



DRAFT TUGAS AKHIR - SF234801

STUDI DAN PENGEMBANGAN SARUNG TANGAN PINTAR TUNAWICARA: MEMANFAATKAN SENSOR GYRO DAN FLEX DENGAN KLASIFIKASI DATA BERBASIS JARINGAN SARAF BUATAN

Muhammad Fauzan Ariyatmoko

NRP 5001211062

Dosen Pembimbing

Diky Anggoro, S.Si., M.Si

NIP 198508092014041003

Program Studi S1 Fisika

Departemen Fisika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025



DRAFT TUGAS AKHIR - SF234801

**STUDI DAN PENGEMBANGAN SARUNG TANGAN
PINTAR TUNAWICARA: MEMANFAATKAN SENSOR
GYRO DAN FLEX DENGAN KLASIFIKASI DATA
BERBASIS JARINGAN SARAF BUATAN**

Muhammad Fauzan Ariyatmoko

NRP 5001211062

Dosen Pembimbing

Diky Anggoro, S.Si., M.Si

NIP 198508092014041003

Program Studi S1 Fisika

Departemen Fisika

Fakultas Sains dan Analitika Data

Institut Teknologi Sepuluh Nopember

Surabaya

2025

[Halaman ini sengaja dikosongkan]



DRAFT FINAL PROJECT - SF234801

***STUDY AND DEVELOPMENT OF DIFABEL SMART
GLOVES: UTILIZING GYRO AND FLEX SENSORS
WITH ARTIFICIAL NEURAL NETWORKS BASED DATA
CLASSIFICATION***

Muhammad Fauzan Ariyatmoko

NRP 5001211062

Advisor

Diky Anggoro, S.Si., M.Si

NIP 198508092014041003

Undergraduate Study Program of Physics

Department of Physics

Faculty of Science and Data Analytics

Sepuluh Nopember Institute of Technology

Surabaya

2025

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Muhammad Fauzan Ariyatmoko / 5001211062
Departemen : Fisika
Dosen Pembimbing / NIP : Diky Anggoro, S.Si., M.Si / 198508092014041003

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "STUDI DAN PENGEMBANGAN SARUNG TANGAN PINTAR TUNAWICARA: MEMANFAATKAN SENSOR GYRO DAN FLEX DENGAN KLASIFIKASI DATA BERBASIS JARINGAN SARAF BUATAN" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, June 2025

Mengetahui
Dosen Pembimbing

Mahasiswa

Diky Anggoro, S.Si., M.Si
NIP. 198508092014041003

Muhammad Fauzan Ariyatmoko
NRP. 5001211062

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Muhammad Fauzan Ariyatmoko / 5001211062
Department : Physics
Advisor / NIP : Diky Anggoro, S.Si., M.Si / 198508092014041003

Hereby declared that the Final Project with the title of "*STUDY AND DEVELOPMENT OF DIFABEL SMART GLOVES: UTILIZING GYRO AND FLEX SENSORS WITH ARTIFICIAL NEURAL NETWORKS BASED DATA CLASSIFICATION*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Sepuluh Nopember Institute of Technology.

Surabaya, June 2025

Acknowledged

Advisor

Student

Diky Anggoro, S.Si., M.Si
NIP. 198508092014041003

Muhammad Fauzan Ariyatmoko
NRP. 5001211062

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama Mahasiswa : Muhammad Fauzan Ariyatmoko
Judul Tugas Akhir : STUDI DAN PENGEMBANGAN SARUNG TANGAN PINTAR TUNAWICARA: MEMANFAATKAN SENSOR GYRO DAN FLEX DENGAN KLASIFIKASI DATA BERBASIS JARINGAN SARAF BUATAN
Pembimbing : Diky Anggoro, S.Si., M.Si

Penelitian ini bertujuan mengembangkan sarung tangan pintar sebagai alat bantu komunikasi bagi penyandang disabilitas tunawicara yang menggunakan Sistem Isyarat Bahasa Indonesia (SIBI). Sarung tangan ini mengintegrasikan sensor giroskop IMU MPU6050 dan lima sensor flex untuk mendeteksi gerakan tangan yang kompleks, merepresentasikan 26 huruf dalam abjad SIBI. Tahap awal melibatkan kalibrasi sensor IMU MPU6050, di mana konfigurasi ke-7 terbukti optimal dengan nilai percepatan sumbu X sebesar $0.01g$, sumbu Y sebesar $0.00g$, dan sumbu Z sebesar $1.00g$, serta kecepatan sudut sumbu X sebesar $-0.01rad/s$, sumbu Y sebesar $0.00rad/s$, dan sumbu Z sebesar $0.00rad/s$, yang esensial untuk akurasi data orientasi dan gerakan tangan. Selanjutnya, dilakukan pengukuran sensitivitas pada kelima sensor flex. Hasil pengukuran menunjukkan hubungan linear yang kuat antara peningkatan resistansi dan sudut tekuk (rata-rata nilai $R^2 > 0.95$), serta hubungan linear terbalik yang konsisten antara penurunan tegangan keluaran dan peningkatan sudut tekuk (rata-rata nilai $R^2 > 0.98$). Karakteristik ini memastikan akuisisi data tekukan jari yang reliabel. Data gabungan dari sensor IMU dan sensor flex kemudian diproses dan diklasifikasikan menggunakan Jaringan Saraf Tiruan (*Artificial Neural Networks*) yang dirancang untuk mengenali gestur abjad SIBI. Evaluasi model jaringan saraf tiruan menunjukkan kinerja yang sangat baik: kurva akurasi pelatihan dan validasi mencapai nilai mendekati 100%, sementara kurva loss pelatihan dan validasi konvergen mendekati nol. Analisis confusion matrix juga memperlihatkan konsentrasi prediksi benar yang tinggi di sepanjang diagonal utama, didukung oleh nilai presisi, recall, dan F1-score yang tinggi (majoritas 0.99 atau 1.00) untuk sebagian besar kelas huruf. Hasil klasifikasi gestur yang dikenali ini kemudian diterjemahkan menjadi keluaran suara melalui modul audio yang terintegrasi dalam sistem sarung tangan. Dengan demikian, sarung tangan pintar ini diharapkan dapat secara efektif memfasilitasi komunikasi dua arah antara penyandang disabilitas tunawicara dan masyarakat yang tidak memahami bahasa isyarat, sekaligus memberikan kontribusi nyata pada kemajuan riset dan pengembangan teknologi asistif di Indonesia.

Kata Kunci : Sarung Tangan Pintar, Sensor Giroskop, Sensor Flex, Jaringan Saraf Tiruan, Sistem Isyarat Bahasa Indonesia (SIBI).

[Halaman ini sengaja dikosongkan]

ABSTRACT

Name : Muhammad Fauzan Ariyatmoko
Title : *STUDY AND DEVELOPMENT OF DIFABEL SMART GLOVES: UTILIZING GYRO AND FLEX SENSORS WITH ARTIFICIAL NEURAL NETWORKS BASED DATA CLASSIFICATION*
Advisors : Diky Anggoro, S.Si., M.Si

This research aims to develop a smart glove as a communication aid for individuals with speech impairments who use the Indonesian Sign Language System (SIBI). The glove integrates an IMU MPU6050 gyroscope sensor and five flex sensors to detect complex hand movements, representing the 26 letters of the SIBI alphabet. The initial phase involved the calibration of the IMU MPU6050 sensor, where the 7th configuration proved optimal, yielding acceleration values of 0.01g for the X-axis, 0.00g for the Y-axis, and 1.00g for the Z-axis, and angular velocity values of -0.01 rad/s for the X-axis, 0.00 rad/s for the Y-axis, and 0.00 rad/s for the Z-axis. These parameters are essential for accurate hand orientation and motion data. Subsequently, sensitivity measurements were performed on the five flex sensors. The measurement results indicated a strong linear relationship between resistance increase and bending angle (average $R^2 > 0.95$), and a consistent inverse linear relationship between output voltage decrease and bending angle increase (average $R^2 > 0.98$). These characteristics ensure reliable acquisition of finger flexion data. The combined data from the IMU and flex sensors were then processed and classified using an Artificial Neural Network (ANN) designed to recognize SIBI alphabet gestures. Evaluation of the ANN model demonstrated excellent performance: training and validation accuracy curves approached 100%, while training and validation loss curves converged towards zero. Confusion matrix analysis also revealed a high concentration of correct predictions along the main diagonal, supported by high precision, recall, and F1-score values (predominantly 0.99 or 1.00) for most letter classes. The classified recognized gestures were subsequently translated into voice output via an audio module integrated into the glove system. Thus, this smart glove is anticipated to effectively facilitate two-way communication between individuals with speech impairments and the community unfamiliar with sign language, while also making a tangible contribution to the advancement of assistive technology research and development in Indonesia.

Keywords : *Smart Glove, Gyro Sensor, Flex Sensor, Artificial Neural Network, Indonesian Sign Language System (SIBI)*.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat rahmat dan penyertaan-Nya penulis dapat menyelesaikan penelitiannya yang hasilnya tertuang dalam laporan penelitian dengan judul "**STUDI DAN PENGEMBANGAN SARUNG TANGAN PINTAR TUNAWICARA: MEMANFAATKAN SENSOR GYRO DAN FLEX DENGAN KLASIFIKASI DATA BERBASIS JARINGAN SARAF BUATAN**". Tentu penulis tidak bisa menyelesaikan ini tanpa dukungan berbagai pihak yaitu:

1. Seorang Ibu sebagai orang tua tunggal penulis yang tercinta, Dra. Endah Ariyani, yang selalu mendoakan dan mendukung putranya dalam kehidupannya.
2. Kedua kakak penulis Anisa Fadillah Widyan dan Dzacky Ariyatmoko yang memberikan bantuan moril dan materil demi adiknya meraih cita-citanya.
3. Bapak Dr. Diky Anggoro,S.Si,M.Si, selaku Dosen Pembimbing selama mengerjakan penelitian tugas akhir.
4. Bapak Dr. Lila Yuwana,S.Si, M.Si, sebagai Kepala Departemen Fisika ITS yang selalu mempermudah mahasiswanya untuk keperluan akademik dan non-akademik.
5. Encek Antonius Wijaya calon pengusaha *Crypto Currency* yang selalu menemani dalam pembuatan progres tugas akhir ini dengan kelakuan lawak dan atensinya menjadi orang yang kaya dan berdampak kelak bagi masyarakat, dia satu-satunya orang yang seefrekuensi dan sefuku, semoga kita berdua sukses untuk kedepannya.
6. Teman-teman Laboratorium Instrumentasi Elektronika yang bersama-sama berbagi cerita dan tawa, meskipun lebih banyak masalahnya yang dibagikan ke saya.
7. Semua pihak yang tidak bisa disebutkan oleh penulis.

Akhir kata, Semoga laporan penelitian ini dapat bermanfaat bagi pembaca yang membutuhkan. Penulis menyadari bahwa laporan penelitian ini masih jauh dari kesempurnaan, oleh karena itu penulis menerima kritik dan saran yang membangun demi perbaikan penelitian ini.

Surabaya, Juni 2025

Muhammad Fauzan Ariyatmoko

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR PROGRAM	xv
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Manfaat	2
2 TINJAUAN PUSTAKA	3
2.1 Penelitian Terdahulu	3
2.2 Sistem Isyarat Bahasa Indonesia (SIBI)	4
2.3 Komunikasi Penyandang Disabilitas Tunawicara	5
2.4 Teknologi Asistif Penerjemah Bahasa Isyarat	6
2.5 Sensor <i>Flex</i>	7
2.6 Sensor Giroskop IMU MPU6050	9
2.7 Mikrokontroler ESP32 DevkitC-V4	10
2.8 Jaringan Saraf Tiruan (<i>Neural Network</i>)	11
2.9 Teknik Evaluasi <i>Confusion Matrix</i>	12
2.9.1 <i>Precision</i>	12
2.9.2 <i>Recall</i>	13
2.9.3 <i>F1-Score</i>	13

2.9.4 Accuracy	13
3 METODE PENELITIAN	15
3.1 Metode yang digunakan	15
3.2 Alat dan Bahan	15
3.3 Perancangan Sarung Tangan Pintar	16
3.3.1 Perancangan Desain Rangkaian Pokok	16
3.3.2 Perancangan Desain <i>PCB</i>	17
3.3.3 Perancangan Desain Wadah Rangkaian	18
3.4 Perancangan Sistem	18
3.5 Alur Perencanaan Pengambilan Data dan Pemodelan	19
3.6 Alur Pelaksanaan Penelitian	20
3.7 Perencanaan Kalibrasi dan Pengukuran Sensitivitas Sensor	21
3.8 Realisasi Perencanaan Pembuatan Sarung Tangan Pintar	23
4 HASIL DAN PEMBAHASAN	25
4.1 Pengukuran Kalibrasi Sensor Giroskop IMU MPU6050	25
4.2 Pengukuran Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i>	27
4.2.1 Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i> ke-1	27
4.2.2 Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i> ke-2	28
4.2.3 Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i> ke-3	29
4.2.4 Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i> ke-4	30
4.2.5 Sensitivitas Resistansi terhadap Sudut pada Sensor <i>Flex</i> ke-5	30
4.3 Pengukuran Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i>	31
4.3.1 Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i> ke-1	31
4.3.2 Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i> ke-2	32
4.3.3 Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i> ke-3	33
4.3.4 Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i> ke-4	34
4.3.5 Sensitivitas Tegangan terhadap Sudut pada Sensor <i>Flex</i> ke-5	35
4.4 Evaluasi Klasifikasi Jaringan Saraf Tiruan Data Abjad SIBI	36
4.5 Pembahasan	39
5 PENUTUP	41
5.1 Kesimpulan	41
5.2 Saran	41

LAMPIRAN	53
DAFTAR PUSTAKA	67
BIOGRAFI PENULIS	69

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

2.1	Contoh gerakan abjad dalam SIBI (Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, 2025).	5
2.2	Bentuk fisik dan rangkaian dasar sensor <i>flex</i> (Ahmed et al., 2018).	7
2.3	Hubungan resistansi dengan sudut kelengkungan pada sensor <i>flex</i> (Syed Mubarak Ali et al., 2019).	8
2.4	Sensor MPU6050.	9
2.5	Detail konfigurasi mikrokontroler ESP32-DevKitC V4 (Espressif, 2025).	10
2.6	Contoh <i>Neural Network</i> (Goodfellow et al., 2020).	11
2.7	Contoh <i>Confusion Matrix</i> sebagai Teknik Evaluasi (Galván-Ruiz et al., 2023). .	12
3.1	Desain rangkaian pokok sarung tangan pintar	16
3.2	Desain skematik rangkaian sarung tangan pintar	17
3.3	Desain <i>PCB</i> sarung tangan pintar	17
3.4	Desain wadah rangkaian sarung tangan pintar	18
3.5	Diagram Blok Sistem Sarung Tangan Pintar	18
3.6	Alur perencanaan pengambilan dan pemodelan data	19
3.7	Alur pelaksanaan penelitian	20
3.8	cara pengukuran sudut sensor <i>flex</i> pada busur derajat	21
3.9	Pengukuran resistansi sensor <i>flex</i> pada multimeter	21
3.10	Pengukuran resistansi sensor <i>flex</i> pada serial monitor pemrograman Arduino .	22
3.11	Kalibrasi sensor giroskop IMU MPU6050 pada permukaan datar	22
3.12	Realisasi sarung tangan pintar	23
4.1	Pengambilan data sensor giroskop IMU MPU6050 setelah kalibrasi menggunakan konfigurasi ke-7.	25
4.2	Pengambilan data sensor giroskop IMU MPU6050 sebelum kalibrasi.	26
4.3	Rangkaian pembagi tegangan sensor <i>flex</i>	27
4.4	Pengukuran sensitivitas resistansi terhadap sudut pada sensor <i>flex</i> ke-1.	28
4.5	Pengukuran sensitivitas resistansi terhadap sudut pada sensor <i>flex</i> ke-2.	28
4.6	Pengukuran sensitivitas resistansi terhadap sudut pada sensor <i>flex</i> ke-3.	29
4.7	Pengukuran sensitivitas resistansi terhadap sudut pada sensor <i>flex</i> ke-4.	30
4.8	Pengukuran sensitivitas resistansi terhadap sudut pada sensor <i>flex</i> ke-5.	31

4.9 Pengukuran sensitivitas tegangan terhadap sudut pada sensor <i>flex</i> ke-1	32
4.10 Pengukuran sensitivitas tegangan terhadap sudut pada sensor <i>flex</i> ke-2	33
4.11 Pengukuran sensitivitas tegangan terhadap sudut pada sensor <i>flex</i> ke-3	33
4.12 Pengukuran sensitivitas tegangan terhadap sudut pada sensor <i>flex</i> ke-4	34
4.13 Pengukuran sensitivitas tegangan terhadap sudut pada sensor <i>flex</i> ke-5	35
4.14 Grafik akurasi pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI	36
4.15 Grafik loss pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI	37
4.16 Matrix confusion hasil pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI	38
5.1 Laporan klasifikasi jaringan syaraf tiruan data abjad SIBI pada program <i>python</i>	51
5.2 Laporan prediksi klasifikasi beberapa semple jaringan syaraf tiruan data abjad SIBI pada program <i>python</i>	51

DAFTAR TABEL

2.1	Analisis Komparatif Sistem Pengenalan Bahasa Isyarat Berbasis Sarung Tangan	6
2.2	Deskripsi Pin MPU6050	9
2.3	Konfigurasi Pin ESP32-DevKitC V4 (Espressif, 2025).	11
3.1	Perangkat keras penelitian pembuatan sarung tangan pintar	15
3.2	Perangkat lunak penelitian pembuatan sarung tangan pintar	15
3.3	Bahan dan komponen pada penelitian pembuatan sarung tangan pintar	16
5.1	Data Kalibrasi Sensor IMU untuk Berbagai Konfigurasi.	43
5.2	Data Kalibrasi Sensor Flex Resistansi terhadap Sudut Tekuk (Sensor 1-3).	43
5.3	Data Kalibrasi Sensor Flex Resistansi terhadap Sudut Tekuk (Sensor 4-5).	44
5.4	Data Kalibrasi Sensor Flex Tegangan terhadap Sudut Tekuk (Sensor 1-3).	44
5.5	Data Kalibrasi Sensor Flex Tegangan terhadap Sudut Tekuk (Sensor 4-5).	44
5.6	Data Huruf A (ClassID = 0) Sarung Tangan Pintar.	44
5.7	Data Huruf B (ClassID = 1) Sarung Tangan Pintar.	45
5.8	Data Huruf C (ClassID = 2) Sarung Tangan Pintar.	45
5.9	Data Huruf D (ClassID = 3) Sarung Tangan Pintar.	45
5.10	Data Huruf E (ClassID = 4) Sarung Tangan Pintar.	45
5.11	Data Huruf F (ClassID = 5) Sarung Tangan Pintar.	45
5.12	Data Huruf G (ClassID = 6) Sarung Tangan Pintar.	46
5.13	Data Huruf H (ClassID = 7) Sarung Tangan Pintar.	46
5.14	Data Huruf I (ClassID = 8) Sarung Tangan Pintar.	46
5.15	Data Huruf J (ClassID = 9) Sarung Tangan Pintar.	46
5.16	Data Huruf K (ClassID = 10) Sarung Tangan Pintar.	46
5.17	Data Huruf L (ClassID = 11) Sarung Tangan Pintar.	47
5.18	Data Huruf M (ClassID = 12) Sarung Tangan Pintar.	47
5.19	Data Huruf N (ClassID = 13) Sarung Tangan Pintar.	47
5.20	Data Huruf O (ClassID = 14) Sarung Tangan Pintar.	47
5.21	Data Huruf P (ClassID = 15) Sarung Tangan Pintar.	47
5.22	Data Huruf Q (ClassID = 16) Sarung Tangan Pintar.	48
5.23	Data Huruf R (ClassID = 17) Sarung Tangan Pintar.	48

5.24 Data Huruf S (ClassID = 18) Sarung Tangan Pintar.	48
5.25 Data Huruf T (ClassID = 19) Sarung Tangan Pintar.	48
5.26 Data Huruf U (ClassID = 20) Sarung Tangan Pintar.	48
5.27 Data Huruf V (ClassID = 21) Sarung Tangan Pintar.	49
5.28 Data Huruf W (ClassID = 22) Sarung Tangan Pintar.	49
5.29 Data Huruf X (ClassID = 23) Sarung Tangan Pintar.	49
5.30 Data Huruf Y (ClassID = 24) Sarung Tangan Pintar.	49
5.31 Data Huruf Z (ClassID = 25) Sarung Tangan Pintar.	49
5.32 Laporan klasifikasi pada setiap data huruf.	50

DAFTAR PROGRAM

5.1	Program Pengukuran Kalibrasi Sensor Giroskop IMU MPU6050	53
5.2	Program Pengukuran Sensitivitas Sensor Flex	54
5.3	Program Sarung Tangan Pintar	55
5.4	Model ANN Sarung Tangan Pintar dalam bahasa C++	59
5.5	Program Pembuatan Model ANN Sarung Tangan Pintar dalam bahasa Python .	61

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berdasarkan data dari *World Health Organization* (WHO) yang dirilis pada 27 Februari 2023, lebih dari 5 persen populasi dunia, atau sekitar 430 juta jiwa, memerlukan rehabilitasi akibat gangguan pendengaran dan berbicara. Dari jumlah tersebut, 432 juta adalah orang dewasa dan 34 juta adalah anak-anak (World-Health-Organization, 2024). Di Indonesia, khususnya di Jawa Timur, terdapat 5.021 penyandang tunawicara dan 4.482 penyandang tunarungu-wicara berdasarkan sensus Badan Pusat Statistik (BPS) Jawa Timur tahun 2019 (BPS-Jatim, 2019). Para penyandang disabilitas ini menghadapi hambatan komunikasi, dan bahasa isyarat menjadi solusi penting dalam interaksi sehari-hari.

Bahasa isyarat merupakan bentuk komunikasi visual non verbal yang menggunakan kombinasi gerakan manual dan non manual (Wu et al., 2016b). Isyarat non manual melibatkan ekspresi wajah, gerakan mulut, dan kepala, sementara isyarat manual terdiri dari gerakan tangan, jari, orientasi tangan, dan bentuk isyarat (Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, 2025). Setiap negara memiliki standar bahasa isyarat yang berbeda, seperti *American Sign Language* (ASL), *German Sign Language* (GSL), dan *Brazilian Sign Language* (Dias et al., 2022). Dalam aktivitas sehari-hari, penyandang tunarungu dan tunawicara sering-kali mengalami kesulitan berkomunikasi dengan orang lain yang tidak memahami bahasa isyarat. Oleh karena itu, diperlukan solusi untuk mengatasi fenomena permasalahan ini. Perangkat asistif (alat bantu) yang mampu menerjemahkan bahasa isyarat ke dalam suara, sehingga dapat meningkatkan kenyamanan dan produktivitas penyandang tunarungu dan tunawicara dalam bersosialisasi (Lee & Min, 2018).

Terdapat dua metode utama untuk mendeteksi gerakan tangan: berbasis visual menggunakan kamera dan berbasis non-visual menggunakan sensor. Metode berbasis visual rentan terhadap pengaruh lingkungan yang dapat memengaruhi kualitas gambar. Selain itu, pengguna harus selalu berada di depan kamera agar gerakan dapat terdeteksi secara akurat. Metode deteksi gerakan tangan berbasis sensor menjadi pilihan yang lebih baik dalam mewujudkan perangkat asistif. Salah satu perangkat asistif yang kurang banyak dikembangkan adalah sarung tangan cerdas (*smart-glove*) (Retnoningsih et al., 2023). Dengan penggunaan sarung tangan cerdas yang menggunakan berbagai sensor, seperti sensor *Gyroscope* dan *Fleks*, untuk mendeteksi gerakan isyarat dan melakukan klasifikasi untuk menghasilkan interpretasi Sistem Isyarat Bahasa Indonesia (SIBI), diharapkan dapat mempermudah komunikasi dua arah pada penyandang disabilitas tunawicara kepada masyarakat sekitar.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ditulis di atas, dapat dituliskan permasalahan yang akan dikaji pada penelitian ini, yaitu:

1. Keterbatasan komunikasi penyandang disabilitas tunawicara terhadap masyarakat yang minim dalam ilmu bahasa isyarat.
2. Kurangnya pembahasan prinsip kerja dan fisis instrumentasi sensor yang digunakan pada penelitian terdahulu.
3. Pengembangan riset teknologi pada perangkat asistif bagi para penyandang disabilitas tunawicara yang masih kurang di Indonesia.

1.3 Tujuan

Berdasarkan rumusan masalah yang ditulis pada bagian sebelumnya, dapat dituliskan tujuan yang akan dicapai pada penelitian ini, yaitu:

1. Membantu mengembangkan sistem komunikasi penyandang disabilitas tunawicara terhadap masyarakat yang minim dalam ilmu bahasa isyarat.
2. Menganalisis prinsip kerja dan fisis instrumentasi sensor.
3. Membuat perangkat asistif dengan harapan membantu perkembangan riset teknologi di Indonesia bagi para penyandang disabilitas tunawicara.

1.4 Batasan Masalah

Penelitian ini dibatasi oleh hal-hal berikut:

1. Sensor yang digunakan pada pembuatan alat asistif *Smart Glove* adalah sensor IMU (MPU6050) dan *Flex*.
2. Temperatur dianggap sebagai suhu ruangan yang konstan sehingga tidak ada perubahan nilai resistansi akibat perubahan suhu.
3. Pada penelitian ini hanya difokuskan menerjemahkan Abjad SIBI, sehingga penerjemahan angka maupun sistem isyarat lain belum diterapkan.
4. Penulis tidak akan mengkaji lebih dalam terkait karakteristik material sensor sehingga tidak dibahas dalam penelitian ini lebih lanjut.

1.5 Manfaat

Penulisan laporan tugas akhir ini diharapkan dapat memberikan manfaat antara lain:

1. Membantu aksesibilitas komunikasi yang lebih luas bagi para penyandang disabilitas tunawicara di Indonesia.
2. Menyediakan solusi teknologi alat asistif untuk menerjemahkan SIBI, sehingga mempermudah interaksi komunikasi penyandang disabilitas tunawicara di lingkungan masyarakat.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian tentang sarung tangan pintar (*smart glove*) sebagai penerjemah bahasa isyarat telah berkembang melalui integrasi beragam teknologi sensor dan algoritma pemrograman sebagai klasifikasi data untuk mendapatkan effisiensi dan akurasi maksimal. Pada penelitian yang dilakukan (Wu et al., 2016a) merancang sistem berbasis *Inertial Measurement Unit* (IMU) dan *surface electromyography* (sEMG) untuk mendeteksi gerakan dinamis *American Sign Language* (ASL) dengan akurasi 89%. Namun, sistem ini terbatas pada 15 kata dasar akibat kompleksitas pemrosesan sinyal sEMG. Di sisi lain, (Lee & Min, 2018) mengombinasikan sensor *flex*, IMU, dan *Force-Sensitive Resistor* (FSR) untuk mengenali 26 huruf ASL, mencapai akurasi 98,2% melalui kalibrasi berbasis *Support Vector Machine* (SVM). Kelemahan utamanya terletak pada ketergantungan terhadap tekanan jari yang konsisten, dimana hal tersebut pastinya sulit dipertahankan dalam penggunaan praktis bagi para penyandang disabilitas tunawicara.

Pada konteks bahasa isyarat Indonesia, penelitian yang dilakukan (Retnoningsih et al., 2023) telah mengembangkan sarung tangan dengan metode *smallest error detection* untuk Sistem Isyarat Bahasa Indonesia (SIBI), tetapi menghadapi kesalahan deteksi 60% pada huruf "N" akibat kemiripan pola tekukan jari manis dan kelingking. Studi komparatif oleh (Wungow et al., 2022) menunjukkan bahwa penerapan *Artificial Neural Network* (ANN) dan *K-Nearest Neighbors* (KNN) pada SIBI mampu meningkatkan akurasi hingga 93% untuk dataset terbatas, meski akurasi *real-time* turun menjadi 75% akibat *noise* sensor. Temuan ini sejalan dengan analisis (Haq et al., 2018) yang mengintegrasikan animasi GIF untuk visualisasi SIBI, tetapi hanya mencapai akurasi 85,24% dalam skenario antar-sesi karena variasi gestur pengguna yang menjadikannya sulit terkласifikasi.

Perkembangan *hardware* turut memengaruhi efektivitas sistem dalam pembuatan sarung tangan penerjemah bahasa isyarat. Pada penelitian yang dilakukan oleh (Darmawan et al., 2023) melakukan perancangan sarung tangan *detachable* dengan konfigurasi modular yang memungkinkan penggantian sensor *flex* 4,5 inci dan IMU 6-DOF (*Degree of Freedom*). Namun, integrasi modul *Bluetooth HC-05* (Electronica60Norte, 2025) meningkatkan konsumsi daya hingga 120 mA, membatasi portabilitas. Karakterisasi sensor yang dilakukan oleh (Saggio & Orengo, 2018) mengungkap bahwa sensor *flex* berbasis *conductive ink* mengalami *hysteresis* 12% setelah 5.000 siklus tekukan, yang menjelaskan penurunan akurasi 15% pada studi longitudinal (Widodo et al., 2020). Temuan ini memperkuat kebutuhan material sensor tahan aus seperti *FSR 402* (Interlink Electronics, 2025) yang mempertahankan linearitas $\pm 5\%$ hingga 10 juta siklus tekan. Akan tetapi hal tersebut tentunya tergantung pada kebutuhan penggunaan dan konfiguasi sistem yang dibuat.

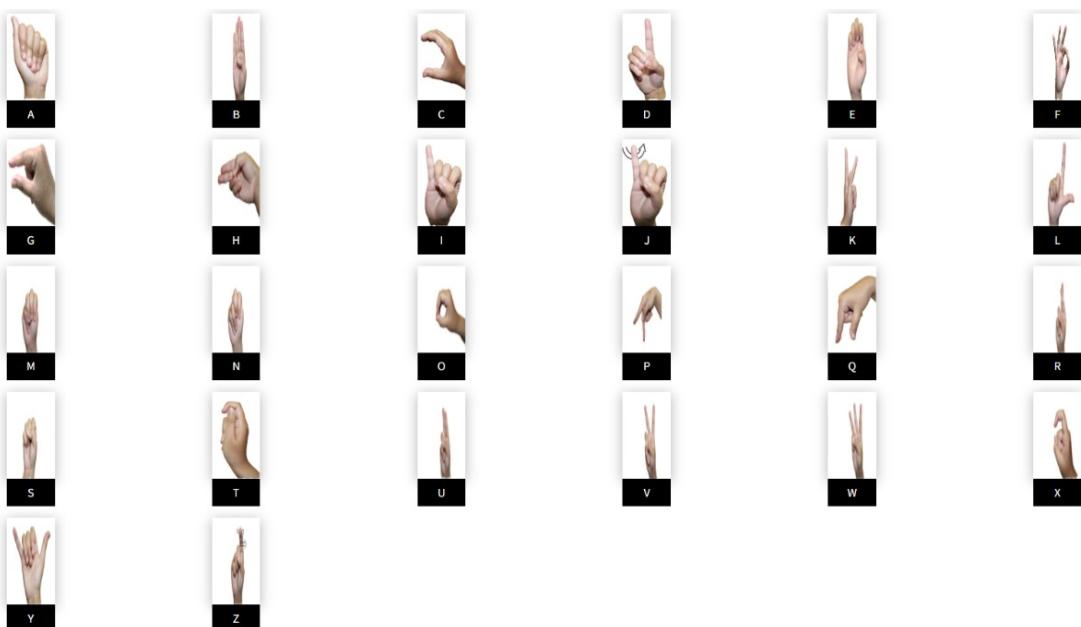
Inovasi algoritma pemrograman menjadi kunci peningkatan kinerja pada pembuatan sarung tangan penerjemah bahasa isyarat. Oleh (Chang, 2020) mengusulkan *sensor fusion* berbasis *Extended Kalman Filter* untuk menggabungkan data akselerometer dan giroskop pada IMU GY-952 (Store Siqma, 2025), mengurangi *drift* sudut dari 8° menjadi $1,2^\circ$. Namun, implementasinya pada mikrokontroler STM32F103C8T6 (Components101, 2023) memerlukan optimasi memori akibat keterbatasan RAM 20KB. Di sisi lain, penelitian yang dilakukan (Alrubayi et al., 2021) menerapkan *Convolutional Neural Network* (CNN) untuk gestur statis bahasa isyarat Malaysia, mencapai akurasi 95%. Integrasi dengan ekosistem *mobile* menghadapi tantangan platform atau media sebagai wadah dari *output* sistem yang telah dibuat. Adapun penelitian yang dilakukan (Bhat & Chayalakshmi, 2020) telah mengembangkan aplikasi Android berbasis *MIT App Inventor* untuk konversi teks-ke-suara, tetapi mengalami *latency* 2,3 detik pada jaringan 3G. Solusi *low-cost* yang dilakukan oleh (Shaheen & Mehmood, 2018) menggunakan sensor resistif kain mencapai akurasi 85% untuk 30 gestur dasar, tetapi gagal mendeteksi gerakan dinamis seperti huruf "J" SIBI yang memerlukan pelacakan lintasan jari. Dalam penelitian yang dilakukan oleh (Ahmed et al., 2018) ditinjau bahwa sistematisnya menyimpulkan bahwa 73% penelitian periode 2007-2017 terfokus pada huruf/angka statis, mengabaikan aspek semantik kalimat.

Tantangan utama terletak pada rekonsiliasi antara akurasi, biaya, dan kompleksitas algoritma pemrograman. Sehingga (Darmawan et al., 2023) memperkirakan biaya produksi *smart glove* modular mencapai \$85 hingga 4 kali lipat harga sarung tangan konvensional. Di sisi lain, optimasi *hardware* oleh (Shaheen & Mehmood, 2018) berhasil menekan biaya hingga \$12 dengan mengorbankan akurasi 15%. Paradoks ini memerlukan pendekatan multidisiplin seperti yang diusulkan (Alrubayi et al., 2021) melalui kolaborasi dengan komunitas tunarungu-wicara untuk menyusun *dataset* gestur kontekstual, sekaligus mengadopsi arsitektur *TinyML* untuk kompresi model ANN yang telah dilakukan (Suleman & Palupi, 2023).

2.2 Sistem Isyarat Bahasa Indonesia (SIBI)

Sistem Isyarat Bahasa Indonesia (SIBI) merupakan standar resmi bahasa isyarat yang dikembangkan oleh Kementerian Pendidikan dan Kebudayaan Republik Indonesia untuk memfasilitasi komunikasi penyandang tunarungu-wicara. Berdasarkan (Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, 2025), SIBI dirancang dengan struktur sistematis yang mencakup kosakata dasar, tata bahasa, dan konvensi isyarat yang terstandarisasi. Sistem ini menggunakan kombinasi gerakan tangan, ekspresi wajah, dan postur tubuh untuk merepresentasikan konsep linguistik, dengan pengelompokan isyarat berdasarkan kategori seperti *abjad, angka, kata kerja, dan kata sifat*.

Salah satu keunikan SIBI adalah penggunaan satu tangan dominan untuk membentuk abjad, berbeda dengan BISINDO yang lebih banyak menggunakan dua tangan, sehingga memudahkan proses pembelajaran dalam konteks pendidikan formal. Berikut merupakan macam-macam gerakan bahasa isyarat yang menginterpretasikan abjad dalam SIBI pada Gambar 2.1.



Gambar 2.1: Contoh gerakan abjad dalam SIBI (Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, 2025).

Implementasi SIBI dalam dunia pendidikan di Indonesia menghadapi beberapa tantangan, meskipun telah tersedia materi pembelajaran digital dan video panduan resmi pada laman tersebut. Analisis terhadap konten *website* menunjukkan bahwa SIBI belum sepenuhnya mengakomodasi variasi dialek regional, sehingga berpotensi menimbulkan kesenjangan komunikasi antardaerah. Namun, upaya integrasi teknologi seperti *smart glove* dengan kamus SIBI ter-digitalisasi (dalam bentuk *database* gerakan) menjadi solusi inovatif untuk meningkatkan aksesibilitas. Laman ini juga menyediakan program pelatihan guru dalam bentuk *workshop* dan modul interaktif, yang sejalan dengan penelitian yang dilakukan oleh (Retnoningsih et al., 2023) tentang pentingnya adaptasi teknologi asistif berbasis SIBI untuk mendukung pembelajaran inklusif. Pengembangan berkelanjutan melalui kolaborasi antara ahli bahasa, komunitas tunawicara, pakar teknologi dan teknisi biomedis diperlukan untuk memastikan evolusi SIBI yang responsif terhadap kebutuhan pengguna.

2.3 Komunikasi Penyandang Disabilitas Tunawicara

Penyandang disabilitas tunarungu dan tunawicara memiliki tantangan komunikasi yang berbeda. Tunarungu mengalami gangguan pendengaran, mulai dari ringan (masih bisa menggunakan bahasa lisan atau alat bantu) hingga berat (bergantung pada bahasa isyarat). Sementara tunawicara adalah individu yang kesulitan berbicara, tetapi umumnya masih mampu mendengar. Keduanya sering menggunakan bahasa isyarat sebagai metode komunikasi utama, menganalisa gerakan tangan, tubuh, dan ekspresi wajah untuk menyampaikan pesan. Namun, interaksi dengan masyarakat non-disabilitas kerap terhambat karena minimnya pemahaman terhadap bahasa isyarat, sehingga memerlukan bantuan penerjemah. Khusus bagi tunawicara, meski dapat mendengar lawan bicara, ketidakmampuan berbicara mengharuskan mereka menggunakan isyarat atau alat bantu teks, yang tidak selalu efektif dalam komunikasi spontan (Retnoningsih et al., 2023).

Saat ini, solusi umum untuk memfasilitasi komunikasi tunawicara adalah penerjemah bahasa isyarat dan teks tertulis. Namun, penerjemah manusia memiliki keterbatasan aksesibilitas dan biaya tinggi, sementara komunikasi via teks (seperti tulisan di kertas) cenderung lambat dan tidak praktis. Teknologi seperti aplikasi smartphone yang mengonversi bahasa isyarat ke suara dan sebaliknya menjadi alternatif inovatif. Sistem ini memungkinkan tunawicara menggunakan isyarat yang diterjemahkan ke suara digital, sementara lawan bicara merespons melalui fitur pengenalan ucapan. Pendekatan ini tidak hanya mengurangi ketergantungan pada penerjemah manusia tetapi juga mendorong kemandirian dan partisipasi aktif dalam interaksi dua arah.

Tantangan pengembangan utama terletak pada aksesibilitas teknologi yang terjangkau dan edukasi masyarakat tentang bahasa isyarat. Dengan kombinasi antara inovasi teknologi (seperti aplikasi interpretasi real-time) dan sosialisasi bahasa isyarat, hambatan komunikasi dapat diminimalisasi. Hal ini tidak hanya memberdayakan penyandang tunawicara untuk berkomunikasi secara mandiri tetapi juga membuka peluang kolaborasi yang setara antara disabilitas dan non-disabilitas dalam berbagai aspek kehidupan (Retnoningsih et al., 2023).

2.4 Teknologi Asistif Penerjemah Bahasa Isyarat

Teknologi asistif mempunyai definisi sebuah perangkat, sistem ataupun metode yang dirancang untuk meningkatkan kemampuan fungsional individu disabilitas dalam menjalankan aktivitas kehidupan sehari-hari (ADL), pendidikan, pekerjaan dan interaksi sosial, dengan memanfaatkan prinsip adaptasi, kompensasi maupun rehabilitasi (Damastuti, 2021).

Penelitian teknologi asistif yang dilakukan oleh (Retnoningsih et al., 2023) bertujuan mengatasi masalah yang timbul dari metode tradisional interpretasi bahasa isyarat sebagai media komunikasi para disabilitas tunawicara, seperti penerjemah bahasa isyarat serta penulisan teks secara manual. Kemajuan teknologi asistif penerjemah bahasa isyarat dapat dibagi menjadi dua jenis, yakni berbasis visual menggunakan kamera dan berbasis non-visual menggunakan sarung tangan pintar yang memiliki sensor instrumentasi.

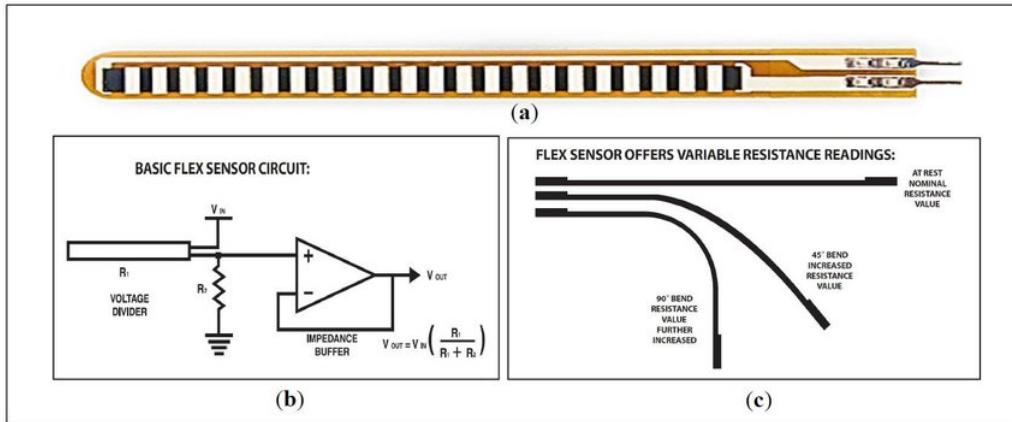
Berbagai penelitian yang telah dilakukan menunjukkan adanya perbedaan yang mencolok dalam aspek bahasa isyarat yang dianalisis, jenis sensor yang dimanfaatkan, metode yang diterapkan untuk menginterpretasikan gerakan isyarat, serta teknik klasifikasi pemrogramman yang digunakan. Adapun pada Tabel 2.1 menjelaskan rangkuman analisis komparatif sistem pengenalan bahasa isyarat berbasis sarung tangan yang telah dilakukan oleh beberapa penelitian terdahulu.

Tabel 2.1: Analisis Komparatif Sistem Pengenalan Bahasa Isyarat Berbasis Sarung Tangan

Penulis (Tahun)	Tipe Sensor	Metode Pengenalan	Bahasa	Akurasi (%)
Retnoningsih et al., 2023	Sensor fleksibel, FSR	DTW	SIBI	89.7
Wu et al., 2016a	IMU, sEMG	HMM	ASL	91.3
Haq et al., 2018	Sensor fleksibel	KNN	SIBI	87.2
Wungow et al., 2022	Sensor fleksibel, akcelerometer	KNN/ANN	SIBI	88.4/90.1
Darmawan et al., 2023	Multi-sensor modular	Hybrid KNN	SIBI	92.3

2.5 Sensor Flex

Flex sensor adalah sensor berbasis resistif yang digunakan untuk mendeteksi tingkat kelengkungan suatu permukaan, terutama dalam aplikasi seperti deteksi gerakan jari dalam bahasa isyarat. Sensor ini bekerja berdasarkan perubahan resistansi akibat perubahan bentuk fisiknya.



Gambar 2.2: Bentuk fisik dan rangkaian dasar sensor *flex* (Ahmed et al., 2018).

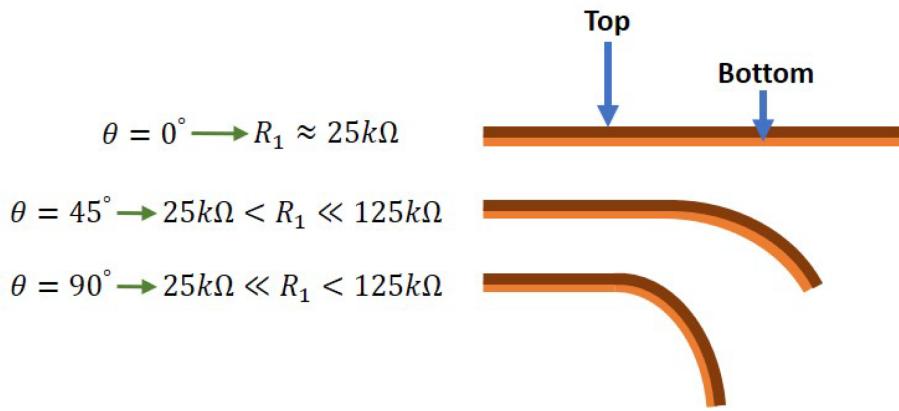
Pada Gambar 2.2(a) menunjukkan tampilan fisik sensor *flex*, yang terdiri dari substrat fleksibel dengan elemen karbon resistif di dalamnya. Sensor ini memiliki desain yang tipis dan ringan, sehingga nyaman digunakan dalam aplikasi yang membutuhkan mobilitas tinggi. Saat sensor ditekuk, resistansinya akan berubah, memungkinkan sistem untuk mendeteksi dan mengukur derajat tekukan.

Lalu pada Gambar 2.2(b) menggambarkan rangkaian elektronik dasar sensor *flex*, yang menggunakan konfigurasi persamaan pembagi tegangan dengan resistor eksternal sebagai pengimbang (R_1). Nilai resistansi sensor berubah sesuai dengan derajat tekukan, dan nilai tegangan keluarannya dihitung menggunakan persamaan 2.1.

$$V_{out} = V_{in} \cdot \frac{R_{flex}}{R_{flex} + R_1} \quad (2.1)$$

Sedangkan pada Gambar 2.2(c) menunjukkan hubungan antara sudut tekukan dan resistansi. Ketika sensor dalam posisi lurus, resistansinya rendah, sementara saat ditekuk hingga 45° atau 90°, resistansi meningkat secara bertahap. Karakteristik ini menjadikan sensor *flex* sangat berguna dalam berbagai aplikasi, termasuk penerjemahan bahasa isyarat dengan mendeteksi gerakan jari.

Sesuai pada Gambar 2.3 yang menunjukkan hubungan antara resistansi sensor *flex* dengan sudut kelengkungan. Sensor *flex* terdiri dari dua lapisan utama, yaitu lapisan atas (*Top layer*) dan lapisan bawah (*Bottom layer*). Lapisan atas mengandung material resistif, seperti tinta karbon atau polimer konduktif, yang sifat resistansinya berubah ketika sensor mengalami tekukan. Material ini sangat responsif terhadap tekanan dan regangan, sehingga ketika sensor ditekuk, resistansinya akan meningkat. Sementara itu, lapisan bawah berfungsi sebagai substrat pendukung yang menjaga fleksibilitas sensor serta menyalurkan perubahan hambatan ke dalam rangkaian listrik. Kedua lapisan ini dipisahkan oleh bahan dielektrik atau insulator tipis untuk mencegah hubungan langsung yang dapat menyebabkan hubungan singkat (*short circuit*).



Gambar 2.3: Hubungan resistansi dengan sudut kelengkungan pada sensor *flex* (Syed Mubarak Ali et al., 2019).

Dimana secara fisis dapat dibahas menggunakan persamaan (Syed Mubarak Ali et al., 2019).

$$R_{flex} = 100 \cdot \frac{\theta}{180} + 25 \quad (2.2)$$

Pada persamaan 2.2 dapat disubtitusikan pada persamaan 2.1, sehingga didapatkan persamaan yang dapat menjelaskan secara utuh fisis yang terjadi pada sensor flex dibawah ini.

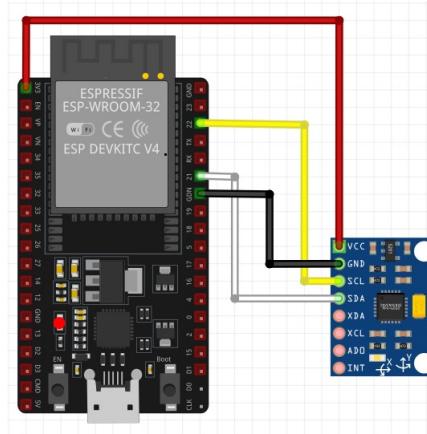
$$V_{out} = V_{in} \cdot \frac{100\theta + 4500}{100\theta + 180(25 + R_1)} \quad (2.3)$$

Hal ini menunjukkan adanya hubungan linier antara θ dan V_{out} pada persamaan 2.3. Ketika sensor dalam keadaan lurus (0°), partikel karbon atau material resistif dalam lapisan atas tersebar secara merata, sehingga menghasilkan resistansi rendah sekitar $25 \text{ k}\Omega$. Namun, ketika sensor mulai melengkung pada sudut tertentu, seperti 45° hingga 90° , perubahan fisik ini memengaruhi distribusi partikel dalam lapisan resistif. Lapisan atas yang meregang menyebabkan partikel konduktifnya semakin menjauh satu sama lain, sehingga jalur konduksi listrik menjadi lebih panjang dan mengakibatkan peningkatan resistansi. Sebaliknya, lapisan bawah mengalami kompresi, tetapi efeknya terhadap resistansi lebih kecil dibandingkan dengan regangan pada lapisan atas. Semakin besar sudut tekukan, semakin tinggi pula resistansinya, yang bisa meningkat dari $25 \text{ k}\Omega$ hingga mencapai $125 \text{ k}\Omega$ atau lebih tergantung pada spesifikasi sensor (Syed Mubarak Ali et al., 2019).

Korelasi antara lapisan atas dan bawah terhadap perubahan resistansi sangat bergantung pada mekanisme regangan dan kompresi yang terjadi saat sensor ditekuk. Regangan pada lapisan atas menyebabkan peningkatan resistansi karena pergerakan elektron menjadi lebih sulit akibat jarak antar partikel konduktif yang membesar. Sebaliknya, kompresi pada lapisan bawah tidak memberikan dampak signifikan terhadap resistansi karena tidak menghambat jalur konduksi secara langsung. Oleh karena itu, semakin besar sudut tekukan, semakin besar pula nilai resistansi yang dihasilkan (Syed Mubarak Ali et al., 2019).

2.6 Sensor Giroskop IMU MPU6050

Sensor IMU (*Inertial Measurement Unit*) MPU6050 adalah modul yang menggabungkan accelerometer 3-sumbu dan gyroscope 3-sumbu dalam satu chip. Sensor ini mampu mengukur percepatan linier (pada sumbu X, Y, Z) menggunakan accelerometer dan kecepatan sudut (rotasi pada sumbu X, Y, Z) menggunakan gyroscope atau yang biasanya disebut dengan *yaw*, *pitch* dan *roll*. MPU6050 juga dilengkapi dengan sensor suhu dan antarmuka komunikasi I^2C (Inter-Integrated Circuit) untuk berinteraksi dengan mikrokontroler seperti Arduino atau ESP32. Sensor ini banyak digunakan dalam aplikasi robotika, drone, sistem navigasi, dan perangkat yang memerlukan pemantauan gerak. Keunggulannya terletak pada akurasi tinggi, konsumsi daya rendah, dan kemampuan kalibrasi internal.



Gambar 2.4: Sensor MPU6050.

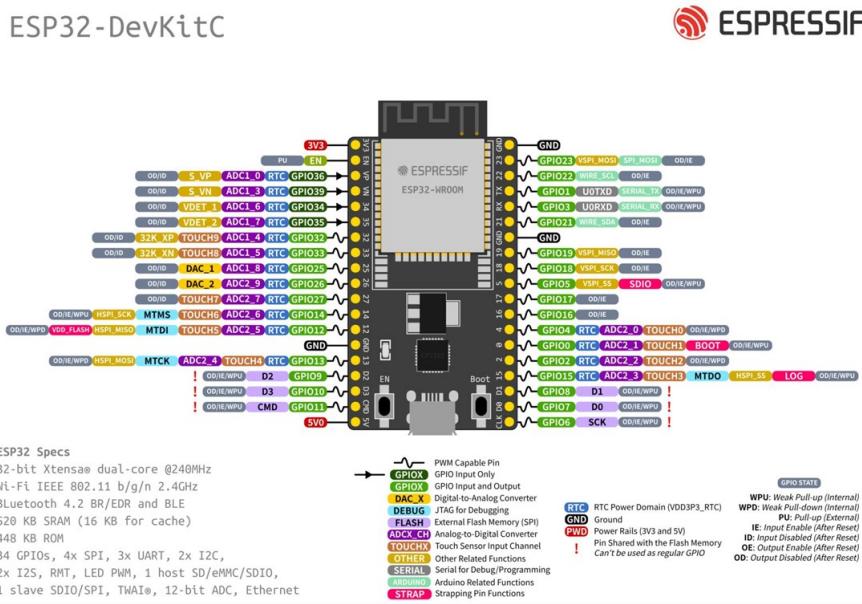
Tabel 2.2: Deskripsi Pin MPU6050

No. Pin	Nama Pin	Fungsi
1	VCC	Sumber daya (3.3V atau 5V)
2	GND	Ground
3	SCL	Clock untuk komunikasi I^2C
4	SDA	Data untuk komunikasi I^2C
5	XDA	Data I^2C tambahan (eksternal sensor)
6	XCL	Clock I^2C tambahan (eksternal sensor)
7	AD0	Alamat I^2C seleksi (low: 0x68, high: 0x69)
8	INT	Interrupt output (data siap dibaca)

Sesuai dengan Tabel 2.2 sensor MPU6050 memiliki 8 pin, berfungsi menghubungkan modul dengan sistem eksternal. Pin utama meliputi VCC (daya), GND (*ground*), SCL (*clock* I^2C), dan SDA (data I^2C) untuk komunikasi. Pin AD0 digunakan untuk mengubah alamat I^2C jika diperlukan, sementara pin INT (Interrupt) memberikan sinyal ketika data siap dibaca. Pin XDA dan XCL berfungsi sebagai antarmuka I^2C tambahan untuk menghubungkan sensor eksternal (misalnya magnetometer). Konfigurasi pin ini memungkinkan integrasi fleksibel dengan berbagai mikrokontroler dan perangkat pendukung (InvenSense, 2022).

2.7 Mikrokontroler ESP32 DevkitC-V4

ESP32-DevKitC V4 adalah sebuah mikrokontroler yang dikembangkan oleh Espressif Systems yang didasarkan pada chipset ESP32-WROOM. Mikrokontroler ini memiliki arsitektur dual-core Xtensa berkecepatan 240MHz dan mendukung koneksi Wi-Fi 802.11 serta Bluetooth 4.2 BLE. Dengan kapasitas memori 520KB SRAM dan 448KB ROM, perangkat ini mampu menjalankan berbagai aplikasi IoT, komunikasi nirkabel, serta sistem tertanam yang membutuhkan daya pemrosesan tinggi. ESP32 juga mendukung berbagai protokol komunikasi seperti UART, SPI, I2C, dan memiliki beberapa kanal ADC serta DAC yang memungkinkan integrasi dengan sensor dan aktuator (Espressif, 2025).



Gambar 2.5: Detail konfigurasi mikrokontroler ESP32-DevKitC V4 (Espressif, 2025).

ESP32-DevKitC V4 memiliki berbagai pin masukan dan keluaran yang mencakup fungsi GPIO, ADC, DAC, komunikasi serial, serta fungsi spesifik lainnya. Adapun Tabel 2.3 yang menjelaskan masing-masing pin berdasarkan Gambar 2.5 yang telah diberikan. Selain fungsi-fungsi pin yang tertera pada Tabel 2.3, terdapat beberapa pin spesial seperti EN (Enable), BOOT (untuk mode flashing), serta pin GND dan 3.3V yang digunakan untuk penyediaan daya.

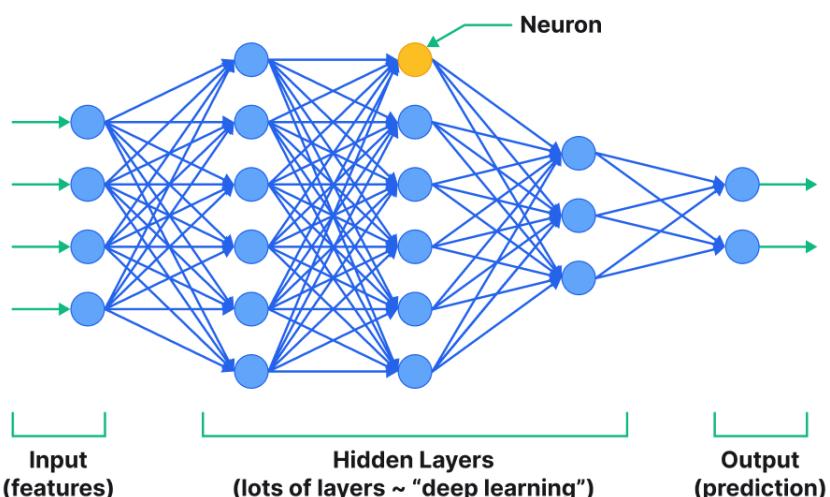
Mikrokontroler ESP32-DevKitC V4 memiliki berbagai keunggulan dibandingkan dengan mikrokontroler lain di kelasnya. Beberapa kelebihannya antara lain adalah performa tinggi dengan prosesor dual-core, dukungan konektivitas Wi-Fi dan Bluetooth yang kuat, serta ketersediaan banyak GPIO dan fitur ADC/DAC untuk aplikasi berbasis sensor. Selain itu, ESP32 memiliki komunitas pengguna yang luas serta dokumentasi yang baik, yang memudahkan pengembang dalam mengimplementasikan proyek berbasis IoT dan sistem tertanam (Espressif, 2025).

Tabel 2.3: Konfigurasi Pin ESP32-DevKitC V4 (Espressif, 2025).

Pin	Fungsi Utama	Tipe I/O	Keterangan
GPIO36	ADC1_0	Input	ADC Channel 1
GPIO39	ADC1_3	Input	ADC Channel 1
GPIO34	ADC1_6	Input	ADC Channel 1
GPIO35	ADC1_7	Input	ADC Channel 1
GPIO32	ADC1_4, Touch9	I/O	ADC dan Touch Sensor
GPIO33	ADC1_5, Touch8	I/O	ADC dan Touch Sensor
GPIO25	DAC1, ADC2_8	I/O	Digital ke Analog Converter
GPIO26	DAC2, ADC2_9	I/O	Digital ke Analog Converter
GPIO12	HSPI_MISO, ADC2_5	I/O	SPI dan ADC Channel 2
GPIO14	HSPI_SCK, ADC2_6	I/O	SPI dan ADC Channel 2
GPIO13	HSPI_MOSI, ADC2_4	I/O	SPI dan ADC Channel 2
GPIO23	VSPI_MOSI	I/O	SPI Master Out Slave In
GPIO22	I2C SCL	I/O	Serial Clock untuk I2C
GPIO21	I2C SDA	I/O	Serial Data untuk I2C
GPIO1	UART TXD0	Output	Serial Data Transmit
GPIO3	UART RXD0	Input	Serial Data Receive
GPIO5	VSPI_SS	Output	SPI Chip Select
GPIO18	VSPI_SCK	Output	SPI Clock

2.8 Jaringan Saraf Tiruan (*Neural Network*)

Jaringan saraf tiruan (*Neural Network*) adalah model matematis yang terinspirasi dari cara kerja otak manusia. Jaringan saraf tiruan dirancang untuk meniru cara otak manusia memproses informasi, menggunakan unit-unit sederhana yang disebut *neuron*. Menurut (Goodfellow et al., 2020), jaringan saraf tiruan sangat efektif untuk tugas-tugas seperti pengenalan pola dan pengambilan keputusan karena kemampuan mereka untuk mempelajari hubungan kompleks dalam data seperti klasifikasi data, regresi, dan prediksi. *Neuron* adalah unit dasar dalam



Gambar 2.6: Contoh *Neural Network* (Goodfellow et al., 2020).

jaringan saraf tiruan yang bertugas untuk memproses informasi. *Neuron* bekerja dengan menerima sinyal atau data, melakukan perhitungan, dan menghasilkan output yang digunakan untuk membuat keputusan atau prediksi. Jaringan ini terdiri dari beberapa lapisan neuron yang saling terhubung dan informasi mengalir melalui lapisan-lapisan ini untuk menyelesaikan tugas, seperti mengenali pola, mengklasifikasikan data, atau membuat prediksi berdasarkan data yang diberikan.

Jaringan saraf ini terdiri dari tiga tipe utama lapisan: *input layer* (lapisan input), *hidden layer* (lapisan tersembunyi), dan *output layer* (lapisan output). *Input layer* menerima data awal, *hidden layer* memproses data tersebut melalui berbagai *neuron*, dan *output layer* menghasilkan hasil akhir berdasarkan pemrosesan yang dilakukan oleh lapisan-lapisan sebelumnya. Setiap *neuron* dalam jaringan saraf ini berfungsi untuk menghitung nilai berdasarkan bobot yang diberikan dan kemudian meneruskan hasilnya ke *neuron* lain dalam jaringan.

2.9 Teknik Evaluasi *Confusion Matrix*

Dalam bidang *Machine Learning* dan *Deep Learning*, *Confusion Matrix* merupakan teknik evaluasi yang sering digunakan untuk menilai performa suatu model klasifikasi data. *Confusion Matrix* menyajikan perbandingan antara hasil prediksi model dengan nilai sebenarnya dalam bentuk matriks yang berisi empat elemen utama, yaitu:

		Prediction	
		TN	FP
Reality	TN		
	FN		TP

Gambar 2.7: Contoh *Confusion Matrix* sebagai Teknik Evaluasi (Galván-Ruiz et al., 2023).

- **True Positive (TP)**: Jumlah data positif yang diprediksi benar oleh model.
- **True Negative (TN)**: Jumlah data negatif yang diprediksi benar oleh model.
- **False Positive (FP)**: Jumlah data negatif yang salah diklasifikasikan sebagai positif.
- **False Negative (FN)**: Jumlah data positif yang salah diklasifikasikan sebagai negatif.

Dari *Confusion Matrix* ini, berbagai metrik evaluasi dapat dihitung untuk menilai performa model, seperti *Precision*, *Recall*, *F1-Score* dan *Accuracy*.

2.9.1 *Precision*

Precision mengukur seberapa banyak prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dibuat oleh model. Persamaannya dinyatakan sebagai:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.4)$$

Metrik ini berguna ketika kesalahan klasifikasi positif (*False Positive*) memiliki dampak yang besar, misalnya dalam diagnosis penyakit kritis.

2.9.2 Recall

Recall mengukur seberapa banyak data positif yang berhasil diidentifikasi dengan benar oleh model. Persamaannya diberikan oleh:

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

Metrik ini penting dalam situasi di mana kehilangan data positif (*False Negative*) lebih berbahaya daripada kesalahan klasifikasi negatif.

2.9.3 F1-Score

F1-Score merupakan rata-rata harmonik dari *Precision* dan *Recall*, memberikan keseimbangan antara keduanya. Persamaannya ditulis sebagai:

$$F1 - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (2.6)$$

Nilai *F1-Score* tinggi menunjukkan keseimbangan antara *Precision* dan *Recall*, terutama dalam dataset yang tidak seimbang.

2.9.4 Accuracy

Accuracy mengukur seberapa sering model memberikan prediksi yang benar dibandingkan dengan keseluruhan data. Persamaannya adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.7)$$

Meskipun *Accuracy* sering digunakan sebagai metrik utama, namun dalam kasus dataset yang tidak seimbang, metrik lain seperti *Precision*, *Recall*, dan *F1-Score* lebih direkomendasikan (Galván-Ruiz et al., 2023).

[Halaman ini sengaja dikosongkan]

BAB III

METODE PENELITIAN

3.1 Metode yang digunakan

Penelitian ini diawali dengan studi literatur untuk mencari referensi yang nantinya akan diimplementasikan dalam pengerjaan penelitian. Pada bab ini terdiri atas perancangan sarung tangan pintar, perancangan sistem, perencanaan pengambilan data dan pemodelan, perencanaan kalibrasi dan pengukuran sensitivitas sensor, alur pelaksanaan penelitian dan realisasi perencanaan pembuatan sarung tangan.

3.2 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini terdiri dari tiga jenis, yakni perangkat lunak, perangkat keras, dan bahan maupun komponen instrumentasi. Berikut merupakan daftar tabel perangkat keras yang digunakan dalam penelitian pembuatan sarung tangan pintar.

Tabel 3.1: Perangkat keras penelitian pembuatan sarung tangan pintar

No	Perangkat Keras	Jumlah
1	Tang Krimping	1 buah
2	Solder	1 buah
3	Penyedot Timah	1 buah
4	Roll Timah	1 buah

Adapun berikut merupakan daftar tabel perangkat lunak yang digunakan dalam penelitian pembuatan sarung tangan pintar.

Tabel 3.2: Perangkat lunak penelitian pembuatan sarung tangan pintar

No	Perangkat Lunak	Fungsi
1	Eagle	Perancangan Desain PCB
2	PlatformIO	Pengembangan perangkat lunak mikrokontroler
3	Visual Studio Code	Editor kode pemrograman
4	Fritzing	Perancangan Desain rangkaian pokok
5	Fusion 360	Perancangan Desain wadah rangkaian
6	Bahasa Pemrograman Python Versi 3.18	Bahasa Pemrograman Pemodelan
7	Library Everywhereml Versi 0.2.40	Konversi Data Model pada C++ atau Arduino
8	Library Matplotlib Versi 3.10.3	Visualisasi data hasil pemodelan
9	Library Numpy Versi 2.2.5	Komputasi numerik data
10	Library Pandas Versi 2.2.3	Pengolahan data pemodelan
11	Library Scikit Learn Versi 1.6.1	Normalisasi data dan pengukuran standar
12	Library Tensorflow Keras Versi 2.18.0	Pembuatan Pemodelan <i>Neural Network</i>
13	Library Seaborn Versi 0.13.2	Visualisasi data evaluasi <i>Confusion Matrix</i>

Sedangkan untuk bahan dan komponen instrumentasi yang digunakan dalam penelitian ini terdiri dari beberapa komponen yang dapat dilihat pada tabel berikut.

Tabel 3.3: Bahan dan komponen pada penelitian pembuatan sarung tangan pintar

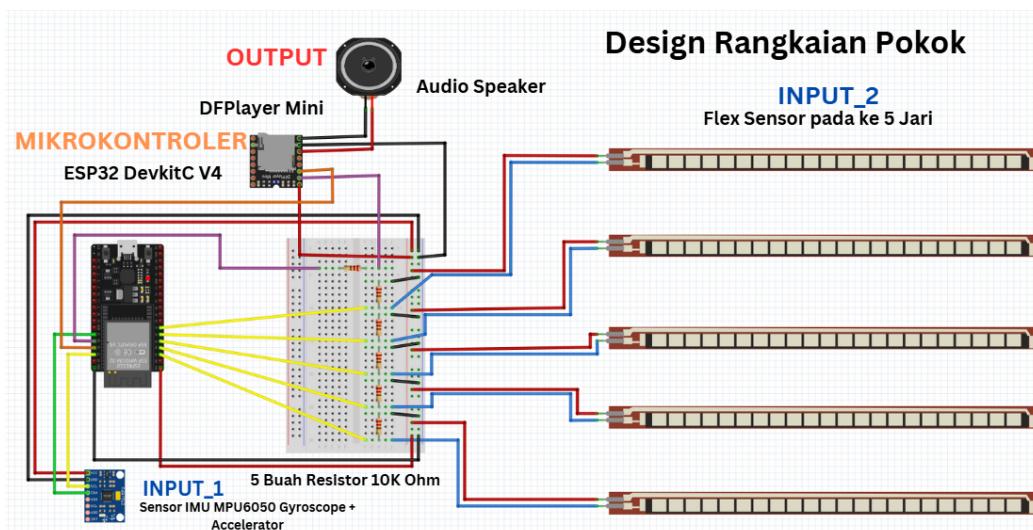
No	Bahan-bahan dan Komponen instrumentasi	Jumlah
1	Konektor JST-HX	9 buah
2	Kabel Konektor 2mm	9 buah
3	PCB Polos	1 buah
4	Sarung Tangan	1 buah
5	ESP32 DevkitC V4	1 buah
6	Sensor Flex	5 buah
7	Sensor Giroskop IMU MPU6050	1 buah
8	<i>DF Player Mini</i>	1 buah
9	Audio Speaker	1 buah
10	Resistor 10K	6 buah
11	Kartu Memori	1 buah
12	Kabel Ties	15 buah
13	Pin Header	4 buah
14	Baterai 5V	1 buah
15	Switch	1 buah

3.3 Perancangan Sarung Tangan Pintar

Pada perancangan sarung tangan ini, penulis melakukan beberapa tahapan yang dilakukan yakni, perancangan desain rangkaian pokok, perancangan desain *PCB* dan perancangan desain wadah rangkaian.

3.3.1 Perancangan Desain Rangkaian Pokok

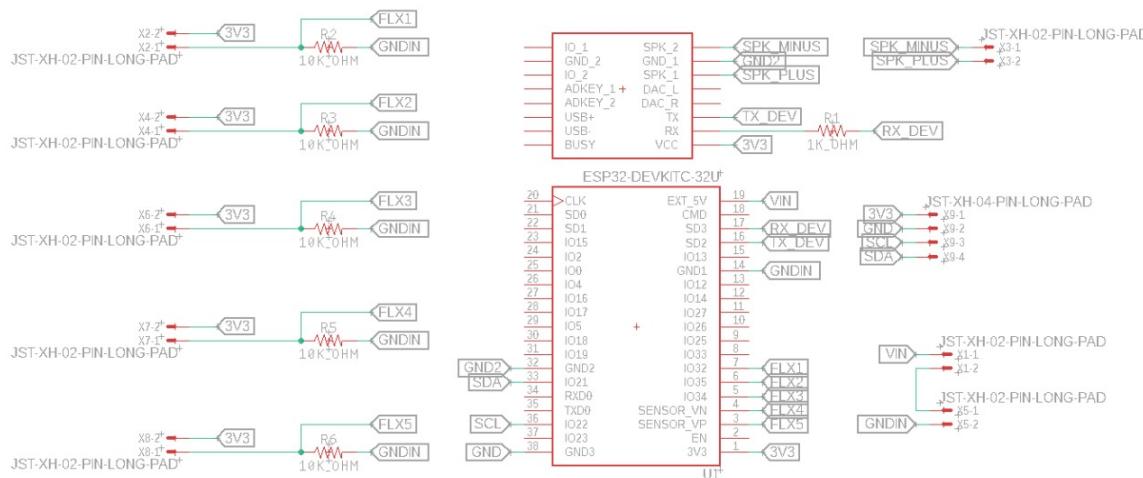
Berikut merupakan desain rangkaian pokok dari sarung tangan pintar yang akan dibuat pada penelitian ini.



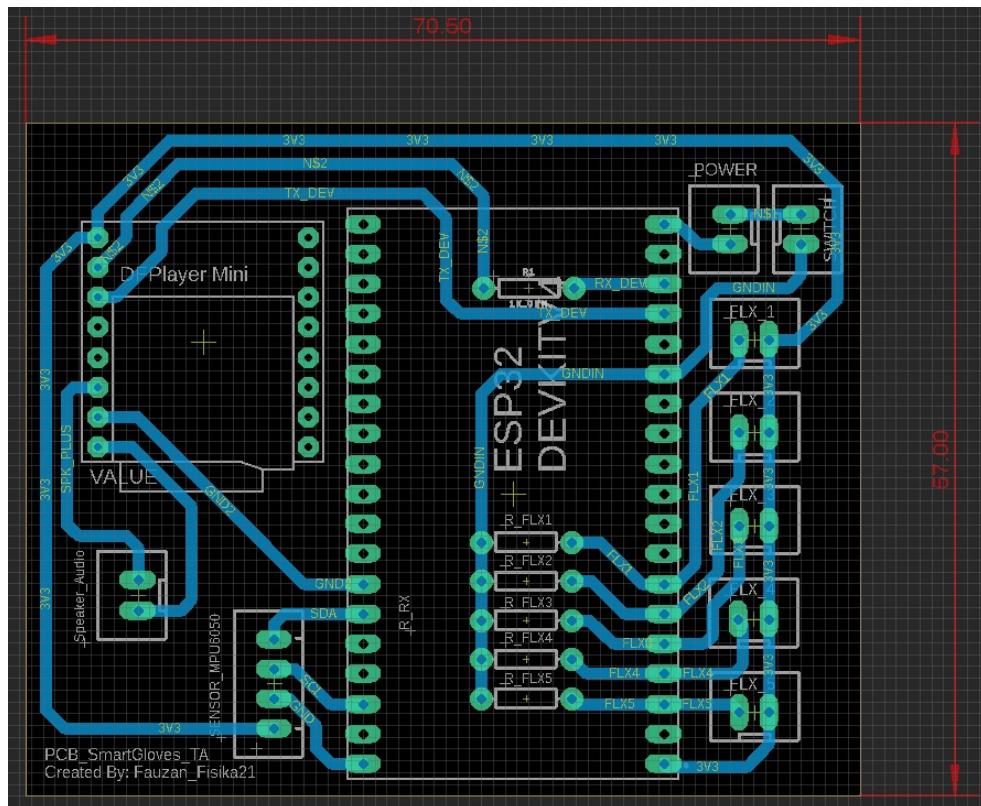
Gambar 3.1: Desain rangkaian pokok sarung tangan pintar

3.3.2 Perancangan Desain PCB

Berikut merupakan desain rangkaian pada *PCB* sarung tangan pintar yang akan dibuat pada penelitian ini.



Gambar 3.2: Desain skematik rangkaian sarung tangan pintar



Gambar 3.3: Desain *PCB* sarung tangan pintar

3.3.3 Perancangan Desain Wadah Rangkaian

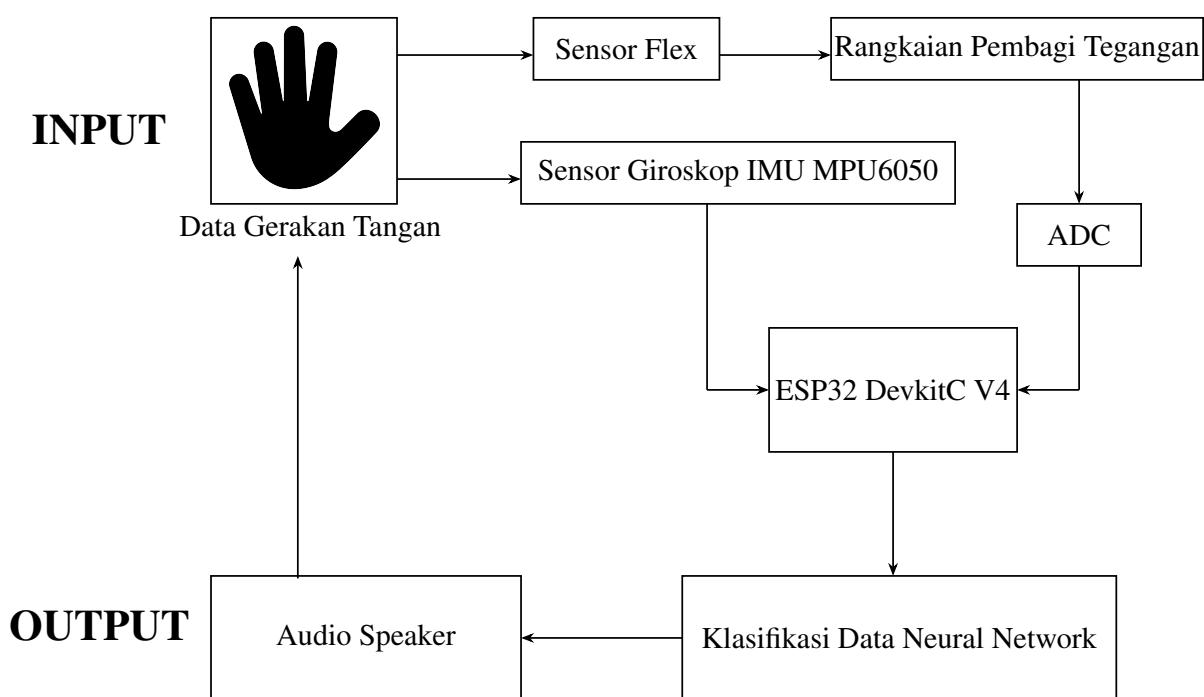
Berikut merupakan desain wadah rangkaian dari sarung tangan pintar yang akan dibuat pada penelitian ini.



Gambar 3.4: Desain wadah rangkaian sarung tangan pintar

3.4 Perancangan Sistem

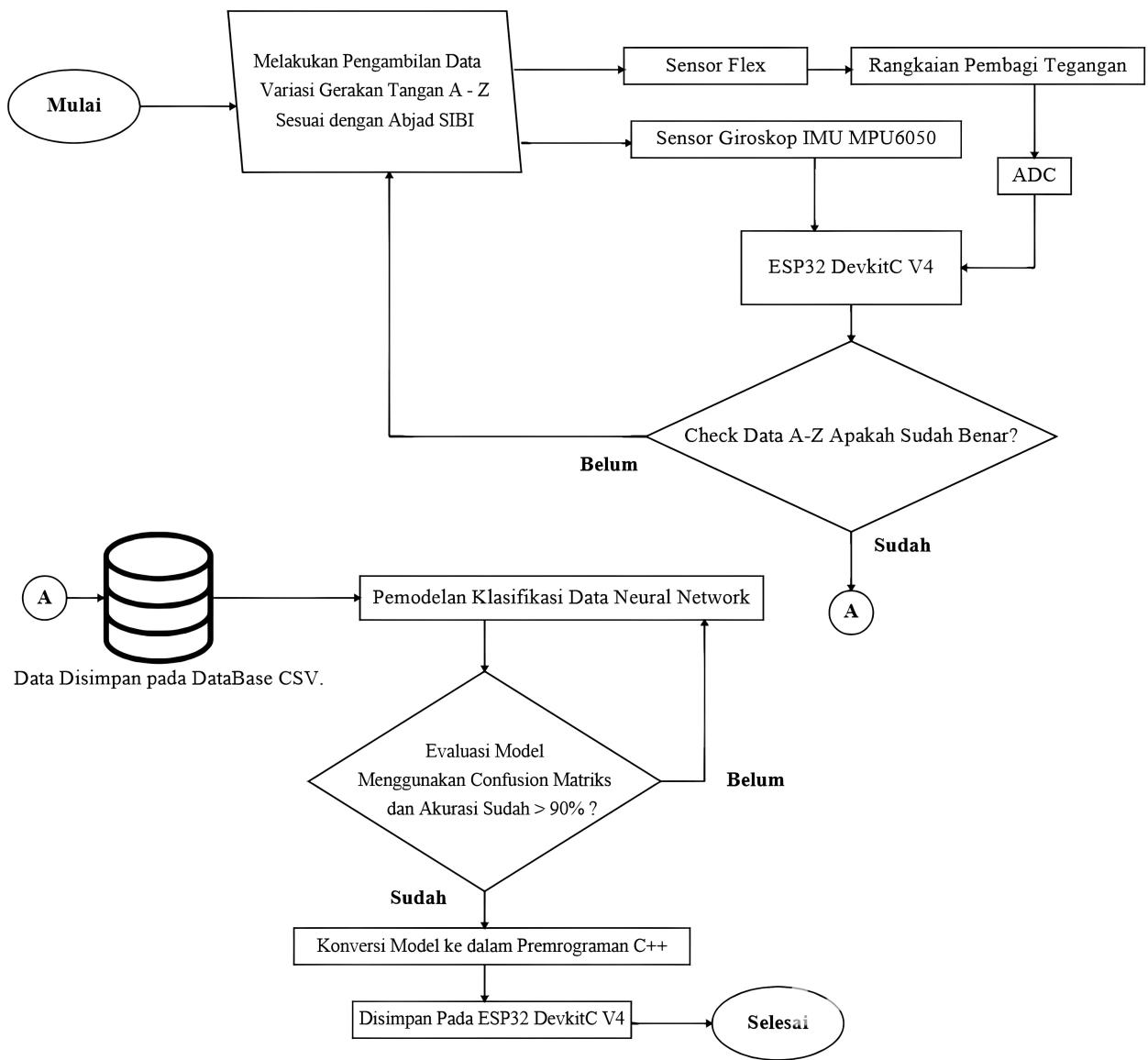
Berikut merupakan desain rancangan sistem yang akan dibuat pada penelitian ini yang dapat dijelaskan menggunakan gambar Diagram Blok dibawah berikut.



Gambar 3.5: Diagram Blok Sistem Sarung Tangan Pintar

3.5 Alur Perencanaan Pengambilan Data dan Pemodelan

Alur perencanaan pengambilan data pada penelitian ini adalah sebagai berikut.



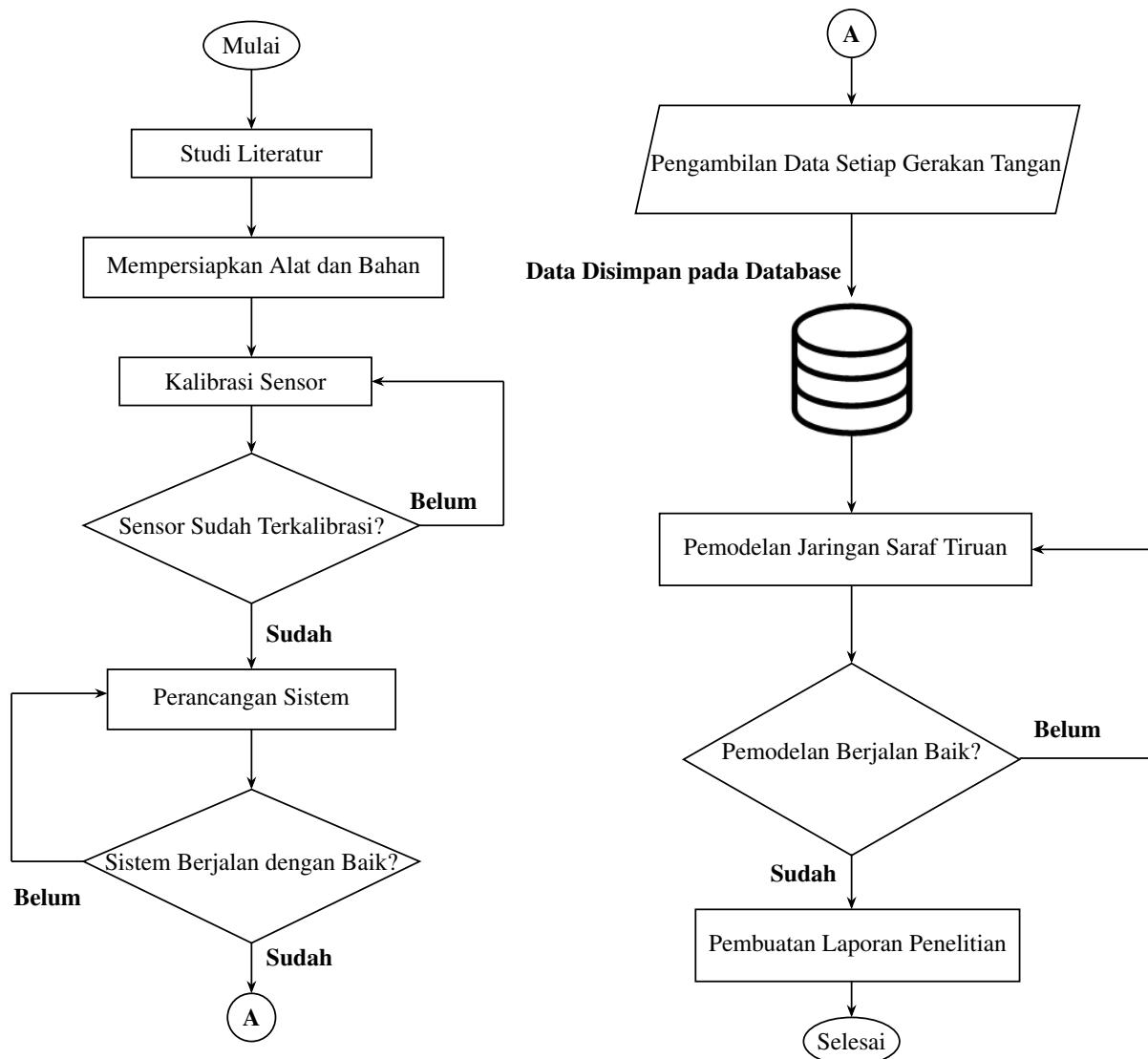
Gambar 3.6: Alur perencanaan pengambilan dan pemodelan data

Pada pengambilan data, penulis melakukan 26 variasi data yakni gerakan Abjad Sistem Isyarat Bahasa Indonesia (SIBI) dari Huruf A sampai Z, lalu dilakukan penyimpanan data pada Database menggunakan file .CSV, yang kemudian data akan diproses pemodelan klasifikasi data menggunakan *Artificial Neural Network* yang dibuat dengan menggunakan bahasa pemrograman Python, dibantu *Library Keras* dan *Tensorflow* untuk pemodelan, serta *Library Matplotlib* untuk visualisasi data hasil pemodelan.

Model selanjutnya dilakukan evaluasi menggunakan *Confusion Matrix* untuk mengetahui tingkat akurasi dari pemodelan yang telah dilakukan. Selanjutnya model akan dikonversi ke dalam bahasa pemrograman C++ atau Arduino menggunakan *Library Everywhereml* agar dapat diimplementasikan pada Sarung Tangan Pintar pada penelitian ini.

3.6 Alur Pelaksanaan Penelitian

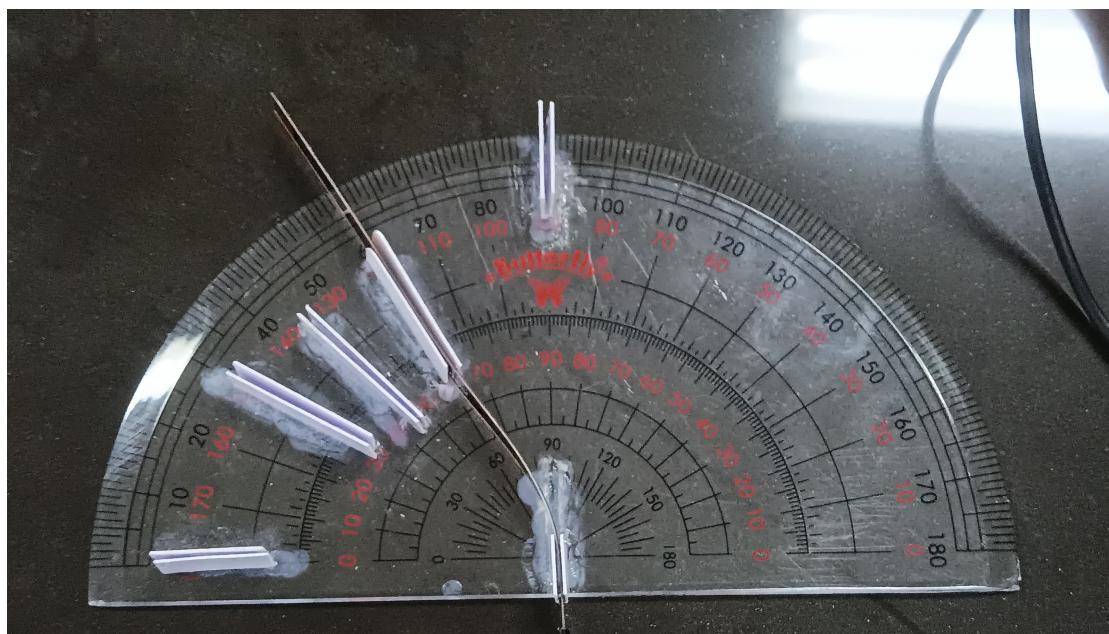
Alur penelitian yang akan dilakukan pada penelitian ini adalah sebagai berikut.



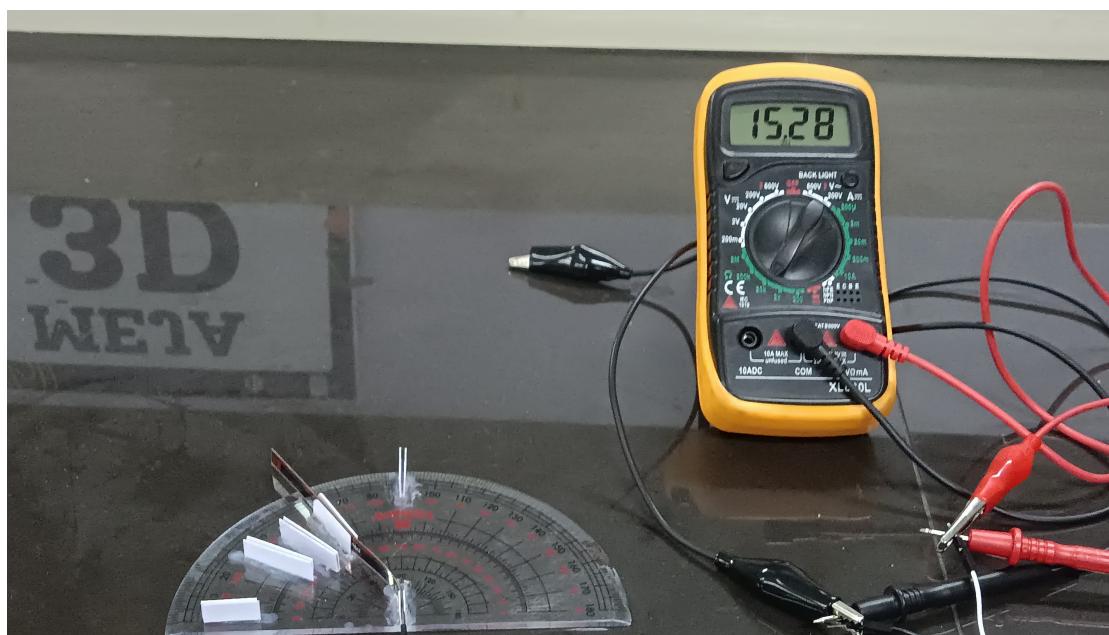
Gambar 3.7: Alur pelaksanaan penelitian

3.7 Perencanaan Kalibrasi dan Pengukuran Sensitivitas Sensor

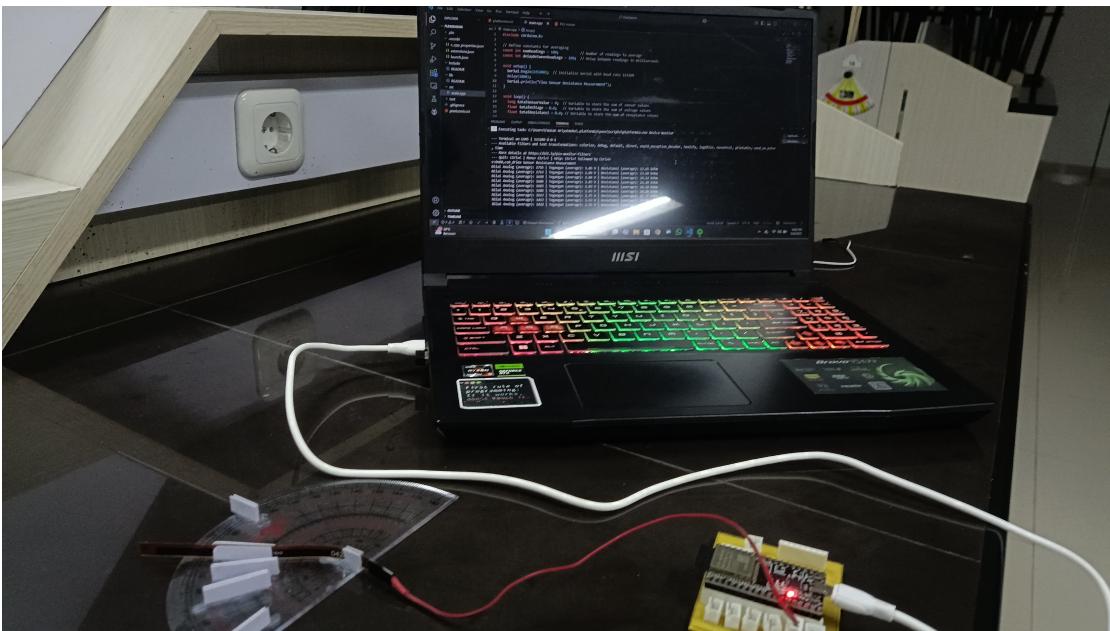
Pada bagian ini penulis melakukan perencanaan kalibrasi dan pengukuran sensitivitas sensor yang terdiri dari dua sensor yakni, sensor flex dan sensor giroskop IMU MPU6050. Pada sensor flex kalibrasi dilakukan menggunakan busur derajat untuk menyelaraskan resistansi kelima sensor flex terhadap sudut tekuk secara manual menggunakan multimeter dan menggunakan serial monitor dengan kode pemrograman Arduino pada Lampiran B 5.2. Berikut merupakan gambaran cara pengukuran sensitivitas kelima sensor *flex* tersebut,



Gambar 3.8: cara pengukuran sudut sensor *flex* pada busur derajat

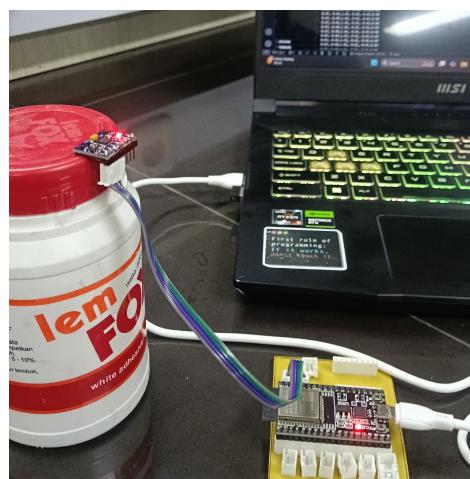


Gambar 3.9: Pengukuran resistansi sensor *flex* pada multimeter



Gambar 3.10: Pengukuran resistansi sensor *flex* pada serial monitor pemrograman Arduino

Pada pengukuran sensitivitas sensor *flex*, penulis melakukan pengukuran resistansi terhadap sudut tekuk dari 0, 30, 45, 60 hingga 90 derajat, sehingga didapatkan 10 data pengukuran untuk setiap sensor *flex*. Lalu untuk sensor giroskop IMU MPU6050, kalibrasi dilakukan dengan cara membuat nilai percepatan sudut (*g*) dan sudut rotasi (*rad/s*) sumbu-X, sumbu-Y dan sumbu-Z (*yaw, pitch, roll*) menjadi nol, terkecuali nilai pada percepatan sumbu-Z yang harus bernilai 1*g*, dikarenakan pada *datasheet* dan beberapa referensi jurnal nilai $1g = 9.81m/s^2$. Berikut merupakan gambar cara kalibrasi sensor giroskop IMU MPU6050,

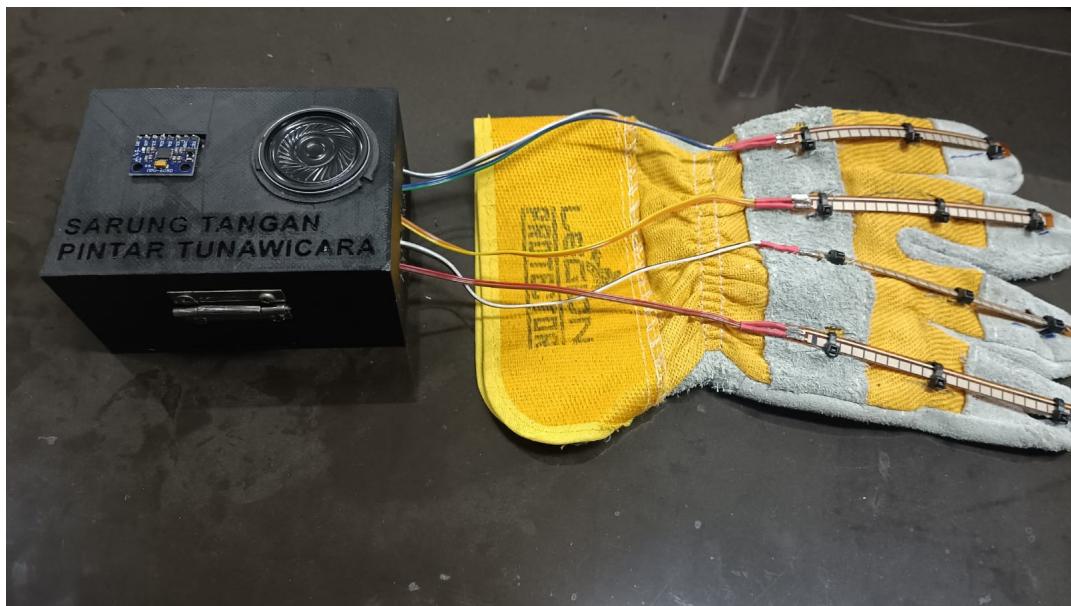


Gambar 3.11: Kalibrasi sensor giroskop IMU MPU6050 pada permukaan datar

Metode ini dilakukan dengan cara meletakkan sensor pada permukaan datar yang stabil, sehingga kita mendapatkan nilai offset kalibrasi (*calibration offset*) dari sensor giroskop IMU MPU6050. Terdapat 7 konfigurasi pengaturan offset kalibrasi yang didapatkan setiap percepatan (*g*) dan sudut (*rad/s*) pada sumbu X, Y dan Z. Tentunya hal tersebut dilakukan menggunakan kode pemrograman Arduino yang telah dibuat pada Lampiran B 5.1.

3.8 Realisasi Perencanaan Pembuatan Sarung Tangan Pintar

Berikut merupakan hasil realisasi dari perencanaan pembuatan sarung tangan pintar yang telah dilakukan pada penelitian ini.



Gambar 3.12: Realisasi sarung tangan pintar

[Halaman ini sengaja dikosongkan]

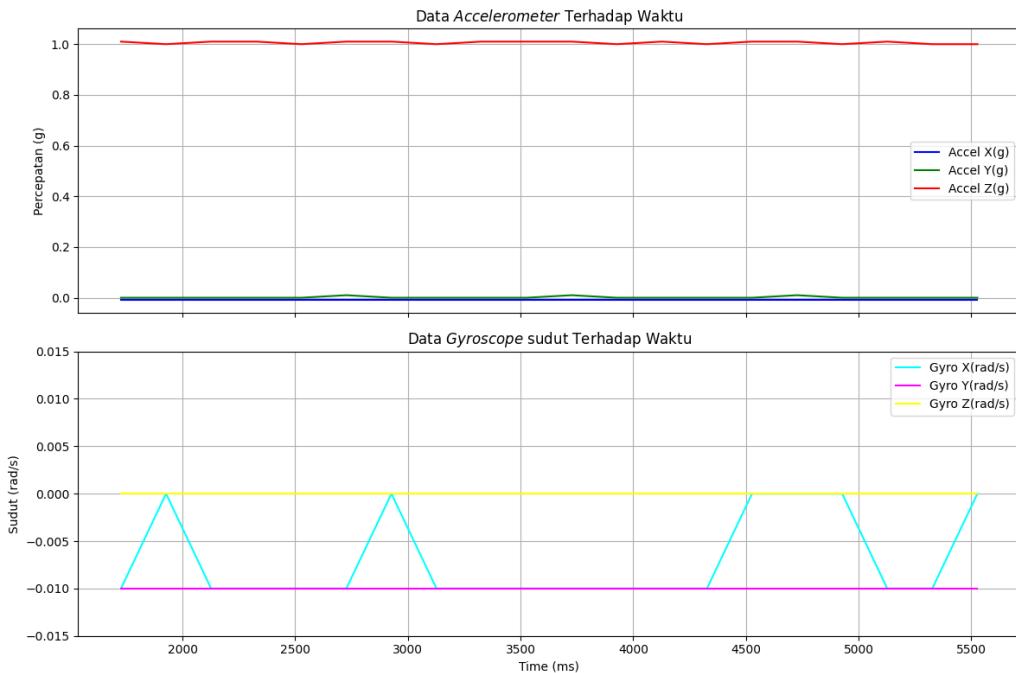
BAB IV

HASIL DAN PEMBAHASAN

Bab ini membahas mengenai hasil data yang diperoleh dari proses kalibrasi dan pengukuran sensitivitas sensor, yang merupakan komponen krusial dalam pengembangan sarung tangan pintar. Analisis mendalam terhadap karakteristik sensor giroskop IMU MPU6050 dan sensor *flex* akan dipaparkan, diikuti dengan evaluasi klasifikasi Jaringan Saraf Tiruan (ANN) untuk data abjad dari huruf A hingga Z yang menganut Sistem Isyarat Bahasa Indonesia (SIBI).

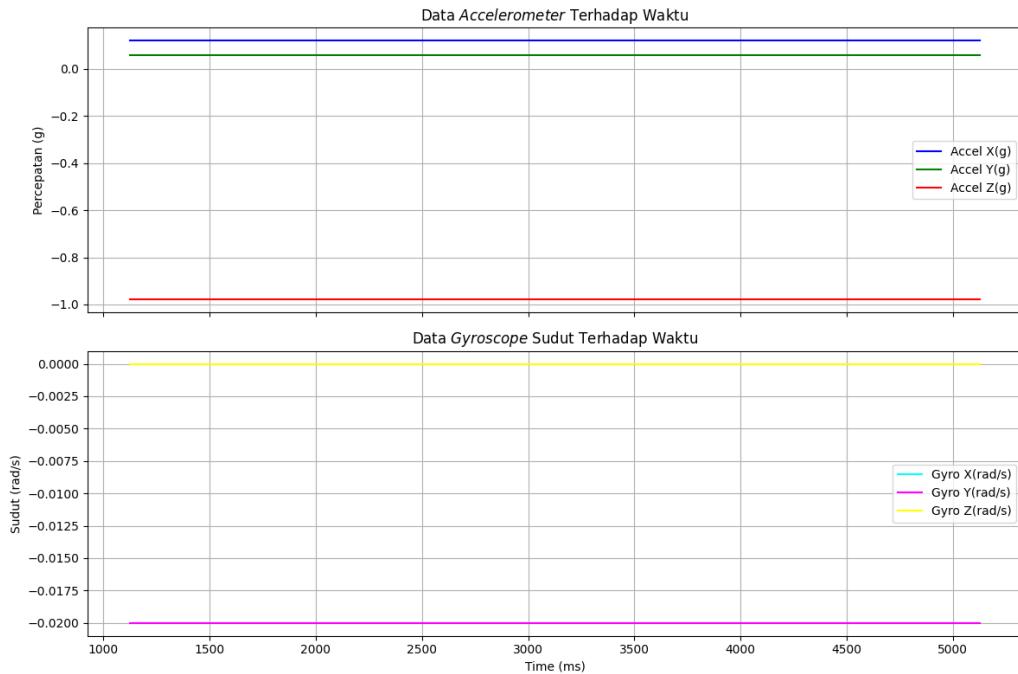
4.1 Pengukuran Kalibrasi Sensor Giroskop IMU MPU6050

Proses kalibrasi sensor giroskop IMU MPU6050 merupakan tahap fundamental untuk memastikan akurasi data orientasi dan gerakan tangan. Sebagaimana dijelaskan pada Bab 3.7, kalibrasi dilakukan dengan mengukur data keluaran giroskop, yaitu sudut rotasi (dalam satuan rad/s) dan percepatan (dalam satuan g) pada sumbu-X, sumbu-Y, dan sumbu-Z. Pengukuran ini dilaksanakan melalui tujuh konfigurasi pengaturan offset kalibrasi yang berbeda, dengan sensor diposisikan pada permukaan datar untuk mendapatkan kondisi referensi yang stabil. Data lengkap dari ketujuh konfigurasi tersebut telah dilampirkan pada Tabel 5.1 di Lampiran A. Dari banyaknya tujuh konfigurasi yang diuji, konfigurasi ke-7 teridentifikasi menghasilkan data keluaran yang paling mendekati nilai ideal, yaitu nilai percepatan sudut (g) dan sudut rotasi (rad/s) untuk sumbu X dan Y mendekati nol, serta nilai percepatan pada sumbu Z mendekati $1g$ (setara dengan $9.81m/s^2$)



Gambar 4.1: Pengambilan data sensor giroskop IMU MPU6050 setelah kalibrasi menggunakan konfigurasi ke-7.

Selanjutnya dari semua konfigurasi tersebut dapat dilihat pada tabel bahwa hasil terbaik atau mendekati nol adalah pada konfigurasi ke-7, dimana pada konfigurasi ini didapatkan nilai percepatan sudut (g) dan sudut rotasi (rad/s) X, Y dan Z (*yaw, pitch, roll*) menjadi nol, serta nilai pada percepatan sumbu-Z yang mendekati nilai $1g$, dikarenakan pada *datasheet* dan beberapa referensi jurnal nilai $1g = 9.81m/s^2$. Adapun dapat dilihat pada Gambar diatas merupakan hasil pengambilan data sensor giroskop IMU MPU6050 setelah kalibrasi pada konfigurasi ke-7. Sedangkan pada Gambar dibawah merupakan hasil pengambilan data sensor giroskop IMU MPU6050 sebelum kalibrasi.

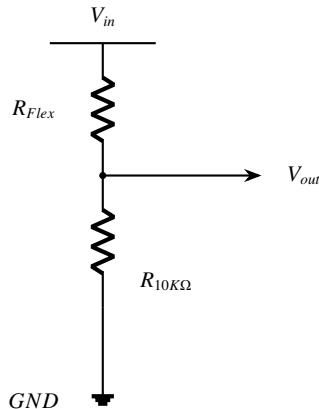


Gambar 4.2: Pengambilan data sensor giroskop IMU MPU6050 sebelum kalibrasi.

Pada gambar 4.2 dapat dilihat bahwa data akselerometer menunjukkan adanya offset yang signifikan dari nilai ideal. Nilai Accel X(g) dan Accel Y(g) menunjukkan deviasi negatif yang cukup besar dari $0g$ (misalnya, Accel X sekitar $-0.1g$ dan Accel Y sekitar $-0.05g$ berdasarkan inspeksi visual grafik). Demikian pula, Accel Z(g) terbaca secara konsisten di bawah nilai $1g$ yang diharapkan (sekitar $-0.8g$), mengindikasikan adanya bias negatif yang substansial. Selanjutnya dapat dilihat bahwa data giroskop sebelum kalibrasi juga menunjukkan deviasi yang nyata dari nol untuk ketiga sumbu rotasi. Nilai Gyro X(rad/s), Gyro Y(rad/s), dan Gyro Z(rad/s) menunjukkan offset negatif, berkisar antara -0.005 hingga -0.015 rad/s . Offset yang teramat pada data sebelum kalibrasi ini merupakan manifestasi dari zero-bias error, yaitu ketidakakuratan inheren yang timbul dari proses fabrikasi sensor di pabrik. Jika data mentah dari Grafik 4.2 ini digunakan secara langsung untuk analisis gestur, sistem akan salah menginterpretasikan kondisi diam sensor. Sebagai contoh, offset negatif pada Accel Z akan menyebabkan sistem menganggap bahwa sarung tangan selalu dalam posisi sedikit miring ke atas, meskipun sebenarnya diletakkan pada permukaan datar. Demikian pula, offset pada data giroskop akan menyebabkan sistem mendeteksi adanya rotasi meskipun tidak ada gerakan aktual.

4.2 Pengukuran Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex*

Sesuai dengan penjelasan metodologi pada Bab 3.7 proses pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* dilakukan setiap 0, 30, 45, 60 dan 90 derajat. Hal ini dilakukan dengan dua metode yakni pengukuran langsung menggunakan multimeter dan juga pada serial monitor menggunakan pemrograman arduino. Rangkaian dasar pada pengukuran flex sensor sendiri merupakan rangkaian pembagi tegangan sebagaimana pada gambar rangkaian dibawah berikut,



Gambar 4.3: Rangkaian pembagi tegangan sensor *flex*.

Maka berlaku hukum pembagi tegangan sebagai berikut,

$$V_{out} = V_{in} \cdot \frac{R_{10K\Omega}}{R_{Flex} + R_{10K\Omega}} \quad (4.1)$$

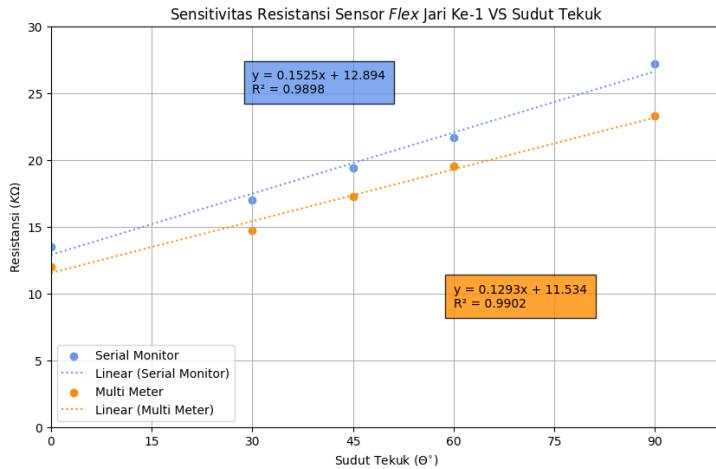
sehingga dapat diturunkan menjadi persamaan seperti dibawah ini untuk mencari nilai resistansi pada sensor *flex*,

$$R_{Flex} = R_{10K\Omega} \cdot \left(\frac{V_{in}}{V_{out}} - 1 \right) \quad (4.2)$$

Dimana V_{in} adalah tegangan masukan, V_{out} adalah tegangan keluaran, $R_{10K\Omega}$ adalah nilai resistor tetap yang digunakan pada rangkaian pembagi tegangan, dan R_{Flex} adalah nilai resistansi pada sensor *flex*. Pengukuran dilakukan pada lima sensor *flex* yang diletakkan pada masing-masing jari tangan kanan, yakni jari kelingking, jari manis, jari tengah, jari telunjuk dan jari jempol. Pengukuran dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun hasil pengukuran resistansi pada kelima sensor *flex* terhadap sudut yang telah dilakukan, yang akan dibahas pada sub bab selanjutnya.

4.2.1 Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex* ke-1

Pada pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-1 dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-1 pada pengukuran ini adalah pada jari kelingking tangan kanan. Hasil pengukuran resistansi pada sensor *flex* ke-1 terhadap sudut dapat dilihat pada Tabel 5.2 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara resistansi terhadap sudut pada sensor *flex* ke-1 seperti pada Gambar 4.4.

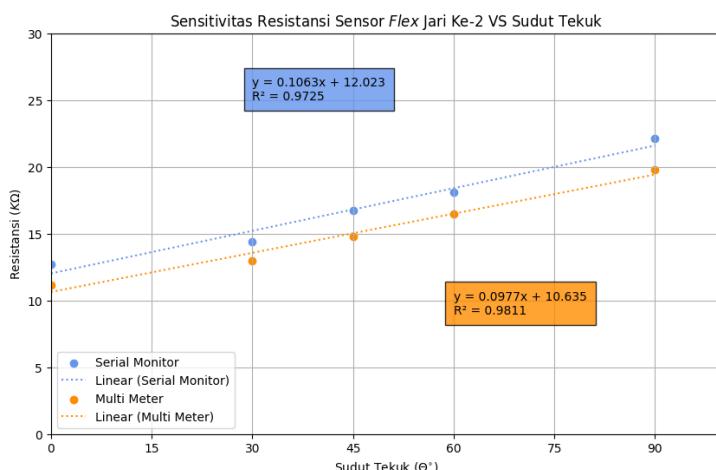


Gambar 4.4: Pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-1.

Dapat dilihat pada grafik di atas, data yang diperoleh melalui serial monitor menunjukkan hubungan yang cenderung linear. Garis regresi linear untuk data ini adalah $y = 0.1525x + 12.894$, dengan nilai $R^2 = 0.9890$. Kemiringan (*slope*) sebesar $0.1525K\Omega/\theta$ merepresentasikan sensitivitas resistansi sensor ini terhadap perubahan sudut; artinya, untuk setiap satu derajat perubahan sudut tekukan, resistansi sensor berubah sebesar $0.1525K\Omega$. Intersep sebesar $12.894K\Omega$ adalah estimasi resistansi sensor pada sudut 0 derajat (posisi lurus). Sedangkan pada data yang diperoleh dari multimeter juga menunjukkan tren linear yang kuat, dengan persamaan regresi $y = 0.1293x + 11.534$ dan nilai $R^2 = 0.9902$.

4.2.2 Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex* ke-2

Pada pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-2 dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-2 pada pengukuran ini adalah pada jari manis tangan kanan. Hasil pengukuran resistansi pada sensor *flex* ke-2 terhadap sudut dapat dilihat pada Tabel 5.2 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara resistansi terhadap sudut pada sensor *flex* ke-2 seperti pada Gambar 4.5.

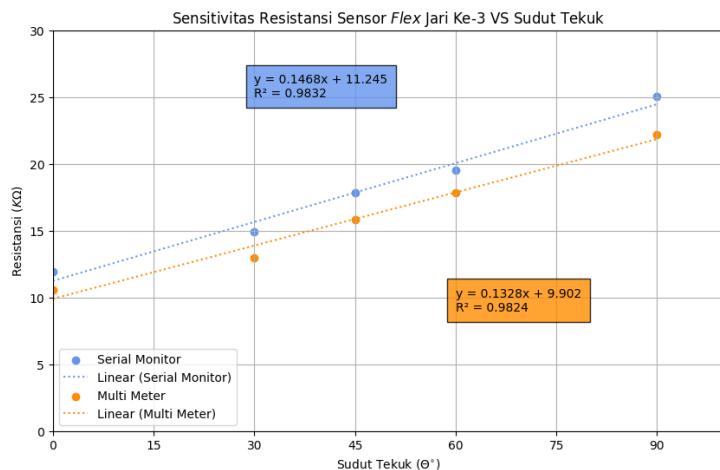


Gambar 4.5: Pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-2.

Dapat dilihat pada grafik sebelumnya, data yang diperoleh melalui serial monitor menunjukkan hubungan yang cenderung linear. Garis regresi linear untuk data ini adalah $y = 0.1063x + 12.023$, dengan nilai $R^2 = 0.9725$. Kemiringan (*slope*) sebesar $0.1063K\Omega/\theta$ merepresentasikan sensitivitas resistansi sensor ini terhadap perubahan sudut; artinya, untuk setiap satu derajat perubahan sudut tekukan, resistansi sensor berubah sebesar $0.1063K\Omega$. Intersep sebesar $12.023K\Omega$ adalah estimasi resistansi sensor pada sudut 0 derajat (posisi lurus). Sedangkan pada data yang diperoleh dari multimeter juga menunjukkan tren linear yang kuat, dengan persamaan regresi $y = 0.0977x + 10.635$ dan nilai $R^2 = 0.9811$. Sensor *flex* kedua juga menunjukkan respons linear yang baik, meskipun sensitivitasnya tampak sedikit lebih rendah dibandingkan sensor *flex*. Nilai R^2 yang tinggi dari kedua metode pengukuran mengkonfirmasi linearitas ini cukup akurat.

4.2.3 Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex* ke-3

Pada pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-3 dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-3 pada pengukuran ini adalah pada jari tengah tangan kanan. Hasil pengukuran resistansi pada sensor *flex* ke-3 terhadap sudut dapat dilihat pada Tabel 5.2 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara resistansi terhadap sudut pada sensor *flex* ke-3 seperti pada Gambar 4.6.

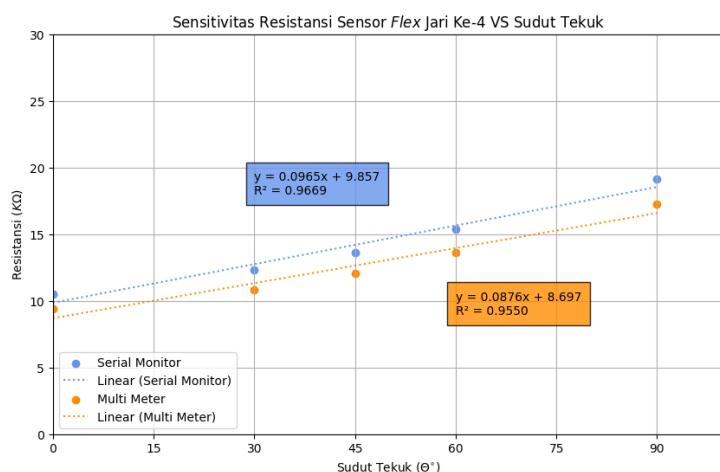


Gambar 4.6: Pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-3.

Dapat dilihat pada grafik di atas, data yang diperoleh melalui serial monitor menunjukkan hubungan yang cenderung linear. Garis regresi linear untuk data ini adalah $y = 0.1468x + 11.245$, dengan nilai $R^2 = 0.9832$. Kemiringan (*slope*) sebesar $0.1468K\Omega/\theta$ merepresentasikan sensitivitas resistansi sensor ini terhadap perubahan sudut; artinya, untuk setiap satu derajat perubahan sudut tekukan, resistansi sensor berubah sebesar $0.1468K\Omega$. Intersep sebesar $11.245K\Omega$ adalah estimasi resistansi sensor pada sudut 0 derajat (posisi lurus). Sedangkan pada data yang diperoleh dari multimeter juga menunjukkan tren linear yang kuat, dengan persamaan regresi $y = 0.1328x + 9.902$ dan nilai $R^2 = 0.9824$. Sensor *flex* ketiga ini menunjukkan sensitivitas yang sebanding dengan sensor *flex* pertama dan juga mempertahankan linearitas yang sangat baik, sebagaimana dapat terbuktikan oleh nilai R^2 yang tinggi.

4.2.4 Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex* ke-4

Pada pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-4 dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-4 pada pengukuran ini adalah pada jari telunjuk tangan kanan. Hasil pengukuran resistansi pada sensor *flex* ke-4 terhadap sudut dapat dilihat pada Tabel 5.3 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara resistansi terhadap sudut pada sensor *flex* ke-4 seperti pada Gambar 4.7.



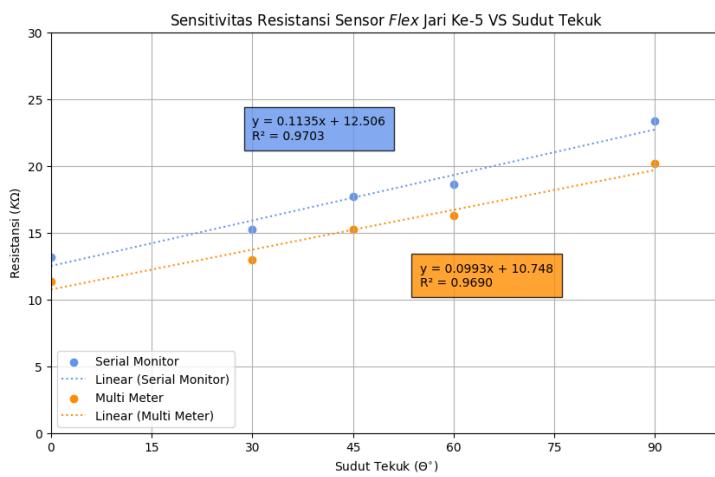
Gambar 4.7: Pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-4.

Dapat dilihat pada grafik di atas, data yang diperoleh melalui serial monitor menunjukkan hubungan yang cenderung linear. Garis regresi linear untuk data ini adalah $y = 0.0965x + 9.857$, dengan nilai $R^2 = 0.9669$. Kemiringan (*slope*) sebesar $0.0965K\Omega/\theta$ merepresentasikan sensitivitas resistansi sensor ini terhadap perubahan sudut; artinya, untuk setiap satu derajat perubahan sudut tekukan, resistansi sensor berubah sebesar $0.0965K\Omega$. Intersep sebesar $9.857K\Omega$ adalah estimasi resistansi sensor pada sudut 0 derajat (posisi lurus). Sedangkan pada data yang diperoleh dari multimeter juga menunjukkan tren linear yang kuat, dengan persamaan regresi $y = 0.0876x + 8.697$ dan nilai $R^2 = 0.9550$. Sensor *flex* keempat menunjukkan sensitivitas yang sedikit lebih rendah dibandingkan sensor *flex* pertama dan ketiga, mirip dengan sensor *flex* kedua. Linearitasnya masih terjaga dengan baik, meskipun nilai R^2 dari data multimeter sedikit lebih rendah dibandingkan sensor lainnya, namun masih menunjukkan kecocokan model linear yang kuat.

4.2.5 Sensitivitas Resistansi terhadap Sudut pada Sensor *Flex* ke-5

Pada pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-5 dilakukan dengan mengukur resistansi pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-5 pada pengukuran ini adalah pada ibu jari tangan kanan. Hasil pengukuran resistansi pada sensor *flex* ke-5 terhadap sudut dapat dilihat pada Tabel 5.3 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara resistansi terhadap sudut pada sensor *flex* ke-5 seperti pada Gambar 4.8.

Dapat dilihat pada grafik di bawah, data yang diperoleh melalui serial monitor menunjukkan hubungan yang cenderung linear. Garis regresi linear untuk data ini adalah $y = 0.1135x + 12.506$, dengan nilai $R^2 = 0.9703$. Kemiringan (*slope*) sebesar $0.1135K\Omega/\theta$ merepresentasikan sensitivitas resistansi sensor ini terhadap perubahan sudut; artinya, untuk setiap satu derajat perubahan sudut tekukan, resistansi sensor berubah sebesar $0.1135K\Omega$. Intersep sebesar $12.506K\Omega$ adalah estimasi resistansi sensor pada sudut 0 derajat (posisi lurus). Sedangkan pada data yang diperoleh dari multimeter juga menunjukkan tren linear yang kuat, dengan persamaan regresi $y = 0.0993x + 10.748$ dan nilai $R^2 = 0.9690$. Sensor *flex* kelima menunjukkan perilaku linear yang konsisten dengan sensor-sensor lainnya, dengan nilai R^2 yang tinggi untuk kedua metode pengukuran.



Gambar 4.8: Pengukuran sensitivitas resistansi terhadap sudut pada sensor *flex* ke-5.

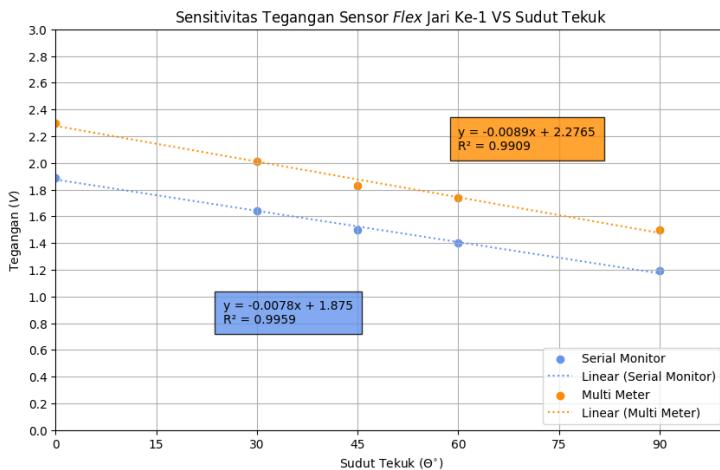
4.3 Pengukuran Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex*

Sesuai dengan penjelasan metodologi pada Bab 3.7 proses pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* dilakukan setiap 0, 30, 45, 60 dan 90 derajat. Hal ini dilakukan dengan dua metode yakni pengukuran langsung menggunakan multimeter dan juga pada serial monitor menggunakan pemrograman arduino. Rangkaian dasar pada pengukuran *flex* sensor sendiri merupakan rangkaian pembagi tegangan sebagaimana pada gambar 4.3. Adapun hasil pengukuran resistansi pada kelima sensor *flex* terhadap sudut yang telah dilakukan, yang akan dibahas pada sub bab selanjutnya.

4.3.1 Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex* ke-1

Pada pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-1 dilakukan dengan mengukur tegangan pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-1 pada pengukuran ini adalah pada jari kelingking tangan kanan. Hasil pengukuran tegangan pada sensor *flex* ke-1 terhadap sudut dapat dilihat pada Tabel 5.4 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara tegangan terhadap sudut pada sensor *flex* ke-1 seperti pada Gambar 4.9.

Dapat dilihat pada grafik dibawah, data dari multimeter menunjukkan hubungan linear terbalik yang sangat kuat. Persamaan regresi linearnya adalah $y = -0.0089x + 2.2765$, dengan nilai $R^2 = 0.9909$. Tanda negatif pada kemiringan ($-0.0089V/\theta$) mengkonfirmasi bahwa tegangan keluaran menurun seiring dengan meningkatnya sudut tekukan. Intersep sebesar 1.875 V adalah estimasi V_{out} pada sudut 0 derajat. Adapun pada data yang diperoleh dari serial monitor juga menunjukkan hubungan linear terbalik yang kuat dengan persamaan regresi $y = -0.0078x + 1.875$ dan $R^2 = 0.9959$.



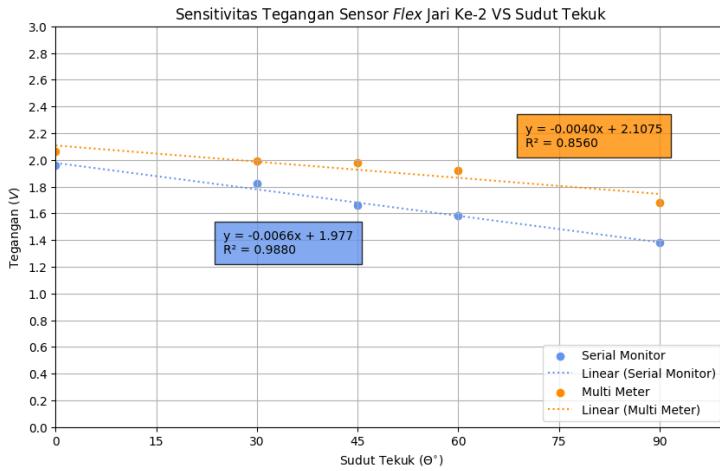
Gambar 4.9: Pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-1.

Hubungan linear terbalik ini sesuai dengan ekspektasi berdasarkan Persamaan 4.2. Ketika sudut tekukan (θ) meningkat, R_{flex} meningkat (sebagaimana ditunjukkan pada analisis resistansi sebelumnya). Karena R_{flex} berada di bagian penyebut dari fraksi dalam persamaan tersebut, peningkatan R_{flex} akan menyebabkan penurunan nilai V_{out} . Nilai R^2 yang sangat tinggi dari kedua metode pengukuran untuk sensor ini menunjukkan linearitas dan reliabilitas yang sangat baik dari respons tegangan sensor terhadap perubahan sudut.

4.3.2 Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex* ke-2

Pada pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-2 dilakukan dengan mengukur tegangan pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-2 pada pengukuran ini adalah pada jari manis tangan kanan. Hasil pengukuran tegangan pada sensor *flex* ke-2 terhadap sudut dapat dilihat pada Tabel 5.4 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara tegangan terhadap sudut pada sensor *flex* ke-2 seperti pada Gambar 4.10.

Dapat dilihat pada grafik, data dari multimeter menunjukkan hubungan linear terbalik yang sangat kuat. Persamaan regresi linearnya adalah $y = -0.0040x + 2.1075$, dengan nilai $R^2 = 0.8560$. Tanda negatif pada kemiringan ($-0.0040V/\theta$) mengkonfirmasi bahwa tegangan keluaran menurun seiring dengan meningkatnya sudut tekukan. Intersep sebesar 2.1075 V adalah estimasi V_{out} pada sudut 0 derajat. Adapun pada data yang diperoleh dari serial monitor juga menunjukkan hubungan linear terbalik yang kuat dengan persamaan regresi $y = -0.0066x + 1.977$ dan $R^2 = 0.9880$.

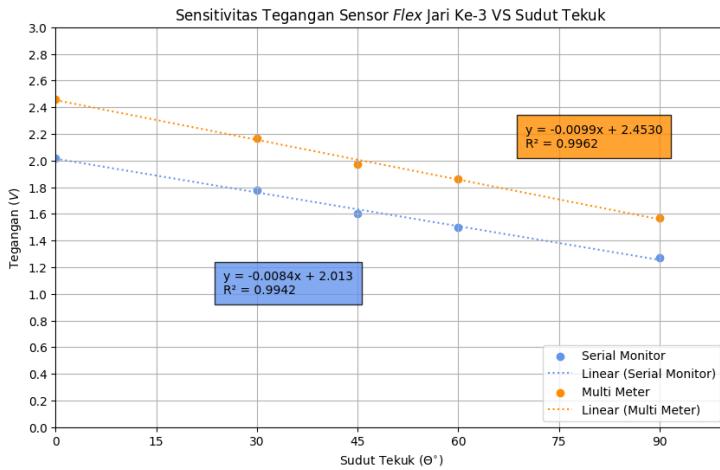


Gambar 4.10: Pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-2.

Hubungan linear terbalik pada sensor *flex* ke-2 ini juga sesuai dengan ekspektasi berdasarkan Persamaan 4.2. Ketika sudut tekukan (θ) meningkat, R_{flex} meningkat, menyebabkan penurunan nilai V_{out} . Nilai R^2 yang tinggi dari kedua metode pengukuran untuk sensor ini menunjukkan linearitas dan reliabilitas yang sangat baik dari respons tegangan sensor terhadap perubahan sudut, meskipun nilai R^2 untuk data serial monitor sedikit lebih rendah dibandingkan sensor *flex* ke-1.

4.3.3 Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex* ke-3

Pada pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-3 dilakukan dengan mengukur tegangan pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakkan sensor *flex* ke-3 pada pengukuran ini adalah pada jari tengah tangan kanan. Hasil pengukuran tegangan pada sensor *flex* ke-3 terhadap sudut dapat dilihat pada Tabel 5.4 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara tegangan terhadap sudut pada sensor *flex* ke-3 seperti pada Gambar 4.11.



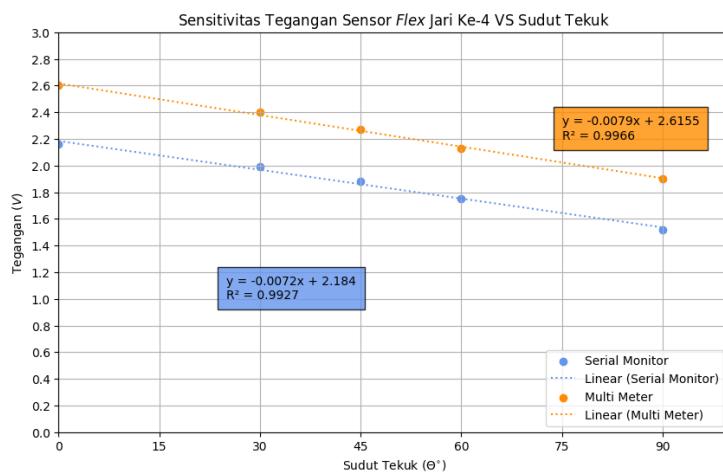
Gambar 4.11: Pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-3.

Dapat dilihat pada grafik, data dari multimeter menunjukkan hubungan linear terbalik yang sangat kuat. Persamaan regresi linearnya adalah $y = -0.0099x + 2.453$, dengan nilai $R^2 = 0.9962$. Tanda negatif pada kemiringan ($-0.0099V/\theta$) mengkonfirmasi bahwa tegangan keluaran menurun seiring dengan meningkatnya sudut tekukan. Intersep sebesar 2.453 V adalah estimasi V_{out} pada sudut 0 derajat. Adapun pada data yang diperoleh dari serial monitor juga menunjukkan hubungan linear terbalik yang kuat dengan persamaan regresi $y = -0.0084x + 2.013$ dan $R^2 = 0.9942$.

Hubungan linear terbalik pada sensor *flex* ke-3 ini juga sesuai dengan ekspektasi berdasarkan Persamaan 4.2. Ketika sudut tekukan (θ) meningkat, R_{flex} meningkat, menyebabkan penurunan nilai V_{out} . Nilai R^2 yang sangat tinggi dari kedua metode pengukuran untuk sensor ini menunjukkan linearitas dan reliabilitas yang sangat baik dari respons tegangan sensor terhadap perubahan sudut, dengan nilai R^2 untuk data serial monitor sedikit lebih rendah dibandingkan sensor *flex* ke-1.

4.3.4 Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex* ke-4

Pada pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-4 dilakukan dengan mengukur tegangan pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-4 pada pengukuran ini adalah pada jari telunjuk tangan kanan. Hasil pengukuran tegangan pada sensor *flex* ke-4 terhadap sudut dapat dilihat pada Tabel 5.5 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara tegangan terhadap sudut pada sensor *flex* ke-4 seperti pada Gambar 4.12.

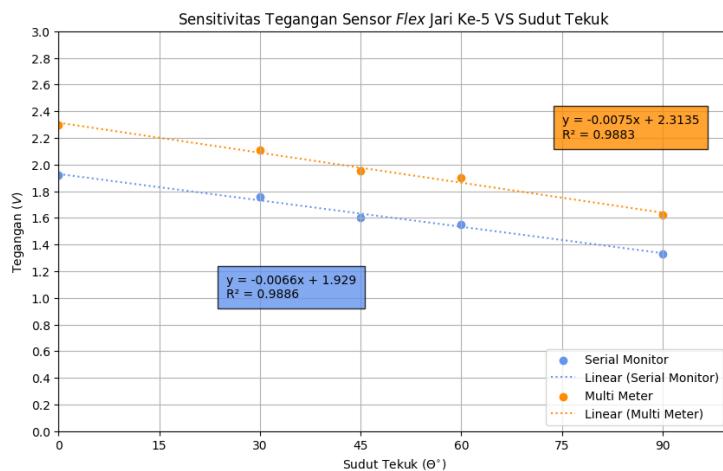


Gambar 4.12: Pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-4.

Dapat dilihat pada grafik, data dari multimeter menunjukkan hubungan linear terbalik yang sangat kuat. Persamaan regresi linearnya adalah $y = -0.0079x + 2.6155$, dengan nilai $R^2 = 0.9966$. Tanda negatif pada kemiringan ($-0.0079V/\theta$) mengkonfirmasi bahwa tegangan keluaran menurun seiring dengan meningkatnya sudut tekukan. Intersep sebesar 2.6155 V adalah estimasi V_{out} pada sudut 0 derajat. Adapun pada data yang diperoleh dari serial monitor juga menunjukkan hubungan linear terbalik yang kuat dengan persamaan regresi $y = -0.0072x + 2.184$ dan $R^2 = 0.9927$. Hubungan linear terbalik pada sensor *flex* ke-4 ini juga sesuai dengan ekspektasi berdasarkan Persamaan 4.2. Ketika sudut tekukan (θ) meningkat, R_{flex} meningkat, menyebabkan penurunan nilai V_{out} .

4.3.5 Sensitivitas Tegangan terhadap Sudut pada Sensor *Flex* ke-5

Pada pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-5 dilakukan dengan mengukur tegangan pada sudut 0, 30, 45, 60 dan 90 derajat. Adapun peletakan sensor *flex* ke-5 pada pengukuran ini adalah pada ibu jari tangan kanan. Hasil pengukuran tegangan pada sensor *flex* ke-5 terhadap sudut dapat dilihat pada Tabel 5.5 dalam Lampiran A. Selanjutnya, dari data tersebut didapatkan grafik hubungan antara tegangan terhadap sudut pada sensor *flex* ke-5 seperti pada Gambar 4.13.



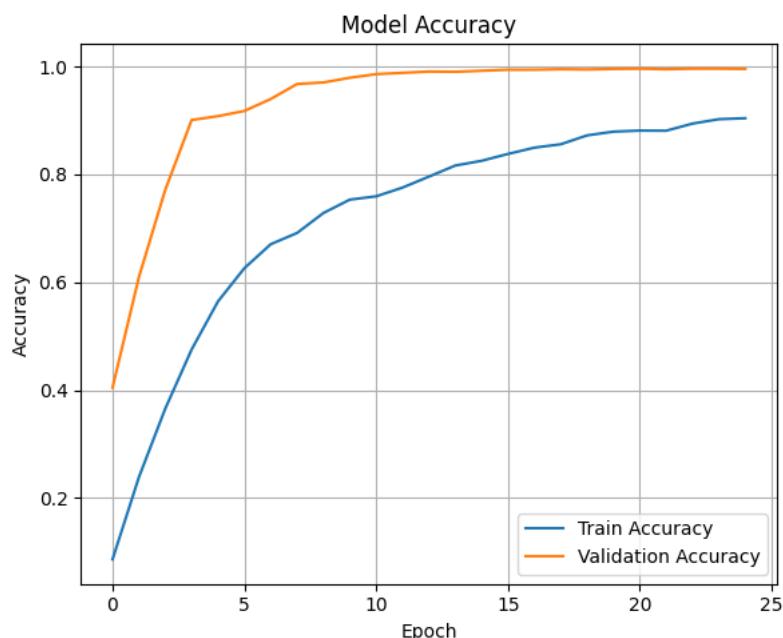
Gambar 4.13: Pengukuran sensitivitas tegangan terhadap sudut pada sensor *flex* ke-5.

Dapat dilihat pada grafik, data dari multimeter menunjukkan hubungan linear terbalik yang sangat kuat. Persamaan regresi linearnya adalah $y = -0.0075x + 2.3135$, dengan nilai $R^2 = 0.9883$. Tanda negatif pada kemiringan ($-0.0075V/\theta$) mengkonfirmasi bahwa tegangan keluaran menurun seiring dengan meningkatnya sudut tekukan. Intersep sebesar 2.3135 V adalah estimasi V_{out} pada sudut 0 derajat. Adapun pada data yang diperoleh dari serial monitor juga menunjukkan hubungan linear terbalik yang kuat dengan persamaan regresi $y = -0.0066x + 1.929$ dan $R^2 = 0.9860$.

Hubungan linear terbalik pada sensor *flex* ke-5 ini juga sesuai dengan ekspektasi berdasarkan Persamaan 4.2. Ketika sudut tekukan (θ) meningkat, R_{flex} meningkat, menyebabkan penurunan nilai V_{out} . Nilai R^2 yang sangat tinggi dari kedua metode pengukuran untuk sensor ini menunjukkan linearitas dan reliabilitas yang sangat baik dari respons tegangan sensor terhadap perubahan sudut, dengan nilai R^2 untuk data serial monitor sedikit lebih rendah dibandingkan sensor *flex* ke-1.

4.4 Evaluasi Klasifikasi Jaringan Saraf Tiruan Data Abjad SIBI

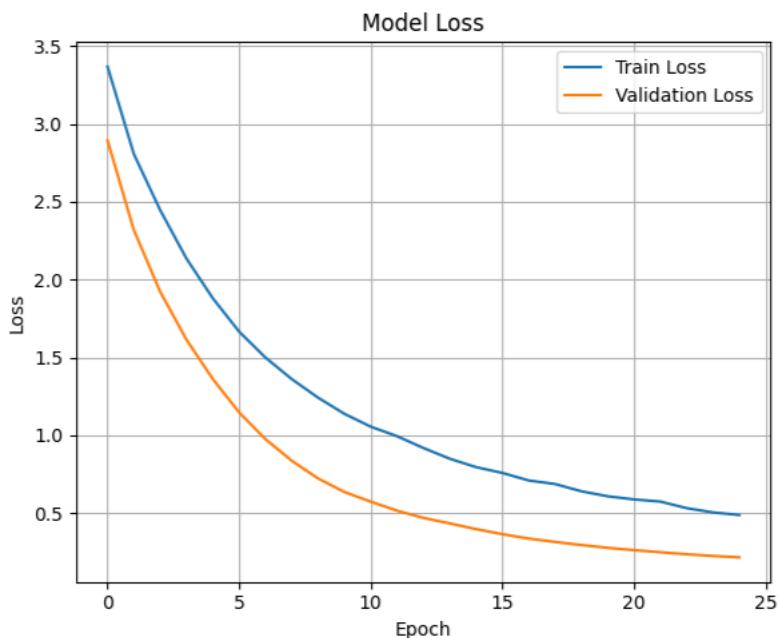
Pada sub bab ini akan dibahas mengenai evaluasi model klasifikasi jaringan saraf tiruan (ANN) yang telah dikembangkan untuk melakukan klasifikasi terhadap 26 abjad dalam Sistem Isyarat Bahasa Indonesia (SIBI). Hasil evaluasi model ini akan menampilkan kinerja model jaringan saraf buatan secara komprehensif, dimana meliputi kurva pembelajaran yakni; grafik *accuracy* dan grafik *loss*, serta *matrix confusion* untuk melihat kualitas klasifikasi data pada setiap data huruf A sampai Z (1 sampai 26). Sebelum masuk pada pembahasan kurva pembelajaran, perlu diketahui bahwa *epoch* dalam grafik adalah serangkaian iterasi total pelatihan pada seluruh data yang berjalan pada jaringan saraf tiruan (ANN) yang telah melewati sesi pelatihan (*train*) dan sesi validasi (*validation*). Adapun istilah *train* dan *validation* pada grafik adalah sesi pelatihan data saat dimasukan pada model jaringan saraf tiruan (ANN) yang disebut dengan sesi *train*, sedangkan *validation* merupakan sesi dimana data akan dimasukan pada model jaringan saraf tiruan (ANN) pada saat selesai sesi pelatihan. Pada evaluasi ini keseluruhan jumlah data yang akan digunakan pada model akan dibagi dengan persentase 20% untuk data validasi dan 80% untuk data pelatihan.



Gambar 4.14: Grafik akurasi pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI.

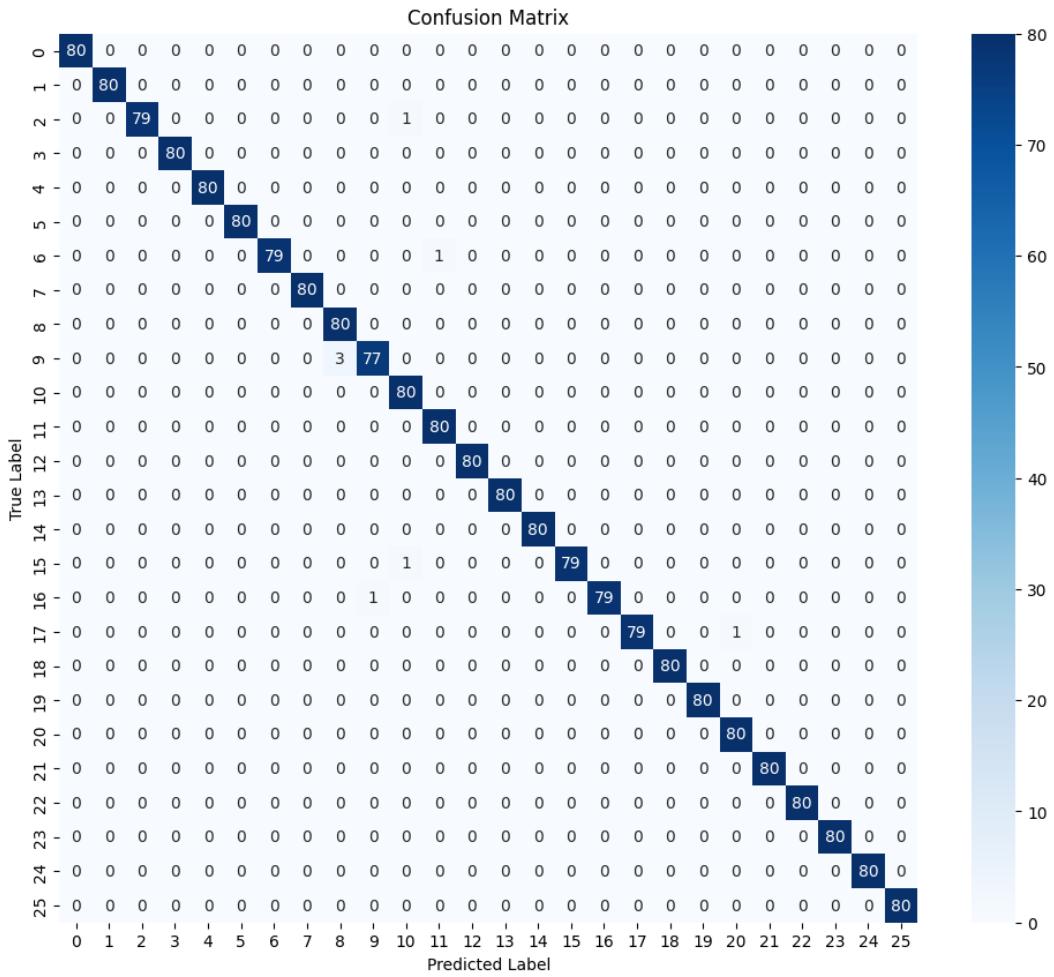
Pada grafik 4.14 dapat dilihat bahwa grafik akurasi pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI menunjukkan peningkatan yang signifikan seiring bertambahnya jumlah *epoch*. Observasi utama dari grafik ini adalah kedua kurva menunjukkan tren peningkatan pada setiap *epoch* dan kemudian konvergen menuju nilai yang sangat tinggi, yaitu mendekati 1.0 atau 100%. Penting untuk dicatat bahwa kurva akurasi validasi mengikuti kurva akurasi pelatihan dengan sangat ketat sepanjang proses. Konvergensi yang cepat menuju akurasi tinggi mengindikasikan bahwa model yang dirancang mampu belajar secara efisien dari data pelatihan yang diberikan.

Adapun terlihat konsisten kenaikan antara kurva akurasi pelatihan dan validasi grafik 4.14, di mana keduanya mencapai tingkat performa puncak yang serupa, merupakan indikasi kuat bahwa model memiliki kemampuan generalisasi yang baik. Artinya, model tidak hanya menghafal data pelatihan tetapi juga mampu mengenali pola yang relevan untuk diterapkan pada data baru yang belum pernah dilihat sebelumnya (data validasi). Pada grafik 4.14, terlihat bahwa tidak ada indikasi terjadinya *overfitting*, di mana model berkinerja sangat baik pada data pelatihan tetapi buruk pada data validasi. Seringkali hal tersebut dapat dilihat dari pergerakan kurva akurasi dan kurva validasi yang mempunyai bentuk yang benar-benar sama tanpa ada gap celah, artinya model benar-benar hanya menghafal data yang diproses tanpa bisa berkembang bila mana saat diberikan data baru diluar data pelatihan, maka model akan menghasilkan prediksi yang bisa jadi salah.



Gambar 4.15: Grafik loss pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI.

Pada grafik 4.15 dapat dilihat bahwa grafik loss pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI menunjukkan penurunan yang signifikan seiring bertambahnya jumlah *epoch*. Observasi utama dari grafik ini adalah kedua kurva menunjukkan tren penurunan pada setiap *epoch* dan kemuadian Konvergensi kedua kurva loss pada nilai yang sangat rendah ini semakin memperkuat kesimpulan yang ditarik dari analisis grafik 4.14, yaitu model tidak mengalami *overfitting* dan telah belajar dengan efektif. Tingkat kurva pelatihan dan kurva validasi pada grafik 4.15 yang sangat rendah (mendekati nol) untuk kedua set data secara kuantitatif, hal ini mendukung klaim kualitatif dari kurva akurasi bahwa model memiliki tingkat akurasi yang sangat tinggi. Ini berarti bahwa perbedaan atau selisih antara prediksi yang dihasilkan oleh model dan nilai sebenarnya sangat minimal. Artinya model dapat benar-benar mengenali pola data yang nantinya akan memberikan hasil respon prediksi akurat. Buktiya dapat dilihat pada Gambar 5.2 di Lampiran A yakni hasil beberapa sample tes pada program *python*.



Gambar 4.16: Matrix confusion hasil pelatihan jaringan saraf tiruan (ANN) untuk klasifikasi data abjad SIBI.

Pada Gambar *matrix confusion* 4.16 adalah sebuah matriks berukuran 26x26, di mana baris merepresentasikan kelas aktual (*True Class*) dan kolom merepresentasikan kelas yang diprediksi oleh model (*Predicted Class*). Observasi utama dari matriks ini adalah bahwa sebagian besar nilai terkonsentrasi di sepanjang diagonal utama, sementara nilai-nilai di luar diagonal (*off-diagonal*) sangat sedikit atau bahkan nol. Setiap sel pada diagonal utama menunjukkan jumlah prediksi yang benar, yaitu *True Positives* (TP) untuk masing-masing kelas abjad. Nilai diagonal utama (dari kiri atas ke kanan bawah) menunjukkan jumlah prediksi yang benar untuk masing-masing kelas, sedangkan nilai di luar diagonal menunjukkan jumlah prediksi yang salah. Dari matriks ini, terlihat bahwa model berhasil mengklasifikasikan sebagian besar abjad SIBI dengan akurasi yang sangat tinggi. Adapun data *precision*, *f1-score*, *recall* dan *support* pada tabel 5.32 di Lampiran A memberikan informasi tambahan tentang kinerja model untuk setiap kelas abjad. Adapun hasil laporan klasifikasi data abjad SIBI yang diperoleh pada program *python* dapat dilihat pada Gambar 5.1.

4.5 Pembahasan

Pembahasan secara keseluruhan pada sarung tangan pintar sebagai penerjemah SIBI ke suara tidak hanya merupakan penjumlahan dari komponen-komponen individualnya, melainkan hasil dari interaksi sinergis antara sensor-sensor yang terkalibrasi secara akurat dan jaringan saraf tiruan dalam mengklasifikasi data. Kinerja sistem ini dapat diibaratkan sebagai sebuah rantai, di mana kegagalan atau ketidakakuratan pada satu komponen baik itu kalibrasi sensor, akuisisi data, maupun model klasifikasi secara signifikan menurunkan kualitas prediksi model maupun hasil keluaran sistem (*output*).

Peran fundamental kalibrasi sensor IMU MPU6050 terletak pada penyediaan kesadaran spasial tangan yang akurat, yang krusial untuk menginterpretasikan gestur yang melibatkan orientasi dan gerakan. Hasil kalibrasi menunjukkan bahwa konfigurasi ke-7 menghasilkan data yang paling mendekati ideal, dengan nilai percepatan untuk sumbu X sebesar $0.01g$, sumbu Y sebesar $0.00g$, dan sumbu Z sebesar $1.00g$, serta kecepatan sudut untuk sumbu X sebesar $-0.01rad/s$, sumbu Y sebesar $0.00rad/s$, dan sumbu Z sebesar $0.00rad/s$. Angka-angka ini menunjukkan keberhasilan dalam meminimalkan zero-bias error, yang sebelum kalibrasi menunjukkan deviasi signifikan seperti Accel Z sekitar $-0.8g$ dan offset pada data giroskop antara -0.005 hingga $-0.015rad/s$. Pencapaian kalibrasi ini bukan sekadar prestasi teknis, melainkan faktor penentu dalam kemampuan sistem untuk membedakan isyarat SIBI. Banyak isyarat SIBI tidak hanya berbeda dalam fleksi jari, tetapi juga dalam kemiringan atau rotasi tangan yang subtil. Tanpa kalibrasi IMU yang presisi, sistem dapat salah menginterpretasikan orientasi statis atau mendeteksi gerakan "hantu", yang akan mengakibatkan tingkat kesalahan tinggi dalam pengenalan gerakan tangan. Sebagai contoh, kemampuan untuk membedakan antara telapak tangan yang datar dan yang sedikit miring sangat bergantung pada akurasi pengukuran akselerasi sumbu Z (vektor gravitasi) dan drift minimal pada sumbu X dan Y. Demikian pula, isyarat dinamis yang melibatkan rotasi, seperti huruf 'J', memerlukan pembacaan giroskop yang akurat. Kalibrasi yang berhasil ini membangun dasar yang reliabel untuk data orientasi dan gerak 3D yang diumpulkan ke jaringan saraf tiruan, secara signifikan mengurangi ambiguitas dalam set fitur masukan.

Reliabilitas sensor flex, yang tecermin dari karakteristik resistansinya, adalah pilar berikutnya dalam kinerja sarung tangan. Pengukuran menunjukkan hubungan linear yang kuat antara perubahan resistansi dan sudut tekuk untuk kelima sensor flex, sebagaimana dibuktikan oleh nilai R^2 yang tinggi, misalnya untuk sensor flex ke-1 (jari kelingking) diperoleh $R^2 = 0.9890$ dari monitor serial dan $R^2 = 0.9902$ dari multimeter, serta untuk sensor flex ke-4 (jari telunjuk) $R^2 = 0.9669$ (serial) dan $R^2 = 0.9550$ (multimeter). Respons linear ini krusial untuk menangkap gerakan jari secara akurat. Meskipun terdapat sedikit variasi dalam sensitivitas antar sensor—misalnya, slope $0.1525K\Omega/\Theta$ untuk sensor flex ke-1 dibandingkan dengan $0.0965K\Omega/\Theta$ untuk sensor flex ke-4 keberhasilan jaringan saraf tiruan dalam sub-bab 4.4 mengindikasikan kemampuannya untuk mempelajari dan mengkompensasi karakteristik individual sensor ini, selama masing-masing sensor itu sendiri konsisten. Jika sistem berbasis ambang batas sederhana yang digunakan, perbedaan sensitivitas ini dapat menyebabkan satu jari terdaftar sebagai "tertekuk" pada sudut fisik yang sedikit berbeda dari yang lain untuk perubahan resistansi yang sama. Namun, jaringan saraf tiruan mempelajari kurva respons unik dari masing-masing 5 input sensor flex sebagai bagian dari vektor input 11 fiturnya. Perubahan resistansi yang reliabel ini secara langsung diterjemahkan menjadi input yang andal untuk rangkaian pembagi tegangan, yang merupakan langkah selanjutnya dalam akuisisi data.

Karakteristik tegangan sensor flex yang juga konsisten dan menunjukkan hubungan linear (terbalik) lebih lanjut memvalidasi kesesuaianya dan memastikan bahwa konversi analog-ke-digital (ADC) secara akurat mencerminkan fleksibilitas jari. Kelima sensor menunjukkan hubungan linear terbalik yang kuat antara tegangan keluaran dan sudut tekuk, dengan nilai R^2 yang tinggi, seperti pada sensor 1 ($R^2 = 0.9959$ serial, $R^2 = 0.9909$ multimeter) dan sensor flex ke-3 ($R^2 = 0.9942$ serial, $R^2 = 0.9962$ multimeter). Perilaku ini konsisten dengan persamaan pembagi tegangan 4.2, di mana peningkatan R_{Flex} akibat tekukan yang lebih besar akan menyebabkan penurunan V_{out} . Linearitas respons tegangan yang tinggi ini memastikan bahwa ADC pada mikrokontroler ESP32 menerima sinyal yang secara prediktif memetakan tekukan fisik jari. Artinya, fitur-fitur yang mewakili tekukan jari yang diumpulkan ke jaringan saraf tiruan adalah representasi yang setia dari gestur fisik. Konsistensi hasil pengukuran resistansi dan tegangan melalui dua metode berbeda (multimeter dan monitor serial) semakin meningkatkan kepercayaan terhadap integritas akuisisi data keseluruhan sistem.

Adapun model jaringan saraf tiruan (ANN) yang dikembangkan menunjukkan kemampuan luar biasa dalam mempelajari pola kompleks dari data sensor gabungan untuk mengklasifikasikan 26 gestur abjad SIBI secara akurat. Kurva akurasi pelatihan dan validasi yang konvergen mendekati 100%, serta kurva kerugian (*loss*) pelatihan dan validasi yang konvergen mendekati nol, adalah bukti kuat akan hal ini. Fakta bahwa akurasi validasi mengikuti akurasi pelatihan dengan ketat mengindikasikan kemampuan generalisasi model yang baik dan tidak adanya overfitting yang signifikan. Kemampuan model untuk menggeneralisasi dengan baik sangat penting karena gestur SIBI dari pengguna di dunia nyata akan memiliki sedikit variasi dibandingkan dengan data pelatihan. Model yang dapat menggeneralisasi dengan baik mampu mengakomodasi nuansa ini, menghasilkan perangkat yang lebih praktis dan ramah pengguna. Akurasi tinggi yang dicapai juga menunjukkan bahwa 11 fitur yang dipilih (5 data sensor flex, 3 data akselerometer, dan 3 data giroskop) cukup deskriptif untuk membedakan antara 26 huruf abjad SIBI.

Kualitas klasifikasi model jaringan saraf tiruan (ANN) yang tinggi semakin diperkuat oleh analisis *confusion matrix*. Matriks ini menunjukkan bahwa nilai-nilai terkonsentrasi kuat di sepanjang diagonal utama, yang mengindikasikan klasifikasi yang benar, dengan sangat sedikit nilai di luar diagonal. Laporan klasifikasi yang terperinci (Tabel 5.32 dan Gambar 5.1) menunjukkan nilai *precision*, *recall*, dan *f1-score* yang tinggi (kebanyakan 0.99 atau 1.00) untuk hampir semua huruf individual. Diagonal yang kuat dalam confusion matrix dan metrik per kelas yang tinggi, seperti *f1-score* 1.00 untuk huruf 'A', 'B', 'D', 'E', 'S', 'T', 'Z', dan lainnya, menandakan bahwa model tidak hanya baik secara rata-rata, tetapi juga berkinerja andal di seluruh rentang gestur yang beragam dalam abjad SIBI. Meskipun beberapa huruf mungkin secara inheren lebih mudah tertukar, data menunjukkan bahwa model menangani sebagian besar perbedaan dengan sangat baik. Sebagai contoh, meskipun huruf 'M' dan 'N' disebut berpotensi mirip dalam bagian saran penelitian, jaringan saraf tiruan saat ini (Tabel 5.32) menunjukkan *f1-score* 1.00 untuk keduanya, menunjukkan bahwa model dengan sensor yang ada dapat membedakannya dengan baik. Kualitas data sensor yang tinggi dan terkalibrasi pada sub-bab sebelumnya merupakan faktor pendukung langsung dari akurasi klasifikasi yang tinggi oleh jaringan saraf tiruan ini; data yang baik memungkinkan jaringan saraf tiruan untuk belajar secara efektif.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada tugas akhir ini dapat disimpulkan sebagai berikut:

1. Sistem komunikasi untuk membantu penyandang disabilitas tunawicara telah berhasil dikembangkan. Ini terbukti dengan sarung tangan pintar yang mampu menerjemahkan gestur abjad SIBI dengan akurasi tinggi menggunakan jaringan saraf tiruan, sehingga berpotensi menjembatani komunikasi dengan masyarakat.
2. Prinsip kerja dan aspek fisis instrumentasi sensor telah berhasil dianalisis. Pemahaman ini dicapai melalui kalibrasi sensor IMU MPU6050 untuk data gerak yang akurat, serta karakterisasi respons resistansi dan tegangan sensor flex terhadap sudut tekuk , yang menunjukkan pemahaman mendalam terhadap cara kerja sensor.
3. Perangkat asistif untuk membantu riset teknologi bagi penyandang disabilitas tunawicara di Indonesia telah berhasil dibuat. Perwujudan ini berupa prototipe sarung tangan pintar yang fungsional dan berkinerja tinggi dalam mengenali abjad SIBI, memberikan kontribusi nyata bagi pengembangan teknologi asistif di Indonesia.

5.2 Saran

Berdasarkan hasil dari tugas akhir yang telah dilakukan untuk pengembangan lebih lanjut dapat dilakukan beberapa saran dan masukan antara lain:

1. Melakukan pengambilan data gerakan tangan dengan orang yang berbeda-beda. Hal ini akan membuat keanekaragaman gerakan pada setiap data abjad SIBI. Sehingga model jaringan syarat tiruan dapat berkembang dan beradaptasi dengan bermacam-macam bentuk tangan orang.
2. Menambahkan sensor tekanan (*Force Sensor*) sehingga dapat menambahkan parameter pada setiap jenis gerakan tangan yang mempunyai bentuk identik, seperti salah satu contohnya yakni gerakan tangan SIBI pada huruf M dan N.
3. Penelitian dapat dikembangkan dengan menambahkan modul sehingga dapat menerjemahkan bahasa isyarat menggunakan kedua tangan. Tentunya perlu beberapa pengaturan mendalam pada program utama sarung tangan pintar tunawicara ini, sehingga dapat keseilarasan pada respon gerakan kedua tangan.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A

Tabel 5.1: Data Kalibrasi Sensor IMU untuk Berbagai Konfigurasi.

Konfigurasi	Jenis Data	Parameter Sumbu Akselerometer			Parameter Sumbu Giroskop		
		X (ax)	Y (ay)	Z (az)	X (gx)	Y (gy)	Z (gz)
		[m/s ²]	[m/s ²]	[m/s ²]	[rad/s]	[rad/s]	[rad/s]
1	Parameter Offset Kalibrasi	-0.07	0.01	-0.03	0.02	0.02	0
	Data Output Gyro	0.14	-0.03	1.07	-0.04	-0.04	0
2	Parameter Offset Kalibrasi	-0.14	0.03	-0.07	0	0	0
	Data Output Gyro	0.19	-0.03	1.11	-0.01	-0.02	0
3	Parameter Offset Kalibrasi	0.14	-0.02	0.02	0	0	0
	Data Output Gyro	-0.13	0.01	1.03	-0.02	-0.02	0
4	Parameter Offset Kalibrasi	0.07	-0.03	0.04	0	0	0
	Data Output Gyro	-0.07	0.02	1.00	-0.02	-0.02	0
5	Parameter Offset Kalibrasi	0.03	-0.04	0.04	-0.01	-0.01	0
	Data Output Gyro	0.14	-0.03	1.07	-0.04	-0.04	0
6	Parameter Offset Kalibrasi	0.02	-0.02	0.04	-0.01	-0.01	0
	Data Output Gyro	-0.03	0.01	1.00	0	-0.01	0
7	Parameter Offset Kalibrasi	0	-0.02	0.04	-0.01	-0.01	0
	Data Output Gyro	0.01	0	1.00	0	-0.01	0

Tabel 5.2: Data Kalibrasi Sensor Flex Resistansi terhadap Sudut Tekuk (Sensor 1-3).

Sudut (θ°)	Sensor Flex ke-1 (K Ω)		Sensor Flex ke-2 (K Ω)		Sensor Flex ke-3 (K Ω)	
	Multimeter	Serial Monitor	Multimeter	Serial Monitor	Multimeter	Serial Monitor
0	11,97	13,48	11,17	12,69	10,55	11,93
30	14,73	17,01	12,97	14,43	12,95	14,89
45	17,23	19,43	14,77	16,74	15,85	17,83
60	19,53	21,70	16,47	18,07	17,83	19,55
90	23,30	27,17	19,77	22,11	22,20	25,06

Tabel 5.3: Data Kalibrasi Sensor Flex Resistansi terhadap Sudut Tekuk (Sensor 4-5).

Sudut (θ°)	Sensor Flex ke-4 ($K\Omega$)		Sensor Flex ke-5 ($K\Omega$)	
	Multimeter	Serial Monitor	Multimeter	Serial Monitor
0	9,42	10,53	11,37	13,14
30	10,82	12,31	12,97	15,21
45	12,07	13,61	15,27	17,74
60	13,65	15,39	16,27	18,63
90	17,24	19,15	20,20	23,35

Tabel 5.4: Data Kalibrasi Sensor Flex Tegangan terhadap Sudut Tekuk (Sensor 1-3).

Sudut (θ°)	Sensor Flex ke-1 (V)		Sensor Flex ke-2 (V)		Sensor Flex ke-3 (V)	
	Multimeter	Serial Monitor	Multimeter	Serial Monitor	Multimeter	Serial Monitor
0	2,3	1,89	2,06	1,96	2,46	2,02
30	2,01	1,64	1,99	1,82	2,17	1,78
45	1,83	1,5	1,98	1,66	1,97	1,6
60	1,74	1,4	1,92	1,58	1,86	1,5
90	1,5	1,19	1,68	1,38	1,57	1,27

Tabel 5.5: Data Kalibrasi Sensor Flex Tegangan terhadap Sudut Tekuk (Sensor 4-5).

Sudut (θ°)	Sensor Flex ke-4 (V)		Sensor Flex ke-5 (V)	
	Multimeter	Serial Monitor	Multimeter	Serial Monitor
0	2,6	2,16	2,3	1,92
30	2,4	1,99	2,11	1,76
45	2,27	1,88	1,95	1,6
60	2,13	1,75	1,9	1,55
90	1,9	1,52	1,62	1,33

Tabel 5.6: Data Huruf A (ClassID = 0) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1421	1200	1114	1109	818	0.15	0.99	0.12	0.03	-0.04	0.01	0
2	1431	1207	1109	1118	805	0.16	1	0.15	-0.01	0.02	0.01	0
3	1456	1232	1118	1113	844	0.17	0.98	0.14	-0.03	-0.03	0	0
4	1417	1269	1079	1136	816	0.14	0.99	0.14	0	-0.04	-0.01	0
...
400	1447	1226	1123	1136	848	0.16	0.95	0.26	-0.06	0.02	0.01	0

Tabel 5.7: Data Huruf B (ClassID = 1) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1200	1904	1766	1728	1535	0.04	0.19	0	0	0.02	-0.02	1
2	1203	1908	1763	1719	1535	0.21	0.98	0.16	0.01	0.04	-0.01	1
3	1263	1914	1764	1717	1562	0.23	0.98	0.15	0.01	0.03	0	1
4	1217	1911	1763	1729	1564	0.21	0.98	0.17	-0.01	0.05	0	1
...
400	1264	1901	1776	1712	1595	0.21	0.97	0.21	-0.02	-0.03	0.01	1

Tabel 5.8: Data Huruf C (ClassID = 2) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1597	1682	1401	1427	1323	0.03	0.17	0.06	0	-0.02	0.03	2
2	1583	1643	1392	1429	1313	0.1	0.87	0.52	0.03	0.01	0.05	2
3	1584	1631	1415	1418	1333	0.13	0.85	0.53	-0.02	-0.03	0.12	2
4	1595	1669	1395	1444	1328	0.15	0.85	0.53	0.04	-0.01	0.05	2
...
400	1616	1681	1403	1442	1287	0.16	0.92	0.37	0	0.02	0.01	2

Tabel 5.9: Data Huruf D (ClassID = 3) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1178	1927	1269	1166	893	0.04	0.19	-0.02	0	0.02	0.02	3
2	1186	1931	1295	1149	918	0.2	0.99	0.06	0.01	-0.02	0	3
3	1168	1918	1280	1116	912	0.21	1	0.05	-0.03	-0.02	-0.01	3
4	1168	1911	1245	1143	909	0.2	0.99	0.07	-0.03	-0.02	0.02	3
...
400	1227	1917	1269	1163	927	0.21	0.98	0.11	0	-0.01	0	3

Tabel 5.10: Data Huruf E (ClassID = 4) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1336	1050	982	1136	867	0.03	0.19	-0.01	-0.04	0.04	0.03	4
2	1328	1040	997	1136	895	0.15	0.99	0.12	-0.03	-0.05	-0.01	4
3	1335	1019	998	1141	901	0.12	0.99	0.13	-0.01	-0.08	0	4
4	1354	1035	1008	1093	906	0.13	0.99	0.13	0.01	-0.06	-0.02	4
...
400	1344	1121	1019	1158	957	0.1	0.98	0.13	-0.01	0	0.01	4

Tabel 5.11: Data Huruf F (ClassID = 5) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1469	1535	1649	1658	1563	0.03	0.18	-0.01	-0.01	-0.09	-0.03	5
2	1462	1535	1657	1666	1578	0.25	0.98	0.13	-0.01	0.04	0.02	5
3	1455	1519	1685	1665	1587	0.25	0.99	0.13	-0.01	-0.02	0.02	5
4	1461	1512	1664	1672	1587	0.21	0.96	0.13	-0.02	0	0.01	5
...
400	1483	1531	1675	1637	1616	0.2	0.99	0.11	-0.01	0.07	0	5

Tabel 5.12: Data Huruf G (ClassID = 6) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1712	1837	1152	1190	977	0.03	0.19	0.03	0	-0.09	-0.03	6
2	1716	1820	1156	1200	982	0.15	0.95	0.31	-0.01	0.03	0	6
3	1717	1825	1155	1158	1007	0.18	0.94	0.31	-0.02	0.03	0.01	6
4	1717	1828	1153	1155	983	0.17	0.94	0.31	0	-0.06	-0.02	6
...
400	1750	1829	1218	1168	933	0.17	0.91	0.38	-0.01	-0.01	0	6

Tabel 5.13: Data Huruf H (ClassID = 7) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1301	1851	1648	1207	923	-0.48	0.75	0.39	-0.03	-0.05	-0.05	7
2	1302	1859	1631	1205	912	-0.48	0.73	0.42	0.01	-0.03	0	7
3	1319	1851	1618	1218	931	-0.47	0.73	0.38	-0.03	0	-0.01	7
4	1303	1856	1644	1210	940	-0.45	0.75	0.42	0	-0.02	0.01	7
...
400	1366	1839	1616	1252	969	-0.5	0.72	0.43	-0.03	0.02	0	7

Tabel 5.14: Data Huruf I (ClassID = 8) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1401	1349	1178	1198	1685	0.28	0.94	0.09	-0.03	-0.01	-0.02	8
2	1402	1418	1168	1250	1680	0.29	0.94	0.1	0.04	0.05	0.03	8
3	1402	1415	1174	1169	1683	0.31	0.97	0.14	-0.04	0.02	0	8
4	1415	1442	1166	1193	1680	0.34	0.95	0.09	0.01	-0.06	0.02	8
...
400	1414	1401	1168	1205	1631	0.37	0.94	0.16	-0.02	-0.02	0	8

Tabel 5.15: Data Huruf J (ClassID = 9) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1273	1171	1118	1118	1604	-0.19	0.24	-0.03	-0.76	4.38	2.56	9
2	1328	1183	1106	1136	1571	-0.16	0.44	0.16	0.08	-1.38	-0.1	9
3	1296	1203	1075	1140	1553	-1.04	1.05	-0.05	1.73	-3.75	-2.98	9
4	1279	1173	1086	1129	1534	-0.69	0.77	0.08	0.68	-4.36	0.54	9
...
400	1259	1202	1154	1166	1558	-1.17	0.91	-0.21	-0.33	4.38	-0.09	9

Tabel 5.16: Data Huruf K (ClassID = 10) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1710	1909	1501	1280	878	0	1	0.03	0	-0.01	-0.01	10
2	1703	1904	1567	1279	944	0	1.01	0.04	-0.02	-0.01	0	10
3	1724	1917	1507	1291	1008	0	1	0.03	0	0.01	0	10
4	1712	1870	1580	1223	940	-0.01	1.01	0.03	0	-0.01	0	10
...
400	1709	1913	1557	1299	981	-0.03	0.99	0.19	-0.04	-0.06	-0.01	10

Tabel 5.17: Data Huruf L (ClassID = 11) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1840	1921	1155	1163	935	0.11	0.95	0.28	-0.02	-0.09	-0.01	11
2	1834	1923	1246	1161	922	0.1	0.97	0.28	-0.01	-0.03	0	11
3	1842	1922	1148	1167	914	0.1	0.96	0.27	0	0	-0.01	11
4	1839	1901	1147	1117	927	0.09	0.96	0.28	-0.02	0.03	-0.03	11
...
400	1831	1907	1162	1199	944	0.16	0.97	0.28	0	-0.01	0.01	11

Tabel 5.18: Data Huruf M (ClassID = 12) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1255	1293	1279	1329	1075	0.05	0.19	0.01	0.01	-0.01	-0.03	12
2	1120	1303	1263	1341	1069	0.29	0.96	0.19	0.01	-0.04	-0.04	12
3	977	1306	1328	1344	1072	0.23	0.95	0.19	0.01	-0.02	0.01	12
4	1115	1294	1280	1331	1072	0.25	0.97	0.2	-0.03	-0.02	0	12
...
400	1147	1326	1309	1319	1009	0.18	0.98	0.12	0.04	0.34	-0.01	12

Tabel 5.19: Data Huruf N (ClassID = 13) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1269	1331	1252	1102	832	0.05	0.99	0.23	0	0.03	0.01	13
2	1206	1311	1254	1118	837	0	1	0.25	0.03	-0.1	-0.01	13
3	1237	1334	1264	1098	833	-0.01	0.97	0.21	-0.03	0.04	0	13
4	1282	1327	1264	1110	832	0.04	1	0.25	-0.05	-0.03	0.01	13
...
400	1232	1394	1327	1163	870	-0.01	0.97	0.23	-0.03	-0.01	0.01	13

Tabel 5.20: Data Huruf O (ClassID = 14) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1264	1410	1299	1249	1195	0.07	0.95	0.35	0.03	0.02	0.01	14
2	1264	1419	1299	1258	1186	0.05	0.94	0.34	-0.05	0.01	0.01	14
3	1268	1423	1300	1278	1168	0.06	0.93	0.35	-0.03	-0.01	0.01	14
4	1275	1424	1338	1296	1168	0.07	0.94	0.36	0	0.01	0.01	14
...
400	1240	1370	1196	1263	1061	0.06	0.92	0.4	-0.04	0.04	0	14

Tabel 5.21: Data Huruf P (ClassID = 15) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1778	1866	1566	1197	892	0.04	-0.06	0.12	-0.03	0.02	-0.01	15
2	1789	1879	1568	1204	862	0.19	-0.47	0.86	-0.03	0.04	-0.01	15
3	1791	1895	1562	1214	875	0.18	-0.46	0.87	0.04	-0.07	-0.01	15
4	1787	1888	1523	1205	895	0.2	-0.45	0.88	-0.03	-0.05	0.03	15
...
400	1782	1886	1523	1221	912	0.08	-0.23	0.96	-0.02	-0.07	-0.01	15

Tabel 5.22: Data Huruf Q (ClassID = 16) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1792	1813	1134	1149	871	0.05	-0.07	0.11	-0.07	0	-0.01	16
2	1795	1814	1118	1140	899	0.28	-0.52	0.8	-0.06	0.08	0.02	16
3	1791	1821	1099	1150	905	0.29	-0.5	0.82	-0.11	0.06	0.01	16
4	1794	1822	1008	1140	873	0.25	-0.53	0.8	0.04	-0.06	0.01	16
...
400	1807	1842	1167	1185	933	0.24	-0.47	0.83	0.01	0.02	0	16

Tabel 5.23: Data Huruf R (ClassID = 17) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1212	1840	1661	1018	915	0.01	0.99	0.17	-0.01	-0.02	0.01	17
2	1318	1826	1653	1044	879	0	0.99	0.16	0.02	-0.01	-0.04	17
3	1262	1847	1648	1054	923	0.06	1	0.14	0.05	-0.22	-0.01	17
4	1269	1940	1674	1053	929	0	0.99	0.15	-0.04	0	0.04	17
...
400	1294	1857	1649	1133	966	0.07	0.98	0.17	0.01	-0.02	-0.01	17

Tabel 5.24: Data Huruf S (ClassID = 18) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1200	1386	1023	1099	823	-0.01	0.98	0.3	0	0.05	0	18
2	1200	1367	1114	1102	859	0.01	0.99	0.31	-0.04	-0.11	-0.02	18
3	1201	1405	1118	1093	855	0.03	0.93	0.29	0.02	0.01	0.02	18
4	1216	1392	1110	1106	851	-0.01	0.95	0.31	0.03	-0.01	0	18
...
400	1264	1392	1162	1153	902	0.06	0.92	0.33	0.01	-0.03	0	18

Tabel 5.25: Data Huruf T (ClassID = 19) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1465	1520	1098	1145	823	-0.01	0.2	-0.01	-0.02	-0.01	0	19
2	1462	1519	1083	1137	814	-0.04	0.99	0.12	0	0	-0.02	19
3	1451	1488	1101	1146	826	-0.07	0.99	0.1	-0.01	0	0	19
4	1456	1506	1088	1072	823	-0.05	0.98	0.11	-0.02	0.03	0	19
...
400	1461	1535	1158	1168	886	-0.05	0.98	0.24	0.01	0.02	0.01	19

Tabel 5.26: Data Huruf U (ClassID = 20) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1166	1908	1716	1184	907	0.01	0.19	0.03	-0.01	-0.06	-0.02	20
2	1190	1904	1692	1170	943	0.08	0.96	0.31	-0.02	-0.06	0.05	20
3	1196	1902	1724	1195	944	0.09	0.94	0.3	-0.04	-0.11	0.03	20
4	1184	1905	1709	1182	929	0.1	0.94	0.34	-0.04	-0.03	0.06	20
...
400	1232	1872	1690	1227	990	0.16	0.95	0.31	-0.07	0.02	0	20

Tabel 5.27: Data Huruf V (ClassID = 21) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1399	1937	1776	1236	1013	-0.01	0.92	0.39	-0.02	0.02	-0.01	21
2	1391	1944	1762	1215	979	-0.03	0.91	0.39	-0.01	-0.09	-0.02	21
3	1405	1934	1774	1232	1008	-0.05	0.9	0.38	-0.01	-0.05	0	21
4	1381	1933	1786	1246	1015	-0.02	0.92	0.4	-0.01	0.01	-0.01	21
...
400	1343	1915	1678	1246	1100	0.01	0.94	0.32	0.05	-0.03	0.03	21

Tabel 5.28: Data Huruf W (ClassID = 22) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1278	1923	1713	1711	935	0.01	0.19	0.02	0.01	0	0	22
2	1275	1936	1680	1718	912	0.11	0.95	0.3	0.01	-0.05	0	22
3	1296	1929	1734	1739	928	0.07	0.95	0.29	-0.01	0.03	0	22
4	1316	1934	1739	1744	931	0.08	0.97	0.31	0.03	-0.04	0.01	22
...
400	1307	1902	1697	1699	1040	0.12	0.95	0.29	-0.02	0.02	-0.02	22

Tabel 5.29: Data Huruf X (ClassID = 23) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1264	1424	985	1089	862	-0.13	0.95	0.18	-0.03	-0.03	0	23
2	1232	1459	1023	1088	880	-0.12	0.96	0.18	0.04	0.01	0.03	23
3	1254	1424	997	1082	880	-0.11	0.97	0.18	0.02	-0.01	0.03	23
4	1228	1408	1023	1104	888	-0.08	0.98	0.18	-0.02	-0.1	0.01	23
...
400	1306	1469	1111	1124	914	0	0.95	0.17	0.01	-0.02	0.01	23

Tabel 5.30: Data Huruf Y (ClassID = 24) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1831	1200	1021	1231	1572	0.01	0.2	-0.01	0.01	-0.07	-0.01	24
2	1856	1211	1075	1235	1535	0.09	0.97	0.09	-0.01	0	0	24
3	1847	1230	1058	1241	1535	0.07	0.98	0.08	0.03	-0.09	0.03	24
4	1846	1177	1043	1218	1561	0.15	1.03	0.12	0.07	-0.19	0.02	24
...
400	1837	1264	1125	1273	1554	0.06	0.97	0.14	0.02	-0.01	-0.05	24

Tabel 5.31: Data Huruf Z (ClassID = 25) Sarung Tangan Pintar.

No	Flex Ke-5	Flex Ke-4	Flex Ke-3	Flex Ke-2	Flex Ke-1	Accel-X	Accel-Y	Accel-Z	Gyro-X	Gyro-Y	Gyro-Z	ClassID
1	1208	1901	1122	1097	899	0.01	0.16	0.07	0.6	-1.16	-0.71	25
2	1215	1902	1138	1121	907	-0.33	0.88	0.25	0.34	0.2	-0.08	25
3	1243	1903	1131	1108	892	0.06	0.85	0.53	-0.16	0.83	0.91	25
4	1235	1904	1087	1094	884	0.37	0.75	0.65	0.15	-0.41	-0.67	25
...
400	1242	1904	1158	1094	912	-0.29	0.43	0.85	-1.03	0.67	1.61	25

Tabel 5.32: Laporan klasifikasi pada setiap data huruf.

<i>Classification Report</i>				
Huruf	Precision	Recall	F1-Score	Support
A	1.00	1.00	1.00	80
B	1.00	1.00	1.00	80
C	1.00	0.99	0.99	80
D	1.00	1.00	1.00	80
E	1.00	1.00	1.00	80
F	1.00	1.00	1.00	80
G	1.00	0.99	0.99	80
H	1.00	1.00	1.00	80
I	0.96	1.00	0.98	80
J	0.99	0.96	0.97	80
K	0.98	1.00	0.99	80
L	0.99	1.00	0.99	80
M	1.00	1.00	1.00	80
N	1.00	1.00	1.00	80
O	1.00	1.00	1.00	80
P	1.00	0.99	0.99	80
Q	1.00	0.99	0.99	80
R	1.00	0.99	0.99	80
S	1.00	1.00	1.00	80
T	1.00	1.00	1.00	80
U	0.99	1.00	0.99	80
V	1.00	1.00	1.00	80
W	1.00	1.00	1.00	80
X	1.00	1.00	1.00	80
Y	0.99	1.00	1.00	80
Z	1.00	1.00	1.00	80

	precision	recall	f1-score	support
0	1.00	1.00	1.00	80
1	1.00	1.00	1.00	80
2	1.00	0.99	0.99	80
3	1.00	1.00	1.00	80
4	1.00	1.00	1.00	80
5	1.00	1.00	1.00	80
6	1.00	0.99	0.99	80
7	1.00	1.00	1.00	80
8	0.96	1.00	0.98	80
9	0.99	0.96	0.97	80
10	0.98	1.00	0.99	80
11	0.99	1.00	0.99	80
12	1.00	1.00	1.00	80
13	1.00	1.00	1.00	80
14	1.00	1.00	1.00	80
15	1.00	0.99	0.99	80
16	1.00	0.99	0.99	80
17	1.00	0.99	0.99	80
18	1.00	1.00	1.00	80
19	1.00	1.00	1.00	80
20	0.99	1.00	0.99	80
21	1.00	1.00	1.00	80
22	1.00	1.00	1.00	80
23	1.00	1.00	1.00	80
24	1.00	1.00	1.00	80
25	1.00	1.00	1.00	80
accuracy			1.00	2080
macro avg		1.00	1.00	2080
weighted avg		1.00	1.00	2080

Gambar 5.1: Laporan klasifikasi jaringan syaraf tiruan data abjad SIBI pada program *python*.

```

--- Sample Predictions ---
Sample 1: Actual ClassID = 5, Predicted ClassID = 5
Sample 2: Actual ClassID = 6, Predicted ClassID = 6
Sample 3: Actual ClassID = 25, Predicted ClassID = 25
Sample 4: Actual ClassID = 18, Predicted ClassID = 18
Sample 5: Actual ClassID = 1, Predicted ClassID = 1
Sample 6: Actual ClassID = 18, Predicted ClassID = 18
Sample 7: Actual ClassID = 7, Predicted ClassID = 7
Sample 8: Actual ClassID = 12, Predicted ClassID = 12
Sample 9: Actual ClassID = 24, Predicted ClassID = 24
Sample 10: Actual ClassID = 10, Predicted ClassID = 10

```

Gambar 5.2: Laporan prediksi klasifikasi beberapa sample jaringan syaraf tiruan data abjad SIBI pada program *python*.

[Halaman ini sengaja dikosongkan]

LAMPIRAN B

Listing 5.1: Program Pengukuran Kalibrasi Sensor Giroskop IMU MPU6050

```
1 #include <Adafruit_MPU6050.h>
2 #include <Adafruit_Sensor.h>
3 #include <Wire.h>
4
5 Adafruit_MPU6050 mpu;
6
7 // Offset dari kalibrasi (adjust sesuai dengan perbandingan hasil)
8 float ax_offset = 0.00; // g
9 float ay_offset = -0.02; // g
10 float az_offset = 0.04; // g
11 float gx_offset = -0.01; // rad/s
12 float gy_offset = -0.01; // rad/s
13 float gz_offset = 0.00; // rad/s
14
15 void setup() {
16     Serial.begin(115200);
17     while (!Serial) {
18         delay(100); // Tunggu Serial Monitor terbuka
19     }
20
21 // Inisialisasi MPU6050
22 if (!mpu.begin()) {
23     Serial.println("Gagal menemukan MPU6050!");
24     while (1);
25 }
26
27 // Konfigurasi sensor
28 mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
29 mpu.setGyroRange(MPU6050_RANGE_250_DEG);
30 mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
31
32 // Cetak header CSV sekali di awal
33 Serial.println("Accel X(g),Accel Y(g),Accel Z(g),Gyro X(rad/s),Gyro Y(rad/s),Gyro Z(
34     rad/s)");
35 }
36
37 void loop() {
38     sensors_event_t a, g, temp;
39     mpu.getEvent(&a, &g, &temp);
40
41     // Koreksi pembacaan dengan offset
42     float corrected_ax = (a.acceleration.x / 9.81) - ax_offset;
43     float corrected_ay = (a.acceleration.y / 9.81) - ay_offset;
44     float corrected_az = (a.acceleration.z / 9.81) - az_offset;
45     float corrected_gx = g.gyro.x - gx_offset;
46     float corrected_gy = g.gyro.y - gy_offset;
47     float corrected_gz = g.gyro.z - gz_offset;
48
49     // Cetak data dalam satu baris, dipisahkan koma, lalu ke baris baru
50     Serial.print(corrected_ax);
51     Serial.print(",");
52     Serial.print(corrected_ay);
```

```

52 Serial.print(",");
53 Serial.print(corrected_az);
54 Serial.print(",");
55 Serial.print(corrected_gx);
56 Serial.print(",");
57 Serial.print(corrected_gy);
58 Serial.print(",");
59 Serial.println(corrected_gz); // Baris baru setelah nilai terakhir
60 delay(200); // Jeda antar pembacaan
61 }

```

Listing 5.2: Program Pengukuran Sensitivitas Sensor Flex

```

1 #include <Arduino.h>
2
3 // Definisikan konstanta untuk perataan
4 const int jumlahPembacaan = 100;           // Jumlah pembacaan untuk dirata-rata
5 const int jedaAntarPembacaan = 100;         // Jeda antar pembacaan dalam milidetik
6 const float TeganganInput = 4.44;           // Tegangan input dalam volt (4.44V untuk
7 ESP32)
8 const float R1 = 10.0;                      // Nilai resistor tetap dalam kOhm
9
10 void setup() {
11     Serial.begin(115200); // Inisialisasi Serial dengan baud rate 115200
12     delay(1000);
13     Serial.println("Pengukuran Resistansi Sensor Fleksibel");
14 }
15
16 void loop() {
17     long totalNilaiSensor = 0;      // Variabel untuk menyimpan jumlah nilai sensor
18     float totalTegangan = 0.0;      // Variabel untuk menyimpan jumlah nilai tegangan
19     float totalResistansi = 0.0;    // Variabel untuk menyimpan jumlah nilai resistansi
20
21     // Lakukan beberapa pembacaan dan jumlahkan
22     for (int i = 0; i < jumlahPembacaan; i++) {
23         int nilaiSensor = analogRead(36);
24         totalNilaiSensor += nilaiSensor;
25
26         // Hitung tegangan untuk setiap pembacaan
27         float tegangan = nilaiSensor * (TeganganInput / 4095.0); // Konversi ke
28         tegangan (0-4.44V)
29         totalTegangan += tegangan;
30
31         // Hitung resistansi untuk setiap pembacaan
32         if (tegangan > 0) { // Hindari pembagian dengan nol
33             float resistansi = (TeganganInput * R1 / tegangan) - R1; // Hitung
34             resistansi dalam kOhm
35             totalResistansi += resistansi;
36         }
37         delay(jedaAntarPembacaan);
38     }
39
40     // Hitung rata-rata
41     int rataNilaiSensor = totalNilaiSensor / jumlahPembacaan;
42     float rataTegangan = totalTegangan / jumlahPembacaan;
43     float rataResistansi = totalResistansi / jumlahPembacaan;

```

```

41 // Cetak hasilnya
42 Serial.print("Nilai Analog (rata-rata): ");
43 Serial.print(rataNilaiSensor);
44 Serial.print(" | Tegangan (rata-rata): ");
45 Serial.print(rataTegangan);
46 Serial.print(" V");
47 Serial.print(" | Resistansi (rata-rata): ");
48 Serial.print(rataResistansi);
49 Serial.println(" kOhm");
50
51 delay(500); // Jeda untuk menjaga waktu loop sekitar 500 ms
52 }

```

Listing 5.3: Program Sarung Tangan Pintar

```

1 // Board: ESP32 Dev Module
2 // Libraries: Adafruit MPU6050, DFRobotDFPlayerMini, TensorFlow Lite Micro
3 #include <Wire.h>
4 #include <Adafruit_MPU6050.h>
5 #include <Adafruit_Sensor.h>
6 #include "DFRobotDFPlayerMini.h"
7
8 // TensorFlow Lite Micro libraries
9 #include <TensorFlowLite_ESP32.h>
10 #include "tensorflow/lite/micro/micro_error_reporter.h"
11 #include "tensorflow/lite/micro/micro_interpreter.h"
12 #include "tensorflow/lite/micro/all_ops_resolver.h"
13 #include "tensorflow/lite/schema/schema_generated.h"
14 #include "TA_Model.h"
15
16 // --- Deklarasi Fungsi (Prototipe) ---
17 char mapClassIDToSIBI(int class_id);
18 void printDetail(uint8_t type, int value);
19
20 // --- Konfigurasi Pin Sensor ---
21 const int flexPin1 = 36; // Flex1
22 const int flexPin2 = 39; // Flex2
23 const int flexPin3 = 34; // Flex3
24 const int flexPin4 = 35; // Flex4
25 const int flexPin5 = 32; // Flex5
26
27
28 const int flexPinArray[] = {flexPin5, flexPin4, flexPin3, flexPin2, flexPin1};
29 const int NUM_FLEX_SENSORS = 5;
30
31 Adafruit_MPU6050 mpu;
32
33 // --- Konfigurasi DFPlayer Mini ---
34 #define MP3_RX_PIN 16
35 #define MP3_TX_PIN 17
36 HardwareSerial dfpSerial(1);
37 DFRobotDFPlayerMini myDFPlayer;
38 bool dfPlayerReady = false;
39
40 // --- Variabel Global untuk TensorFlow Lite ---

```

```

41 tflite::MicroErrorReporter micro_error_reporter;
42 tflite::ErrorReporter* error_reporter = &micro_error_reporter;
43
44 const tflite::Model* model = nullptr;
45 tflite::MicroInterpreter* interpreter = nullptr;
46 TfLiteTensor* input_tensor = nullptr;
47 TfLiteTensor* output_tensor = nullptr;
48
49 const int TENSOR_ARENA_SIZE = 20 * 1024;
50 uint8_t tensor_arena[TENSOR_ARENA_SIZE] __attribute__((aligned(16)));
51
52 // --- Parameter Model dan Data ---
53 const int NUM_FEATURES = 11;
54 const int NUM_CLASSES = 26; // A hingga Z
55
56 float current_sensor_readings[NUM_FEATURES];
57
58 const float MEANS[NUM_FEATURES] = {
59     1428.4070, 1627.8153, 1347.8956, 1253.6952, 1075.7409, // Flex5 -> Flex1
60     0.0590, 0.8313, 0.2896,                                     // AccelX, AccelY, AccelZ
61     -0.0088, -0.0091, 0.0043                                    // GyroX, GyroY, GyroZ
62 };
63 const float STD_DEVS[NUM_FEATURES] = {
64     222.1246, 281.9704, 246.7781, 179.0070, 270.2374, // Flex5 -> Flex1
65     0.2106, 0.3776, 0.2201,                                // AccelX, AccelY, AccelZ
66     0.2350, 0.6496, 0.2731                                // GyroX, GyroY, GyroZ
67 };
68
69 // MPU6050 Offsets (sesuaikan jika perlu berdasarkan kalibrasi Anda)
70 const float ax_offset = 0.00; const float ay_offset = -0.02; const float az_offset =
71     0.04;
71 const float gx_offset = -0.01; const float gy_offset = -0.01; const float gz_offset =
72     0.00;
73
73 // --- Fungsi Setup ---
74 void setup() {
75     Serial.begin(9600);
76     while (!Serial) delay(10);
77     Serial.println(F("--- SIBI Gesture Recognition (ANN & DFPlayer - Verified) ---"));
78
79     if (!mpu.begin()) {
80         Serial.println(F("MPU6050 Gagal!"));
81         while (1) delay(10);
82     }
83     Serial.println(F("MPU6050 OK!"));
84     mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
85     mpu.setGyroRange(MPU6050_RANGE_250_DEG);
86     mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
87     delay(100);
88
89     Serial.println(F("Inisialisasi DFPlayer..."));
90     dfpSerial.begin(9600, SERIAL_8N1, MP3_RX_PIN, MP3_TX_PIN);
91     if (myDFPlayer.begin(dfpSerial, true, false)) {
92         Serial.println(F("DFPlayer OK."));
93         myDFPlayer.volume(28);
94         dfPlayerReady = true;
95     } else {

```

```

96     Serial.println(F("DFPlayer Gagal. Periksa kartu SD & koneksi."));
97 }
98
99 model = tflite::GetModel(TA_Model);
100
101 if (model->version() != TFLITE_SCHEMA_VERSION) {
102     error_reporter->Report("Versi skema model tidak cocok!");
103     return;
104 }
105 Serial.println(F("Model TFLite dimuat."));
106
107 static tflite::AllOpsResolver resolver;
108 static tflite::MicroInterpreter static_interpreter(
109     model, resolver, tensor_arena, TENSOR_ARENA_SIZE, error_reporter);
110 interpreter = &static_interpreter;
111
112 if (interpreter->AllocateTensors() != kTfLiteOk) {
113     error_reporter->Report("AllocateTensors() gagal!");
114     return;
115 }
116 Serial.println(F("Tensor dialokasikan."));
117
118 input_tensor = interpreter->input(0);
119 output_tensor = interpreter->output(0);
120
121 Serial.println(F("--- Info Tensor ---"));
122 Serial.print(F("Input Tensor Dims Count: ")); Serial.println(input_tensor->dims->size);
123 Serial.print(F("Input Tensor Dims: "));
124 for (int i = 0; i < input_tensor->dims->size; i++) {
125     Serial.print(input_tensor->dims->data[i]); Serial.print(F(" "));
126 }
127 Serial.println();
128
129 Serial.print(F("Output Tensor Dims Count: ")); Serial.println(output_tensor->dims->size);
130 Serial.print(F("Output Tensor Dims: "));
131 for (int i = 0; i < output_tensor->dims->size; i++) {
132     Serial.print(output_tensor->dims->data[i]); Serial.print(F(" "));
133 }
134 Serial.println();
135
136 Serial.println(F("Setup selesai."));
137 if (dfPlayerReady) myDFPlayer.play(26); // Suara "Sistem Siap" (file 0026.mp3)
138 }
139
140 // --- Fungsi Loop Utama ---
141 void loop() {
142     static int last_predicted_class_id = -1; // Menyimpan prediksi sebelumnya
143
144     // 1. Baca Flex Sensors (Flex5, Flex4, Flex3, Flex2, Flex1)
145     current_sensor_readings[0] = (float)analogRead(flexPinArray[0]);
146     current_sensor_readings[1] = (float)analogRead(flexPinArray[1]);
147     current_sensor_readings[2] = (float)analogRead(flexPinArray[2]);
148     current_sensor_readings[3] = (float)analogRead(flexPinArray[3]);
149     current_sensor_readings[4] = (float)analogRead(flexPinArray[4]);
150

```

```

151 // 2. Baca MPU6050 & Preprocess
152 sensors_event_t a_event, g_event, temp_event;
153 mpu.getEvent(&a_event, &g_event, &temp_event);
154
155 float ax = (a_event.acceleration.x / 9.80665F) - ax_offset;
156 float ay = (a_event.acceleration.y / 9.80665F) - ay_offset;
157 float az = (a_event.acceleration.z / 9.80665F) - az_offset;
158 float gx = g_event.gyro.x - gx_offset;
159 float gy = g_event.gyro.y - gy_offset;
160 float gz = g_event.gyro.z - gz_offset;
161
162 current_sensor_readings[5] = ax;
163 current_sensor_readings[6] = ay;
164 current_sensor_readings[7] = az;
165 current_sensor_readings[8] = gx;
166 current_sensor_readings[9] = gy;
167 current_sensor_readings[10] = gz;
168
169 // 3. Normalisasi Data Sensor (Z-score)
170 for (int i = 0; i < NUM_FEATURES; i++) {
171     if (fabs(STD_DEVS[i]) > 1e-7) {
172         current_sensor_readings[i] = (current_sensor_readings[i] - MEANS[i]) /
173             STD_DEVS[i];
174     } else {
175         current_sensor_readings[i] = (current_sensor_readings[i] - MEANS[i]);
176     }
177 }
178
179 // 4. Salin data ke tensor input
180 for (int i = 0; i < NUM_FEATURES; i++) {
181     input_tensor->data.f[i] = current_sensor_readings[i];
182 }
183
184 // 5. Jalankan inferensi
185 if (interpreter->Invoke() != kTfLiteOk) {
186     error_reporter->Report("Invoke failed!");
187     return;
188 }
189
190 // 6. Dapatkan output
191 int predicted_class_id = -1;
192 float max_prob = 0.0f;
193
194 int actual_output_units = output_tensor->dims->data[output_tensor->dims->size -
195 1];
196 if (actual_output_units != NUM_CLASSES) {
197     Serial.printf("PERINGATAN: Model output units (%d) TIDAK SAMA dengan
198     NUM_CLASSES (%d)!\n", actual_output_units, NUM_CLASSES);
199 }
200
201 for (int i = 0; i < actual_output_units && i < NUM_CLASSES; i++) {
202     float current_prob = output_tensor->data.f[i];
203     if (current_prob > max_prob) {
204         max_prob = current_prob;
205         predicted_class_id = i;
206     }
207 }
208

```

```

205
206 // 7. Tampilkan hasil dan mainkan suara hanya jika prediksi berubah
207 if (predicted_class_id != -1) {
208     char predicted_char = mapClassIDToSIBI(predicted_class_id);
209     Serial.printf("Prediksi: %c (ID: %d, Conf: %.2f%%) ",
210                 predicted_char, predicted_class_id, max_prob * 100.0f);
211
212     const float CONFIDENCE_THRESHOLD = 0.50;
213     if (dfPlayerReady && max_prob > CONFIDENCE_THRESHOLD) {
214         if (predicted_class_id != last_predicted_class_id) {
215             int fileToPlay = predicted_class_id + 1;
216             myDFPlayer.play(fileToPlay);
217             last_predicted_class_id = predicted_class_id;
218         }
219         Serial.println();
220     } else {
221         last_predicted_class_id = -1; // Reset jika confidence turun
222         Serial.println();
223     }
224 } else {
225     last_predicted_class_id = -1; // Reset jika tidak ada prediksi
226     Serial.println("Tidak ada prediksi yang cukup meyakinkan.");
227 }
228
229 delay(1000);
230 }
231
232 // --- Fungsi Helper untuk Memetakan ClassID ke Huruf SIBI ---
233 char mapClassIDToSIBI(int class_id) {
234     if (class_id >= 0 && class_id < 26) return (char)('A' + class_id);
235     return '?';
236 }
237
238
239
240 // --- Fungsi Helper untuk Detail DFPlayer ---
241 void printDetail(uint8_t type, int value) {
242     switch (type) {
243         case TimeOut: Serial.println(F("DFPlayer: Time Out!")); break;
244         case WrongStack: Serial.println(F("DFPlayer: Stack Wrong!")); break;
245         case DFPlayerCardInserted: Serial.println(F("DFPlayer: Card Inserted!")); break;
246         default: break;
247     }
248 }
```

Listing 5.4: Model ANN Sarung Tangan Pintar dalam bahasa C++

```

1 #pragma once
2
3 #ifdef __has_attribute
4 #define HAVE_ATTRIBUTE(x) __has_attribute(x)
5 #else
6 #define HAVE_ATTRIBUTE(x) 0
7 #endif
8 #if HAVE_ATTRIBUTE(aligned) || (defined(__GNUC__) && !defined(__clang__))
9 #define DATA_ALIGN_ATTRIBUTE __attribute__((aligned(4)))

```

```

10 #else
11 #define DATA_ALIGN_ATTRIBUTE
12 #endif
13
14 // automatically configure network
15 #define TF_NUM_INPUTS 11
16 #define TF_NUM_OUTPUTS 1
17 #define TF_NUM_OPS 2
18 #define TF_OP_FULLYCONNECTED
19 #define TF_OP_RELU
20
21
22 // these layers are used in Python
23 // but are not allowed in Arduino
24 // - Dropout
25 // - BatchNormalization
26
27
28 // sample data
29 float x0[11] = {0.03868563220f, -1.39310836689f, -0.89917031216f, -0.69659384218f,
   -0.89454998555f, 0.52672881089f, 0.42037804434f, -0.31627838028f, 0.03740889290f,
   -0.00144319521f, -0.12560991638f};
30 float x1[11] = {-0.75816458470f, 0.97948145295f, 1.80771479984f, 2.56026155438f,
   1.96589786521f, 0.71662530022f, 0.39389352723f, -0.58891897332f, -0.04771130098f,
   0.04473905136f, 0.02085136934f};
31 float x2[11] = {0.74999825520f, 0.22053642837f, 0.17872106477f, 1.05752724619f,
   0.75585071489f, 0.43178056623f, 0.26147094171f, 0.36532310234f, 0.03740889290f,
   -0.10920177053f, -0.05237927352f};
32 float x3[11] = {-0.96075362289f, 0.94756320426f, -0.29944127537f, -0.56810726565f,
   -0.63921893548f, 0.81157354488f, 0.36740901013f, -0.77067936869f, -0.04771130098f,
   0.01395088698f, -0.01576395209f};
33 float x4[11] = {-0.22693110676f, -1.88252151358f, -1.37733265230f, -0.47872529936f,
   -0.52450498545f, 0.33683232156f, 0.42037804434f, -0.77067936869f, 0.07996898983f,
   0.02934496917f, 0.09408201221f};
34 float x5[11] = {0.30880434980f, -0.40009618519f, 1.28903022748f, 2.30328840130f,
   2.09541361524f, 0.76409942255f, 0.39389352723f, -0.58891897332f, 0.03740889290f,
   -0.09380768834f, -0.01576395209f};
35 float x6[11] = {1.51533462171f, 0.67448485429f, -0.38453796302f, -0.57369363854f,
   -0.52080453545f, 0.28935819923f, 0.26147094171f, 0.41076320118f, -0.09027139791f,
   0.04473905136f, -0.05237927352f};
36 float x7[11] = {-0.33497859380f, 0.72058899130f, 1.11073240573f, -0.19382028182f,
   -0.48009958544f, -2.51161501837f, -0.21525036618f, 0.54708349771f, 0.07996898983f,
   -0.09380768834f, -0.05237927352f};
37 float x8[11] = {0.01167376044f, -0.81857989034f, -0.65198469564f, -0.58486638432f,
   1.99920191522f, 1.52368537987f, 0.23498642460f, -0.36171847912f, -0.00515120404f,
   -0.12459585272f, -0.08899459495f};
38 float x9[11] = {-0.70864281981f, -1.46403780844f, -0.98831922303f, -0.59603913011f,
   1.88078751519f, -2.93888211936f, 1.24139807460f, -1.81580164204f, 1.35677189790f,
   -6.69786894730f, 5.95253344130f};
39 float x10[11] = {1.25421986137f, 1.00430675750f, 1.12288907539f, 0.19163944779f,
   -0.36908608541f, -0.66012424741f, 0.39389352723f, -1.22508035710f, -0.04771130098f
   , -0.01683727739f, -0.01576395209f};
40 float x11[11] = {1.83047312557f, 1.03267853412f, -0.67629803497f, -0.38375696018f,
   -0.49120093544f, 0.33683232156f, 0.34092449302f, 0.00180231161f, -0.00515120404f,
   -0.03223135958f, -0.05237927352f};
41 float x12[11] = {-1.41095148555f, -1.18386651432f, -0.38048573980f, 0.34805788879f,
   -0.40239013542f, 0.52672881089f, 0.28795545881f, -0.40715857796f, -0.30307188259f,

```

```

    0.07552721574f, 0.02085136934f};
42 float x13[11] = {-0.78067447783f, -0.77956869748f, -0.14140456973f, -0.59603913011f,
   -0.74283153551f, -0.32780539109f, 0.36740901013f, -0.36171847912f, -0.13283149485f
   , -0.12459585272f, -0.08899459495f};
43 float x14[11] = {-0.92473779388f, -0.95689230136f, -0.34806795403f, 0.13018934597f,
   0.01946116469f, -0.18538302409f, 0.26147094171f, 0.31988300350f, -0.00515120404f,
   0.01395088698f, -0.01576395209f};
44 float x15[11] = {1.60537419424f, 0.92273789971f, 0.70145786035f, -0.22733851918f,
   -0.77243513552f, 0.43178056623f, -3.49933048721f, 2.59188794556f, -0.13283149485f,
   -0.00144319521f, 0.02085136934f};
45 float x16[11] = {1.70441772403f, 0.69576368675f, -1.05315479458f, -0.47872529936f,
   -0.58371218547f, 1.14389240121f, -3.52581500432f, 2.41012755020f, 0.03740889290f,
   0.01395088698f, 0.09408201221f};
46 float x17[11] = {-0.88421998624f, 0.41559239263f, 1.22014243271f, -0.73011207954f,
   -0.25807258538f, 0.00451346524f, 0.39389352723f, -0.36171847912f, 0.03740889290f,
   -0.00144319521f, -0.08899459495f};
47 float x18[11] = {-0.88421998624f, -0.69799983970f, -0.72897693685f, -0.86977140186f,
   -0.72062883550f, 0.05198758757f, 0.39389352723f, 0.09268250930f, -0.13283149485f,
   0.07552721574f, -0.05237927352f};
48 float x19[11] = {0.13322718336f, -0.34689910403f, -0.91132698182f, -0.35582509572f,
   -0.83904323554f, -0.56517600275f, 0.34092449302f, -0.45259867680f, -0.00515120404f
   , -0.01683727739f, -0.01576395209f};
49 float x20[11] = {-0.90222790075f, 0.84826198609f, 1.41870137057f, -0.27202950232f,
   -0.43939463543f, 0.52672881089f, 0.39389352723f, -0.17995808375f, 0.03740889290f,
   -0.01683727739f, -0.01576395209f};
50 float x21[11] = {-0.23593506402f, 0.98657439711f, 1.65778254064f, -0.07092007818f,
   -0.13225728535f, -0.28033126876f, 0.31443997592f, 0.13812260814f, -0.00515120404f,
   -0.01683727739f, -0.01576395209f};
51 float x22[11] = {-0.58258741826f, 1.04331795035f, 1.52405917433f, 2.61053891041f,
   -0.55780903546f, 0.05198758757f, 0.31443997592f, -0.13451798491f, -0.09027139791f,
   -0.15538401709f, -0.08899459495f};
52 float x23[11] = {-0.78067447783f, -0.68736042347f, -1.12204258934f, -0.91446238501f,
   -0.65772118549f, -0.37527951342f, 0.42037804434f, -0.54347887448f, 0.03740889290f,
   0.04473905136f, 0.13069733364f};
53 float x24[11] = {1.85298301871f, -1.24415653963f, -0.98426699981f, 0.05756649836f,
   1.58845196511f, 0.24188407690f, 0.44686256144f, -0.49803877564f, -0.00515120404f,
   -0.04762544177f, 0.09408201221f};
54 float x25[11] = {-0.80318437096f, 0.94756320426f, -0.72087249041f, -0.65190285904f,
   -0.72432928551f, -0.61265012508f, -0.40064198592f, 1.63764586990f, 2.03773344887f,
   -1.15599935936f, -4.88560170252f};

55
56
57 /* model size = 65392 bytes */
58 const unsigned char TA_Model[] DATA_ALIGN_ATTRIBUTE = { 0x1c, 0x00, 0x00, 0x00, 0x54,
   0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00, 0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10
   , 0x00, 0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0
   x1c, 0x00, 0x00, 0x00, 0x94, dst.};

```

Listing 5.5: Program Pembuatan Model ANN Sarung Tangan Pintar dalam bahasa Python

```

1 # -*- coding: utf-8 -*-
2 """NewModel.ipynb
3
4 Automatically generated by Colab.
5
6 Original file is located at

```

```

7   https://colab.research.google.com/drive/1iNjakptndCHBrfGw4d0Iec9mPehI8Vn0
8
9
10 # Impor library yang diperlukan
11 import numpy as np
12 import pandas as pd
13 import tensorflow as tf
14 from tensorflow.keras.models import Sequential
15 from tensorflow.keras.layers import Input, Dense, Dropout, BatchNormalization
16 from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
17 from tensorflow.keras.regularizers import l2
18 from sklearn.model_selection import train_test_split
19 from sklearn.preprocessing import StandardScaler
20 from scipy.stats import mode
21 import matplotlib.pyplot as plt
22 import seaborn as sns
23
24 # --- 1. Memuat dan Mempersiapkan Data ---
25 # Gunakan file CSV baru Anda dengan label 0-25
26 file_path = 'DataTA_new2.csv'
27 # Specify the semicolon delimiter
28 df = pd.read_csv(file_path, delimiter=';')
29
30 print("Contoh Data:")
31 print(df.head())
32 print("\nInfo Data:")
33 df.info()
34 print("\nJumlah Sampel untuk Setiap Kelas:")
35 print(df['ClassID'].value_counts().sort_index())
36
37 # Pisahkan fitur (X) dan label (y)
38 X = df.drop('ClassID', axis=1).values
39 y = df['ClassID'].values
40
41 # Tentukan kolom fitur untuk digunakan saat menyimpan nilai normalisasi
42 features_columns = df.drop('ClassID', axis=1).columns.tolist()
43
44 # --- 2. Normalisasi Fitur ---
45 # Pisahkan data menjadi set training dan testing/validasi
46 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state
        =42, stratify=y)
47
48 # Buat dan fit StandardScaler HANYA pada data training
49 scaler = StandardScaler()
50 X_train_scaled = scaler.fit_transform(X_train)
51 X_test_scaled = scaler.transform(X_test)
52
53 # Simpan nilai means dan standard deviations untuk digunakan di ESP32
54 means = scaler.mean_
55 std_devs = scaler.scale_
56
57 print("\n--- Nilai untuk Normalisasi di ESP32 ---")
58 print("const float MEANS[] = {")
59 print(", ".join([f"{val:.4f}" for val in means]) + "};")
60 print("const float STD_DEVS[] = {")
61 print(", ".join([f"{val:.4f}" for val in std_devs]) + "};")
62 print("Urutan Fitur:", features_columns)

```

```

63 print("-----\n")
64
65 # --- 3. [Opsional] Windowing dan Flattening (jika Anda ingin menangkap pola waktu)
66
66 X_train_final = X_train_scaled
67 y_train_final = y_train
68 X_test_final = X_test_scaled
69 y_test_final = y_test
70
71 NUM_FEATURES = X_train_final.shape[1]
72 NUM_CLASSES = len(np.unique(y))
73
74 # --- 4. Membangun Model ANN yang Ditingkatkan ---
75 def create_enhanced_ann_model(num_features, num_classes):
76     model = Sequential([
77         Input(shape=(num_features,), name='input_layer'),
78
79         Dense(128, kernel_regularizer=l2(0.001)),
80         BatchNormalization(),
81         tf.keras.layers.ReLU(),
82         Dropout(0.4),
83
84         Dense(64, kernel_regularizer=l2(0.001)),
85         BatchNormalization(),
86         tf.keras.layers.ReLU(),
87         Dropout(0.3),
88
89         Dense(64, kernel_regularizer=l2(0.001)),
90         BatchNormalization(),
91         tf.keras.layers.ReLU(),
92         Dropout(0.2),
93
94         Dense(num_classes, activation='softmax', name='output_layer')
95     ])
96     return model
97
98 # Buat model
99 model = create_enhanced_ann_model(NUM_FEATURES, NUM_CLASSES)
100
101 # Kompilasi model dengan learning rate yang mungkin sedikit lebih rendah
102 optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001)
103 model.compile(optimizer=optimizer,
104                 loss='sparse_categorical_crossentropy',
105                 metrics=['accuracy'])
106
107 model.summary()
108
109 # Generate and save the model plot
110 from tensorflow.keras.utils import plot_model
111 plot_model(model, to_file='model.png', show_shapes=True, show_layer_names=True)
112
113 # Display the generated image in the notebook
114 from IPython.display import Image
115 display(Image(filename='model.png', width=400, height=1800)) # Adjusted width and
116     height
117 # --- 5. Melatih Model ---

```

```

118 # Callbacks untuk menyimpan model terbaik dan menghentikan training lebih awal
119 model_checkpoint = ModelCheckpoint('sibi_ann_model_enhanced.keras', save_best_only=
120     True, monitor='val_accuracy', mode='max')
120 early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=
121     True)
122
122 history = model.fit(
123     X_train_final,
124     y_train_final,
125     epochs=25,
126     batch_size=32,
127     validation_data=(X_test_final, y_test_final),
128     callbacks=[model_checkpoint, early_stopping]
129 )
130
131 # --- 6. Evaluasi Model ---
132 # Muat model terbaik yang disimpan
133 best_model = tf.keras.models.load_model('sibi_ann_model_enhanced.keras')
134
135 # Evaluasi pada data test
136 test_loss, test_accuracy = best_model.evaluate(X_test_final, y_test_final)
137 print(f"\nAkurasi pada data test: {test_accuracy*100:.2f}%")
138
139 # Plot hasil training
140 import matplotlib.pyplot as plt
141
142 plt.figure(figsize=(12, 5)) # Sedikit diperbesar untuk kejelasan
143
144 plt.subplot(1, 2, 1)
145 plt.plot(history.history['accuracy'], label='Train Accuracy')
146 plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
147 plt.title('Model Accuracy')
148 plt.xlabel('Epoch')
149 plt.ylabel('Accuracy')
150 plt.legend(loc='lower right')
151 plt.grid(True)
152
153
154 plt.subplot(1, 2, 2)
155 plt.plot(history.history['loss'], label='Train Loss')
156 plt.plot(history.history['val_loss'], label='Validation Loss')
157 plt.title('Model Loss')
158 plt.xlabel('Epoch')
159 plt.ylabel('Loss')
160 plt.legend(loc='upper right')
161 plt.grid(True)
162
163 plt.tight_layout()
164 plt.show()
165
166 # --- 7. Prediksi dan Analisis Tambahan (Optional) ---
167 from sklearn.metrics import confusion_matrix, classification_report
168 import seaborn as sns
169 import matplotlib.pyplot as plt
170 import numpy as np
171
172 # Get predictions for the test set

```

```

173 y_pred_probs = best_model.predict(X_test_final)
174 y_pred = np.argmax(y_pred_probs, axis=1)
175
176 # --- Confusion Matrix ---
177 print("\n--- Confusion Matrix ---")
178 cm = confusion_matrix(y_test_final, y_pred)
179
180 # Define class_names based on the unique values in the original 'ClassID' column
181 # Assuming 'df' contains the original data and 'ClassID' is the column with class
182 # labels
183 if 'df' in locals() and 'ClassID' in df.columns:
184     class_names = sorted(df['ClassID'].unique())
185 else:
186     # Fallback if df or 'ClassID' is not available, though it should be based on the
187     # notebook state
188     # In this case, we'll just create generic class names
189     class_names = [str(i) for i in range(NUM_CLASSES)] # Use NUM_CLASSES defined
190     # earlier
191
192
193 # Plot confusion matrix
194 plt.figure(figsize=(12, 10))
195 sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
196             xticklabels=class_names, yticklabels=class_names)
197 plt.xlabel('Predicted Label')
198 plt.ylabel('True Label')
199 plt.title('Confusion Matrix')
200 plt.show()
201
202
203 # --- Classification Report ---
204 print("\n--- Classification Report ---")
205 # Generate class labels based on the unique classes in y_test_final
206 # Assuming y_test_final contains the actual class indices (0-25)
207 class_labels = [str(i) for i in sorted(np.unique(y_test_final))]
208 print(classification_report(y_test_final, y_pred, target_names=class_labels))
209
210
211 # Tampilkan beberapa prediksi vs label asli
212 print("\n--- Sample Predictions ---")
213 for i in range(10): # Tampilkan 10 sampel pertama
214     # Access y_test using positional index .iloc[i]
215     print(f"Sample {i+1}: Actual ClassID = {y_test_final[i]+1}, Predicted ClassID = {y_pred[i]+1}")
216
217 !pip install "everywhereml>=0.2.32"
218
219 """
220 Export NN to C++
221 Copy-paste the generated code inside a file named model.h or irisModel.h
222 in your Arduino project
223 """
224 from everywhereml.code_generators.tensorflow import convert_model
225 import tensorflow as tf # Import tensorflow for one-hot encoding
226
227 # Use the normalized training data as the representative dataset for C++ export
228 # convert_model needs input data to understand feature scaling/ranges for C++
229 # generation
230 X_np = X_test_final # Changed from X.values to X_train_normalized

```

```
225
226 # One-hot encode the target variable for the C++ conversion
227 # Assuming NUM_CLASSES is the total number of classes
228 y_train_one_hot = tf.keras.utils.to_categorical(y_test_final, num_classes=26)
229
230 # Use the loaded best model (best_model_ann_loaded) for conversion
231 # Pass the one-hot encoded y to the convert_model function
232 c_header = convert_model(best_model, X_np, y_train_one_hot, model_name='TA_Model')
233 print(c_header)
234
235 with open('MODEL.h', 'w') as f:
236     f.write(c_header)
```

DAFTAR PUSTAKA

- Ahmed, M. A., Bahaa, B., Zaidan, A., Salih, M., & Lakulu, M. M. (2018). A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors*, 18, 2208. <https://doi.org/10.3390/s18072208>
- Alrubayi, A. H., Ahmed, M. A., Zaidan, A. A., Albahri, A. S., Zaidan, B. B., Albahri, O. S., & Alazab, M. (2021). A pattern recognition model for static gestures in malaysian sign language based on machine learning techniques. *Computers and Electrical Engineering*, 95, 107383.
- Bhat, K., & Chayalakshmi, C. L. (2020). Advanced glove for deaf and dumb with speech and text message on android cell phone. *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 1–7.
- BPS-Jatim. (2019). *Banyaknya desa/kelurahan menurut keberadaan penyandang cacat*. Retrieved February 20, 2025, from <https://jatim.bps.go.id/statictable/2019/10/04/1557/banyaknya-desa%20kelurahan-menurut-keberadaan-penyandang-cacat-.html>
- Chang, T., H. (2020). Sensor glove based on novel inertial sensor fusion control algorithm for 3-d real-time hand gestures measurements. *IEEE Transactions on Industrial Electronics*, 67(1), 658–666.
- Components101. (2023). Stm32f103c8t6 blue pill arduino guide [Diakses pada 17 maret 2025]. https://components101.com/sites/default/files/component_datasheet/STM32F103C8T6-Blue-Pill-Development-Board-Datasheet.pdf
- Damastuti, E. (2021). Teknologi asistif: Konsep dan aplikasi untuk disabilitas [Diakses pada 18 maret 2025]. https://repo-dosen.ulm.ac.id/bitstream/handle/123456789/25362/Teknologi%20Asistif%20_Eviani%20Damastuti.pdf
- Darmawan, Y. A., Arifin, A., & Arrofiqi, F. (2023). Rancang bangun detachable multi-sensor glove penerjemah sistem isyarat bahasa indonesia bagi individu tunarungu-wicara. *Jurnal Teknik POMITS*, 1(1), 1–6.
- Dias, T. S., Júnior, J. J. A. M., & Pichorim, S. F. (2022). An instrumented glove for recognition of brazilian sign language alphabet. *IEEE sensors journal*, 22(3), 2518–2529.
- Electronica60Norte. (2025). Datasheet bluetooth to serial port module hc05 [Diakses pada 17 maret 2025]. <https://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>
- Espressif. (2025). *Esp32-devkitc v4 datasheet* [Available: <https://www.espressif.com/>, Diakses pada 17 maret 2025].
- Galván-Ruiz, J., Travieso-González, C. M., Pinan-Roescher, A., & Alonso-Hernández, J. B. (2023). Robust identification system for spanish sign language based on three-dimensional frame information. *Sensors*, 23(1). <https://doi.org/10.3390/s23010481>
- Goodfellow, I., Bengio, Y., & Courville, A. (2020). *Deep learning* [<http://www.deeplearningbook.org>]. MIT Press.
- Haq, E. S., Suwardiyanto, D., & Huda, M. (2018). Indonesian sign language recognition application for two-way communication deaf-mute people. *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, 313–318.
- Interlink Electronics. (2025). Fsr 402 data sheet [Diakses pada 17 maret 2025]. <http://www.interlinkelectronics.com/FSR402short.php>

- InvenSense. (2022). *Mpu-6000 and mpu-6050 register map and descriptions* (I. Inc., Ed.) [Revision 4.0, Document Number: RM-MPU-6000A-00]. www.invensense.com
- Lee, B. G., & Min, L. S. (2018). Smart wearable hand device for sign language interpretation system with sensors fusion. *IEEE Sensors Journal*, 18(3), 1224–1232.
- Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia, K. (2025). *Sistem isyarat bahasa indonesia (profil)*. Retrieved February 20, 2025, from <https://pmpk.kemdikbud.go.id/sibi/profil>
- Retnoningsih, A., Arifin, A., & Arrofqi, F. (2023). Desain komunikasi dua arah sistem isyarat bahasa indonesia untuk penyandang disabilitas tunarungu wicara berbasis smart glove. *Jurnal Teknik POMITS*, 1(1), 1–7.
- Saggio, G., & Orengo, G. (2018). Flex sensor characterization against shape and curvature changes. *Sensors and Actuators A: Physical*, 273, 221–231.
- Shaheen, H., & Mehmood, T. (2018). Talking gloves: Low-cost gesture recognition system for sign language translation. *2018 IEEE Region Ten Symposium (TENSYMP)*, 219–224.
- Store Siqma. (2025). Gy-952 6-dof imu sensor module [3v-5v] [Diakses pada 17 maret 2025]. <https://store.siqma.com/gy-952-imu-module.html>
- Suleman, A. R., & Palupi, I. (2023). Penerapan artificial neural network (ann) untuk prediksi prestasi akhir mahasiswa melalui nilai mata kuliah dasar tingkat 1. *eProceedings of Engineering*, 10(2).
- Syed Mubarak Ali, S. A. A., Ahmad, N. S., & Goh, P. (2019). Flex sensor compensator via hammerstein-wiener modeling approach for improved dynamic goniometry and constrained control of a bionic hand. *Sensors*, 19(18). <https://doi.org/10.3390/s19183896>
- Widodo, R. B., Swastika, W., & Haryasena, A. B. (2020). Studi sensor dan akuisisi data hand gesture dengan sarung tangan. *Conference on Innovation and Application of Science and Technology (CIASTECH)*, 561–568.
- World-Health-Organization. (2024). *Deafness and hearing loss*. Retrieved February 20, 2025, from <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>
- Wu, J., Sun, L., & Jafari, R. (2016a). A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. *IEEE Journal of Biomedical and Health Informatics*, 20(5), 1281–1290.
- Wu, J., Sun, L., & Jafari, R. (2016b). A wearable system for recognizing american sign language in real-time using imu and surface emg sensors. *IEEE J Biomed Health Inform*, 20(5), 1281–1290.
- Wungow, K. C., Widodo, R. B., & Subianto, M. (2022). Studi klasifikasi dengan knn dan ann pada sarung tangan penerjemah angka dan alfabet bahasa isyarat sibi. *Prosiding Seminar Nasional Universitas Ma Chung*, 2.

BIOGRAFI PENULIS



Muhammad Fauzan Ariyatmoko, lahir pada 02 November 2002 di Kota Bogor. Penulis adalah anak bungsu dari 3 bersaudara. Saat ini penulis sedang menempuh pendidikan Strata-1 di Departemen Fisika Institut Teknologi Sepuluh Nopember dengan Nomor Registrasi Pokok 5001211062. Sebelumnya penulis menempuh pendidikan sekolah dasar di SDN Bantarjati 5 Kota Bogor. Kemudian penulis melanjutkan pendidikan di sekolah di SMPN 3 Kota Bogor dan SMAN 4 Kota Bogor. Selama menjadi mahasiswa, penulis aktif di berbagai kegiatan akademik dan non akademik di luar jam kuliah. Di ranah kegiatan akademik penulis aktif menjadi Asisten dan Koordinator Asisten di Laboratorium Fisika Dasar. Untuk kegiatan non-akademik penulis aktif berorganisasi di INTEL yakni Organisasi Laboratorium Instrumentasi Elektronika ITS, yang mempunyai peran sebagai ketua Divisi RnD dan menjadi Alumni Bangkit Academy 2024 Batch 1 pada bidang Machine Learning dengan predikat A (Sangat Memuaskan).

[Halaman ini sengaja dikosongkan]