi *SD.begin(SD_CS_PIN)*, sedangkan proses pembuatan dan penyimpanan file CSV dilakukan menggunakan fungsi *SD.open()* dan *dataFile.println()*. Pada program ini, SD card digunakan untuk menyimpan data *accelerometer*, *gyroscope*, *magnetometer*, *quaternion*, serta orientasi sensor dalam format CSV secara otomatis selama proses pengambilan *dataset* berlangsung. Selain itu, program juga menggunakan mekanisme *buffer* penyimpanan melalui variabel *sdBuffer[]* untuk mengurangi frekuensi penulisan langsung ke SD card sehingga proses penyimpanan data menjadi lebih stabil dan efisien. Hasil Pengujian SD Card dapat dilihat pada Gambar 4.10.

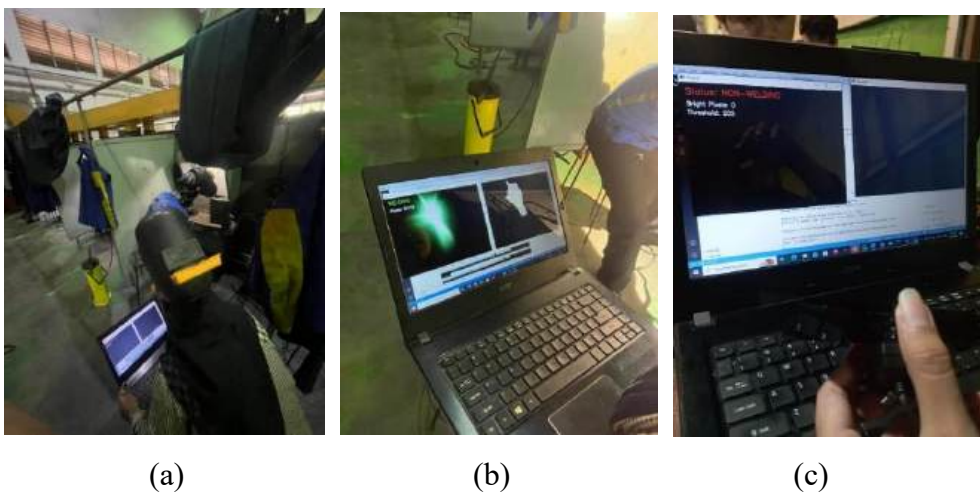
time	seconds	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz
1.76E+18	-0.22084	0.001808	-0.00096	-0.00117	-0.0088	0.011	0.002475	-12.4875	-45.9563	-5.5125	1	-0.00015	-0.00044	0.000466
1.76E+18	0.175211	0.001808	-0.00096	-0.00117	-0.0088	0.011	0.002475	-12.0563	-46.9125	-4.2	0.999999	-0.00031	-0.00087	0.00094
1.76E+18	0.216008	-0.02039	0.0064	-0.00877	-0.01128	0.003575	-0.00481	-12.1125	-46.7438	-4.44375	0.999999	-0.00012	-0.00048	0.001433
1.76E+18	0.257603	0.019752	0.005793	-0.02116	-0.01719	0.011275	-0.00468	-12.3563	-46.7438	-4.55625	0.999998	2.56E-05	-0.00088	0.001929
1.76E+18	0.299201	0.058663	0.004652	-0.05832	-0.02874	0.025438	-0.00784	-12.2438	-46.6125	-4.74375	0.999996	9.96E-05	-0.00131	0.002423
1.76E+18	0.340804	0.028152	-0.00986	-0.11949	-0.04015	0.041662	-0.01444	-12.2438	-46.6125	-4.85625	0.999994	0.000105	-0.00153	0.003039
1.76E+18	0.382406	-0.01958	0.003541	-0.07654	-0.0605	0.062837	-0.02695	-12.1875	-46.3875	-5.15625	0.999992	0.000195	-0.00144	0.003682
1.76E+18	0.424008	0.016317	0.004006	-0.03754	-0.0759	0.0671	-0.03506	-12.0563	-46.3125	-5.45625	0.999989	0.000306	-0.00175	0.00424
1.76E+18	0.465615	0.012074	0.004593	-0.04101	-0.08497	0.06215	-0.03919	-11.8875	-46.1438	-5.45625	0.999986	0.000427	-0.002	0.004823
1.76E+18	0.507222	-0.00475	0.00507	-0.08059	-0.0792	0.05225	-0.03644	-11.6438	-46.2563	-5.4	0.999983	0.000545	-0.00204	0.005462
1.76E+18	0.548832	0.006131	0.018927	-0.05418	-0.00083	0.000825	-0.00083	-11.5125	-46.2563	-5.45625	0.999979	0.000814	-0.00219	0.006045
1.76E+18	0.59043	0.003617	0.018672	-0.03501	-0.00083	0.000825	-0.00083	-11.2875	-46.2563	-5.4	0.999975	0.001149	-0.00232	0.006597
1.76E+18	0.632024	-0.01243	0.017697	-0.01657	-0.0484	0.033	-0.01691	-11.2125	-46.2563	-5.45625	0.999971	0.001538	-0.00215	0.007095
1.76E+18	0.673635	-0.04163	0.002754	0.041418	-0.04799	0.026812	-0.01622	-11.0438	-46.2563	-5.64375	0.999968	0.001623	-0.00182	0.007655
1.76E+18	0.715239	-0.02936	0.015184	0.080006	-0.05239	0.023375	-0.0187	-11.0438	-46.3875	-5.4	0.999964	0.001863	-0.00152	0.008189
1.76E+18	0.756837	-0.00246	0.014001	-0.03818	-0.066	0.01815	-0.02461	-11.2125	-46.5563	-5.15625	0.999958	0.002133	-0.00156	0.008779
1.76E+18	0.798443	-0.0157	0.026539	-0.0262	-0.06242	0.0132	-0.02365	-11.3438	-46.5563	-5.34375	0.999953	0.002525	-0.00143	0.00928
1.76E+18	0.840054	-0.01556	0.011805	0.013544	-0.00083	0.000825	-0.00083	-11.5875	-46.6125	-5.7	0.999947	0.002847	-0.00116	0.009788
1.76E+18	0.881661	-0.00227	0.009889	-0.00069	-0.00591	0.000825	0.002063	-11.6438	-46.6875	-5.64375	0.999942	0.00334	-0.00112	0.010224
1.76E+18	0.923259	-0.00275	0.00856	-0.04144	0.011962	-0.00261	0.010175	-11.8125	-46.6125	-5.75625	0.999934	0.003556	-0.00117	0.010847
1.76E+18	0.964856	-0.01712	0.007441	-0.0269	0.0374	-0.00743	0.020488	-11.8125	-46.4438	-5.64375	0.999927	0.003773	-0.00095	0.011433
1.76E+18	1.006453	-0.03155	0.020241	0.015197	0.058162	-0.01416	0.028462	-11.8125	-46.2563	-5.45625	0.99992	0.00409	-0.00063	0.011923
1.76E+18	1.048051	-0.01855	0.019531	0.002481	0.068963	-0.01801	0.032313	-11.8125	-46.3875	-5.75625	0.999913	0.004501	-0.00037	0.012386
1.76E+18	1.089645	-0.00533	0.005114	0.01642	0.067787	-0.0209	0.033	-11.7563	-46.3875	-6.05625	0.999904	0.004765	-0.00028	0.012995

Gambar 4.10 Hasil Pengujian SD Card

Hasil pengujian SD Card menunjukkan bahwa sistem berhasil menyimpan data sensor IMU ICM-20948 secara kontinu dalam format CSV yang terdiri dari data *accelerometer*, *gyroscope*, *magnetometer*, *quaternion*. Secara keseluruhan, hasil pengujian menunjukkan bahwa modul SD Card mampu berfungsi dengan baik sebagai media penyimpanan data sensor secara lokal dan terstruktur sesuai kebutuhan penelitian.

4.3 Pengujian Kamera

Pengujian kamera USB dilakukan dengan mengambil tangkapan cahaya yang dihasilkan dari proses pengelasan menggunakan kamera yang dikenali melalui program python. Tujuan utama dari pengujian untuk menentukan *setpoint* intensitas cahaya yang akan digunakan sebagai parameter acuan dalam sistem. Selain menentukan *setpoint*, pengujian juga dilakukan dengan mengintegrasikan kamera pada raspberry Pi untuk memastikan bahwa kamera dapat berfungsi dengan baik, pada proses pengujian, kamera dilengkapi dengan kaca las sebagai filter optik yang sangat diperlukan untuk mengurangi gangguan dari cahaya lingkungan sekitar dan mereduksi intensitas cahaya berlebihan yang masuk pada kamera. Penggunaan kaca las sebagai filter optik sangat diperlukan karena proses pengelasan menghasilkan cahaya dengan intensitas yang sangat tinggi dan dapat merusak kamera apabila cahaya tersebut masuk secara langsung tanpa adanya perlindungan. Selain itu, kaca las dapat melindungi kamera dari panas las.

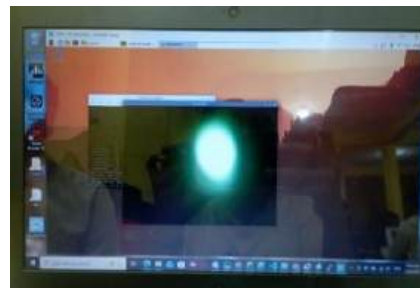


Gambar 4.11 a) Pengambilan Tangkapan Cahaya Las b) Tampilan Ketika Cahaya Las terdeteksi
c) Tampilan Ketika Cahaya Las Tidak Terdeteksi

Pada Gambar 4.11 a) ditunjukkan proses pengambilan tangkapan cahaya las menghitung durasi waktu sejak *welder* mulai melakukan pengelasan. hingga benar-benar mengelas Pengukuran waktu ini bertujuan untuk memastikan bahwa data citra yang diambil benar-benar merepresentasikan kondisi saat pengelasan berlangsung, Berdasarkan hasil pengujian, diperoleh waktu sebesar 4.52 detik hingga *welder* benar-benar mengelas. Pada Gambar 4.11 b) adalah tampilan yang dihasilkan ketika *welder* melakukan pengelasan. Kamera menangkap cahaya las dengan status *Welding* dan Bright 26.105. lalu Pada Gambar 4.11 c) adalah tampilan ketika kamera tidak menangkap cahaya apapun dengan status *Non-*.



(a)



(b)

Gambar 4.12 a) Raspberry Pi yang Tersambung dengan Kamera
b) Tampilan kamera SSH Raspberry Pi

Pada Gambar 4.12 a) ditunjukkan rangkaian Raspberry Pi disambungkan dengan Power Suplai bertegangan 5V dan arus 3A untuk memastikan suplai daya yang stabil selama proses pengujian berlangsung. Kamera USB juga dihubungkan langsung ke Raspberry Pi sebagai *input* untuk menangkap cahaya las. Selanjutnya, pada Gambar 4.12 b) ditampilkan hasil visualisasi berupa tampilan cahaya yang diakses melalui koneksi SSH ketika Raspberry Pi telah berhasil terhubung dengan laptop. Tampilan yang dihasilkan menunjukkan bahwa sistem telah berjalan dengan baik, dimana cahaya dapat terdeteksi dengan jelas. Hal ini menandakan bahwa proses komunikasi telah terintegrasi dan siap digunakan.

4.4 Pengujian Baterai

Pengujian ini bertujuan untuk mengetahui waktu operasi aktual dari kedua baterai pada kondisi penggunaan sebenarnya, sehingga diperoleh gambaran nyata mengenai daya tahan di lapangan.

a. Pengujian Baterai ESP32

Pengujian baterai ESP32 dilakukan dengan menjalankan sistem ESP32 mengirimkan data sensor IMU secara kontinu melalui protokol MQTT, terhitung sejak baterai terisi penuh hingga sistem berhenti mengirim data akibat baterai habis. Selama pengujian, ESP32 bekerja terus-menerus tanpa jeda, meliputi pembacaan sensor, pengiriman data ke broker MQTT, sehingga kondisi pengujian merepresentasikan beban kerja aktual sistem. Waktu berakhir dicatat saat sistem berhenti mengirimkan data, yang mengindikasikan baterai telah mencapai kondisi habis atau batas tegangan minimum (*cut-off*), sehingga diperoleh durasi operasi aktual sistem ESP32 dalam kondisi kerja kontinu.

Berdasarkan Tabel 4.11, kelima pengujian menunjukkan hasil yang konsisten, dengan durasi operasi aktual sebesar 24 jam 20 menit pada seluruh pengujian. Tegangan awal baterai berkisar antara 4,00 V hingga 4,10 V, dan tegangan akhir saat sistem mati berkisar antara 3,20 V hingga 3,33 V. Konsistensi durasi operasi pada kelima pengujian menunjukkan bahwa performa baterai ESP32 cukup stabil dan dapat diandalkan untuk pemakaian berulang. Berdasarkan hasil pengujian tersebut, arus rata-rata yang ditarik oleh sistem ESP32 dapat dihitung menggunakan persamaan 4.13:

$$I = \frac{Q}{t} \quad (4.13)$$

Keterangan:

I: arus rata-rata yang ditarik sistem (mA)

Q: kapasitas baterai (mAh)

t: lama waktu baterai bertahan (jam)

Kapasitas: 4000 mAh

Waktu bertahan: 24 jam 20 menit = $24 + \frac{20}{60} = 24,33$ jam

$$I = \frac{4000mAh}{24,33 jam} = 163,27mA$$

Tabel 4.11 Hasil Pengujian Waktu Operasi Baterai ESP32

Pengujian Ke-	Tegangan Awal (V)	Waktu Mulai	Waktu Sistem Mati	Tegangan Akhir (V)	Durasi Operasi Aktual
1	 4,10V	08.00	20.35	 3,33V	24 jam 20 menit
2	 4,04V	08.15	20.43	 3,33V	24 jam 20 menit
3	 4,0V	08.30	21.02	 3,30V	24 jam 20 menit
4	 4.0V	08.20	20.56	 3.20V	24 jam 20 menit

Pengujian Ke-	Tegangan Awal (V)	Waktu Mulai	Waktu Sistem Mati	Tegangan Akhir (V)	Durasi Operasi Aktual
5	 4,04V	08.10	20.42	 3,33V	24 jam 20 menit

b. Pengujian Baterai Raspberry Pi

Pengujian Baterai Raspberry Pi dilakukan dengan menjalankan sistem Raspberry Pi beserta kamera USB dalam kondisi aktif melakukan proses *monitoring* dan mengirim data *brightness* ke MQTT, terhitung sejak baterai terisi penuh hingga sistem mati.

Berdasarkan Tabel 4.12, kelima pengujian menunjukkan hasil yang konsisten, dengan durasi operasi aktual sebesar 2 jam 19 menit pada seluruh pengujian. Tegangan awal baterai berkisar antara 12,04 V hingga 12,09 V, dan tegangan akhir saat sistem mati berkisar antara 4,40 V hingga 4,57 V. Hasil ini menunjukkan bahwa baterai Raspberry Pi memiliki waktu operasi yang jauh lebih singkat dibandingkan ESP32, sejalan dengan kebutuhan daya yang lebih tinggi akibat beban kerja Raspberry Pi dan kamera USB yang berjalan secara bersamaan.

Berdasarkan hasil pengujian tersebut, arus rata-rata yang ditarik oleh sistem Raspberry Pi dapat dihitung menggunakan persamaan 4.14:

$$I = \frac{Q}{t} \quad (4.14)$$

Keterangan:

I: arus rata-rata yang ditarik sistem (mA)

Q: kapasitas baterai (mAh)

t: lama waktu baterai bertahan (jam)

Kapasitas: 1200 mAh

Waktu bertahan: 2 jam 19 menit = $2 + \frac{19}{60} = 2,32$ jam

$$I = \frac{1200mAh}{2,32 jam} = 517,99 mA$$

Tabel 4.12 Hasil Pengujian Waktu Operasi Baterai Raspberry Pi

Pengujian Ke-	Tegangan Awal (V)	Waktu Mulai	Waktu Sistem Mati	Tegangan Akhir (V)	Durasi Operasi Aktual
1	 12,07 V	08.00	09.19	 4,51V	2 jam 19 menit
2	 12,06V	08.15	10.34	 4.57V	2 jam 19 menit
3	 12,09V	08.30	10.49	 4,43V	2 jam 19 menit

Pengujian Ke-	Tegangan Awal (V)	Waktu Mulai	Waktu Sistem Mati	Tegangan Akhir (V)	Durasi Operasi Aktual
4	 12,04V	08.20	10.39	 4,40V	2 jam 19 menit
5	 12,09 V	08.10	10.29	 4,44V	2 jam 19 menit

4.5 Berat Keseluruhan Komponen

Pengujian berat keseluruhan komponen dilakukan untuk mengetahui total massa perangkat *wearable* yang digunakan pada penelitian. Pengukuran dilakukan menggunakan timbangan digital setelah seluruh komponen elektronik dipasang pada jaket *welder*. Hasil pengukuran ini digunakan untuk mengevaluasi apakah perangkat masih nyaman digunakan selama aktivitas pengelasan dan tidak mengganggu pergerakan pengguna. Hasil pengukuran berat masing-masing komponen ditunjukkan pada Tabel 4.13.

Tabel 4.13 Berat Keseluruhan Komponen

No	Komponen	Jumlah	Berat (gram)
1	PCB + Baterai	1	 57 gram

No	Komponen	Jumlah	Berat (gram)
2	Raspberry Pi + Baterai	1	 367 gram
3	Kamera	1	 32 gram
4	Kabel	1	 33 gram
5	Jaket	1	 1000 gram
Total			489 gram

Berdasarkan Tabel 4.13, total berat jaket *welder* 1000 gram dan total berat perangkat *wearable* yang dipasang pada lengan atas *welder* adalah 57 gram. Berat tersebut merupakan akumulasi dari seluruh komponen elektronik, meliputi ESP32,

sensor IMU, baterai Li-Po, modul *charger* dan *casing* pelindung. Lalu total berat yang dipasang pada saku bawah *welder* adalah 400 gram. Dan total berat kamera yang dipasang di dada *welder* adalah 32 gram. Dengan pembagian berat secara merata tersebut, perangkat masih tergolong ringan sehingga tidak memberikan beban yang signifikan maupun mengganggu kenyamanan serta fleksibilitas gerak *welder* selama proses pengambilan data dan pengelasan berlangsung.

4.6 Pengujian Komunikasi MQTT

Pengujian komunikasi MQTT dilakukan untuk mengetahui apakah data dari ESP32 dan Raspberry Pi dapat dikirim (*publish*) dan diterima (*subscribe*) oleh laptop melalui MQTT Broker secara *real-time* tanpa kehilangan data.

```
Connecting WiFi .....
WiFi Connected
ESP32 IP: 10.233.207.238
ICM20948 READY
Connecting MQTT...connected
-----
{"timestamp":13.058,"accel_x":0.115234,"accel_y":-0.059570,"accel_z":1.025391,
MQTT CONNECTED : 1
PUBLISH STATUS : 1
MQTT STATE : 0
-----
```

Gambar 4.13 ESP32 *Publish* data ke MQTT

Pada Gambar 4.13 adalah tampilan serial pada Arduino.IDE ketika ESP32 berhasil *connect* ke MQTT dan berhasil *publish* data sensor.

```
welder/sensor {"timestamp":13.058,"accel_x":0.115234,"accel_y":-0.059570,"accel_z":
1.025391,"gyro_x":-0.004263,"gyro_y":-0.008926,"gyro_z":-0.012790,"mag_x":8.400000,
"mag_y":49.950001,"mag_z":-62.099998,"qw":0.004136,"qx":0.006621,"qy":-0.006268,"qz
":0.999950}
welder/sensor {"timestamp":13.159,"accel_x":0.112793,"accel_y":-0.054199,"accel_z":
1.031738,"gyro_x":-0.005063,"gyro_y":0.011724,"gyro_z":-0.031576,"mag_x":12.150000,
"mag_y":52.200001,"mag_z":-63.900002,"qw":0.008611,"qx":0.012941,"qy":-0.012600,"qz
":0.999800}
welder/sensor {"timestamp":13.260,"accel_x":0.100098,"accel_y":-0.043945,"accel_z":
1.020508,"gyro_x":0.011191,"gyro_y":-0.021051,"gyro_z":0.016920,"mag_x":8.100000,"m
ag_y":50.400002,"mag_z":-62.400002,"qw":0.013116,"qx":0.019323,"qy":-0.018817,"qz":
0.999550}
```

Gambar 4.14 Laptop *Subscribe* data dari ESP32

Pada Gambar 4.14 adalah tampilan laptop yang berhasil *subscribe* data dari ESP32 ke MQTT.

```

-----
STATUS      : WELDING
WHITE PIXEL : 4668
CONFIDENCE  : 1.52
FPS         : 7.52
-----

STATUS      : NON-WELDING
WHITE PIXEL : 4592
CONFIDENCE  : 1.49
FPS         : 7.52
-----

```

Gambar 4.15 Raspberry Pi *Publish* data ke MQTT

Pada Gambar 4.15 adalah tampilan serial pada Arduino.IDE ketika ESP32 berhasil *connect* ke MQTT dan berhasil *publish* data sensor, selanjutnya Pada Gambar 4.16 adalah tampilan laptop yang berhasil *subscribe* data dari ESP332.

```

7.69, "timestamp": "2026-07-23T13:23:19.154970"}
elder/camera {"device": "RPI_CAM_01", "status": "WELDING", "label": 1, "white_pixels": 9430, "confidence": 3.07, "fps"
7.69, "timestamp": "2026-07-23T13:23:19.275066"}
elder/camera {"device": "RPI_CAM_01", "status": "WELDING", "label": 1, "white_pixels": 13612, "confidence": 4.43, "fps"
7.69, "timestamp": "2026-07-23T13:23:19.395144"}
elder/camera {"device": "RPI_CAM_01", "status": "WELDING", "label": 1, "white_pixels": 1661, "confidence": 0.54, "fps"
7.7, "timestamp": "2026-07-23T13:23:19.514977"}
elder/camera {"device": "RPI_CAM_01", "status": "WELDING", "label": 1, "white_pixels": 0, "confidence": 0.0, "fps": 7.
"timestamp": "2026-07-23T13:23:19.635295"}
elder/camera {"device": "RPI_CAM_01", "status": "NON-WELDING", "label": 0, "white_pixels": 0, "confidence": 0.0, "fps"
7.7, "timestamp": "2026-07-23T13:23:19.755017"}
elder/camera {"device": "RPI_CAM_01", "status": "NON-WELDING", "label": 0, "white_pixels": 0, "confidence": 0.0, "fps"

```

Gambar 4.16 Laptop *Subscribe* data dari Raspberry Pi

Berdasarkan hasil pengujian komunikasi MQTT pada Gambar 4.13, 4.14, 4.15, dan 4.16. seluruh data yang dikirimkan oleh ESP32 dan Raspberry Pi berhasil diterima oleh laptop melalui broker MQTT. Data IMU dan hasil validasi kamera dapat diproses secara bersamaan tanpa kehilangan data sehingga proses klasifikasi aktivitas dan validasi pengelasan dapat berlangsung secara *real-time*. Hasil pengolahan selanjutnya berhasil disimpan pada MongoDB dan ditampilkan pada *dashboard* web sebagai media *monitoring* aktivitas *welder*.

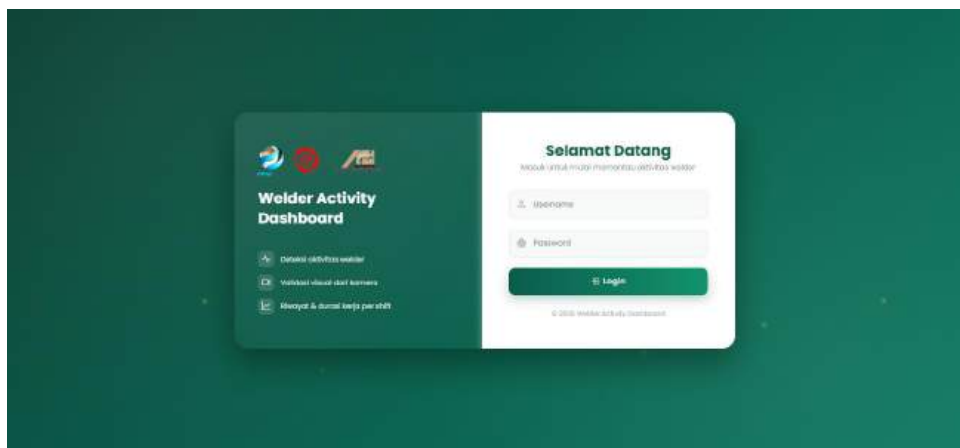
4.7 Pengujian *Dashboard*

Pada tahap ini dilakukan pengujian *Website Welder Activity Dashboard* yang telah dibuat. Pengujian ini dibagi menjadi 2 jenis pengujian yaitu, pengujian sistem *login website* dan pengujian tampilan *homepage dashboard*. Pengujian ini bertujuan untuk memastikan bahwa sistem autentikasi berjalan dengan baik.

4.7.1 Pengujian sistem *login dashboard*

Pada tahap ini, peneliti melakukan pengujian dengan melakukan *login* pada *dashboard* yang telah dibuat. Pengujian dilakukan dengan memasukkan *username* dan *password* yang telah didaftarkan sebelumnya ke dalam sistem, sebagai langkah awal untuk memastikan hanya pihak yang berwenang yang dapat mengakses data pemantauan aktivitas *welder*. Pada halaman ini tidak disediakan menu registrasi mandiri, dikarenakan sistem dirancang hanya dapat diakses oleh supervisor atau kepala bengkel yang akunnya telah didaftarkan terlebih dahulu oleh admin, sehingga kontrol terhadap siapa saja yang berhak mengakses sistem tetap terjaga dan tidak sembarang pihak dapat membuat akun secara bebas.

Proses *login* berhasil dilakukan menggunakan *username* dan *password* yang telah terdaftar pada sistem. Setelah proses autentikasi berhasil, pengguna secara otomatis diarahkan ke halaman *dashboard* tanpa memerlukan langkah tambahan. Sistem juga mencatat sesi pengguna yang sedang aktif, sehingga identitas pengguna yang *login* dapat ditampilkan secara konsisten pada antarmuka *dashboard*. Hasil pengujian menunjukkan bahwa mekanisme autentikasi berjalan sesuai dengan perancangan, sehingga hanya pengguna yang memiliki hak akses dan akun terdaftar yang dapat mengakses sistem, sementara upaya *login* dengan kredensial yang tidak sesuai akan ditolak dan diarahkan kembali ke halaman *login*.

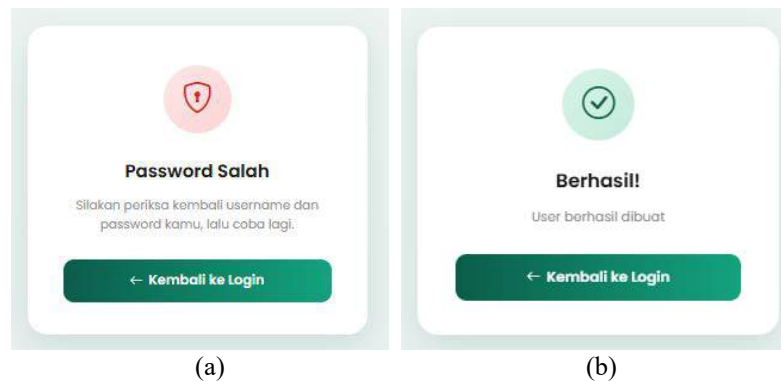


Gambar 4.17 Pengujian Halaman *Login*
(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan Gambar 4.17, proses *login* berhasil dilakukan menggunakan *username* dan *password* yang telah terdaftar pada sistem. Setelah proses autentikasi

berhasil, pengguna secara otomatis diarahkan ke halaman *dashboard*. Hasil pengujian menunjukkan bahwa mekanisme autentikasi berjalan sesuai dengan perancangan sehingga hanya pengguna yang memiliki hak akses yang dapat mengakses sistem.

4.7.2 Pengujian Gagal *Login* dan *User* Berhasil Dibuat



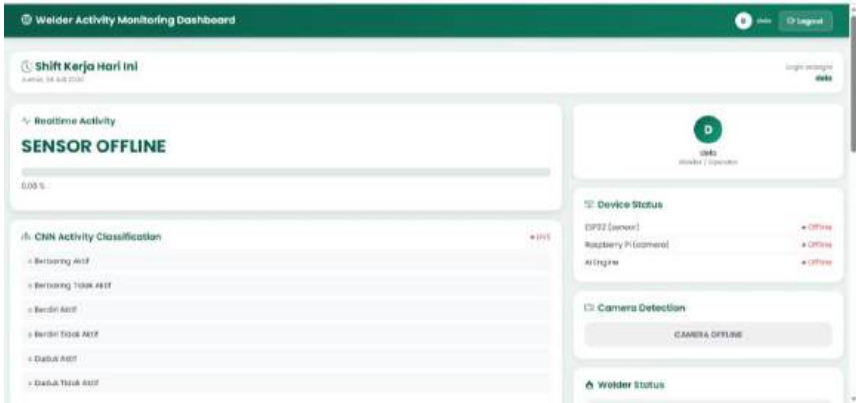
Gambar 4.18 Tampilan Gagal *Login* dan b) *user* berhasil dibuat
(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan Gambar 4.18 a) sistem menolak proses *login* ketika *username* atau *password* yang dimasukkan tidak sesuai dengan data yang tersimpan pada basis data. Sistem kemudian menampilkan pesan kesalahan kepada pengguna dan tetap berada pada halaman *login*. Hal ini menunjukkan bahwa mekanisme validasi autentikasi telah berfungsi dengan baik. Berdasarkan Gambar 4.18 b), proses penambahan pengguna baru berhasil dilakukan oleh admin. Data pengguna yang dimasukkan berhasil tersimpan ke dalam basis data sehingga akun tersebut dapat digunakan untuk melakukan *login* ke dalam sistem. Hasil pengujian menunjukkan bahwa fitur manajemen pengguna telah berjalan sesuai dengan fungsinya.

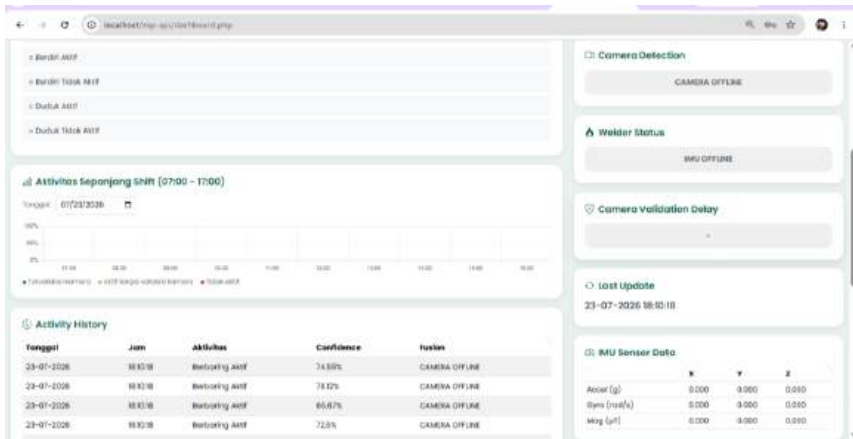
4.7.3 Hasil Pengujian Tampilan *Homepage Supervisor*

Pada tahap ini, peneliti melakukan pengujian tampilan *dashboard* setelah mendapatkan akses *login* untuk memastikan seluruh elemen antarmuka menampilkan informasi secara benar dan sesuai dengan data aktual dari sistem. Gambar 4.19, 4.20, dan 4.21 menunjukkan hasil pengujian tampilan *dashboard* setelah *login*. Pada tampilan tersebut, *dashboard* menampilkan aktivitas *welder* secara *real-time* beserta ID dan Name pengguna yang sedang *login*, sehingga supervisor dapat mengetahui *welder* mana yang sedang dipantau. Status *device* juga

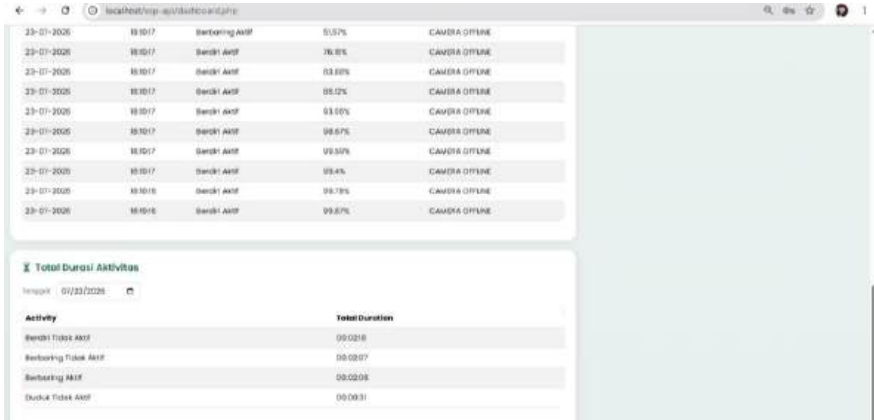
ditampilkan untuk menunjukkan kondisi koneksi masing-masing perangkat (ESP32, Raspberry Pi, dan AI Engine), serta *work duration* yang mencatat lamanya waktu kerja *welder*.



Gambar 4.19 Tampilan *Dashboard* Setelah *Login*
(Dokumentasi Pribadi Penulis, 2026)



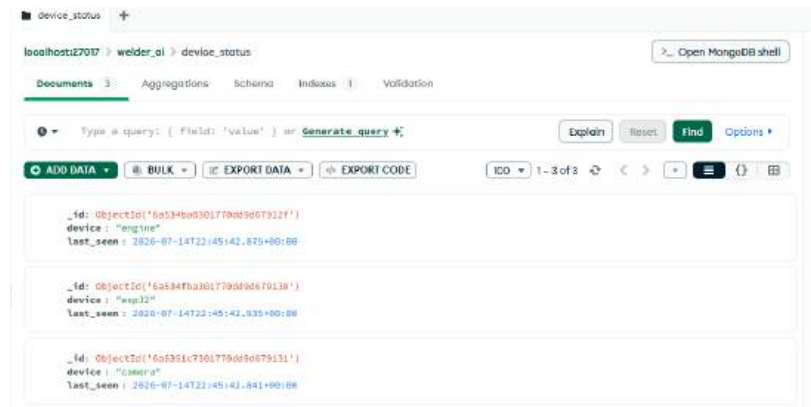
Gambar 4.20 Tampilan *Dashboard* Setelah *Login*
(Dokumentasi Pribadi Penulis, 2026)



Gambar 4.21 Tampilan *Dashboard* Setelah *Login*
(Dokumentasi Pribadi Penulis, 2026)

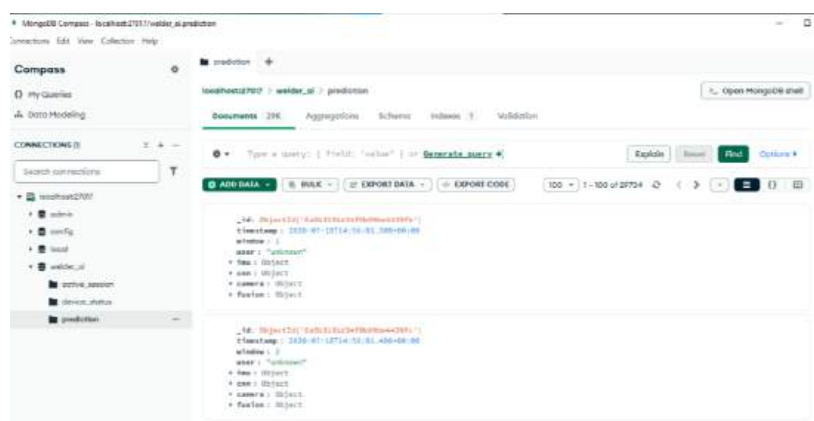
4.7.4 Hasil Pengujian *Database* Mongo DB

Pengujian *database* MongoDB dilakukan untuk memastikan data hasil klasifikasi aktivitas *welder* dan hasil validasi kamera dapat tersimpan dengan baik ke dalam *database* sebelum ditampilkan pada *dashboard* web.



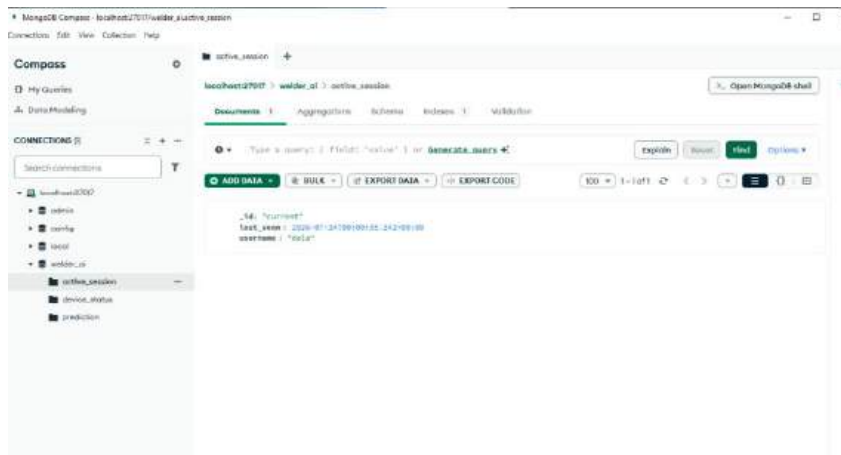
Gambar 4.22 *Database* Menyimpan Status *Device*
(Dokumentasi Pribadi Penulis, 2026)

Gambar 4.22 menunjukkan tampilan data pada MongoDB ketika sistem berhasil menyimpan status perangkat (*device status*). Data tersebut menunjukkan bahwa *engine* Python, ESP32, dan kamera tercatat dalam kondisi *online* atau aktif, yang menandakan seluruh subsistem berhasil terhubung dan berjalan dengan baik secara bersamaan. Penyimpanan status ini berfungsi sebagai indikator pemantauan kondisi sistem, untuk memastikan proses pengambilan dan pengiriman data berjalan tanpa gangguan selama pengujian berlangsung.



Gambar 4.23 *Database* Menyimpan Hasil *Prediction*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.23 tampilan Mongoddb ketika berhasil menyimpan hasil prediksi yang dikirim oleh pembacaan model pada python.



Gambar 4.24 Database Menyimpan Hasil *Active_session*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.24 tampilan MongoDB ketika berhasil menyimpan hasil *active_session* untuk menandakan sensor benar sedang aktif atau tidak.

4.8 Pengujian *Latency* Sistem

Pengujian ini bertujuan untuk mengetahui *latency* pada setiap tahapan alur data sistem, mulai dari proses inferensi model CNN, penyimpanan hasil ke basis data, hingga penyajian data pada *dashboard* web. Pengujian *latency* penting dilakukan untuk memastikan sistem mampu menyajikan informasi aktivitas *welder* secara *near real-time*.

4.8.1 Pengujian *Latency* Inferensi CNN

Pengujian *latency* inferensi dilakukan dengan mencatat waktu yang dibutuhkan model CNN untuk melakukan prediksi terhadap satu *window* data (100 *sample*). Pengujian dilakukan secara kontinu sebanyak 14.866 *window* prediksi.

Tabel 4.14 Hasil Pengujian *Latency* Inferensi CNN

Metrik	Nilai
Jumlah Sampel	14.866 <i>window</i>
Rata-rata	110,58 ms
Standar Daviasi	37,47 ms
Nilai Minimum	78,79 ms
Nilai Maksimum	654,41 ms

Berdasarkan Tabel 4.14, proses inferensi model CNN membutuhkan waktu rata-rata sebesar 110,58 ms dengan standar deviasi 37,47 ms per *window* prediksi. Nilai ini menunjukkan bahwa proses klasifikasi aktivitas *welder* oleh model CNN berlangsung sangat cepat, jauh di bawah durasi satu *window* data 10 detik untuk 100 *sample* dengan interval pengiriman 100 ms, sehingga tidak menjadi *bottleneck* dalam alur pemrosesan data secara keseluruhan. Nilai maksimum yang lebih tinggi 654,41 ms kemungkinan terjadi akibat adanya proses lain pada laptop yang berjalan bersamaan *resource contention*, namun tetap berada dalam rentang waktu yang dapat diterima untuk aplikasi *real-time monitoring*.

4.8.2 Pengujian *Latency* Penyimpanan Basis Data

Pengujian *latency* penyimpanan dilakukan dengan mencatat waktu yang dibutuhkan sistem untuk menyimpan satu dokumen hasil klasifikasi ke dalam basis data MongoDB melalui fungsi *insert_one()*.

Tabel 4.15 Hasil Pengujian *Latency* Penyimpanan MongoDB

Metrik	Nilai
Jumlah Sampel	14.866 <i>window</i>
Rata-rata	1,07 ms
Standar Daviasi	2,94 ms
Nilai Minimum	0,00 ms
Nilai Maksimum	155,29 ms

Berdasarkan Tabel 4.15, proses penyimpanan data ke MongoDB berlangsung sangat cepat, dengan rata-rata hanya 1,07 ms. Hal ini menunjukkan bahwa MongoDB sebagai basis data NoSQL berbasis dokumen mampu menangani proses *insert* data secara efisien pada skala data yang digunakan dalam penelitian ini, sehingga tidak memberikan kontribusi signifikan terhadap total *latency* sistem.

4.8.3 Pengujian *Latency* Penyajian *Dashboard*

Pengujian *latency* penyajian data pada *dashboard* dilakukan dengan mencatat waktu yang dibutuhkan antara PHP selesai memproses *response* hingga

data diterima dan diproses oleh browser (*JavaScript*), pada mekanisme *polling* dengan interval 2 detik.

Tabel 4.16 Hasil Pengujian *Latency* PHP ke Browser

Metrik	Nilai
Jumlah Sampel	343 <i>polling</i>
Rata-rata	1,06 ms
Standar Daviasi	1,85 ms
Nilai Maksimum	28,75 ms

Berdasarkan Tabel 4.16, waktu yang dibutuhkan data untuk berpindah dari server PHP ke browser sangat singkat, dengan rata-rata 1,06 ms. Hal ini wajar karena PHP (*server*) dan browser dijalankan pada satu perangkat yang sama (*localhost*) selama pengujian, sehingga tidak ada *latency* jaringan eksternal yang memengaruhi proses ini

4.9 Penerapan Metode CNN

Simulasi pengolahan data menggunakan metode CNN dilakukan untuk merancang tahapan pengolahan data sensor sebelum digunakan untuk merancang tahapan pengolahan data sensor sebelum digunakan pada proses klasifikasi aktivitas. Tahap ini meliputi *pre-processing* data sebelum akhirnya dilakukan *split* data *training* dan data *test* untuk dilakukan simulasi klasifikasi menggunakan metode *cnn*. Data yang digunakan pada tahap ini berasal dari hasil rekaman sensor *accelesometer*, *gyroscope*, dan *magnetometer* yang telah memiliki label aktivitas serta postur tubuh.

4.9.1 Pengambilan dan Pengumpulan *Dataset*

Dataset merupakan sekumpulan data yang diperoleh dari proses akuisisi sensor dan digunakan sebagai masukan dalam proses pelatihan serta pengujian model klasifikasi aktivitas. Pada penelitian ini, *dataset* diperoleh dari sensor IMU ICM-20948 yang terdiri atas data *accelerometer*, *gyroscope*, *magnetometer*, dan *quaternion* dengan frekuensi pengambilan sampel sebesar 50 Hz. Selama proses pengambilan data, sensor dipasang pada jaket lengan kanan atas yang digunakan

oleh *welder* untuk merekam gerakan tubuh sesuai dengan aktivitas yang dilakukan. *Dataset* yang telah terbentuk terdiri atas enam kelas aktivitas, yaitu duduk aktif, duduk tidak aktif, berdiri aktif, berdiri tidak aktif, berbaring aktif, dan berbaring tidak aktif. Klasifikasi kelas dan jumlah *dataset* yang akan digunakan dapat dilihat pada Tabel 4.17.

Tabel 4.17 Durasi dan Jumlah *Dataset* yang digunakan

Aktivitas	Durasi	Jumlah sampel	File
Duduk Aktif	10 Menit	30.000	CSV
Duduk Tidak Aktif	10 Menit	30.000	CSV
Berdiri Aktif	10 Menit	30.000	CSV
Berdiri Tidak Aktif	10 Menit	30.000	CSV
Berbaring Aktif	10 Menit	30.000	CSV
Berbaring Tidak Aktif	10 Menit	30.000	CSV

Tabel 4.17 menunjukkan hasil pengumpulan *dataset* untuk masing-masing aktivitas. Setiap aktivitas direkam selama 10 menit dengan frekuensi pengambilan sampel sebesar 50 Hz sehingga menghasilkan 30.000 sampel untuk setiap kelas aktivitas. Seluruh data disimpan dalam format CSV sebagai data mentah (*raw data*) yang selanjutnya digunakan pada proses pelabelan dan *pre-processing* sebelum digunakan sebagai masukan pada model CNN. Dibawah ini pada Tabel 4.18 adalah contoh *Raw Data* dalam format CSV.

Tabel 4.18 Contoh *Raw Data* dalam format CSV

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	...	qz
0	0	0.141031	-0.29775	-0.00398	-0.00543	-0.0019	-0.00134	5.78963	...	0.610664
2000000	0.02	0.286944	-0.59616	-0.006	-0.01588	-0.0023	-0.000415	7.253675	...	0.610634
4000000	0.04	0.429602	-0.89993	-0.00444	-0.02086	-0.002418	-0.00433	6.322011	...	0.610611
6000000	0.06	0.429602	-0.90498	-0.005	-0.01273	-0.007214	-0.005751	5.78963	...	0.610625
1.76E+18	0.08752	0.12088	0.16206	0.010002	0.275137	0.229762	-0.19003	-8.94375	...	-0.00014

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	...	qz
1.76E+18	0.16266	0.12088	0.16206	0.010002	0.275137	0.229762	-0.19003	-16.5	...	-0.00039
1.76E+18	0.203438	-0.08668	-0.11558	-0.00594	-0.00894	-0.01196	-0.00055	-16.5563	...	-0.00063
1.76E+18	0.244996	-0.11316	-0.08827	0.172021	0.019662	-0.06078	0.001375	-16.5	...	-0.00096
1.76E+18	0.286557	-0.04346	-0.07499	0.127026	0.017738	-0.07563	0.00385	-16.2	...	-0.00125
1.76E+18	0.328115	-0.06857	-0.06229	-0.00987	0.004813	-0.06628	0.00605	-16.2	...	-0.00152

Tabel 4.18 menunjukkan contoh data mentah (*raw data*) yang dihasilkan dari proses akuisisi data menggunakan sensor IMU ICM-20948, yang disimpan dalam format CSV. Kolom *time* dan *seconds_elapsed* sama-sama merepresentasikan waktu relatif sejak dimulainya sesi perekaman pada mikrokontroler ESP32, dengan satuan yang berbeda *time* dalam *nanosecond*, sedangkan *seconds_elapsed* merupakan hasil konversinya ke satuan detik agar lebih mudah dibaca. Selisih antar baris pada kedua kolom tersebut konsisten sebesar 20.000.000 *nanosecond* atau 0,02 detik, sesuai dengan frekuensi *sampling* sistem sebesar 50 Hz. Setiap baris data merepresentasikan satu titik waktu pengukuran yang mencakup nilai *accelerometer* (*accel_x*, *accel_y*, *accel_z*), *gyroscope* (*gyro_x*, *gyro_y*, *gyro_z*), *magnetometer* (*mag_x*, *mag_y*, *mag_z*), serta data orientasi *quaternion* (*qw*, *qx*, *qy*, *qz*) yang dihitung langsung oleh mikrokontroler ESP32 melalui filter Madgwick. Kolom label menunjukkan kelas aktivitas *welder* pada saat data tersebut direkam, yang digunakan sebagai target klasifikasi pada tahap pelatihan model CNN. Data lengkap disertakan pada bagian lampiran.

4.9.2 Pelabelan *Dataset*

Pelabelan *dataset* merupakan proses pemberian identitas atau kelas pada setiap data hasil akuisisi sensor IMU. Pada penelitian ini, pelabelan dilakukan berdasarkan aktivitas yang diperagakan oleh *welder* selama proses pengambilan data. Setiap kali perekaman dilakukan, hanya satu jenis aktivitas yang diperagakan, sehingga seluruh data yang diperoleh pada rekaman tersebut diberikan label sesuai

aktivitas yang dilakukan. Label yang digunakan terdiri atas enam kelas, yaitu duduk aktif, duduk tidak aktif, berdiri aktif, berdiri tidak aktif, berbaring aktif, dan berbaring tidak aktif. Data yang telah diberi label kemudian disimpan dalam format CSV dan digunakan pada tahapan *pre-processing*. Klasifikasi label yang digunakan pada penelitian ini dapat dilihat pada Tabel 4.19.

Tabel 4.19 Contoh Labeling Kelas Berdiri Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	...	label
0	0	0.141031	-0.29775	-0.00398	-0.00543	-0.0019	...	Berdiri Aktif
20000 000	0.02	0.286944	-0.59616	-0.006	-0.01588	-0.0023	...	Berdiri Aktif
40000 000	0.04	0.429602	-0.89993	-0.00444	-0.02086	0.002418	...	Berdiri Aktif
60000 000	0.06	0.429602	-0.90498	0.005	-0.01273	0.007214	...	Berdiri Aktif
80000 000	0.08	0.423905	-0.90937	0.008418	-0.00092	0.001708	...	Berdiri Aktif
1E+08	0.1	0.424231	-0.90953	0.007279	0.003832	-0.01699	...	Berdiri Aktif
1.2E+0 8	0.12	0.423417	-0.91067	-0.00086	0.001522	-0.03076	...	Berdiri Aktif
1.4E+0 8	0.14	0.425533	-0.91132	-0.0033	0.002544	-0.03431	...	Berdiri Aktif
1.6E+0 8	0.16	0.430904	-0.90628	-0.00525	0.010804	-0.02032	...	Berdiri Aktif
1.8E+0 8	0.18	0.43481	-0.89749	-0.00086	0.012847	-0.00802	...	Berdiri Aktif

4.9.3 Pre-Processing Data

Tahap *pre-processing* bertujuan untuk meningkatkan kualitas data dengan mengurangi *noise* serta menyeragamkan skala nilai setiap fitur sehingga proses pelatihan model CNN menjadi lebih stabil dan akurat. Pada penelitian ini, data berasal dari sensor *accelerometer*, *gyroscope*, *magnetometer* dan *quaternion* diproses melalui tahap pembersihan data, normalisasi menggunakan *Z-score*, dan segmentasi menggunakan *sliding window*. Setelah proses *pre-processing* selesai, data dibagi menjadi data *training* dan *testing* untuk melatih serta mengevaluasi performa model CNN dalam mengklasifikasikan enam aktivitas *welder*.

Filtering

Sebelum dilakukan proses *windowing*, data sensor terlebih dahulu melalui tahap *filtering* untuk mengurangi *noise*. Penyaringan dilakukan menggunakan *low-*

pass filter Butterworth. Tahapan ini penting dilakukan sebelum *windowing* agar setiap segmen data yang dihasilkan memiliki kualitas sinyal yang baik dan konsisten untuk proses pembelajaran model CNN. *Low-pass filter* Butterworth digunakan untuk menghaluskan sinyal *accelerometer*, *gyroscope*, *magnetometer*, dan *quaternion*, dengan fungsi respons magnitude filter ditunjukkan pada Persamaan 4.15.

$$|H(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} \quad (4.15)$$

Dimana:

- $|H(j\omega)|$ = respons magnitude filter
- ω = frekuensi sinyal *input* (rad/s)
- ω_c = frekuensi *cutoff* (rad/s)
- n = orde filter

Frekuensi *cutoff* ternormalisasi dihitung menggunakan persamaan 4.16:

$$W_n \frac{f_c}{f_s/2} \quad (4.16)$$

Dimana:

- f_c = frekuensi *cutoff* = 5Hz
- f_s = frekuensi *sampling* = 50 Hz

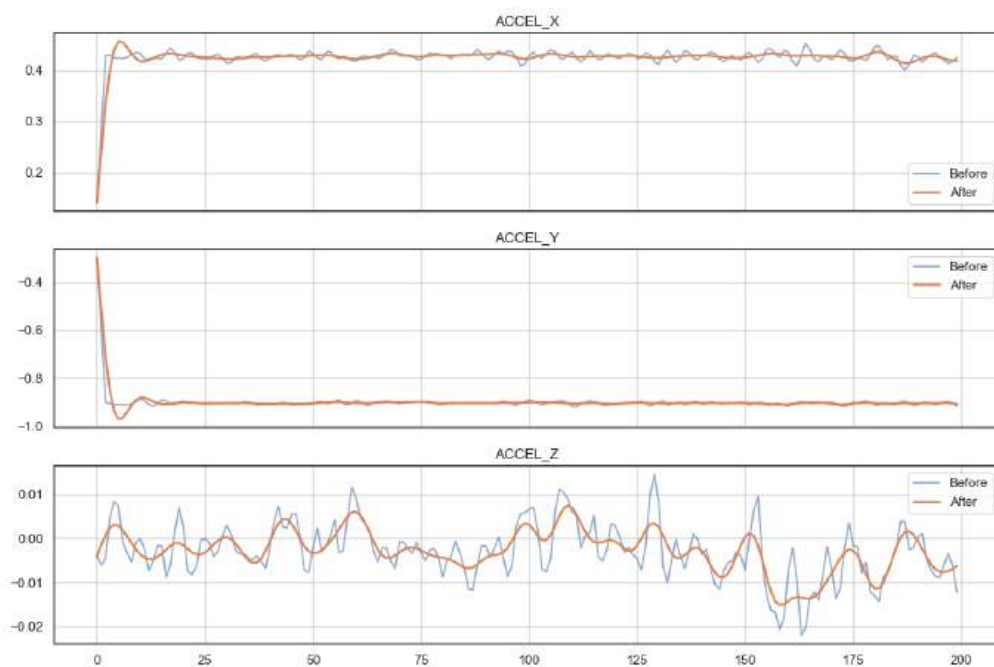
Sehingga nilai frekuensi *cutoff* ternormalisasikan adalah :

$$W_n \frac{5}{50/2} = \frac{5}{25} = 0,2$$

Filter Butterworth *low-pass* orde ke-4 diterapkan dengan frekuensi *cutoff* ternormalisasi sebesar 0,2, diperoleh dari *cutoff* 5 Hz dibagi frekuensi *Nyquist* 25 Hz. Orde ke-4 dipilih karena memberikan *roll-off* lebih tajam pada frekuensi *cutoff*, sehingga meredam *noise* frekuensi tinggi secara efektif tanpa terlalu mendistorsi sinyal gerakan. Penyaringan dilakukan menggunakan fungsi *filtfilt* dari *library scipy.signal*, yang menerapkan filter dua arah (*forward-backward*) sehingga tidak menimbulkan *phase shift*. Pendekatan ini dipilih dibanding filter satu arah seperti

lfilter, karena pergeseran fase pada data *time-series* IMU dapat mengaburkan hubungan temporal antar-sampel yang menjadi dasar klasifikasi aktivitas.

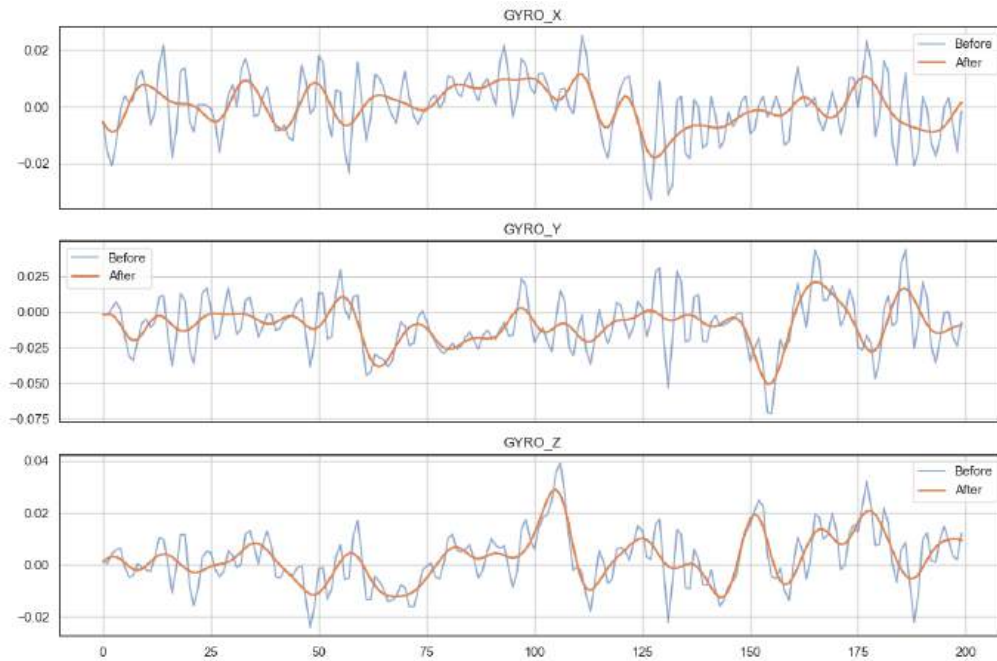
Dengan metode ini, sinyal hasil filtering menjadi lebih halus, komponen *noise* frekuensi tinggi (seperti getaran mekanis atau interferensi elektronik) dapat dikurangi secara signifikan, sementara bentuk maupun waktu (*timing*) pola data tetap sesuai dengan kondisi sinyal asli. Konsistensi temporal ini penting untuk menjaga integritas data sebelum memasuki tahap segmentasi *sliding window* dan proses klasifikasi menggunakan model CNN, sehingga performa model tidak terpengaruh oleh distorsi sinyal akibat proses pra-pemrosesan.



Gambar 4.25 Hasil *Filtering Accelerometer*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.25 menunjukkan perbandingan sinyal *accelerometer* pada sumbu X, Y, dan Z sebelum dan sesudah proses *filtering*. Garis *Before* merepresentasikan data mentah, sedangkan garis *After* menunjukkan data yang telah difilter menggunakan *low-pass filter Butterworth*. Setelah proses *filtering*, terlihat bahwa sinyal menjadi lebih halus dan stabil, dengan *noise* frekuensi tinggi yang berkurang secara signifikan. Meskipun demikian, pola utama gerakan tetap terjaga, sehingga informasi penting dari aktivitas tidak hilang. Hal ini penting karena informasi mengenai bentuk dan tren sinyal bukan sekadar nilai instan

menjadi dasar bagi model CNN dalam mengenali pola aktivitas pada tahap klasifikasi. Dengan berkurangnya *noise*, segmen data yang dihasilkan pada tahap *windowing* selanjutnya diharapkan memiliki kualitas sinyal yang lebih konsisten, sehingga dapat meningkatkan akurasi model dalam membedakan keenam kelas aktivitas *welder*.

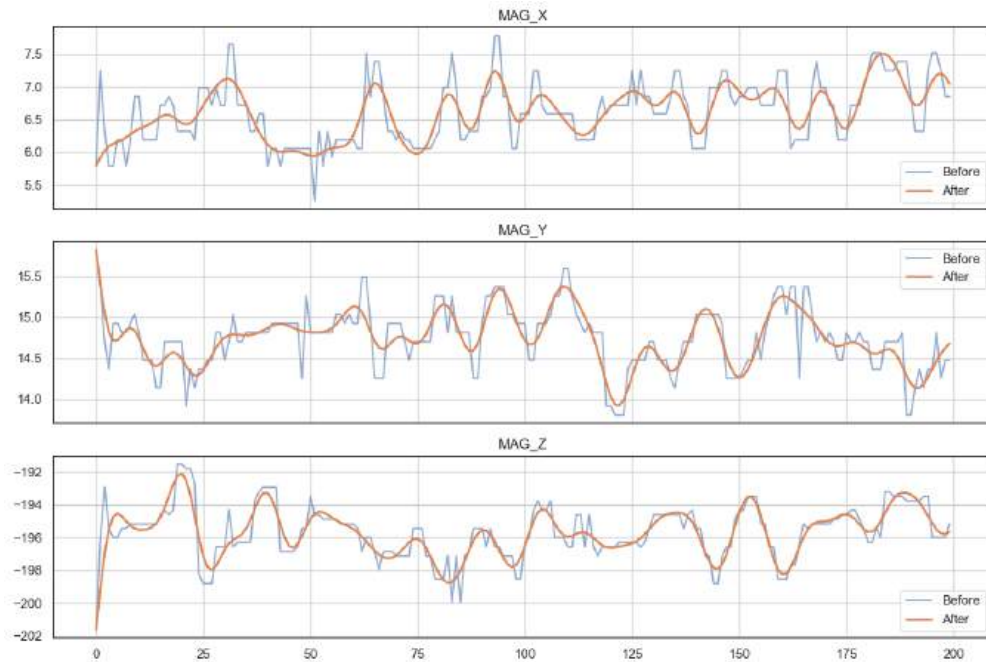


Gambar 4.26 Filtering Gyroscope
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.26 menunjukkan perbandingan sinyal *gyroscope* pada sumbu X, Y, dan Z sebelum dan sesudah proses *filtering*. Garis *Before* merepresentasikan data mentah yang masih mengandung *noise* dan fluktuasi tidak stabil, sedangkan garis *After* menunjukkan sinyal yang telah difilter menggunakan *low-pass filter* Butterworth dengan frekuensi *cutoff* 5 Hz dan orde 4.

Pada ketiga sumbu, sinyal mentah menunjukkan osilasi cepat dengan amplitudo yang bervariasi, terutama terlihat pada sumbu Y yang memiliki lonjakan nilai cukup besar pada beberapa titik. Setelah proses *filtering*, osilasi berfrekuensi tinggi tersebut berhasil diredam, sementara pola perubahan orientasi dan kecepatan sudut lengan atas *welder* tetap dipertahankan. Proses ini penting karena data *gyroscope* sangat sensitif terhadap getaran kecil dan gangguan mekanis, sehingga tanpa *filtering*, pola rotasi sebenarnya dapat tertutup oleh *noise*. Dengan

berkurangnya *noise* frekuensi tinggi, karakteristik sinyal menjadi lebih mudah dikenali, dan pola rotasi lengan pada tiap aktivitas menjadi lebih jelas dibedakan. Hal ini meningkatkan kualitas data *gyroscope* sehingga lebih optimal digunakan pada tahap *windowing* dan proses klasifikasi aktivitas oleh model CNN selanjutnya.



Gambar 4.27 Filtering *Magnetometer*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.27 menunjukkan perbandingan sinyal *magnetometer* pada sumbu X, Y, dan Z sebelum dan sesudah proses *filtering*. Garis *Before* merepresentasikan data mentah yang menunjukkan variasi nilai mengikuti perubahan medan magnet selama aktivitas berlangsung, dengan pola pergerakan berbentuk tangga *staircase* yang cukup tajam berbeda dari karakteristik sinyal *accelerometer* dan *gyroscope* yang lebih halus meski tetap ber-*noise*. Hal ini dapat disebabkan oleh laju pembaruan *update rate* sensor *magnetometer* yang relatif lebih rendah, sehingga nilai antar sampel cenderung berubah secara diskrit. Setelah proses *filtering* menggunakan *low-pass filter Butterworth*, pola tangga tersebut berhasil dihaluskan menjadi kurva yang lebih kontinu tanpa menghilangkan tren perubahan medan magnet yang sebenarnya.

Selain itu, nilai sinyal pada sumbu Z berada pada kisaran yang jauh berbeda (sekitar -192 hingga -202) dibandingkan sumbu X dan Y, yang mengindikasikan

adanya pengaruh gangguan medan elektromagnetik di lingkungan bengkel las, terutama dari busur listrik yang dihasilkan selama proses pengelasan. Meskipun demikian, proses *filtering* tetap mampu meredam *noise* frekuensi tinggi tanpa menghilangkan karakteristik variasi medan magnet tersebut, sehingga data *magnetometer* yang dihasilkan menjadi lebih stabil dan siap digunakan pada tahap *windowing* selanjutnya.

Normalisasi

Setelah proses *filtering*, data sensor dinormalisasikan menggunakan metode *z-score* untuk menyeragamkan data sebelum masuk ke model CNN. Normalisasi ini bertujuan agar setiap fitur memiliki skala yang sebanding sehingga proses pelatihan model menjadi lebih stabil dan akurat. Metode *z-score* bekerja dengan mengubah data berdasarkan nilai rata-rata dan standar deviasi dari masing masing fitur. Persamaan normalisasi *Z-score* ditunjukkan pada persamaan 4.17:

$$z = \frac{x - \mu}{\sigma} \quad (4.17)$$

Dimana:

- z = nilai setelah normalisasi
- x = nilai asli
- μ = nilai rata – rata (*mean*) *dataset*
- σ = standar deviasi *dataset*

Sebagai contoh perhitungan normalisasi *Z-score*, digunakan data sensor *accelerometer* sumbu X (*accel_x*) dari *dataset*. Misalnya pada salah satu sampel diperoleh nilai $x = 0,12088$, dengan nilai rata-rata *dataset* $\mu = -0,01313$ dan standar deviasi $\sigma = 0,99757$. Maka proses normalisasi dilakukan menggunakan Persamaan 4.16 sebagai berikut:

$$\begin{aligned} z &= \frac{0,12088 - (-0,01313)}{0,99757} \\ z &= \frac{0,13401}{0,99757} \\ z &= 0,13434 \end{aligned}$$

Hasil normalisasi menunjukkan bahwa nilai data berada sedikit di atas rata-rata distribusi data. Dengan normalisasi *Z-score*, data menjadi memiliki skala yang lebih beragam sehingga membantu meningkatkan performa model CNN dalam proses pelatihan dan klasifikasi aktivitas. Pada proses ini, menggunakan *library StandardScaler* dari *sklearn.preprocessing* pada python

4.9.4 Sliding Window

Sliding Window merupakan proses segmentasi data *time-series* menjadi beberapa segmen dengan panjang tetap. Pada penelitian ini, proses tersebut bertujuan agar model CNN-1D mampu mempelajari pola temporal aktivitas *welder* dalam interval waktu tertentu. Metode yang digunakan adalah *sliding window* dengan overlap sebesar 50%, sehingga setiap *window* baru bergeser sejauh 50 sampel dari *window* sebelumnya. Dengan panjang *window* sebesar 100 sampel, maka setiap dua *window* yang berurutan memiliki 50 sampel yang saling tumpang tindih. Persamaan *sliding window* ditunjukkan pada Persamaan 4.18:

$$w_i = \{x(t): t \in [i \cdot s, i \cdot s + L - 1]\} \quad (4.18)$$

Dimana:

- w_i = *window* ke- i
- L = panjang *window* (jumlah sampel per segmen)
- s = langkah geser
- t = indeks waktu

Jumlah total *window* yang dihasilkan menggunakan perhitungan Persamaan 4.19:

$$N_w = \left\lfloor \frac{N_{total} - L}{s} \right\rfloor + 1 \quad (4.19)$$

Dimana N_{total} adalah total jumlah sampel dalam *dataset*. Pada penelitian ini digunakan Panjang *window* $L = 100$ sampel yang setara dengan durasi 2 detik pada frekuensi *sampling* 50 Hz:

$$L = f_s \times t = 50 \times 2 = 100 \text{ sampel}$$

Sedangkan nilai *stride* yang digunakan adalah 50 sampel, sehingga menghasilkan overlap sebesar 50% antar-*window*. Setelah proses *windowing*, penentuan label setiap *window* dilakukan menggunakan metode *majority voting* dengan bantuan fungsi *np.bincount().argMax()* pada Python untuk memilih label yang paling dominan pada setiap segmen data. Proses ini menghasilkan pemetaan enam kelas aktivitas, yaitu Berbaring Aktif (0), Berbaring Tidak Aktif (1), Berdiri Aktif (2), Berdiri Tidak Aktif (3), Duduk Aktif (4), dan Duduk Tidak Aktif (5).

Hasil segmentasi menghasilkan data masukan berukuran (3619, 100, 13) dan label (3619,), yang menunjukkan bahwa diperoleh 3.619 *window*, masing-masing terdiri atas 100 *timestep* dengan 13 fitur (9 data IMU dan 4 komponen *quaternion* serta satu label yang merepresentasikan aktivitas dominan pada setiap *window*).

4.9.5 Split Data Training dan Data Test

Pembagian data setelah proses *windowing* merupakan tahap penting sebelum pelatihan model CNN-1D. Tujuan pembagian data adalah memisahkan data yang digunakan untuk proses *training* dan data yang digunakan untuk *testing* sehingga kemampuan model dalam melakukan generalisasi terhadap data baru dapat dievaluasi secara objektif. Pada penelitian ini, pembagian *dataset* dilakukan menggunakan fungsi *train_test_split()* dari pustaka *Scikit-learn* dengan rasio 80:20 serta menerapkan parameter *stratify* agar proporsi setiap kelas tetap seimbang pada data latih maupun data uji yang mengikuti hasil pembagian otomatis oleh sistem. Jumlah data *training* dihitung menggunakan Persamaan 4.20:

$$N_{train} = [N_{total} \times 0,8] \quad (4.20)$$

$$N_{train} = [3619 \times 0,8] = [2895,2] = 2895 \text{ window}$$

Sedangkan jumlah data *test* dihitung menggunakan Persamaan 4.21:

$$N_{test} = [N_{total} - N_{train}] \quad (4.21)$$

$$N_{test} = 3619 - 2895 = 724 \text{ window}$$

Berdasarkan Tabel 4.20 dibawah ini, total data *training* berjumlah 2895 *window* dengan bentuk (2895, 100, 13) dan data *test* berjumlah 724 *window* dengan bentuk(724, 100, 13) merupakan hasil dari pembagian data mengikuti rasio 80:20 dari jumlah total data masing-masing kelas, sehingga representasi setiap kelas pada data *training* dan *test* sudah seimbang dan dapat digunakan untuk melatih serta mengevaluasi model CNN.

Tabel 4.20 Hasil Pembagian Data

Kelas Aktivitas	Data <i>Training</i>	Data <i>Test</i>
Berdiri Aktif	497	120
Berdiri Tidak Aktif	497	120
Duduk Aktif	480	120
Duduk Tidak Aktif	480	120
Berbaring Aktif	480	120
Berbaring Tidak Aktif	479	120
Total	2895	724

(Dokumentasi Pribadi Penulis, 2026)

4.10 Penerapan Metode CNN Dengan *Quaternion*

Pada tahap pengujian ini, dilakukan dengan menambahkan orientasi berbaris *quaternion* yang terdiri dari empat komponen, yaitu q_w, q_x, q_y, q_z , sehingga total kanal *input* menjadi 13. Penambahan *quaternion* bertujuan untuk merepresentasikan orientasi rotasi secara lebih stabil dan menghindari permasalahan *gimbal lock* yang sering terjadi pada representasi sudut *Euler*. Dengan penambahan kanal tersebut maka,

$$\text{Total kanal input} = 13$$

$$\text{Sampling rate IMU} = 50 \text{ Hz}$$

$$\text{Durasi window} = 2 \text{ detik}$$

$$\text{Window length} = 50 \times 2 = 100 \text{ sampel}$$

Jadi bentuk *input* setelah segmentasi *time-series*:

$$(100 \times 13)$$

Arsitektur CNN yang digunakan pada tahap ini tetap mengacu pada perancangan awal dengan dua lapisan Conv1D yang diikuti oleh *Max Pooling*.

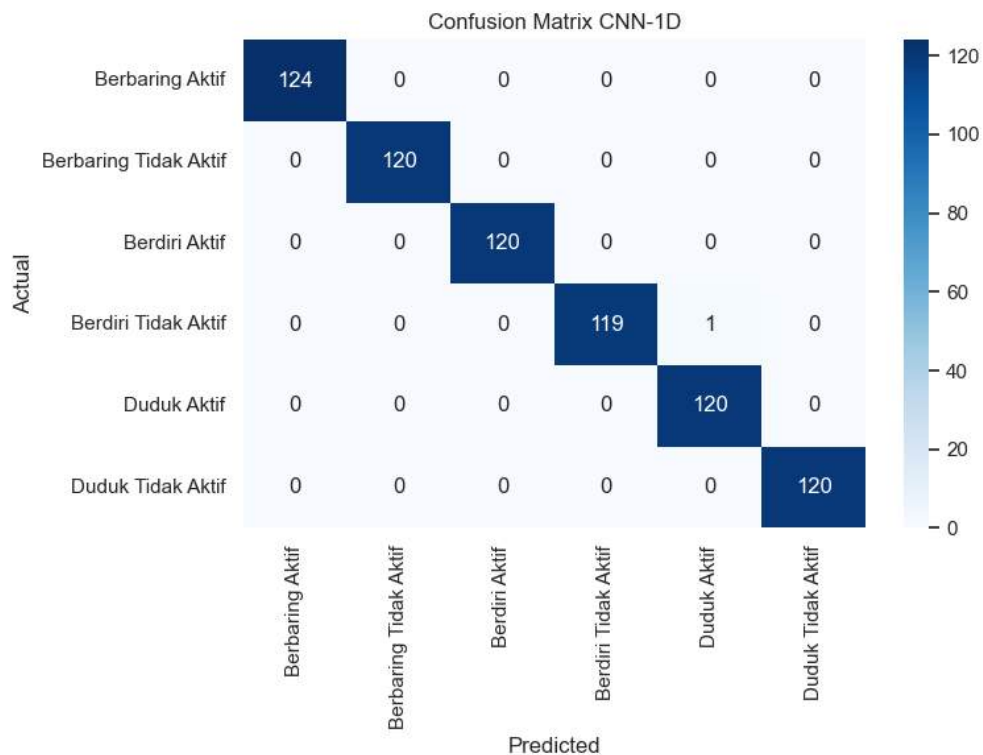
Tabel 4.21 Hasil Model Sequential

Layer	Output Shape
Conv1D	(100, 16)
<i>Max Pooling 1D</i>	(50, 16)
Conv1D	(50, 32)
<i>Max pooling 1D</i>	(25, 32)
<i>Flatten</i>	(800)
<i>Dense</i>	(6)
<i>Dense</i>	(6)

(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil *summary* model *sequential* pada Tabel 4.21, terlihat bahwa lapisan pertama Conv1D menghasilkan *Output* sebesar (100, 16), yang kemudian direduksi melalui *Max pooling* menjadi (50, 16). Lapisan konvolusi kedua meningkatkan kompleksitas fitur dengan menghasilkan (50,32) yang selanjutnya direduksi kembali menjadi (24, 32). Setelah proses *flatten*, diperoleh vector fitur sepanjang 800 yang menjadi *input* bagi *dense* layer. Lapisan pertama menghasilkan 6 neuron yang berfungsi sebagai representasi fitur klasifikasi hasil pembelajaran CNN, selanjutnya, lapisan *Output* juga terdiri dari 6 neuron yang merepresentasikan jumlah kelas aktivitas yang digunakan dalam penelitian.

Penambahan kanal menyebabkan peningkatan jumlah parameter pada lapisan konvolusi pertama, karena jumlah bobot bergantung langsung pada jumlah kanal *input*. Hal ini memungkinkan model untuk mempelajari hubungan yang lebih kompleks antara data sensor dan orientasi gerakan. Dengan demikian, hasil pengujian menunjukkan bahwa integrasi data *quaternion* ke dalam *input* CNN mampu meningkatkan kualitas representasi fitur dan berpotensi meningkatkan performa klasifikasi dibandingkan dengan penggunaan 9 kanal IMU saja. Sebelum dilakukan evaluasi, penelitian ini menggunakan *library sklearn.metrics*. Evaluasi menggunakan metrik akurasi yang ditunjukkan pada Gambar 4.28:



Gambar 4.28 *Confusion matrix*
(Dokumentasi Pribadi Penulis, 2026)

Confusion matrix digunakan untuk mengevaluasi kinerja model CNN-1D dalam mengklasifikasikan enam kelas aktivitas *welder* dengan membandingkan label aktual terhadap hasil prediksi model. Berdasarkan hasil *confusion matrix* pada Gambar 4.28, hampir seluruh data berhasil diklasifikasikan dengan benar. Kelas Berbaring Aktif, Berbaring Tidak Aktif, Berdiri Aktif, Duduk Aktif, dan Duduk Tidak Aktif memperoleh hasil klasifikasi sempurna, di mana seluruh data pada masing-masing kelas berhasil dikenali tanpa kesalahan. Kesalahan klasifikasi hanya terjadi pada kelas Berdiri Tidak Aktif, yaitu dari 120 data uji terdapat 119 data yang berhasil diklasifikasikan dengan benar, sedangkan 1 data salah diklasifikasikan sebagai Duduk Aktif. Kesalahan tersebut diduga disebabkan oleh kemiripan pola orientasi tubuh dan minimnya pergerakan pada kedua aktivitas sehingga karakteristik data yang dihasilkan sensor menjadi hampir serupa. Selain menggunakan *confusion matrix*, evaluasi kinerja model juga dilakukan menggunakan *metrik precision, recall, dan F1-score* yang dihitung menggunakan pustaka *sklearn.metrics*, sebagaimana ditunjukkan pada Persamaan 4.22-4.25:

$$Akurasi = \frac{\sum TP}{Total Data} \times 100\% \quad (4.22)$$

$$Akurasi = \frac{124 + 120 + 120 + 119 + 120 + 120}{124 + 120 + 120 + 120 + 120 + 120} = \frac{723}{724} \times 100\% \\ = 99,34\%$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (4.23)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.24)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.25)$$

Dimana:

- TP = *True positive* (prediksi benar positif)
- TN = *True Negative* (prediksi benar negatif)
- FP = *False Positive* (prediksi salah positif)
- FN = *False Negative* (prediksi salah negatif)

Tabel 4.22 *Classification Report*

Classification Report				
	Precision	recall	f1-score	Support
Berbaring Aktif	1.00	1.00	1.00	124
Berbaring Tidak Aktif	1.00	1.00	1.00	120
Berdiri Aktif	1.00	1.00	1.00	120
Berdiri Tidak Aktif	1.00	0.99	1.00	120
Duduk Aktif	0.99	1.00	1.00	120
Duduk Tidak Aktif	1.00	1.00	1.00	120
Accuracy			1.00	724
Macro Avg	1.00	1.00	1.00	724
Weighted Avg	1.00	1.00	1.00	724

(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 4.22, hampir seluruh kelas aktivitas memiliki nilai *precision*, *recall*, dan *F1-score* yang mencapai 1,00, dengan akurasi keseluruhan sebesar 99,86%. Hasil tersebut menunjukkan bahwa model CNN-1D mampu mengklasifikasikan aktivitas *welder* dengan tingkat

ketepatan yang sangat tinggi. Kelas Berbaring Aktif, Berbaring Tidak Aktif, Berdiri Aktif, dan Duduk Tidak Aktif memperoleh nilai *precision*, *recall*, dan *F1-score* sebesar 1,00, yang menunjukkan seluruh data pada kelas tersebut berhasil diklasifikasikan dengan benar. Pada kelas Berdiri Tidak Aktif, nilai *recall* sebesar 0,99 menunjukkan terdapat satu data yang tidak berhasil dikenali sebagai kelas sebenarnya dan salah diprediksi sebagai Duduk Aktif. Sebaliknya, pada kelas Duduk Aktif diperoleh nilai *precision* sebesar 0,99, yang menunjukkan terdapat satu data dari kelas lain yang diprediksi sebagai Duduk Aktif.

Meskipun demikian, nilai *macro average* dan *weighted average* masing-masing sebesar 1,00, sehingga dapat disimpulkan bahwa model memiliki performa yang sangat baik dan konsisten dalam mengenali seluruh aktivitas *welder*.

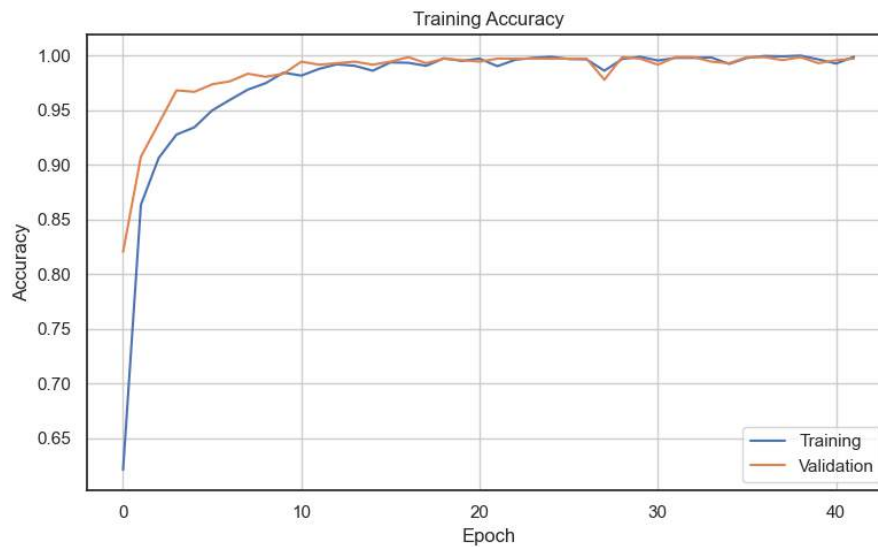
Tabel 4.23 Hasil Evaluasi

Hasil Evaluasi				
<i>Accuracy</i>	<i>Precision</i>	<i>recall</i>	<i>f1-score</i>	<i>Latency (s)</i>
99.86%	99.86%	99.86%	99.86%	0.158374

(Dokumentasi Pribadi Penulis, 2026)

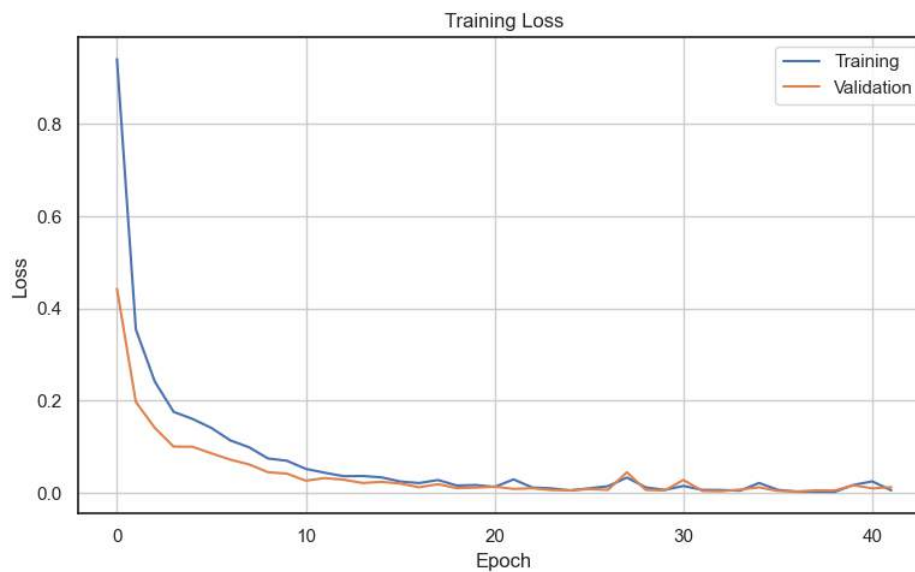
Secara keseluruhan, dapat dilihat pada Tabel 4.23 model memperoleh nilai *accuracy* sebesar 99,86%, yang menunjukkan bahwa model mampu mengklasifikasikan aktivitas *welder* dengan tingkat ketepatan yang sangat tinggi. Nilai *precision* sebesar 99,86%, *recall* sebesar 99,86%, dan *F1-score* sebesar 99,86% menunjukkan bahwa model memiliki keseimbangan yang baik antara ketepatan prediksi dan kemampuan mengenali setiap kelas aktivitas. Selain itu, waktu inferensi *latency* yang diperoleh sebesar 0.158374 detik menunjukkan bahwa model mampu melakukan proses klasifikasi secara cepat sehingga mendukung implementasi pemantauan aktivitas secara *real-time*.

Hasil evaluasi tersebut menunjukkan bahwa model CNN yang dikombinasikan dengan representasi data *quaternion* serta tahapan *pre-processing*, seperti *filtering*, normalisasi, dan *sliding window*, mampu mempelajari karakteristik data sensor IMU ICM-20948 secara optimal sehingga dapat mengklasifikasikan keenam aktivitas *welder* dengan performa yang sangat baik.



Gambar 4.29 Hasil *Accuracy* CNN
(Dokumentasi Pribadi Penulis, 2026)

Gambar 4.29 menunjukkan grafik akurasi model CNN-1D dengan data *quaternion* selama proses *training* dan validasi. Akurasi *training* meningkat dari sekitar 62% menjadi 99%, sedangkan akurasi validasi meningkat dari sekitar 82% hingga mendekati 100%. Peningkatan akurasi pada kedua kurva menunjukkan bahwa model mampu mempelajari pola aktivitas *welder* dengan baik sehingga menghasilkan performa klasifikasi yang tinggi.



Gambar 4.30 Hasil *Loss* CNN
(Dokumentasi Pribadi Penulis, 2026)

Gambar 4.30 menunjukkan grafik *loss* model CNN-1D dengan data *quaternion* selama proses *training* dan validasi. Nilai *training loss* menurun dari sekitar 0,94 menjadi mendekati 0,01, sedangkan *validation loss* menurun dari sekitar 0,44 hingga mendekati 0,01. Penurunan *loss* pada kedua kurva menunjukkan bahwa model semakin baik dalam mempelajari pola aktivitas *welder* selama proses pelatihan.

4.11 Penerapan Metode CNN Tanpa *Quaternion*

Pada tahap pengujian ini juga dilakukan pengujian untuk membandingkan hasil pengujian metode CNN dengan *quaternion* dan tanpa *quaternion* arsitektur awal CNN tanpa *quaternion* menggunakan 9 kanal *input* yaitu *accelerometer* sumbu x, y, z, *gyroscope* sumbu x, y, z, *magnetometer* sumbu x, y, z maka,

Total kanal *input* = 9

Sampling rate IMU = 50 Hz

Durasi *window* = 2 detik

Window length = $50 \times 2 = 100$ sampel

Jadi bentuk *input* setelah segmentasi *time-series*:

(100×9)

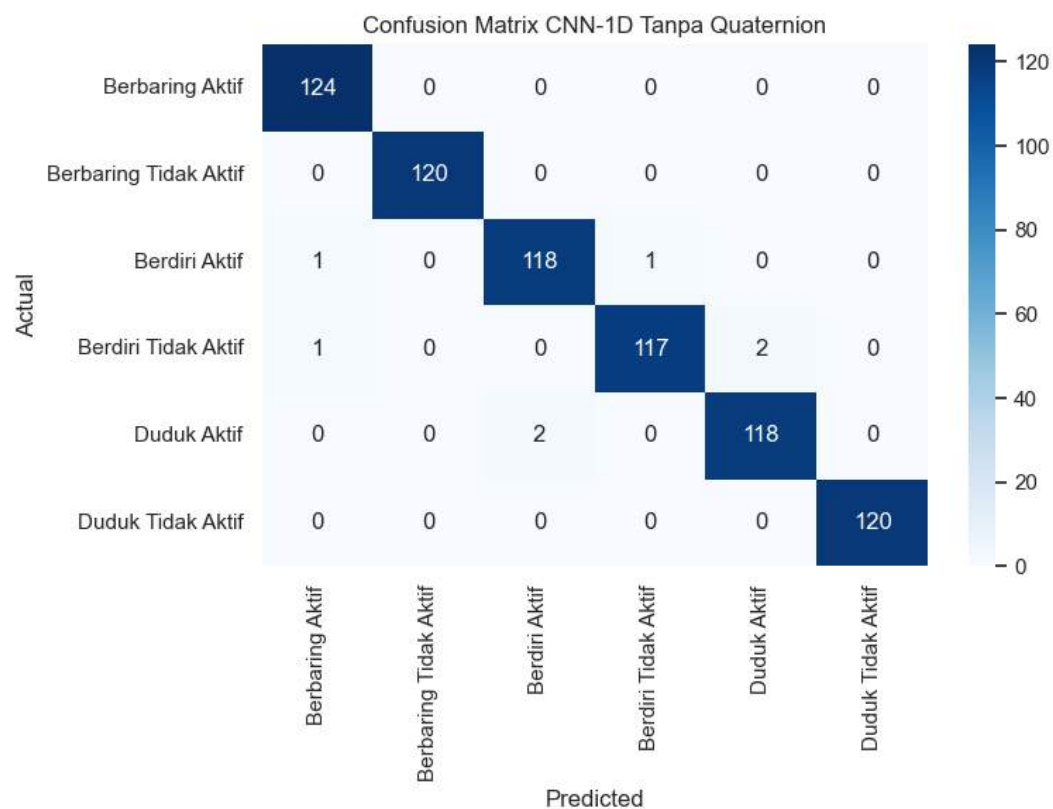
Arsitektur CNN yang digunakan pada tahap ini tetap mengacu pada perancangan awal dengan dua lapisan Conv1D yang diikuti oleh *Max Pooling*.

Tabel 4.24 Hasil Model Sequential Tanpa *Quaternion*

Layer	Output Shape
Conv1D	(100, 16)
<i>Max Pooling</i> 1D	(50, 16)
Conv1D	(50, 32)
<i>Max pooling</i> 1D	(25, 32)
<i>Flatten</i>	(800)
<i>Dense</i>	(6)
<i>Dense</i>	(6)

(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil *summary* model sequential pada Tabel 4.24, lapisan pertama Conv1D menghasilkan *Output* sebesar (100,9) yang kemudian direduksi melalui *Max pooling* menjadi (50,9). Lapisan konvolusi kedua meningkatkan kompleksitas fitur dengan menghasilkan *Output* (48,18) yang kembali direduksi menjadi (24,18). Setelah proses *flatten*, diperoleh vektor fitur sepanjang 432 yang digunakan sebagai *input* pada *dense* layer, dengan lapisan *Output* terdiri dari 6 neuron sesuai jumlah kelas aktivitas. Dibandingkan model dengan *quaternion* yang menghasilkan vektor fitur sepanjang 800, model tanpa *quaternion* memiliki representasi fitur yang lebih sedikit sehingga kemampuan dalam menangkap kompleksitas pola gerakan menjadi lebih terbatas. Sebelum proses evaluasi dilakukan, penelitian ini menggunakan *library sklearn.metrics*



Gambar 4.31 *Confusion matrix* Tanpa *Quaternion*
(Dokumentasi Pribadi Penulis, 2026)

Confusion matrix digunakan untuk mengevaluasi kinerja model CNN dalam mengklasifikasikan enam kelas aktivitas berdasarkan perbandingan antara label sebenarnya dan hasil prediksi model. Berdasarkan Gambar 4.31, sebagian besar data berhasil diklasifikasikan dengan benar yang ditunjukkan oleh nilai pada

diagonal utama. Kesalahan klasifikasi hanya terjadi pada kelas Berdiri Aktif, Berdiri Tidak Aktif, dan Duduk Aktif, dengan jumlah kesalahan yang relatif sedikit. Hal ini menunjukkan bahwa model CNN tanpa *quaternion* masih mengalami kesulitan membedakan beberapa aktivitas yang memiliki pola gerakan serupa. Selanjutnya, performa model dievaluasi menggunakan metrik *precision*, *recall*, dan *F1-score* dengan bantuan *library* sklearn.metrics yang ditunjukkan pada Persamaan 4.26–4.29:

$$Akurasi = Akurasi = \frac{\sum TP}{Total\ Data} \times 100\% \quad (4.26)$$

$$Akurasi = \frac{124 + 120 + 118 + 117 + 118 + 120}{124 + 120 + 120 + 120 + 120 + 120} = \frac{717}{724} \times 100\% \\ = 99,03\%$$

$$Precision = \frac{TP}{TP + FP} \quad (4.27)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.28)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.29)$$

Dimana:

- *TP*= *True positive* (prediksi benar positif)
- *TN*= *True Negative* (prediksi benar negatif)
- *FP*= *False Positive* (prediksi salah positif)
- *FN*= *False Negative* (prediksi salah negatif)

Tabel 4.25 *Classification Report* Tanpa *Quaternion*

Classification Report				
	<i>Precision</i>	<i>recall</i>	f1-score	Support
Berbaring Aktif	0.98	1.00	0.99	124
Berbaring Tidak Aktif	1.00	1.00	1.00	120
Berdiri Aktif	0.98	0.98	0.98	120
Berdiri Tidak Aktif	0.98	0.97	0.98	120
Duduk Aktif	1.00	0.98	0.98	120
Duduk Tidak Aktif		1.00	1.00	120

<i>Classification Report</i>				
<i>Accuracy</i>			0.99	724
Macro Avg	0.99	0.99	0.99	724
Weighted Avg	0.99	0.99	0.99	724

(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil *classification report* pada Tabel 4.25, model CNN tanpa menggunakan data *quaternion* menunjukkan performa klasifikasi yang sangat baik pada enam kelas aktivitas *welder*. Kelas Berbaring Tidak Aktif dan Duduk Tidak Aktif memperoleh nilai *precision*, *recall*, dan *F1-score* sebesar 1,00, yang menunjukkan seluruh data pada kedua kelas tersebut berhasil diklasifikasikan dengan benar. Sementara itu, kelas Berbaring Aktif, Berdiri Aktif, Berdiri Tidak Aktif, dan Duduk Aktif memperoleh nilai metrik antara 0,97–0,99, yang menunjukkan masih terdapat sedikit kesalahan klasifikasi. Secara keseluruhan, model CNN tanpa *quaternion* memperoleh akurasi sebesar 99% dengan nilai *macro average* dan *weighted average* masing-masing sebesar 0,99, sehingga model mampu mengenali pola aktivitas *welder* dengan tingkat ketepatan yang tinggi meskipun hanya menggunakan data sensor IMU tanpa representasi orientasi *quaternion*. Hasil evaluasi ditunjukkan pada Tabel 4.26.

Tabel 4.26 Hasil Evaluasi

Hasil Evaluasi				
<i>Accuracy</i>	<i>Precision</i>	<i>recall</i>	<i>f1-score</i>	<i>Latency (s)</i>
99.03%	99.04%	99.03%	99.03%	0.171510

(Dokumentasi Pribadi Penulis, 2026)

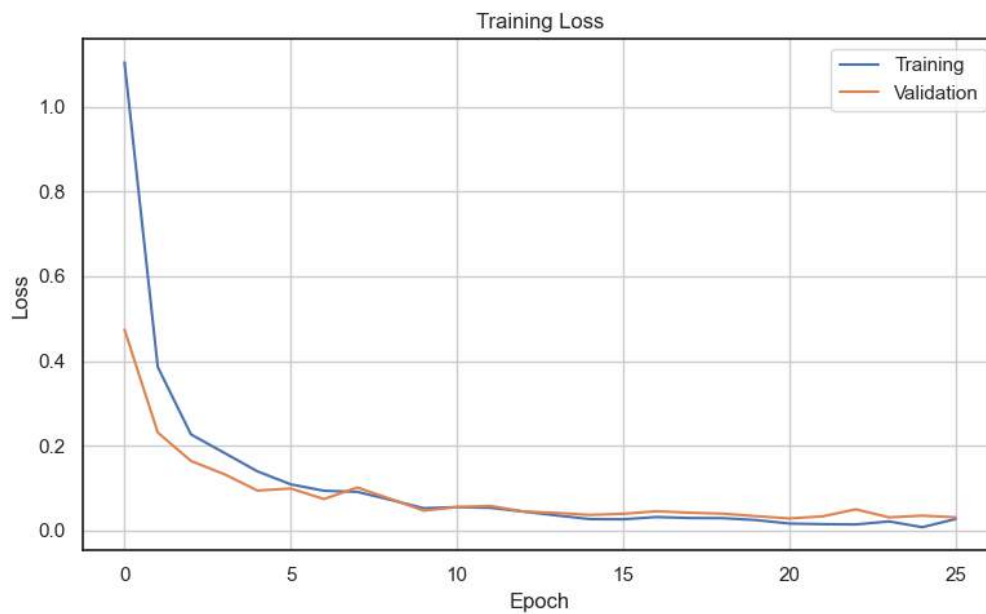
Secara keseluruhan, model CNN tanpa *quaternion* memperoleh *accuracy* sebesar 99,03%, *precision* sebesar 99,04%, *recall* sebesar 99,03%, dan *F1-score* sebesar 99,03%. Nilai *precision*, *recall*, dan *F1-score* yang hampir sama menunjukkan bahwa model memiliki performa yang konsisten dalam mengklasifikasikan setiap kelas aktivitas *welder*. Selain itu, waktu inferensi (*latency*) sebesar 0,171510 detik menunjukkan bahwa model mampu melakukan proses klasifikasi dengan cepat sehingga mendukung penerapan sistem untuk pemantauan aktivitas *welder* secara *real-time*.



Gambar 4.32 Hasil *Accuracy* CNN Tanpa *Quaternion*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.32 ditunjukkan grafik akurasi model CNN tanpa *quaternion* selama proses *training* dan validasi. Garis biru menunjukkan akurasi data *training*, sedangkan garis oranye menunjukkan akurasi data validasi pada setiap *epoch*. Pada awal pelatihan, akurasi *training* berada di sekitar 58%, kemudian meningkat secara signifikan hingga mencapai lebih dari 90% dalam beberapa *epoch* pertama. Akurasi validasi juga menunjukkan peningkatan yang cepat, yaitu dari sekitar 86% hingga mencapai lebih dari 95% pada awal proses pelatihan. Hal ini menunjukkan bahwa model mampu mempelajari karakteristik data sensor IMU dengan baik sejak tahap awal pelatihan.

Seiring bertambahnya jumlah *epoch*, akurasi *training* dan validasi terus meningkat secara bertahap hingga mencapai nilai sekitar 99% pada akhir proses pelatihan. Kedua kurva memiliki pola yang hampir sama dan cenderung stabil mulai sekitar *epoch* ke-10, yang menunjukkan bahwa proses pembelajaran model telah mencapai konvergensi. Meskipun hanya menggunakan sembilan fitur sensor IMU tanpa informasi orientasi *quaternion*, model CNN tetap mampu mengenali pola gerakan dari data *accelerometer*, *gyroscope*, dan *magnetometer* dengan sangat baik, sehingga menghasilkan performa klasifikasi aktivitas *welder* yang tinggi.



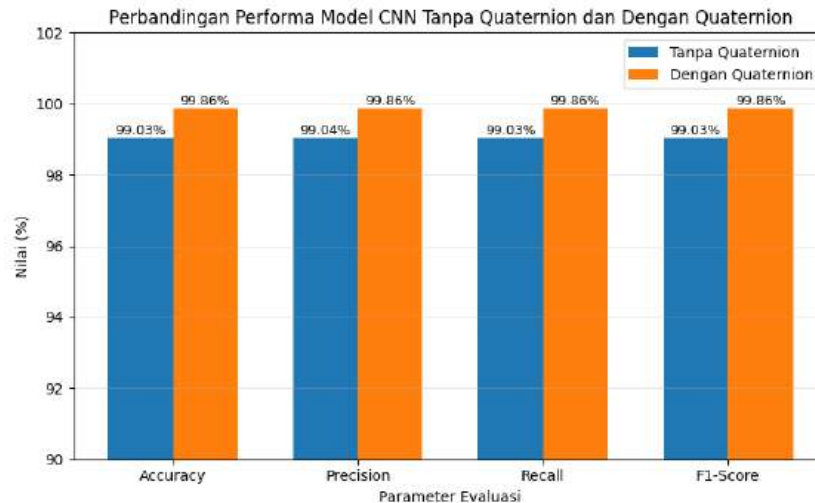
Gambar 4.33 Hasil *Loss* CNN Tanpa *Quaternion*
(Dokumentasi Pribadi Penulis, 2026)

Pada Gambar 4.33 ditunjukkan grafik *loss* model CNN tanpa *quaternion* selama proses *training* dan validasi pada setiap *epoch*. Garis biru menunjukkan nilai *training loss*, sedangkan garis oranye menunjukkan *validation loss*. Pada awal proses pelatihan, nilai *training loss* masih berada di atas 1,0, sedangkan *validation loss* berada di sekitar 0,47. Seiring bertambahnya jumlah *epoch*, kedua nilai *loss* mengalami penurunan yang cukup signifikan, terutama pada beberapa *epoch* pertama. Penurunan ini menunjukkan bahwa model semakin mampu menyesuaikan bobot dan meminimalkan kesalahan prediksi terhadap data yang dipelajari.

Setelah memasuki sekitar *epoch* ke-10, nilai *training loss* dan *validation loss* terus menurun secara bertahap hingga mencapai nilai yang sangat kecil, yaitu mendekati 0,03 pada akhir proses pelatihan. Kedua kurva juga memiliki pola yang hampir sama dan cenderung stabil hingga pelatihan selesai. Hasil tersebut menunjukkan bahwa model CNN tanpa *quaternion* mampu melakukan ekstraksi fitur dari data *accelerometer*, *gyroscope*, dan *magnetometer* secara efektif sehingga menghasilkan kesalahan prediksi yang rendah dalam proses klasifikasi aktivitas *welder*.

4.12 Perbandingan CNN Dengan *Quaternion* dan Tanpa *Quaternion*

Berdasarkan seluruh hasil evaluasi yang telah dilakukan, terdapat perbedaan performa yang sangat signifikan antara model CNN dengan *quaternion* dan tanpa *quaternion*.



Gambar 4.34 Hasil Perbandingan performa Model
(Dokumentasi Pribadi Penulis, 2026)

Berdasarkan hasil perbandingan performa pada Gambar 4.34, model CNN dengan *quaternion* menunjukkan peningkatan kinerja dibandingkan model tanpa *quaternion* pada seluruh metrik evaluasi. Penambahan data *quaternion* meningkatkan nilai akurasi dari 96,16% menjadi 99,34%, *precision* dari 96,38% menjadi 99,35%, *recall* dari 96,16% menjadi 99,34%, dan f1-score dari 96,16% menjadi 99,34%. Peningkatan tersebut menunjukkan bahwa informasi orientasi dalam bentuk *quaternion* mampu memberikan representasi gerakan yang lebih baik sehingga model CNN dapat membedakan pola aktivitas *welder* dengan lebih akurat.

Perbedaan performa antara kedua model disebabkan oleh kemampuan *quaternion* dalam merepresentasikan orientasi sensor secara tiga dimensi dan lebih stabil dibandingkan penggunaan data sensor tanpa representasi orientasi. Dengan adanya informasi *quaternion*, model memperoleh fitur tambahan yang membantu mengenali perbedaan pola gerakan antar kelas aktivitas, terutama pada aktivitas yang memiliki karakteristik gerakan yang serupa. Hasil tersebut menunjukkan bahwa penggunaan data *quaternion* memberikan kontribusi dalam meningkatkan performa klasifikasi aktivitas *welder* berbasis sensor IMU.

4.13 Hasil Pengujian *Real-time*

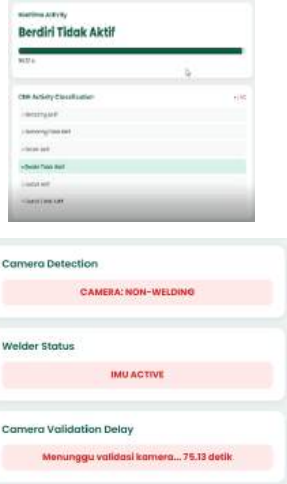
Pengujian *real-time* dilakukan untuk mengetahui kemampuan sistem *embedded* dalam melakukan akuisisi data sensor IMU, proses klasifikasi aktivitas *welder* menggunakan model CNN-1D, serta menampilkan hasil klasifikasi melalui *dashboard* web secara langsung. Pengujian dilakukan dengan menggunakan jaket *welder* yang telah terintegrasi sensor IMU, ESP32 sebagai perangkat akuisisi data, Raspberry Pi sebagai perangkat akuisisi kamera USB sebagai data pendukung validasi aktivitas. Pengujian dilakukan secara kontinu selama 10 menit untuk merekam aktivitas *welder*.

Berdasarkan hasil pengujian, sistem berhasil mengklasifikasikan enam kelas aktivitas secara *real-time*, yaitu Duduk Aktif, Duduk Tidak Aktif, Berdiri Aktif, Berdiri Tidak Aktif, Berbaring Aktif, dan Berbaring Tidak Aktif. Total durasi yang tercatat untuk masing-masing aktivitas adalah Duduk Aktif selama 2 menit, Duduk Tidak Aktif selama 1,2 menit, Berdiri Aktif selama 1 menit, Berdiri Tidak Aktif selama 2 menit, Berbaring Aktif selama 1,2 menit, dan Berbaring Tidak Aktif selama 2 menit. Hasil klasifikasi tersebut kemudian disimpan ke dalam basis data dan ditampilkan secara langsung melalui *dashboard* web, sehingga aktivitas *welder* dapat dipantau secara *real-time* selama proses pengelasan berlangsung. Hasil pengujian dapat dilihat pada Tabel 4.27.

Tabel 4.27 Hasil Pengujian *Real-time*

N o	Aktivitas <i>Welder</i>	<i>Dashboard</i> Prediksi CNN dan Validasi Kamera	Total Durasi Aktivitas
1			

N o	Aktivitas <i>Welder</i>	<i>Dashboard</i> Prediksi CNN dan Validasi Kamera	Total Durasi Aktivitas														
2		Realtime Activity Berdiri Aktif CNN Activity Classification +1.00 Welding: 0.00% Working: 100.00% Rest: 0.00% Idle: 0.00% Waiting: 0.00% Camera Detection CAMERA: WELDING DETECTED Welder Status WELDING Camera Validation Delay Tervalidasi dalam 1.40 detik Total Durasi Aktivitas Tanggal: 01/10/2023 <table><tr><th>Activity</th><th>Total Duration</th></tr><tr><td>Duduk Tidak Aktif</td><td>03:00:45</td></tr><tr><td>Duduk Aktif</td><td>03:01:18</td></tr><tr><td>Berbaring Aktif</td><td>03:00:10</td></tr><tr><td>Berdiri Aktif</td><td>03:00:28</td></tr><tr><td>Berbaring Tidak Aktif</td><td>03:00:22</td></tr><tr><td>Berdiri Tidak Aktif</td><td>03:00:00</td></tr></table>	Activity	Total Duration	Duduk Tidak Aktif	03:00:45	Duduk Aktif	03:01:18	Berbaring Aktif	03:00:10	Berdiri Aktif	03:00:28	Berbaring Tidak Aktif	03:00:22	Berdiri Tidak Aktif	03:00:00	
Activity	Total Duration																
Duduk Tidak Aktif	03:00:45																
Duduk Aktif	03:01:18																
Berbaring Aktif	03:00:10																
Berdiri Aktif	03:00:28																
Berbaring Tidak Aktif	03:00:22																
Berdiri Tidak Aktif	03:00:00																
3		Realtime Activity Duduk Aktif CNN Activity Classification +1.00 Welding: 0.00% Working: 100.00% Rest: 0.00% Idle: 0.00% Waiting: 0.00% Camera Detection CAMERA: WELDING DETECTED Welder Status WELDING Camera Validation Delay Tervalidasi dalam 1.40 detik Total Durasi Aktivitas Tanggal: 01/10/2023 <table><tr><th>Activity</th><th>Total Duration</th></tr><tr><td>Duduk Tidak Aktif</td><td>03:00:45</td></tr><tr><td>Duduk Aktif</td><td>03:01:18</td></tr><tr><td>Berbaring Aktif</td><td>03:00:10</td></tr><tr><td>Berdiri Aktif</td><td>03:00:28</td></tr><tr><td>Berbaring Tidak Aktif</td><td>03:00:22</td></tr><tr><td>Berdiri Tidak Aktif</td><td>03:00:00</td></tr></table>	Activity	Total Duration	Duduk Tidak Aktif	03:00:45	Duduk Aktif	03:01:18	Berbaring Aktif	03:00:10	Berdiri Aktif	03:00:28	Berbaring Tidak Aktif	03:00:22	Berdiri Tidak Aktif	03:00:00	
Activity	Total Duration																
Duduk Tidak Aktif	03:00:45																
Duduk Aktif	03:01:18																
Berbaring Aktif	03:00:10																
Berdiri Aktif	03:00:28																
Berbaring Tidak Aktif	03:00:22																
Berdiri Tidak Aktif	03:00:00																
4		Realtime Activity Berbaring Tidak Aktif CNN Activity Classification +1.00 Welding: 0.00% Working: 100.00% Rest: 0.00% Idle: 0.00% Waiting: 0.00% Camera Detection CAMERA: NON-WELDING Welder Status IMU ACTIVE Camera Validation Delay Menunggu validasi kamera... 75.13 detik Total Durasi Aktivitas Tanggal: 01/10/2023 <table><tr><th>Activity</th><th>Total Duration</th></tr><tr><td>Duduk Tidak Aktif</td><td>03:00:45</td></tr><tr><td>Duduk Aktif</td><td>03:01:18</td></tr><tr><td>Berbaring Aktif</td><td>03:00:10</td></tr><tr><td>Berdiri Aktif</td><td>03:00:28</td></tr><tr><td>Berbaring Tidak Aktif</td><td>03:00:22</td></tr><tr><td>Berdiri Tidak Aktif</td><td>03:00:00</td></tr></table>	Activity	Total Duration	Duduk Tidak Aktif	03:00:45	Duduk Aktif	03:01:18	Berbaring Aktif	03:00:10	Berdiri Aktif	03:00:28	Berbaring Tidak Aktif	03:00:22	Berdiri Tidak Aktif	03:00:00	
Activity	Total Duration																
Duduk Tidak Aktif	03:00:45																
Duduk Aktif	03:01:18																
Berbaring Aktif	03:00:10																
Berdiri Aktif	03:00:28																
Berbaring Tidak Aktif	03:00:22																
Berdiri Tidak Aktif	03:00:00																

No	Aktivitas Welder	Dashboard Prediksi CNN dan Validasi Kamera	Total Durasi Aktivitas
5			
6			

BAB 5

KESIMPULAN DAN SARAN

Bab penutup dalam laporan Tugas Akhir merupakan bagian akhir yang berfungsi untuk merangkum keseluruhan hasil penelitian serta menyampaikan saran yang dapat dijadikan acuan untuk penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, diperoleh kesimpulan sebagai berikut :

1. Sistem *embedded* berbasis sensor IMU berhasil dirancang dan diintegrasikan pada jaket *welder* untuk memperoleh data pola gerakan tubuh *welder*. Sensor IMU ICM-20948 mampu melakukan akuisisi data gerakan lengan kanan atas berupa data *accelerometer*, *gyroscope*, *magnetometer*, dan *quaternion* yang digunakan sebagai data masukan dalam proses pengenalan aktivitas *welder*.
2. Sistem mampu mengklasifikasikan enam kelas aktivitas *welder* secara *real-time* menggunakan metode CNN-1D berdasarkan pola data sensor IMU, dengan performa sangat baik: *accuracy* sebesar 99,86%, serta *precision*, *recall*, dan *F1-score (macro average dan weighted average)* rata-rata 1,00 dari 724 data uji. Nilai *precision* terendah sebesar 0,99 tercatat pada kelas Duduk Aktif, dan *recall* terendah sebesar 0,99 pada kelas Berdiri Tidak Aktif, dengan waktu inferensi rata-rata 110,58 ms per *window*.
3. Validasi dilakukan dengan membandingkan status aktif/tidak aktif hasil prediksi sistem terhadap kondisi aktual mengelas, menggunakan deteksi cahaya busur las sebagai *ground truth*. Hasil validasi menunjukkan sistem mampu membedakan aktivitas mengelas dan tidak mengelas dengan cukup baik, meskipun terdapat *delay* 1,06 ms antara kondisi aktual dan status pada *dashboard*, akibat proses akuisisi data, pengiriman MQTT, inferensi model, hingga pembaruan tampilan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Disarankan menggunakan layar display yang terintegrasi pada Raspberry Pi agar proses *monitoring* dapat dilakukan secara langsung tanpa memerlukan *remote* SSH melalui *Real VNC viewer*, *xrdp*, atau penggunaan kabel LAN tambahan.
2. Disarankan menggunakan baterai dengan kapasitas yang lebih besar untuk meningkatkan durasi operasional Raspberry Pi, sehingga sistem dapat digunakan dalam waktu yang lebih lama selama proses *monitoring* aktivitas *welder* berlangsung.
3. Disarankan menggunakan kamera dengan ukuran yang lebih kecil agar lebih mudah diintegrasikan pada jaket *welder* serta penempatan kamera pada *helm welder*.
4. Penelitian selanjutnya disarankan melakukan pengambilan data dalam kondisi kerja yang lebih beragam, seperti variasi posisi benda kerja, jenis proses pengelasan, lokasi pengelasan, dan durasi aktivitas *welder*.

DAFTAR PUSTAKA

- Adafruit. (2023). *Adafruit TDK InvenSense ICM-20948 9-DoF IMU (MPU-9250 Upgrade)*.
- Alarfaj, M., Al Madini, A., Alsafran, A., Farag, M., Chtourou, S., Afifi, A., Ahmad, A., Al Rubayyi, O., Al Harbi, A., & Al Thunaian, M. (2024). Wearable sensors based on artificial intelligence models for human activity *recognition*. *Frontiers in Artificial Intelligence*, 7(June), 1–17. <https://doi.org/10.3389/frai.2024.1424190>
- Alibaba. (2024). *Mini USB Camera 5MP CMOS OmniVision*. <https://www.aliexpress.com/item/1005005616112869.html>
- Arifien, Z., Bachtiar, F. A., & Yudistira, N. (2022). Pengenalan Aktivitas Manusia Menggunakan Sensor Akselerometer dan Girooskop pada Smatphone dengan Metode K-Nearest Neighbor. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), 173–180. <https://doi.org/10.25126/jtiik.2022915593>
- Baek, S. Y., & Nam, J. H. (2021). Physical *welding* factors for reclassified *welding* positions in shipbuilding assembly process based on muscle activity measured by surface electromyography. *Journal of Marine Science and Engineering*, 9(11). <https://doi.org/10.3390/jmse9111211>
- Bieproduction. (2026). *Fungsi SD Card*. <https://bieproduction.com/pengertian-dan-fungsi-sd-card/>
- Fan, B., Zhang, L., Cai, S., Du, M., Liu, T., Li, Q., & Shull, P. (2025). Influence of *Sampling Rate* on Wearable IMU Orientation Estimation *Accuracy* for Human Movement Analysis. *Sensors*, 25(7), 1–16. <https://doi.org/10.3390/s25071976>
- Farahan, S. B., Machado, J. J. M., de Almeida, F. G., & Tavares, J. M. R. S. (2022). 9-DOF IMU-Based Attitude and Heading Estimation Using an Extended Kalman Filter with *Bias* Consideration. *Sensors*, 22(9). <https://doi.org/10.3390/s22093416>

- Foundation, R. P. (2023). *Raspberry Pi* 5. <https://www.raspberrypi.com/products/raspberry-pi-5/>
- GeeksforGeeks. (2026). *MongoDB vs MySQL: Difference Between MongoDB and MySQL*. <https://www.geeksforgeeks.org/mongodb/mongodb-vs-mysql/>
- Hakim, A. A. (2021). *CNN*. 3(2).
- Himanshi Singh. (2022). *A Guide to Convolutional Neural Network (CNNs)*.
- Jaramillo, I. E., Jeong, J. G., Lopez, P. R., Lee, C. H., Kang, D. Y., Ha, T. J., Oh, J. H., Jung, H., Lee, J. H., Lee, W. H., & Kim, T. S. (2022). *Real-time Human Activity Recognition with IMU and Encoder Sensors in Wearable Exoskeleton Robot via Deep Learning Networks*. *Sensors*, 22(24). <https://doi.org/10.3390/s22249690>
- Kim, Y. W., & Lee, S. (2023). Data Valuation Algorithm for Inertial Measurement Unit-Based Human Activity Recognition. *Sensors*, 23(1), 1–18. <https://doi.org/10.3390/s23010184>
- Lamablawa, F., & Aritonang, S. (2024). *KARAKTERISTIK LITHIUM-POLYMER BATTERY UNTUK APLIKASI RADIO YANG DI GUNAKAN PERSONIL TNI DALAM MENDUKUNG IKN (LITERATURE REVIEW)*. 2(4), 592–602. <https://doi.org/10.53866/jimi.v2i4.162>
- Martínez-zarzuela, M., González-alonso, J., Antón-rodríguez, M., Díaz-pernas, F. J., Müller, H., & Simón-martínez, C. (2023). *Multimodal video and IMU kinematic dataset on daily life activities using affordable devices*. 1–11. <https://doi.org/10.1038/s41597-023-02554-9>
- MATLAB, P. (2024). *Apa itu artificial intelligence, deep learning, dan convolutional neural network*.
- Pribadi, T. W., & Shinoda, T. (2022). Hand Motion Analysis for Recognition of Qualified and Unqualified Welders using 9-DOF IMU Sensors and Support Vector Machine (SVM) Approach. *International Journal of Technology*, 13(1), 38–47. <https://doi.org/10.14716/ijtech.v13i1.4813>

- Pribadi, T. W., & Shinoda, T. (2025). The Improvement concerning *Recognition Accuracy* of Six Skill Levels of Flux-Cored Arc *Welding* Hand *Welder* using *Convolutional Neural Network-Long Short-Term Memory*. *International Journal of Technology*, 16(4), 1296–1305. <https://doi.org/10.14716/ijtech.v16i4.7649>
- Rahman, S., Sembiring, A., Siregar, D., Khair, H., Prahmana, G., Puspadini, R., & Zen, M. (2023). Python : Dasar Dan Pemrograman Berorientasi Objek Tahta Media Group. In *Python: Dasar Dan Pemrograman Berorientasi Objek*.
- Santi, E. (2025). *Fitur Utama Visual Studio Code yang Membuat Pengembangan Kode Lebih Mudah*. <https://idwebhost.com/blog/vscode-adalah/>
- Tseng, Y. H., & Wen, C. Y. (2023). Hybrid Learning Models for IMU-Based HAR with Feature Analysis and Data Correction. *Sensors*, 23(18). <https://doi.org/10.3390/s23187802>
- Virmansyah, V. M., Pribadi, T. W., & Baihaqi, I. (2024). Deep Learning CNN-LSTM Approach for Identifying *Welder's* Hand Motion Gestures Using Wearable Sensors. *IOP Conference Series: Earth and Environmental Science*, 1423(1). <https://doi.org/10.1088/1755-1315/1423/1/012036>
- Zenhadi. (2022). Praktek Modul 1 : Pengenalan ESP32 Board. *MK Internet of Things*, 6, 1–16.
- Zhou, X., Li, S., Liu, J., Wu, Z., & Chen, Y. F. (2024). Construction Activity Analysis of Workers Based on Human Posture Estimation Information. *Engineering*, 33, 225–236. <https://doi.org/10.1016/j.eng.2023.10.004>

(Halaman ini sengaja dikosongkan)

LAMPIRAN

1. Berbaring Aktif

time	seconds_elpased	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	- 0.31 779	0.105 568	0.032 971	0.014 112	- 0.000 48	- 0.000 53	- 15.5 056	- 29.8 532	- 281. 837	0.088 581	- 0.51 324	0.270 987	0.809 509	Berbarin g Aktif
2000 0000	0.02	- 0.63 998	0.220 902	0.061 873	0.027 868	- 0.007 14	0.003 701	- 14.7 07	- 32.2 1	- 284. 912	0.089 149	- 0.51 279	0.271 566	0.809 536	Berbarin g Aktif
4000 0000	0.04	- 0.95 5	0.343 234	0.089 473	0.027 325	- 0.015 97	0.004 286	- 16.7 034	- 33.1 078	- 289. 386	0.089 692	- 0.51 234	0.272 197	0.809 548	Berbarin g Aktif
6000 0000	0.06	- 0.94 8	0.353 651	0.090 449	0.004 853	- 0.015 92	0.002 465	- 19.2 322	- 38.9 438	- 291. 623	0.090 17	- 0.51 188	0.272 855	0.809 567	Berbarin g Aktif
8000 0000	0.08	- 0.94 345	0.344 048	0.097 936	- 0.011 58	0.003 884	- 0.000 29	- 16.3 041	- 33.1 078	- 287. 429	0.090 697	- 0.51 14	0.273 47	0.809 602	Berbarin g Aktif
1E+0 8	0.1	- 0.95 907	0.328 748	0.101 192	- 0.007 85	0.020 982	0.006 107	- 16.4 372	- 33.8 935	- 289. 386	0.091 272	- 0.51 095	0.274 022	0.809 637	Berbarin g Aktif
1.2E+ 08	0.12	- 0.97 486	0.327 935	0.099 564	- 0.006 38	0.014 365	0.017 875	- 16.7 034	- 34.9 035	- 289. 386	0.091 859	- 0.51 051	0.274 566	0.809 665	Berbarin g Aktif

1.4E+08	0.14	- 0.97 453	0.333 469	0.103 47	- 0.018 6	- 0.011 79	0.022 183	- 16.4 372	- 34.9 035	- 291. 064	0.092 434	- 0.51 004	0.275 116	0.809 705	Berbarin g Aktif
1.6E+08	0.16	- 0.95 614	0.330 213	0.117 305	- 0.023 79	- 0.020 63	0.019 696	- 16.3 041	- 34.6 791	- 288. 827	0.092 995	- 0.50 954	0.275 636	0.809 781	Berbarin g Aktif
1.8E+08	0.18	- 0.95 142	0.317 844	0.124 466	0.001 966	- 0.009 13	0.012 546	- 16.3 041	- 34.6 791	- 288. 827	0.093 581	- 0.50 903	0.276 112	0.809 87	Berbarin g Aktif
2E+08	0.2	- 0.95 907	0.318 495	0.107 377	0.032 032	- 0.004 55	0.010 859	- 16.3 041	- 34.2 301	- 288. 827	0.094 194	- 0.50 857	0.276 632	0.809 913	Berbarin g Aktif
2.2E+08	0.22	- 0.96 574	0.342 746	0.083 451	0.029 678	- 0.015 7	0.011 347	- 16.3 041	- 33.3 323	- 287. 429	0.094 781	- 0.50 813	0.277 28	0.809 9	Berbarin g Aktif
2.4E+08	0.24	- 0.95 809	0.355 278	0.076 615	- 0.006 69	- 0.022 81	0.012 857	- 16.3 041	- 33.1 078	- 286. 31	0.095 334	- 0.50 768	0.277 984	0.809 874	Berbarin g Aktif
2.6E+08	0.26	- 0.94 458	0.346 489	0.093 054	- 0.032 14	- 0.007 18	0.010 059	- 16.3 041	- 33.3 323	- 287. 429	0.095 887	- 0.50 72	0.278 627	0.809 89	Berbarin g Aktif
2.8E+08	0.28	- 0.94 947	0.325 819	0.113 236	- 0.019 04	- 0.019 383	0.010 459	- 17.6 351	- 34.2 301	- 290. 225	0.096 485	- 0.50 671	0.279 161	0.809 939	Berbarin g Aktif
3E+08	0.3	- 0.96 363	0.320 773	0.114 864	0.009 738	0.020 804	0.012 679	- 16.3 041	- 33.6 69	- 287. 429	0.097 11	- 0.50 623	0.279 669	0.809 99	Berbarin g Aktif

3.2E+08	0.32	-0.97356	0.332004	0.10461	0.026214	-0.00011	0.017032	-17.1027	-35.4647	-291.902	0.097723	-0.50576	0.280232	0.810016	Berbaring Aktif
3.4E+08	0.34	-0.96054	0.343234	0.094681	0.014312	-0.02543	0.015921	-17.1027	-36.9237	-294.419	0.098299	-0.50528	0.280857	0.81003	Berbaring Aktif
3.6E+08	0.36	-0.94328	0.342909	0.100215	0.002943	-0.01535	0.012013	-17.1027	-36.9237	-291.902	0.098857	-0.50478	0.28148	0.81006	Berbaring Aktif

2. Berbaring Tidak Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	-0.24064	0.050881	0.221122	-0.01373	0.050013	-0.05702	28.94816	-28.1697	-209.42	-0.91259	-0.12734	-0.3827	0.067038	Berbaring Tidak Aktif
2000000	0.02	-0.4847	0.10583	0.444848	-0.00313	0.12756	-0.11807	29.61363	-29.0676	-209.42	-0.91265	-0.12551	-0.38321	0.066875	Berbaring Tidak Aktif
4000000	0.04	-0.74927	0.165011	0.665808	0.016178	0.224426	-0.17771	28.41578	-29.4043	-208.582	-0.91256	-0.12391	-0.38398	0.06668	Berbaring Tidak Aktif

60000 000	0.06	- 0.78 182	0.179 985	0.659 948	0.062 409	0.228 201	- 0.176 64	27.88 34	- 28.0 575	- 211. 098	- 0.91 228	- 0.12 274	- 0.38 505	0.066 404	Berba ring Tidak Aktif
80000 000	0.08	- 0.80 901	0.205 376	0.637 65	0.085 369	0.176 507	- 0.164 34	28.41 578	- 28.6 187	- 209. 42	- 0.91 184	- 0.12 223	- 0.38 631	0.066 054	Berba ring Tidak Aktif
1E+08	0.1	- 0.80 868	0.232 557	0.613 887	0.089 055	0.103 585	- 0.154 84	28.41 578	- 29.0 676	- 210. 539	- 0.91 136	- 0.12 219	- 0.38 755	0.065 589	Berba ring Tidak Aktif
1.2E+ 08	0.12	- 0.79 11	0.249 321	0.596 634	0.068 36	0.101 587	- 0.148 35	28.01 65	- 28.7 309	- 210. 818	- 0.91 087	- 0.12 24	- 0.38 872	0.065 013	Berba ring Tidak Aktif
1.4E+ 08	0.14	- 0.78 02	0.248 182	0.592 24	0.039 982	0.138 98	- 0.159 86	27.88 34	- 28.7 309	- 211. 098	- 0.91 04	- 0.12 26	- 0.38 987	0.064 401	Berba ring Tidak Aktif
1.6E+ 08	0.16	- 0.78 329	0.227 349	0.593 216	0.012 092	0.164 427	- 0.174 51	27.61 721	- 29.2 92	- 211. 098	- 0.90 992	- 0.12 255	- 0.39 108	0.063 897	Berba ring Tidak Aktif
1.8E+ 08	0.18	- 0.79 354	0.209 608	0.592 24	0.002 855	0.163 672	- 0.182 99	27.35 102	- 29.2 92	- 212. 216	- 0.90 944	- 0.12 227	- 0.39 234	0.063 518	Berba ring Tidak Aktif

2E+08	0.2	- 0.80 136	0.203 423	0.593 867	0.007 651	0.143 155	- 0.171 94	27.21 793	- 29.1 798	- 212. 216	- 0.90 897	- 0.12 191	- 0.39 36	0.063 18	Berba ring Tidak Aktif
2.3E+ 08	0.23	- 0.79 322	0.218 722	0.592 24	0.009 472	0.138 447	- 0.159 63	27.21 793	- 29.8 532	- 212. 216	- 0.90 85	- 0.12 177	- 0.39 48	0.062 724	Berba ring Tidak Aktif
2.5E+ 08	0.25	- 0.79 664	0.209 933	0.598 425	0.001 877	0.156 256	- 0.145 64	26.81 864	- 29.6 287	- 211. 098	- 0.90 804	- 0.12 148	- 0.39 601	0.062 301	Berba ring Tidak Aktif
2.7E+ 08	0.27	- 0.81 096	0.190 728	0.603 308	0.006 896	0.195 07	- 0.117 13	26.68 555	- 29.6 287	- 211. 098	- 0.90 759	- 0.12 095	- 0.39 724	0.062 011	Berba ring Tidak Aktif
3E+08	0.3	- 0.83 602	0.172 661	0.609 492	0.026 792	0.222 916	- 0.075 16	26.68 555	- 29.6 287	- 210. 818	- 0.90 715	- 0.12 026	- 0.39 85	0.061 853	Berba ring Tidak Aktif
3.2E+ 08	0.32	- 0.85 295	0.174 614	0.612 259	0.039 36	0.175 308	- 0.040 44	26.02 007	- 29.8 532	- 210. 818	- 0.90 669	- 0.11 962	- 0.39 975	0.061 694	Berba ring Tidak Aktif
3.5E+ 08	0.35	- 0.86 499	0.171 685	0.611 446	0.046 377	0.081 646	- 0.021 38	26.02 007	- 29.8 532	- 210. 818	- 0.90 623	- 0.11 9	- 0.40 1	0.061 557	Berba ring Tidak Aktif

3.7E+08	0.37	-0.87964	0.162896	0.59989	0.038649	-0.00811	0.006018	26.02007	-29.8532	-212.216	-0.90575	-0.11838	-0.40226	0.061468	Berbaring Tidak Aktif
3.9E+08	0.39	-0.87769	0.159966	0.590124	0.041003	-0.03089	0.059444	25.88698	-29.8532	-212.216	-0.90528	-0.1178	-0.40353	0.061376	Berbaring Tidak Aktif

3. Berdiri Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	0.141031	-0.29775	-0.00398	-0.00543	-0.0019	0.00134	5.78963	15.82443	-201.592	-0.36153	0.577664	-0.40335	0.610664	Berdiri Aktif
2000000	0.02	0.286944	-0.59616	-0.006	-0.01588	-0.0023	0.000415	7.253675	15.37551	-196	-0.36139	0.577831	-0.40328	0.610634	Berdiri Aktif
4000000	0.04	0.429602	-0.89993	-0.00444	-0.02086	-0.002418	0.00433	6.322011	14.70213	-192.924	-0.3613	0.577972	-0.4032	0.610611	Berdiri Aktif
6000000	0.06	0.429602	-0.90498	-0.005	-0.01273	-0.007214	0.005751	5.78963	14.36544	-195.44	-0.36125	0.578067	-0.40308	0.610625	Berdiri Aktif

80000 000	0.08	0.423 905	- 0.909 37	0.008 418	- 0.000 92	0.001 708	0.006 418	5.789 63	14.92 659	-196	- 0.361 25	0.578 118	- 0.402 99	0.610 636	Berdiri Aktif
1E+0 8	0.1	0.424 231	- 0.909 53	0.007 279	0.003 832	- 0.016 99	- 0.000 82	6.188 916	14.92 659	-196	- 0.361 24	0.578 174	- 0.402 91	0.610 644	Berdiri Aktif
1.2E+ 08	0.12	0.423 417	- 0.910 67	- 0.000 86	0.001 522	- 0.030 76	- 0.004 64	6.188 916	14.81 436	- 195.4 4	- 0.361 22	0.578 264	- 0.402 86	0.610 604	Berdiri Aktif
1.4E+ 08	0.14	0.425 533	- 0.911 32	- 0.003 3	0.002 544	- 0.034 31	- 0.003 97	5.789 63	14.81 436	- 195.4 4	- 0.361 19	0.578 373	- 0.402 8	0.610 555	Berdiri Aktif
1.6E+ 08	0.16	0.430 904	- 0.906 28	- 0.005 25	0.010 804	- 0.020 32	0.000 378	6.188 916	14.92 659	- 195.1 61	- 0.361 11	0.578 507	- 0.402 73	0.610 523	Berdiri Aktif
1.8E+ 08	0.18	0.434 81	- 0.897 49	- 0.000 86	0.012 847	- 0.008 02	- 0.000 87	6.854 391	15.03 882	- 195.1 61	- 0.361	0.578 637	- 0.402 62	0.610 54	Berdiri Aktif
2E+0 8	0.2	0.433 671	- 0.886 58	- 4.50E -05	0.005 697	- 0.005 13	- 0.001 89	6.854 391	14.81 436	- 195.1 61	- 0.360 87	0.578 776	- 0.402 48	0.610 572	Berdiri Aktif
2.3E+ 08	0.23	0.421 464	- 0.892 93	- 0.002 32	- 0.006 52	- 0.011 08	- 0.002 6	6.188 916	14.47 767	- 195.1 61	- 0.360 82	0.578 883	- 0.402 42	0.610 543	Berdiri Aktif
2.5E+ 08	0.25	0.422 278	- 0.908 56	- 0.007 21	- 0.001 54	- 0.006 82	0.003 353	6.188 916	14.47 767	- 195.1 61	- 0.360 79	0.578 989	- 0.402 4	0.610 474	Berdiri Aktif

2.7E+08	0.27	0.42358	-0.91637	-0.00509	0.015067	0.010545	0.010503	6.188916	14.47767	-195.161	-0.36078	0.579073	-0.40238	0.610412	Berdiri Aktif
2.9E+08	0.29	0.423905	-0.90579	-0.00167	0.021773	0.0117	0.009837	6.188916	14.14098	-195.161	-0.36075	0.579158	-0.40233	0.610381	Berdiri Aktif
3.1E+08	0.31	0.422441	-0.88951	-0.00167	0.003254	-0.01623	0.000822	6.721295	14.14098	-194.602	-0.36069	0.579261	-0.40226	0.610368	Berdiri Aktif
3.3E+08	0.33	0.434159	-0.89456	-0.00883	0.01789	0.03808	-0.0109	6.721295	14.70213	-194.322	-0.36056	0.579424	-0.40217	0.610343	Berdiri Aktif
3.6E+08	0.36	0.443599	-0.90856	-0.00591	0.00816	0.01766	-0.00069	6.854391	14.70213	-194.602	-0.36044	0.579575	-0.40207	0.610343	Berdiri Aktif
3.8E+08	0.38	0.43481	-0.9097	0.001582	0.012625	0.013077	0.011614	6.721295	14.70213	-194.322	-0.36037	0.579659	-0.40198	0.610359	Berdiri Aktif

4. Berdiri Tidak Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	0.02661	-0.32917	0.03997	0.044666	0.007823	0.01786	22.5596	7.631639	-115.754	-0.44964	0.44395	-0.49728	0.594517	Berdiri Tidak Aktif

2000 0000	0.02	0.040 199	- 0.65 784	0.075 219	0.092 663	0.011 782	0.037 275	21.89 413	7.070 489	- 119. 11	- 0.44 98	0.443 814	- 0.49 736	0.594 426	Berdiri Tidak Aktif
4000 0000	0.04	0.055 253	- 0.98 261	0.113 561	0.137 152	0.013 165	0.055 225	20.69 627	6.621 57	- 117. 432	- 0.45 003	0.443 666	- 0.49 743	0.594 311	Berdiri Tidak Aktif
6000 0000	0.06	0.048 254	- 0.97 106	0.110 143	0.126 315	0.016 585	0.060 687	22.02 722	7.519 41	- 114. 077	- 0.45 023	0.443 427	- 0.49 76	0.594 185	Berdiri Tidak Aktif
8000 0000	0.08	0.044 836	- 0.96 878	0.124 629	0.103 533	0.039 101	0.073 744	21.89 413	7.070 489	- 116. 593	- 0.45 056	0.443 079	- 0.49 769	0.594 125	Berdiri Tidak Aktif
1E+0 8	0.1	0.028 56	- 0.97 513	0.138 301	0.085 769	0.052 735	0.091 952	21.22 865	7.070 489	- 116. 593	- 0.45 11	0.442 54	- 0.49 777	0.594 051	Berdiri Tidak Aktif
1.2E +08	0.12	- 0.008 06	- 1.00 084	0.154 577	0.062 853	0.029 819	0.084 935	21.22 865	7.070 489	- 115. 195	- 0.45 191	0.441 683	- 0.49 8	0.593 877	Berdiri Tidak Aktif
1.4E +08	0.14	- 0.028 57	- 1.01 028	0.155 228	0.039 671	- 0.070 02	0.033 952	22.02 722	7.519 41	- 115. 195	- 0.45 278	0.440 745	- 0.49 834	0.593 622	Berdiri Tidak Aktif
1.6E +08	0.16	- 0.008 22	- 0.98 896	0.149 043	- 0.010 87	- 0.188 68	- 0.051 45	21.89 413	7.631 639	- 115. 475	- 0.45 353	0.439 968	- 0.49 858	0.593 428	Berdiri Tidak Aktif
1.8E +08	0.18	0.049 719	- 0.95 559	0.142 696	- 0.066 34	- 0.283 54	- 0.133 3	22.02 722	7.631 639	- 115. 195	- 0.45 387	0.439 63	- 0.49 843	0.593 552	Berdiri Tidak Aktif

2E+0 8	0.2	0.101 151	- 0.94 778	0.117 468	- 0.105 82	- 0.313 56	- 0.172 07	22.69 27	7.519 41	- 115. 475	- 0.45 364	0.439 884	- 0.49 815	0.593 766	Berdiri Tidak Aktif
2.2E +08	0.22	0.128 658	- 0.97 806	0.081 498	- 0.103 51	- 0.304 37	- 0.159 01	22.02 722	7.519 41	- 115. 475	- 0.45 321	0.440 423	- 0.49 803	0.593 803	Berdiri Tidak Aktif
2.4E +08	0.24	0.149 003	- 1.00 149	0.053 828	- 0.076 42	- 0.240 42	- 0.114 91	21.89 413	7.519 41	- 114. 636	- 0.45 267	0.441 066	- 0.49 799	0.593 765	Berdiri Tidak Aktif
2.6E +08	0.26	0.159 582	- 0.98 863	0.057 572	- 0.075	- 0.140 1	- 0.073 92	22.02 722	7.182 72	- 114. 636	- 0.45 209	0.441 727	- 0.49 789	0.593 793	Berdiri Tidak Aktif
2.8E +08	0.28	0.143 144	- 0.96 975	0.096 472	- 0.108 35	- 0.037 86	- 0.052 03	21.49 484	7.182 72	- 114. 636	- 0.45 164	0.442 241	- 0.49 762	0.593 989	Berdiri Tidak Aktif
3E+0 8	0.3	0.107 174	- 0.97 171	0.135 371	- 0.133 26	- 0.010 42	- 0.038 48	21.49 484	7.182 72	- 114. 636	- 0.45 153	0.442 355	- 0.49 722	0.594 324	Berdiri Tidak Aktif
3.2E +08	0.32	0.080 318	- 1.00 393	0.168 412	- 0.114 57	- 0.038 71	- 0.020 27	21.49 484	7.407 18	- 114. 636	- 0.45 184	0.441 994	- 0.49 674	0.594 756	Berdiri Tidak Aktif
3.4E +08	0.34	0.077 388	- 1.03 209	0.176 061	- 0.050 97	- 0.082 45	0.003 531	21.22 865	7.407 18	- 114. 916	- 0.45 222	0.441 588	- 0.49 627	0.595 159	Berdiri Tidak Aktif
3.6E +08	0.36	0.070 878	- 1.04 528	0.159 46	0.016 666	- 0.108 87	0.023 471	21.09 556	7.519 41	- 114. 916	- 0.45 252	0.441 299	- 0.49 604	0.595 336	Berdiri Tidak Aktif

5. Duduk Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	0.111245	-0.31305	0.086356	0.002654	0.027408	0.013686	13.37605	21.21147	-213.894	-0.43947	0.514108	-0.39431	0.622156	Duduk Aktif
2000000	0.02	0.211098	-0.61976	0.16604	0.009038	0.02284	0.026084	14.707	21.66039	-216.69	-0.43956	0.513187	-0.39367	0.623257	Duduk Aktif
4000000	0.04	0.325273	-0.91458	0.235795	0.006674	-0.0153	0.024004	13.90843	21.54816	-217.808	-0.43948	0.512539	-0.39299	0.624272	Duduk Aktif
6000000	0.06	0.346594	-0.897	0.212845	-0.013	-0.08427	0.002909	13.90843	21.54816	-216.41	-0.43909	0.512344	-0.39231	0.625135	Duduk Aktif
8000000	0.08	0.387773	-0.89716	0.201452	-0.02659	-0.0765	0.00966	13.90843	21.54816	-216.69	-0.43845	0.512512	-0.39154	0.625931	Duduk Aktif
1E+08	0.1	0.39233	-0.91084	0.210567	-0.01589	-0.00074	0.003398	14.44081	21.54816	-216.69	-0.43782	0.512606	-0.39076	0.626782	Duduk Aktif
1.2E+08	0.12	0.373938	-0.91035	0.222774	0.002055	0.045274	0.003753	13.90843	21.54816	-217.808	-0.43734	0.512483	-0.38998	0.627707	Duduk Aktif
1.4E+08	0.14	0.350663	-0.90188	0.228308	0.002499	0.019561	-0.01148	14.44081	21.54816	-218.368	-0.43701	0.51213	-0.38927	0.628661	Duduk Aktif

1.6E+08	0.16	0.365312	-0.89375	0.218379	-0.01638	-0.03786	-0.04039	14.44081	22.67046	-222.562	-0.43654	0.512009	-0.38854	0.629539	Duduk Aktif
1.8E+08	0.18	0.377193	-0.90172	0.212194	-0.01593	-0.03666	-0.03599	14.44081	22.67046	-222.562	-0.436	0.512017	-0.38785	0.630329	Duduk Aktif
2.1E+08	0.21	0.371334	-0.90725	0.205033	-0.00856	-0.00726	-0.02018	13.90843	21.99708	-219.486	-0.43549	0.51204	-0.38731	0.630996	Duduk Aktif
2.3E+08	0.23	0.350175	-0.90791	0.206009	-0.01056	-0.001619	-0.00375	14.30771	21.99708	-219.486	-0.43512	0.511902	-0.38691	0.631603	Duduk Aktif
2.6E+08	0.26	0.344804	-0.90742	0.208939	-0.03103	-0.00984	-0.002687	14.30771	21.99708	-217.249	-0.4348	0.511698	-0.38655	0.632211	Duduk Aktif
2.8E+08	0.28	0.350663	-0.90872	0.220983	-0.03059	-0.01104	-0.016277	13.77533	21.43593	-216.131	-0.4345	0.511431	-0.38606	0.632932	Duduk Aktif
3E+08	0.3	0.354732	-0.91035	0.224564	-0.0182	-0.003662	-0.032087	14.30771	21.43593	-216.131	-0.43419	0.511151	-0.38554	0.633687	Duduk Aktif
3.3E+08	0.33	0.352616	-0.90612	0.222774	-0.01153	-0.00433	-0.034041	14.30771	21.43593	-216.131	-0.43389	0.510881	-0.38505	0.634412	Duduk Aktif
3.5E+08	0.35	0.357987	-0.90807	0.213496	-0.02366	-0.02552	-0.03111	13.77533	20.87478	-215.572	-0.43353	0.510761	-0.38463	0.635012	Duduk Aktif

3.8E+08	0.38	0.356848	-0.90872	0.213496	-0.03378	-0.03569	0.028934	12.17819	20.31363	-204.108	-0.4332	0.510647	-0.38422	0.635579	Duduk Aktif
4E+08	0.4	0.355872	-0.90921	0.218542	-0.03445	-0.02685	0.033064	12.0451	19.75248	-204.108	-0.4329	0.510485	-0.38378	0.636171	Duduk Aktif

6. Duduk Tidak Aktif

time	seconds_elapsed	accel_x	accel_y	accel_z	gyro_x	gyro_y	gyro_z	mag_x	mag_y	mag_z	qw	qx	qy	qz	label
0	0	-0.00741	-0.3168	0.10719	-0.00583	-0.0055	0.000407	30.01292	13.91652	-117.432	-0.5146	0.328271	-0.46634	0.640274	Duduk Tidak Aktif
2000000	0.02	-0.01514	-0.63522	0.213891	-0.01272	-0.00585	0.003834	30.5453	14.70213	-119.948	-0.51418	0.328958	-0.46636	0.640243	Duduk Tidak Aktif
4000000	0.04	-0.02743	-0.95266	0.322709	-0.01997	-0.00598	0.002509	30.41221	14.81436	-120.228	-0.51381	0.329588	-0.46639	0.640192	Duduk Tidak Aktif
6000000	0.06	-0.03345	-0.95755	0.321407	-0.02681	-0.00651	0.003664	31.87625	13.57983	-117.152	-0.5134	0.330081	-0.46659	0.64012	Duduk Tidak Aktif
8000000	0.08	-0.03264	-0.95559	0.322383	-0.02686	-0.00229	0.002554	30.5453	14.70213	-118.55	-0.51306	0.330619	-0.4667	0.640041	Duduk Tidak Aktif

1E+08	0.1	- 0.03 264	- 0.95 478	0.321 081	- 0.028 86	0.009 168	0.008 638	31.21 078	14.70 213	- 119. 948	- 0.51 271	0.331 134	- 0.46 683	0.639 961	Duduk Tidak Aktif
1.2E+08	0.12	- 0.02 938	- 0.95 446	0.320 755	- 0.021 97	0.020 582	0.009 793	31.21 078	13.91 652	- 120. 228	- 0.51 236	0.331 666	- 0.46 692	0.639 903	Duduk Tidak Aktif
1.4E+08	0.14	- 0.03 345	- 0.95 657	0.321 244	- 0.024 41	0.029 242	0.006 64	31.21 078	13.91 652	- 120. 508	- 0.51 203	0.332 143	- 0.46 705	0.639 818	Duduk Tidak Aktif
1.6E+08	0.16	- 0.03 231	- 0.96 308	0.320 593	- 0.025 66	0.025 245	0.001 044	31.21 078	13.91 652	- 120. 508	- 0.51 17	0.332 632	- 0.46 719	0.639 727	Duduk Tidak Aktif
1.8E+08	0.18	- 0.03 182	- 0.96 243	0.323 034	- 0.027 35	0.024 446	0.002 865	31.21 078	13.80 429	- 120. 508	- 0.51 139	0.333 103	- 0.46 73	0.639 658	Duduk Tidak Aktif
2E+08	0.2	- 0.03 394	- 0.95 901	0.322 22	- 0.024 86	0.010 368	0.004 33	30.94 458	13.91 652	- 118. 55	- 0.51 108	0.333 53	- 0.46 743	0.639 581	Duduk Tidak Aktif
2.2E+08	0.22	- 0.02 938	- 0.95 511	0.320 918	- 0.028 01	0.003 795	0.002 865	30.94 458	13.80 429	- 118. 55	- 0.51 076	0.333 993	- 0.46 752	0.639 535	Duduk Tidak Aktif
2.4E+08	0.24	- 0.02 678	- 0.95 706	0.317 5	- 0.033 83	0.007 259	- 0.003 97	30.94 458	13.80 429	- 118. 55	- 0.51 041	0.334 485	- 0.46 762	0.639 48	Duduk Tidak Aktif
2.6E+08	0.26	- 0.02 238	- 0.96 276	0.321 732	- 0.031 56	0.026 133	- 0.007 84	31.47 697	14.70 213	- 119. 11	- 0.51 002	0.334 993	- 0.46 768	0.639 48	Duduk Tidak Aktif

2.8E+08	0.28	- 0.02 922	- 0.96 455	0.325 15	- 0.022 15	0.028 931	- 0.005 57	31.34 387	14.70 213	- 119. 11	- 0.50 97	0.335 401	- 0.46 777	0.639 456	Duduk Tidak Aktif
3E+08	0.3	- 0.03 573	- 0.96 194	0.329 87	- 0.012 91	0.012 988	- 0.009 84	31.47 697	13.91 652	- 120. 508	- 0.50 948	0.335 695	- 0.46 784	0.639 427	Duduk Tidak Aktif
3.2E+08	0.32	- 0.04 078	- 0.96 406	0.326 452	- 0.006 56	- 0.000 91	- 0.009 53	31.47 697	14.70 213	- 120. 787	- 0.50 929	0.335 947	- 0.46 799	0.639 332	Duduk Tidak Aktif
3.4E+08	0.34	- 0.04 208	- 0.95 918	0.325 964	0.000 545	- 0.002 47	- 0.014 1	31.34 387	14.25 321	- 120. 787	- 0.50 914	0.336 17	- 0.46 813	0.639 238	Duduk Tidak Aktif
3.6E+08	0.36	- 0.03 85	- 0.95 901	0.322 871	0.001 433	0.001 796	- 0.008 33	31.34 387	13.91 652	- 118. 83	- 0.50 892	0.336 426	- 0.46 829	0.639 157	Duduk Tidak Aktif

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Nama : Dela Almal Liani
NRP : 0922040009
Program Studi : D4 – Teknik Otomasi
Agama : Islam
Status : Belum Menikah
Alamat Asal : PTM Jl.Kenanga Raya Blok D19 N0.10, Wanasari,
Cibitung, Kab.Bekasi, Jawa Barat
Nomor Telepon : 0895411577380
Jenis Kelamin : Perempuan
Email : dealaalmaal@student.ppns.ac.id
Tempat, Tanggal Lahir : Brebes, 25 Juni 2004
Nama Orang Tua/Wali : Imam Budiyo
Alamat Orang Tua/Wali : PTM Jl.Kenanga Raya Blok D19 N0.10, Wanasari,
Cibitung, Kab.Bekasi, Jawa Barat
Riwayat Pendidikan

PENDIDIKAN FORMAL			
Pendidikan	Tahun	Tempat Pendidikan	Jurusan
Diploma 4	2022-sekarang	Politeknik Perkapalan Negeri Surabaya	Teknik Otomasi
SMA	2019-2022	SMAN 1 Cibitung	MIPA
SMP	2016-2019	SMPIT Permata Hati	-
SD	2010-2016	SDN Wanasari 09	-

(Halaman ini sengaja dikosongkan)